DISTRIBUTED ENVIRONMENT OF MACHINE LEARNING WITH OPTIONAL PRIVACY PRESERVING FOR SUPPLY CHAIN MANAGEMENT

Submitted by
Tejashwini Neralekere Appanna Gowda

A thesis submitted in fulfilment for the requirements of the degree
Master's by Research-2024

Candidate Declaration

This thesis is my own original work and does not contain material that has been accepted for the award of any other degree or diploma.

To the best of the knowledge this thesis contains no material previously published or written by another person except where due reference is made in the text.

Tejashwini Neralekere Appanna Gowda

22 September 2023

Distributed Environment of Machine Learning with Optional Privacy Preserving for Supply Chain Management

Tejashwini Neralekere Appanna Gowda

Abstract

This thesis delves into the intricate challenges inherent in supply chain collaboration, seeking to provide effective solutions for the modern era. It introduces innovative privacy-preserving strategies designed to facilitate collaboration among multiple parties within the supply chain ecosystem while safeguarding the integrity of raw data. The primary focus of our research revolves around optimizing the complexities of last-mile delivery, a crucial aspect of contemporary logistics that involves multiple participants and the transportation of various categorized goods.

To tackle these multifaceted challenges, we have developed a pioneering Bilevel framework that leverages DBSCAN and PSO algorithms. This framework has yielded remarkable results by reducing the total travel distance by an impressive 44%, significantly streamlining the delivery process. What sets our approach apart is the emphasis on data privacy through the innovative Radius-Sector strategy, which ensures that sensitive information remains protected during collaborative operations. Furthermore, we recognize the pressing need for timely delivery of perishable products, such as groceries, in the context of the booming online shopping industry. To address this challenge, we have crafted a hybrid framework that seamlessly combines classification and clustering techniques. This hybrid approach significantly enhances last-mile delivery efficiency, reducing the total travel time by 24% when compared to conventional baseline methods.

This research represents a substantial contribution to the field of supply chain management, offering practical and innovative solutions that are tailored to the contemporary landscape of logistics and distribution.

Supervisory Panel

The work of this thesis was supervised by the following Swinburne staffs:

Principal supervisor: Dr.Pei Wei Tsai
Co-Supervisors: Prof Hussein Dia
External Supervisor: Dr Kewen Liao (Australian Catholic university)

Acknowledgements

I wish to convey my profound gratitude to my esteemed supervisory team, Dr. Pei Wei Tsai, Professor Hussein Dia, and Dr. Kewen Liao, whose unwavering guidance and support have been of immeasurable value throughout the progression of this research endeavor.

It is worth noting that the execution of this work occurred amidst the backdrop of the formidable challenges posed by the global COVID-19 pandemic, which introduced intricate hurdles into the research process.

Lastly, I extend my heartfelt appreciation to my life partner and my family, whose steadfast encouragement, unwavering love, and exceptional understanding have played an indispensable role in facilitating the realization of this research. Their steadfast support has been instrumental in this academic journey.

Contents

List of Figures

List of Tables

List of Algorithms

1 List of Acronyms

2 Introduction

Supply chain management stands as a formidable economic force, making substantial contributions to national revenue, with billions of dollars at stake. This intricate network encompasses a multitude of sectors, spanning from manufacturers to end customers. Within this multifaceted ecosystem, collaboration assumes a pivotal role in enhancing visibility, streamlining processes, and ensuring efficiency to meet the ever-evolving demands of customers. In this intricate web of supply chain operations, the collaborative involvement of multiple parties from diverse companies has become commonplace, accentuating the imperative for data exchange to optimize outcomes. However, this exchange of information has engendered valid concerns regarding data security, potentially breaching the confidentiality of sensitive corporate data and customer records.

The supply chain industry constitutes a substantial economic force, contributing billions of dollars to the national revenue [1]. Supply chain management encompasses the orchestration of various sectors, including suppliers, manufacturers, distributors, retailers, and end customers. Collaboration within the supply chain ecosystem enhances visibility, streamlines processes, and facilitates efficiency, thereby meeting customer demands. Effective collaboration among supply chain partners involves planning, execution, and optimization, resulting in reduced time, distance, and resource consumption. The ever-increasing demand in today's market has led to a significant expansion of supply chain operations, challenging companies to maintain affordable and reliable logistics [2].

In this intricate network, multiple parties from different companies collaborate to optimize results, necessitating data exchange. However, security concerns arise when dealing with confidential data, as the exchange of information may lead to privacy breaches. Competent companies may gain access to internal data or customer records. Preserving privacy has become a paramount concern across industries due to significant losses incurred from cyberattacks and data breaches in the 21st century [45].

The supply chain operates in a distributed environment, underscoring the critical need to manage data flow while preserving privacy. Various methods exist to protect data in distributed machine learning environments. Differential privacy [23], homomorphic encryption [28, 29], secure multi-party computation [30], and federated learning [44] are some of the approaches performing effectively in different scenarios, each with its own trade-offs and substantial implementation costs [9].

Among all stages in the supply chain, last-mile delivery stands out as exceptionally complex. Increasing demands result in higher order volumes, complicating the movement of goods [3]. Consequently, last-mile delivery represents the most challenging, inefficient, and costly segment of the supply chain, accounting for almost 28 percent of total transportation costs [4]. Multiple hubs and parties involved in deliveries make optimization challenging. Although some researchers have focused on optimizing last-mile delivery [15, 16, 17, 18], very few have addressed multi-party delivery. Logistic congestion and inaccurate route planning contribute to extended delivery times, consuming a significant portion of e-commerce logistics costs [6]. The exchange of tasks among different parties, based on constraints, has the potential to reduce time, distance, vehicle numbers, and resource usage, thereby enhancing overall customer satisfaction.

Modern delivery services cater to a wide range of products, from food items to furniture, necessitating customized approaches based on product type. For example, supermarket deliveries involve diverse products, each with unique storage requirements. Classification techniques have been explored in supply chain optimization [60, 61, 62, 63], but a clear framework for classification and obtaining optimized results remains an active area of research.

To address these challenges and formulate them as convex optimization problems, significant progress has been made in machine learning. Bilevel optimization, a systematic search of hyperparameters, has proven efficient in various domains, including transportation, manufacturing, economics, decision sciences, and engineering [7, 12]. In our thesis, we aim to tackle supply chain collaboration using bilevel optimization, focusing on multi-party collaboration.

Our research addresses these challenges by developing a comprehensive framework for optimizing last-mile delivery services through multi-party collaboration using bilevel optimization. We have also devised a method to protect sensitive data while fostering collaboration. Primarily focused on controlled environment deliveries, our hybrid framework leverages classification and clustering techniques.

The development of our framework unfolds in different phases:

- **Phase 1:** A Radius-Sector Information Blur Strategy for preserving real data.
- **Phase 2:** A Bilevel framework for multi-party collaboration in the supply chain with privacy preservation for end-users.
- **Phase 3:** A Hybrid framework using classification and clustering for multi-party deliveries.

In summary, our research endeavors to tackle critical challenges within the supply chain industry, offering innovative solutions and methodologies to optimize last-mile delivery and safeguard sensitive data while promoting multi-party collaboration.

2.1 Aim & Research Question

Upon discussing the research gap in the previous section, we have concluded to the below research questions(RQ)

- RQ 1: How to hide/Preserve the real information in Optimisation process?
- RQ 2: How to obtain Collaborative last mile delivery with multiple parties involved and by preserving the private information?
- RQ 3: How to solve the Cooperative shipment under the category limitations in supply chain?

2.2 Contribution of the Research

- A Radius Sector strategy (R-S) to calculate the distance in the clusters by preserving the privacy of users
- A Privacy preserving framework using Bilevel optimisation for supply chain management
- A Hybrid approach using the classification and clustering to deliver the jobs between multi parties.

2.3 Research Methodology

Supply chain management has various phases from Suppliers to the End customers. While our research is mainly focused in the last two phases, where the products reach from the Hubs to the end customers. Multiple delivery partners and Hub locations will have their own delivery jobs to deliver. Most of the times, all the delivery jobs are not similar, we need a different delivery process depending upon the product. While the years of research confirmed that collaboration and cooperation between the entities makes the process efficient.So, we are trying to achieve the efficient delivery with minimal time and distance with collaboration between multiple parties. Collaboration between the parties also needs information exchange, but exchanging between the competent parties leads to data privacy problems, so we have also achieved preserving the privacy by hiding the real data until the end stage. We have created a fuzzy technique to preserve the real data as discussed in Section: 4.1.1. We have created a two different framework for delivering the jobs using multiple party collaboration using the soft computing techniques.

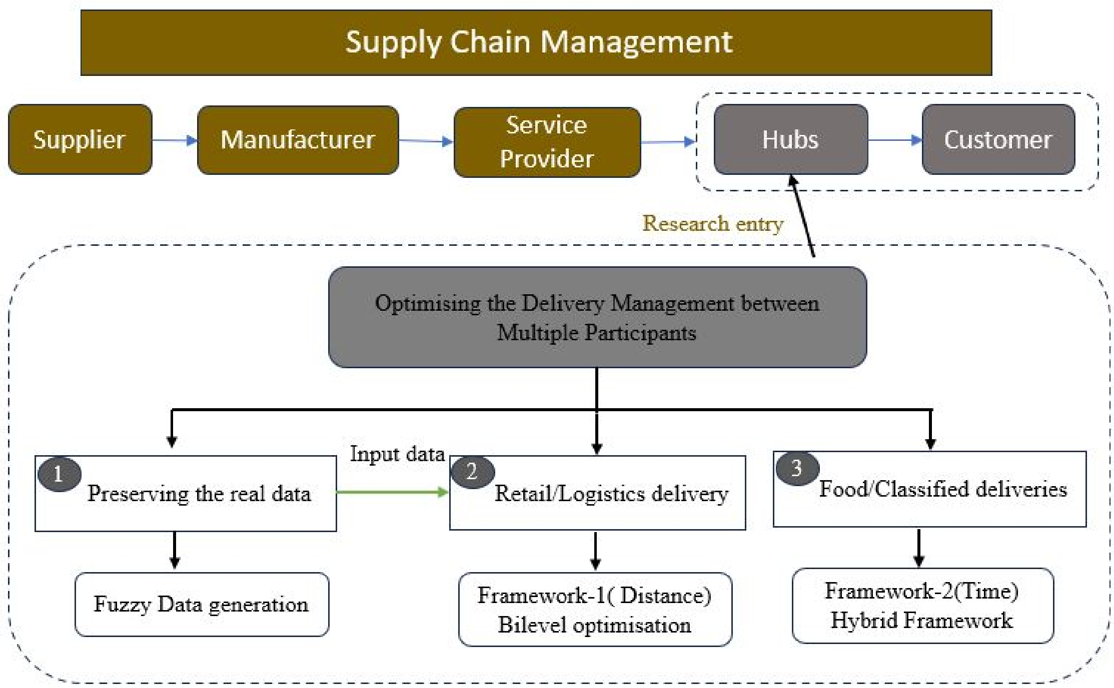


Figure 1: Overview of Research Methodology

Our first framework is suitable for regular delivery jobs like logistics or Retail. we have solved it using the Bilevel approach where the problem is divided into two layers and we are using Density based spatial clustering for applications with noise (DBSCAN) clustering in upper layer and Particle swarm optimisation in lower layer. We also hidden the real data using the fuzzy technique we created. Using this framework, We have achieved the efficient delivery list to deliver by reducing the total distance travelled by 65.22% by comparing with the baselines.

Our second framework is focused on the supermarket delivery or any sensitive parcels delivery where it needs to be classified before delivering. We have created a Hybrid framework with classification and clustering. We classify the products based on the rules defined and then we cluster them accordingly between the Hubs. Once the final clusters are achieved we also deliver the products to the end customer locations. This approach is built using the time constraint. We have reduced the total travelling time by 24% using this approach. The overall flow is been detailed in the Fig: 1 and explained in the chapters below.

2.4 Thesis Structure

The proposed structure of my thesis is illustrated in Fig: 2

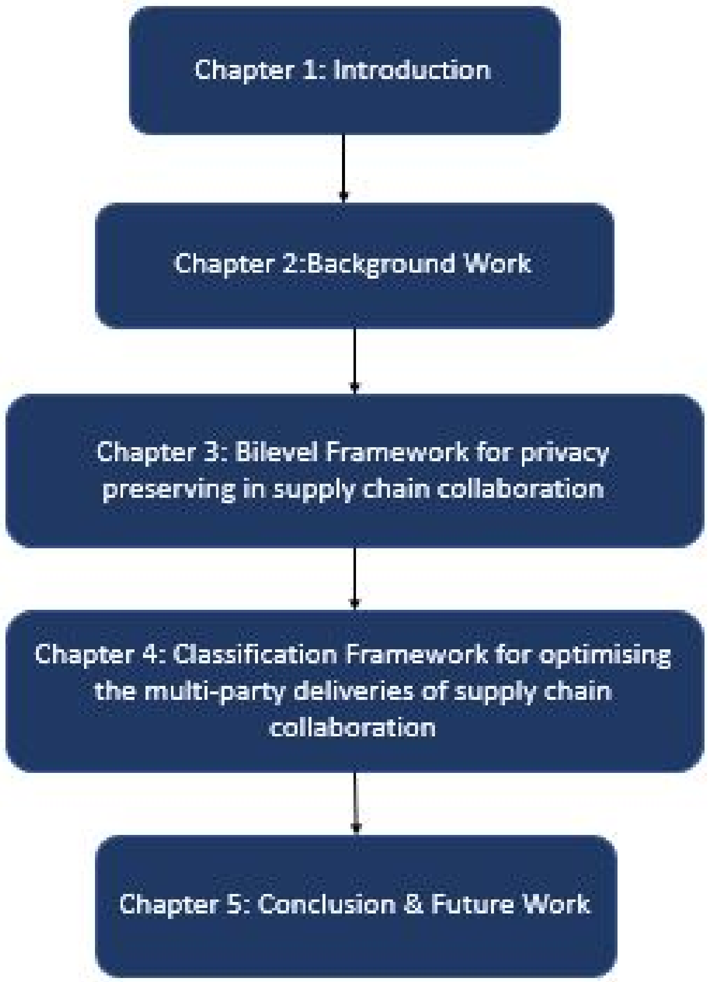


Figure 2: Proposed Thesis Structure

3 Background and Related Work

While our study is in supply chain collaboration, collaboration includes multiple parties, background on the multi party collaboration in supply chain explains us methods and importance of having a multiple parties involved. As the supply chain growing in numbers, collaboration extension has increased with multiple companies and even competitors. While coordinating between them, taking care of the privacy of each other is priority. We have discussed various methods available in the distributed environments of machine learning and how it could facilitate supply chain in preserving the privacy and confidential data while still achieving the overall effective collaboration. We have also discussed on the optimisation of supply chain using which we could facilitate the available resources and obtain the maximum efficiency, among the available optimisation techniques we have focused on the advanced optimisation methods and also surveyed over the literature's in past how they have used the optimisations and obtained the effective results. While dealing with data driven in last mile delivery services, we need effective methods like clustering and classification to handle the data and make it ready for the optimisation process, we have discussed about few data driven approaches and how they could alleviate the performance of the last mile delivery services.

3.1 Supply Chain Collaboration in Multi Party Management

Supply chain is one of the biggest economy and most in-demand sector. Supply chain collaboration includes the process of various entities working together to achieve common objective. various entities includes suppliers, manufacturers, distributors, retailers, and other stakeholders who could contribute for the smooth operation of supply chain collaboration. The main goal being to improve the overall efficiency, performance, reduce the cost, reduce the distance and time and obtain higher customer satisfaction.

There are various benefits of doing collaboration in Supply chain management (SCM).

- Operation planning- Sharing the information between each other will ensure better performance together leading to smooth operations.
- Economical- Collaboration achieves working in harmony leading to greater economical benefits [107]
- Customer satisfaction- Speed of delivery or the response time to customers and meeting their demand requirements with ease and timely manners leads to greater customer satisfaction and expanding the business [109].

- Cooperative optimisation- Exchanging the required information between the various parties leads to getting optimised results in cooperative way benefiting supply chain
- Waste reduction- Having the collaboration leads to dividing the works in efficient ways reducing the repetitive tasks and wastage of resources [108]
- Reduce supply chain uncertainty- Supply, demand and technological uncertainty could be reduced by coordinating the customer demands with suppliers, manufacturers and production teams with good collaboration [110]

A systematic literature review [105] with the field research was conducted in the areas of medical and hospitals in the similar supply chain, and bullwhip effect was been observed over the years. Bullwhip effect defined as ”the effects of uncertainty in demand and lead time cause order sizes and lead times to be inflated the further up the supply chain and away from the end customer actually get” [89]. This causes an impairment in the industry and leads to loss of billions of dollars [91]. Lack of mutual trust, honesty, information sharing, credibility and compliance mainly between the competence companies were the primary reason. By collaborating and exchanging the information and coordinating among the supply chain participants the bullwhip effect could be improvised [106].

Few of the literature's where the successful collaborations have implemented is been discussed below- SCM includes transportation and logistics planning decisions which can be classified as strategic, tactical and operational planning [92]. Collaborative transportation has been studied which can be classified as vertical or horizontal collaborations [93]. Under horizontal collaboration an investigation was made by combining the depot location and vehicle routing decisions in urban road freight transportation for collaborating optimisation with an objective to minimise the total cost and the CO2 emissions and increase the job opportunities. By comparing the results of collaborative and non collaborative approach, by collaborating there was a reduction in the rate of transportation cost, number of vehicles used, total distance travelled, CO2 emitted, and improvement in the load rate of vehicles [111]. Partners serve distinct network regions in vertical collaborations, whereas they can serve the same or overlapping network in a horizontal way. Several challenges have been identified in the horizontal collaboration [94] such that partners need to agree on a mechanism to share their cost and benefits, decisions about which orders are remained (with the partner) or to be exchanged (with other partners) must be decided operationally subject to order requests [95]. Cooperative game theory and simulation modeling approaches have been applied for horizontal collaboration [93].

For a successful collaboration between the various parties, sharing of information is really important. Whereas, with the increase in threats and privacy concerns sharing the information also leads to leaking the customer privacy or confidential data [91]. With the evolution of machine learning there are various techniques which could be used for the distributed environments like supply chain. Clustering [24, 52], supervised learning [25, 53] and privacy-preserving [26] from the field of data mining exist for handling such privacy leak with the objective of information exchange minimization through partial data sharing. On the other hand, from the security perspective, Trusted Third Party and Secure Multiparty Computation [45] are two typical approaches involving cryptography protocols and algorithms that encrypt messages or data for sharing. The security approaches without allowing the decrypted messages would make the associated cost optimization objective inferior. We will be discussing various privacy preserving approaches in machine learning in the next section.

3.1.1 Conclusion

Early SCM primarily revolved around optimizing individual processes. However, as business environments expanded in scale and complexity, a more intricate and collaborative approach became imperative. The evolution towards collaborative supply chain management has been well-documented in the literature, highlighting its benefits in terms of improved demand forecasting, reduction of the bullwhip effect, enhanced customer service, adaptability to market changes, and effective risk management.

In the contemporary landscape, the integration of recent technological advancements, particularly in machine learning, is paramount to address various supply chain challenges. These challenges encompass data privacy concerns, issues of trust, complexities arising from diverse stakeholder interests, and resistance to change. Embracing these technologies is crucial for staying competitive and resilient in the dynamic supply chain environment. In response to the rapid increase in the interest mainly in waste reduction and the pursuit of multi-party collaboration with a focus on privacy preservation, our research offers a valuable and effective approach. Our work contributes to the ongoing discourse on supply chain optimization, emphasizing the importance of harnessing advanced technologies to tackle evolving complexities and meet the ever-changing demands of modern SCM.

3.2 Privacy Preserving in Supply chain collaboration

Supply chain being one of the widely distributed environments includes multiple parties like suppliers, manufacturers, retailers, distributors during the collaboration. As the growth of supply chain and customer demands multiple companies in the same party is been competing every day. There's a lot of information flow and exchange happening between the parties. Protecting these information is vital to manage the trust, relationships, rules and regulations and company profits. While exchanging the information we cannot provide all the confidential data or customer data to each other for collaboration. There should be a way where we can filter out what we could provide each other and still obtain the optimisation and collaboration for benefits of each other. The recent years the privacy in distributed environments has given most importance and with the help of machine learning developments there are few of the technologies and frameworks discussed below which we could use in the supply chain and secure the information flow.

The growth in the distributed environments and collaborative training in machine learning has exponentially increased in recent years, which leads to challenges in preserving the sensitive or raw data. Obtaining the optimised results with an effective training includes enormous amount of data handling, increasing the higher risks of cyber attacks and data theft. Having a strong privacy preserving methods is at most important than ever in the history. Few of the efficient privacy preserving for distributed environments are discussed in [79, 96] also tabulated in Table: 4.

3.2.1 Federated Learning

Traditional way of Machine learning approaches requires to accumulate all the date at one single place of any machine or biggest to a data center, whereas with the increasing applications of machine learning day by day, data has been growing enormous and in order to accumulate all of them in one data center, and then apply machine learning algorithms would be a tedious job and expensive. Federated learning is a privacy preserving decentralised approach in machine learning has set up a new era in Machine learning where we can apply the Machine learning algorithms and obtain the required results without having to accumulate all the data at single point. All the raw data would be kept on devices, the federated server send the updated model to the clients and based on the clients local data the model gets trained and only the gradient updates would be sent to the server [44]. Federated learning (FL) mainly used in the prediction model of the mobile phones by keeping the user data local. Working of this Federated learning on the mobile devices is explained as follows, mobile device downloads the current model, improves it by learning from data on the phone, and then

Table 1: Privacy Preserving Techniques in Machine Learning

Privacy Preserving Techniques	**Explanation**	**Example**
Differential privacy (DP)	A way to measure and quantify an individuals privacy when their data is used for analysis by focusing on minimising the risk of re identifying individuals within the released or analysed data while maintaining the meaningful results by introducing the controlled amount of noise or the randomness in the data.	Random shuffling of data to remove the association between individuals and their data entries [23].
Homomorphic encryption	Cryptographic technique that preserves the ability to perform mathematical operations on data as if it was unencrypted (plain text).	Performing neural network computations on encrypted data without first decrypting it [28, 29].
Secure multi party computation (SMPC)	It is a cryptographic technique where multiple parties can collaboratively compute an analye the results from their combined data without exposing the individual data to each other. It ensures the sensitive information to retain private during the computation process.	Determining which patients two hospitals have in common without revealing their respective patient list (private set intersection) [30].
Hardware security implementation	Collection of techniques whereby specialized computer hardware provides guarantees of privacy or security.	Secure storage or processing enclaves in mobile phones or computers [112]
Anonymisation	Removal of personally identifiable information from a dataset.	Removing information related to age, gender and so on [113].
Pseudonymisation	Replacement of personally identifiable information in a dataset with a dummy/synthetic entry with separate storage of the linkage record (look-up table).	Replacing names with randomly generated text [114].

summarizes the changes as a small focused update. Only this update to the model is sent to the cloud, using encrypted communication, where it is immediately averaged with other user updates to improve the shared model. All the training data remains on mobile device, and no individual updates are stored in the cloud [21]. This technique aids with smarter models, training deep neural networks [14] which takes less power consumption and low latency with privacy. Alongside providing the update to the shared model, the improved model can be used immediately.

Failure of Conventional FL reasons as mentioned in [35]

- One single model will not be able to fit local data distribution and learn the global model to achieve their goal.
- All the clients are treated equally and only one model is learned which is not capable of achieving the goal.
- Conventional FL assumes all the clients are congruent which is one central model can fit all clients distributions at same time.

As FL been progressing from 2017 [46, 48, 49, 51] and has various applications in large environments, various attacks have been notified on the FL environment as well. When data is been scattered with so many heterogeneous devices and though the raw data would remain on the device and only the updates of the model reach the server through encrypted communication, still the privacy of the data is under treat with various attacks such as Membership inference attacks- which exploits the vulnerabilities of ML models as well as coordinating servers to retrieve private data [13]. Model memorization, Model Inversion attacks, Inference attacks. Hence Federated learning needs more additional privacy apart from existing ones. In [43] FL is been extended to secure FL including horizontal, vertical and Federated transfer learning. Various research interests on Federated learning since 2017 includes investigating on server aspects for FL, system perspective, personalised models scalability [50], communication efficiency [50] and Privacy Few works in [28] have surveyed the privacy protection and security threats of FL

While training the models across decentralised environments, distributed learning and federated learning are major techniques. While Federated learning is a specific form of distributed learning there are few differences in between these depending on the data sharing , privacy, communication pattern and centralisation. The differences are tabulated in below Table:2.

While Federated learning is an excellent solution for various distributed environments, it may not be always a best solution for using on supply chain collaboration with multiple

Factors	Distributed Learning	Federated Learning
Data Sharing	Data distributed across multiple nodes, they share gradient updates [14]	Data remains at decentralised devices, only model updates shared with centralised server [46]
Privacy	Sharing partial or complete data with other nodes raises privacy concerns [115]	Model developed with privacy concern so all the raw data remains at devices [48]
Communication	Frequent communication is needed to synchronise the model parameters [116]	Periodic updates reducing frequent communication [49]
Use case	Traditional parallel computing environments, data centers [116]	Mobile devices, IoT devices [51]

Table 2: Distributed learning vs Federated learning

parties. Few of the major problems are- While supply chain contains various partners from manufacturers to retailers, they all have different data formats and infrastructure leading to incompatible data. Data quality and formats varies significantly in supply chains, leading to difficulty in achieving reliable and consistent training for model convergence. Encryption and privacy authentications requires complex mechanisms in FL. While FL needs frequent communication, its hard to manage frequent communications between the participating parties in supply chain which leads to having latency issues and can increase operational costs. Supply chain is a complex problem, having to train models in federated learning environment requires significant computational resources and not be feasible for all participants. Thus after careful evaluation of our specific use case, we cannot use Federated learning in ourr model and have explored other privacy preserving techniques.

3.2.2 Federated Clustering

With the distributed environments mainly with supply chain as the clients are widely distributed across the locations, clustering the nodes and then processing it is beneficial. With Federated learning framework being served for preserving the data privacy, clustering with federated learning is achieved with Federated clustering.

Although machine learning models have been so efficient and solved the worlds complex problems. The food for ML model to work appropriately is a huge data, whereas the leakage of data from these models raises a huge privacy risks. For instance the last term from the neural network layer contains the information on the label distribution of the training data. Thus the various inversion attacks will result in obtaining the clients input data [58, 59, 64, 65, 66] Federated clustering is a branch of FL research that focuses on grouping data that is globally

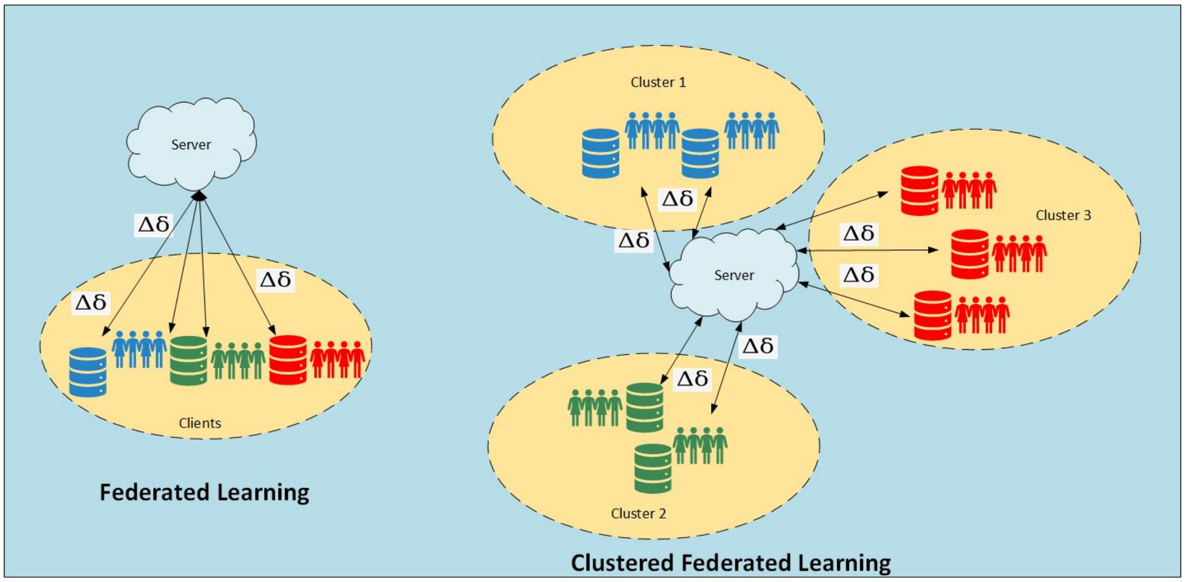


Figure 3: Difference between Federated Learning and Clustered Federated Learning

related while keeping all data local. In supervised FL frameworks, it solves difficulties such as non-independently-identically-distributed (i.i.d.) data. Federated Multi task learning (FMTL) framework, which groups the client population into clusters with jointly trainable data distributions [35], Clustering would be an addendum to FL as it is performed after the convergence obtained by FL. Flexible enough to handle the varying client populations over time and main reason is implemented in preserving privacy. It is applicable to non convex objectives and no changes needed to the FL communication protocol. Few of the literature's where federated clustering have successfully excelled the expectations are discussed below.

In [38], FL where users are distributed and partitioned into clusters. This setup captures settings where different groups of users have their own objectives(learning tasks) but by aggregating their data with others in the same cluster (same learning task), the can leverage the strength in numbers in order to perform more efficient federated learning. As shown in the Fig: 3 For this new framework of Clustered Federated learning (CFL), they have proposed the Iterative Federated Clustering Algorithm(IFCA), which alternately estimates the cluster identities of the users and optimizes model parameters for the user clusters via gradient descent. When the clustering structure is ambiguous, they propose to train the models by combining iterative federated clustering algorithm (IFCA) with the weight sharing technique in multi-task learning

The FL training procedure happens in five steps:

- Client Selection: Choose clients who will take part in the training process.
- Broadcasting: A central server creates a global model and distributes it to clients.

- Client Computation: Each client applies a training protocol to the global model and shares the results with the central server.
- Aggregation: To update the global model, the central server uses an aggregation function.
- Model update: The clients are given access to the updated global model.

Few of the review papers detailing on the federated learning are [76, 77, 78]. While most of the surveys and the literature's focuses mainly on the supervised learning of federated model whereas the unsupervised learning including the federated clustering has not been focused a lot in the research. With the main goal of federated clustering is to group together (local) data-points that are globally similar to each other. That is, data points are dispersed across different clients and grouped using a global similarity metric, while all data remains local on client devices. To our knowledge, there are only a few works that address this issue. As like Federated learning, Clustered federated learning can be easily adapted to other techniques with an encryption mechanism that achieves this end. As both the cosine similarity between two clients' weight-updates and the norms of these updates are invariant to orthonormal transformations. Federated learning could also be expanded to fuzzy c-meas [75, 72]

While Federated clustering similar to Federated learning can allow multiple parties to collaborate in decentralised environments, it does have some limitations to implement in supply chain collaborations. Federated clustering relies on the diverse datasets to identify the clusters and patterns, whereas in supply chain the data is distributed across partners, which may result in incomplete biased clustering results with partial data. As Federated Clustering (FC) operates on the local data, it doesn't create a global view which is challenging to identify patterns and trends in supply chain. As the data is varied in data formats, accuracy and completeness. FC may exacerbate these problems resulting in noisy and unreliable clustering results. While supply chain contains complex and huge data, FC may struggle to handle scalability and complexity leading to sub optimal results. Though federated clustering may have many applications in various scenarios where data privacy and decentralisation's are paramount. It doesn't fit into our model, so we didn't use Federated clustering.

3.2.3 XGBoost Federated Clustering

Federated learning is introduced into Extreme Gradient Boosting (XGBoost) by Yang [162] where the gradient information on each tree node will be communicated frequently from central server to all other tree nodes. While there been few updates on Federated learning still the privacy preserving was challenging. A new privacy preserving machine learning algorithm -XGboost Federated learning was introduced [163] XGBoost known as Extreme Gradient Boosting is an ensemble method actually referring to push the limit of boosted tree algorithms computational resources. A complete story and the evolution of XGBoost can be accessed from [158]. Under the distributed environments of machine learning ,XGBoost is a software which is easily downloaded and installed on the machine [157]. XGBoost environment is exceptionally known for it's speed and performance, it runs ten times faster on single machine than existing popular solutions [161]. XGboost uses gradient boosting decision tree algorithm [159]. According to [160], few of the supported features are

- Regularisation: to avoid over fitting
- Parallelisation: train the model with multiple CPU cores
- Cross-validation: Built in & comes out of box
- Non-linearity: detect and learn from non linear data patterns
- Scalability: process huge data, distributed servers, many programming language

With all the benefits of XGBoost mainly with predictive modeling and optimisation tasks. There are some limitations to be aware while using in supply chain collaboration. Supply chain data can be very noisy, incomplete and can have various errors. In order to ensure the reliability and quality, significant effort is required to clean and preprocess the data. While sharing the sensitive information is hard in supply chain, whereas XGBoost needs an access to entire dataset which is not feasible in collaborations due to compliance issues. Requirement for significant computational resources and data transfer between collaborators leads to communication and latency issues. Thus, XGBoost fails to fit into our model.

3.2.4 Secure Multi Party Computation

With the growth of cooperative computation where most of the sectors or people jointly computing tasks together based on each other private data, maintaining the privacy between the untrusted parties is an issue. SMPC is been developed to save from these situations where multiple parties can jointly collaborate over their personal or private data without sharing their confidentiality with each other by using a cryptographic technique. While for supply chain collaboration where the multiple parties are involved, SMPC can be used to obtain the efficient collaboration with securing the private data [117].

SMPC can be used in various context of supply chain collaboration [118], supply chain includes various entities and data flow between the various entities or parties has to be clear for efficient cooperation. While SMPC can allow these data flow and manage to secure privacy.

Inventory management also includes sharing of data like each other inventory stocks, raw materials needed, safety levels, reorder details or even suppliers information. While doing the inventory with a third party or the competing company, sharing these details could result in problems, hence SMPC would be a good solution. Similarly in demand forecasting, risk assessment, cost sharing process and mainly with supplier collaboration where we don't have to share the quotations or production capacity, buyers involved. All of these crucial data could be secured using the SMPC.

Few of the literature's where the SMPC was applied on the supply chain collaboration are discussed. A secure multi party computation protocols for joint ordering policy in supply chain collaboration between a single supplier and single retailer was developed in [119] where they successfully obtained the collaboration without revealing any of the participants data to each other.The use of SMPC is not just limited in activities of supply chain collaboration, the applicability is been extended to online business collaboration as well, where the confidentiality of the private data is secured while rapidly adapting to changing the business needs which is been demonstrated in [120]. The case study conducted on the actual supply chain collaboration for aeroengine parts manufacturer's using the SecureSCM discussed in [120] has overcome all the critical steps and have maintained the high privacy between the various actors in supply chain. In this case study they have used a protocol based on linear secret sharing which is more efficient than homomorphic encryption.Thus,Using secure multi party computation in supply chain relieves the stress of handling the data breach, trust issues, data privacy while collaborating between the companies by providing the enhanced security.

Implementing SMPC in supply chain collaboration needs addressing few complexities. While supply chain contains most of the sensitive and proprietary information in various stages, ensuring the data remaining private and secure from breaches amd cyber attacks, and only accessible to authorised parties is a significant challenge with SMPC. With widely distributed supply chain parties, integrating the data to enable secure collaboration and determining the data ownership, access rights is challenging. Different regions have different regulatory compliance regarding data sharing, privacy and security. Collaborative efforts must create trust among partners and comply to regulations to avoid legal issues. Implementing SMPC requires technical infrastructure, including encryption, secure communication protocols and expensive which is not possible to implement by small scale supply chains. With all these issues, we are not using SMPC for our model, we are focused on more economic, reliable and scalable privacy preserving methods.

3.2.5 Differential Privacy

As the big data been emerged with the collaboration and sharing of data various privacy preserving techniques also emerged. Adding to the line of privacy preserving techniques, a rigorous mathematical based framework for enhancing the security and privacy of the confidential data while dealing with large amount of personal data or sensitive information collected or shared between various organisations DP could be a better fit [121]. This algorithm basically injects the controlled noise into the data or the results and introduces more inaccuracy while still providing useful aggregate information. It's mainly used in the areas where sensitive data analysis or sharing is required. While differential privacy been used in various sectors of supply chain collaboration [122] in data sharing between the multiple partners, demand forecasting, inventory management, production planning, supplier performance monitoring by annonymizing the performance metrics , facilitating the real time tracking by securing the confidentiality of shipment details in transportation and logistics, and mainly following the rules for compliance and regulations. Few of the works where the DP was applied in the areas of supply chain and obtained satisfactory results are discussed in [123]. During the collaboration when the external parties needs statistical reports of number of products sold in time period or produced, companies need to share product reports or production reports, or releasing details on the raw materials used in beverages as per the government rules. These are the sensitive model parameters and there's a need to preserve these for the system model. A novel mechanism of differential privacy model is proposed in [123] by analysing the situations of supply chain where they have obtained an analytical expression and utility function in their setup using which the differential privacy can be obtained with minimum noise. protecting the dynamic supply chain model by generating the minimum noise required using the differential privacy is also studied in [124].

In summary, while differential privacy is a valuable tool for protecting individual privacy in data analysis, it's application in supply chain collaboration should be carefully considered. DP introduces noise or random perturbations into data to protect individual privacy, which could significantly affect data utility making it challenging and almost a trade off between privacy and accuracy. Sharing the data between the parties in supply chain is important, but implementing DP may hinder the ability to share and analyse the data effectively with the added noise. Implementing DP by itself requires expertise and resources for understanding the parameters and specialised algorithms which is resource intensive in computational and economical. Thus we couldn't use DP in our model.

3.2.6 Conclusion

In the contemporary landscape marked by extensive data sharing and collaborative endeavors, the paramount importance of preserving privacy cannot be overstated. Within the context of supply chain collaboration, we have diligently explored a range of privacy preserving techniques which are aimed at safeguarding sensitive information. While the data privacy remains a critical concern, our research has successfully demonstrated that effective collaboration can be achieved without compromising on privacy through the application of various techniques, including Federated learning, Federated clustering and Extreme Gradient Boosting, Along with the encryption techniques and noise addition techniques like differential privacy preserving and secure multi party collaboration.

While these methods exhibit impressive performance under specific circumstances, it is important to acknowledge their inherent limitations. For instance, FL, while effective in decentralized environments, is best suited for simpler models like neural networks with relatively small parameter sizes. Complex models may encounter challenges when distributed across decentralized networks. Similarly, CFL raises legitimate security concerns, and ensuring synchronization when dealing with dynamic datasets remains a challenging endeavor. XGBoost, although a powerful model, is often considered a ”Black box,” making the interpretation of predictions and optimization computationally intensive and reliant on expert knowledge. Secure Multi-Party Computation introduces the overhead of cryptographic operations, necessitates specialized expertise, and presents challenges in adapting to real-time operations. Furthermore, Differential Privacy, while effective, introduces data-dependent noise into query results, potentially compromising the accuracy of aggregate data.

In light of these challenges and the quest for a cost-effective approach that preserves privacy while enabling optimization without the need for additional infrastructure or a team of experts, our proposed model, the Radius-Sector Information Blur Strategy (discussed in Section 4.1.1), comes to the forefront. This innovative strategy represents a streamlined and pragmatic solution, offering a simplified yet effective approach to achieving optimization while preserving the privacy of sensitive information.

3.3 Optimisation in Supply Chain

Optimisation techniques helps supply chain to streamline all the operations, make right decisions, decreases the costs, and in turn helps in obtaining customer satisfaction. Finding the best possible solutions helps in increase the efficiency and maximize the profits. As supply chain contains various components from manufacture's to delivery drivers. Obtaining optimised solutions at every stage is important. Transporting the goods from the hubs to their final destinations is called Last Mile Delivery. Extensive evolution of logisticians to resolve the last mile delivery problems are discussed in [131], [133]. Being the last leg in supply chain, it incurs the highest transportation cost across all distribution networks [155]. So optimising the last mile deliveries to reduce the overall cost and increase the efficiency is vital. In our Thesis we are dealing with last mile delivery services, where the delivery jobs from hubs to customer location's had to be optimised. While this problem is considered as the NP hard due to the complexity involved and various factors considered. It's a Bilevel problem and needs to be optimised. There has been various researchers trying to solve the problem in an efficient way. We discuss the literature's of the ways we are trying to solve this problem below.

3.3.1 Bilevel Optimisation

It contains 2 levels of mathematical problems called as upper level (Leader) and lower level (Follower), they both are interrelated to each other where one serves as a constraint to the other problem. Bilevel problems (BLP) are a hierarchy of two different optimisation problems. Optimisation of objective function happens independently between two different parties without considering each other's objective function but objective function and it's decision space of two parties get affected depending on the decisions made by leader and follower [20]. Based on the objective function and its constraints, leader will select its first decision and sends it to follower. Based on the leader's choice, follower will compute an optimal solution and sends it back to leader. leader should determine whether his initial selection was feasible or not. Thus, it is an iterative approach between leader and follower to determine the optimal solution for the bilevel problems. BLP are one of the most complex problems to solve due to its non-linear, non convex, non-differentiable [19], discontinuous functionalities, which made the researchers busy to explore the best and efficient methodologies to solve bi-level problems. Though the exploration of this topic was started in 1960s, there was a decade gap with not much focus on to this area. The first formulation of the Bilevel programming was proposed in 1973 by J. Bracken and J. McGill [20]. After 1980s significant research outcomes could be seen. Bilevel problems(BLP) includes linear BLP, Linear Quadratic BLP, Nonlinear BLP.

Enormous research is been carried in Bilevel area from 1980s but still it's been complicated problem on universe to solve, two reasons which makes it so complex are defined in [20]. Firstly, the upper level solution evaluation is complex because it cannot be optimised by its own objective function without depending on the reactions of the lower level problem. Secondly, the interactions between the upper level and the lower level are never continuous at all places nor been convex for the objective function. Even if the whole problem, the objective functions, and the constraints are determined as linear functions, the interaction between the upper and the lower levels still shows the non-convex and discontinuous characteristics. Supply chain consists of various members independently controlling the decision variables, every one makes decision based on their interest. Hence, Bilevel programming problem is used to model the pricing and lot-sizing problems in supply chain [134].

While the BLP are hard to solve, there are many methods in literature's for solving this problem. There were four ways of division of problems discussed in [135]

- Vertex enumeration
- Kuhn-Tucker Conditions
- Fuzzy approach
- Metaheuristics- Genetic algorithms, Simulated annealing

Most of the higher complex non-linear problems are solved using the Metaheuristics methods with no limiting conditions. Few of them are Genetic Algorithms was developed from group of Hejazia and colleagues in [135], few other literatures supporting them are [136], [137]. Tabu search algorithms from [138], [139]. Simulated annealing is been discussed in [68]. Neural network alogrithms approach in [87], [90]

3.3.2 Particle Swarm Optimisation

There are various optimisation problems in mixed-integer quadratic programming which is a NP hard problem, where the size of problem increases gets harder to obtain the exact solution with the direct Portfolio optimisation (PO), thus a greater attention was focused on Metaheuristics/heuristic algorithms. Where the Evolutionary and swarm intelligence (SI) are most outstanding approaches. An extended survey of the various swarm algorithms is been conducted in [140]. The concept of swarm intelligence firstly discussed by Benti and Wang [151]. Swarm intelligence are mimics for the coordination behavior of birds and fish moving in flocks, which will transfer the information in synchronised way across the group to help in decision making process [140]. Few of the swarm based algorithms are Particle swarm optimisation (PSO) which was first introduced by Kennedy and Eberhart in [141], ant Colony optimisation(ACO) [142], bacterial foraging optimisation(BFO) [143], artificial bee colony(ABC) [144], cat swarm optimisation(CSO) [145], firefly algorithm(FA) [149], invasive weed optimisation(IWO) [147], bat algorithm(BA) [148] and fireworks algorithm(FA) [149]. In PSO technique, particles moves in multi dimensional solution space from one position to another seeking to converge on a optimal solution, particles behaviours depends on compromise between group to individual memory [140]. The basic steps are mentioned below:

1. Initial population creation
2. Individuals fitness evaluation
3. For every individual set the individual best position
4. In population for best individual set the global best position
5. For each individual update Velocity
6. For each individual update position
7. Until the stopping criteria is met, keep repeating from Step 2

Among the literature's on using multi objective swarm algorithms, PSO is most preferred. Multi objective models built using the Multi objective PSO have shown significant results over the other evolutionary algorithms [152], [153]. Bilevel problems are included with two different objective and NP hard problems, where PSO had been excelling itself in obtaining the optimal solutions [84] & [86]. While there are many applications in Supply chain management, Pricing problem in supply chain using a PSO is been discussed in [82]. Joint Pricing and lot-sizing using the bilevel programming where the leader is the Manufacturer and Follower is Retailer of supply chain, with an objective of determining the optimal values for replenishment's number,retail and wholesale price using the bilevel PSO based algorithm (BPSO) is discussed in [134]. A bilevel PSO for solving Vehicle routing and location routing problem is been discussed in [132].

3.3.3 Conclusion

Optimising being an important factor in any industry, supply chain collaboration needs optimisation in every sector to maintain the efficient and smooth operations.We are more interested in last mile operations optimisation in supply chain and we had discussed about two different optimisation techniques which would suit for our problem statements. Bilevel and PSO are the advanced techniques which is proven to be mot efficient in supply chain environments as per the various literature's discussed above. Bilevel helps in simultaneously optimising decisions both strategic and operational levels which strikes the balance between the cost minimisation and distance minimisation in supply chain. While PSO being a powerful tool for fine tuning and decision variables, which navigates the vast solution space and find the optimal solutions. The dynamic nature of these algorithms helps in adapting to the situations. These together provides a data driven decision making allowing to align the supply chain strategy with real world dynamics.

3.4 Last Mile Delivery Services in Supply Chain

Last mile is the final leg of delivery process in supply chain management which reaches from the Hubs/distribution centers to the customers. This stage is the most expensive and time consuming among all the supply chain activities which is assumed to reach upto USD 84.72 billion dollars by 2030 for global autonomous last mile delivery market size. Being the biggest industry having the right collaboration between the retailers, manufacturers, technology providers is very crucial to reduce the delivery times, optimise the best routes and obtain the customer satisfaction. The entire cost or efficiency of supply chain can be higher or lower depending upon the last mile service, as it benefits directly in obtaining cost reduction, enhanced customer satisfaction and overall increases the efficiency. Few of the literature's aiming to solve the problems of Last mile delivery logistics are discussed in here. A novel approach in logistic planning involving the decision making with higher customer engagement is been introduced in [8]. Vehicle routing and scheduling problems research was conducted in [10] where they have modeled the problem in 2 stages of mixed integer programming problem. First stage handling with multiple suppliers with varying production speeds, second stage involving multiple vehicles with varying capacity of vehicle and speed. The study was conducted on the short life span products production and delivery in [125] where they tried to solve the integrated production and transportation scheduling problem (PTSP) by simultaneously doing both production and scheduling. Clustering technology have been utilised in many researches to simplify the supply chain network and assist the decision-making. In [40],they proposed a k-means clustering algorithm for reducing the complexity , optimisation factors in supply chain process communication, product variability and Inaccurate forecast. Set of rules to discover the cluster centres of different supply chain levels, inclusive of clients,retailers, distribution centres and producers, to assist the enterprise selections. so as to simplify the delivery chain and manufacturing community. In [41] labored on a okay-approach clustering approach for grouping of state spaces of production community. In [42] applied neural community-based fuzzy clustering to have a look at the deliver chain best management at the same time as thinking about macro variables.

3.4.1 Clustering

Clustering in last mile supply chain helps in grouping delivery destinations that are geographically close to each other. By clustering deliveries based on proximity, delivery routes can be optimized, reducing the overall travel distance and time. This leads to cost savings in terms of fuel, labor, and vehicle maintenance. Instead of sending multiple vehicles to deliver packages to scattered locations, they can send one vehicle to handle deliveries within a cluster.

This reduces the number of vehicles on the road and minimizes congestion. With optimized routes, deliveries can be made more quickly. This is particularly important for last mile delivery services, where timely deliveries are crucial to customer satisfaction. Clustering helps in ensuring that packages reach their destinations faster, which can lead to improved customer retention and loyalty.

We frequently want to find clusters with elements that are similar to those found within the cluster but not to those found outside of it. This goal, which underpins the majority of early clustering algorithms, is to reduce the complexity of a multivariate data set by dividing it into a manageable number of clusters, each with members who are similar to one another. We are often interested in the relationships between the clusters themselves, in addition to splitting enormous data sets into subsets of relevance. From this perspective, a clustering is more than just a collection of interesting or useful clusters in a sample. It is a structure that, when taken as a whole, provides crucial insights into the system under investigation. These structures, which include not just subsets of data but also essential relationships between them. A classical framework where points are assigned to one and only one cluster, this problem is typically formulated as a difficult integer optimization problem where solutions must meet constraints have values restricted to be either 0 or 1. In the richer fuzzy extension, however, the optimization problem is a continuous-variable problem tractable using the powerful tools of continuous mathematical analysis.

The terms ”k-means” and ”c-means” are used to describe a variety of clustering models and techniques. The batch (global) hard c-means (HCM) model and the sequential (local) sHCM model are the most well-documented and commonly used of these. Many publications confuse the two models and algorithms for optimising them by simply referring to them as ”k-means” without specifying which version they are referring to. The key distinction between HCM and sHCM is that the sequential version takes into account local data structure one point at a time, whereas the batch HCM model aims to capture the global relationship between all the vectors in X [73].

Clustering is used in various applications, while clustering in supply chain plays a vital role in improving the efficiency of the process. The complexity of algorithm in the computational is reduced just by downsizing the network through clustering, which helps in obtaining the optimal solutions for the supply chain network [102]. During the last mile delivery, grouping the similar locations together and delivering them will increase the effectiveness and efficiency of the delivery services. Fresh food delivery service from distribution centers to the customer locations using the clustering based routing heuristic(CRH) algorithm having to do repetitive clustering of K-means is used to achieve optimal solution with less computational time [102]. A study conducted in China regarding to decide the best way to fulfil the multiple orders received online from customers using clustering algorithms and product categories is been noted in [156].

Clustering in supply chain plays a vital role and alleviates the problems in route optimisation, resource allocation, faster deliveries, cost savings, reducing environmental impact and obviously increasing customer satisfactions.

3.4.2 K Means Clustering

K means is most popular and simpler clustering method used in machine learning. K-means algorithm identifies the k number of centroids, then assign the data point around it to the closest cluster. K-means is explained in detail in [104]. Let X is a dataset $X = x_i, i = 1, ..., n$ is the set of n points to be clustered. Set of clusters K, $C = c_k, k = 1, ..., K$. The objective is to minimize total inertia, within cluster sum of squared distances, $J = \sum_i \sum_x \|X - C_i\|^2$. where $|X - C_i\|^2$ is the squared Euclidean distance between data point X to Centroid C_k.

- K means keeps assigning each data point to the nearest centroid and updating the cluster centroids by calculating the mean for data points in the clusters, until the convergence is achieved

- It Keeps iterating until no longer it could change the assignments or based on number of iterations specified

K-means clustering is used in various applications in supply chain. Clustering the delivery locations with well separated distances in a hierarchical way is been discussed in [103]

3.4.3 DBSCAN Clustering

DBSCAN stands for Density Based Noisy Application Spatial Clustering. You can find randomly shaped clusters and noisy clusters (i.e. outliers).The basic idea of DB-SCAN is that a point belongs to a cluster if it is close to many points in the cluster.

DB-SCAN is been illustrated with an example in the Fig: 4

There are couple of parameters of DBSCAN shown in Fig: 4 are explained below [85]

- core point: There are at-least minPts number of points (including the point itself) in its surrounding area with radius eps (Red points)

- Border point: A point is a border point if it is reachable from a core point and there are less than minPts number of points within its surrounding area (yellow points)

- Outlier: A point is an outlier if it is not a core point and not reachable from any core points (Blue point-N)

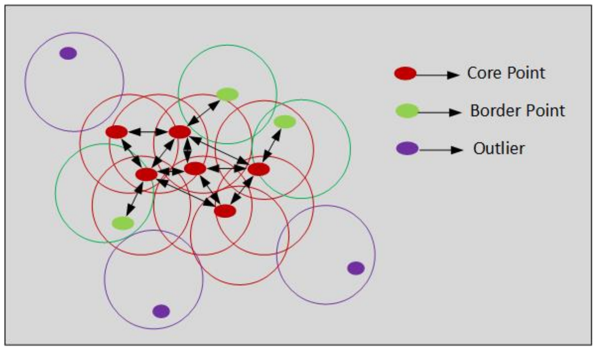


Figure 4: DB SCAN Clustering

Initially minPts and eps are would be chosen. Later using the radius eps neighbourhood area will be chosen from the starting point. Later based on the condition of core points if there are at least min points, the point will be marked as core point if not it will be called noise. once the formation of cluster starts all the points within the neighbourhood of starting point would be in same cluster. Later the next steps the next point would be randomly chosen which is not been visited before and the same process continues. This process will be finished once the entire range of points are visited at least once. distance measurement methods similar to k-means will be used to calculate the distance. Mostly the simple euclidean distance calculation will be used, with all these steps, DB-SCAN algorithm is able to differentiate between the high and low density areas [83]

There's various differences between the K-means and DB-SCAN algorithm is tabulated in Table: 4

3.4.4 Classification

The process of classifying or categorising the data into a set of pre defined classes or categories is called classification.

Depending upon the type of problem the classification is divided into five types [69], [70]

1. Classification Predictive Modeling: Spam filtering is mainly based on this, where we have a training dataset with sufficient examples to train the model and then for any given example, the model will predict the class label

2. Binary Classification: This involves the class with normal and abnormal state, like Cancer detection, If detected-Abnormal, Not detected-Normal. K-nearest neighbor, Decision trees, Logistic Regression, Support Vector Machine, naive Bayes are most popular algorithms used for binary classification.

3. Multi Class Classification: If the task requires more than two class labels like Facial

K-means Clustering	**DB-SCAN Clustering**
Most of the clusters would resemble the shape of either spherical or convex Number of clusters are specified.	Arbitrary clusters are formed and don't have same feature size. Number of clusters need not be specified.
Works efficiently with large datasets	It's not efficient with large dimensional data.
Noisy data, data with outliers doesn't work well using K-means.	DBSCAN clustering efficiently handles outliers and noisy data sets
In the domain of anomaly detection, this algorithm causes problems as anomalous points will be assigned to the same cluster as "normal" data points.	DBSCAN algorithm, on the other hand, locates regions of high density that are separated from one another by regions of low density.
Parameters : Number of clusters (K)	Parameters : Radius(R) and Minimum Points(M)
Clustering doesn't get affected by varying data density.	Clustering does get affected with varying density and sparse dataset.

Table 3: Difference between K-means and DB-SCAN clustering

classification, Animal or Plant species classification. A model with multiple class labels will be defined, and each input will fall into one of the class label. Algorithms like Gradient Boosting, Decision trees, Naive Bayes, K-Nearest neighbour, Random forest can be used for this classification.

4. Multi Label Classification: The tasks involving two or more labels like photo classification, where for each example, one or more class labels may be predicted. Algorithms used for multi label classification are specialised versions - Multi-label Decision Trees, Multi-label Random Forests, Multi-label Gradient Boosting.

5. Imbalanced Classification: If the examples in each class is not equally distributed. Examples like Outlier detection, Medical diagnostic tests and Fraud detection. It needs a specialised techniques used to sample the training data by undersampling the majority and oversampling the minority class. Algorithms used are Cost-sensitive logistic regression, cost-sensitive support vector machines,Cost-sensitive decision trees.

Classification in Supply chain is used in various areas like product categorisation, customer segmentation, demand forecasting, supplier categorisation, Risk assessment, transportation optimisation etc. As my thesis dealing with solving last mile delivery in supply chain collaboration, there are few researchers who have developed a classification techniques in supply chain. In [47] they have used the chemical reaction optimisation for analysing the nature of vehicle scheduling problem, where initially they have used classification to classify the transportation into three nodes, then collaborative scheduling strategy was implemented. Supervised classification algorithms like logistic regression,k-nearest neighbors,naive bayes,support vector machines are used to train the model and obtained the efficient way for supplier assessment problem [97]. A collaborative supply chain model using decision trees and clustering techniques for predicting the model performance was given in [150]

3.4.5 Conclusion

In this section we embarked on the in-depth analysis of the last mile delivery services which are going to help in building our framework below, including the clustering techniques with K-means and DBSCAN, as well as classification methods. Identifying the distinct clusters in the geographical distribution of delivery points can be done with K-means and DBSCAN which helps in obtaining the deliver hot spots, optimise the routing efficiency and delivery preferences on location and order frequency. While classification enables us to provide more precise delivery with the products by estimating the delivery time. Integration of clustering and classification was also important in last mile delivery service which we will be using in the hybrid framework in Section: 5. Clustering results helps in designing the classification model by tailoring the delivery time slots. This personalised approach helps us in personalised delivery service and increase the customer satisfaction. Thus, clustering and classification techniques can revolutionize the route planning, customer segmentation, delivery time predictions, reduce the costs and build the great customer relationships.

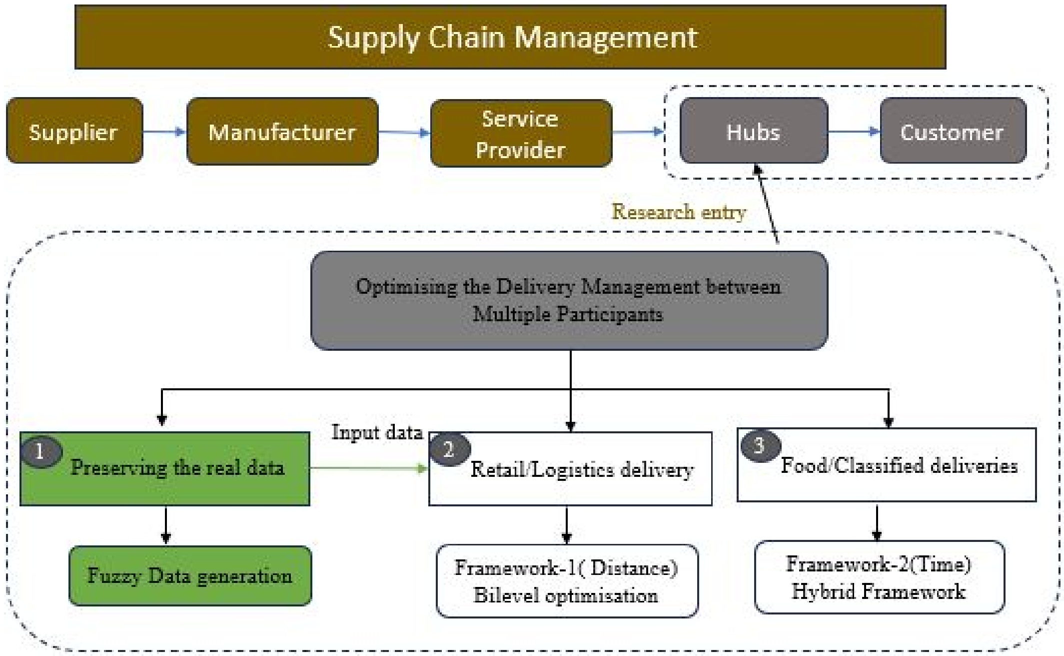
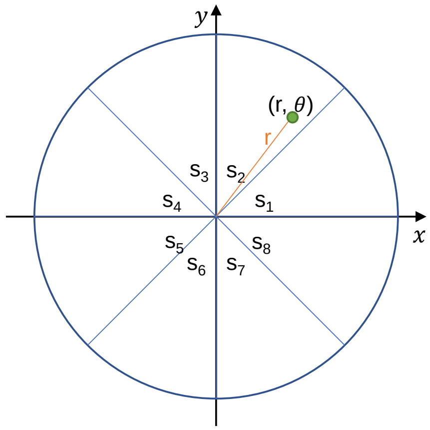


Figure 5: RQ-1 in Research Methodology

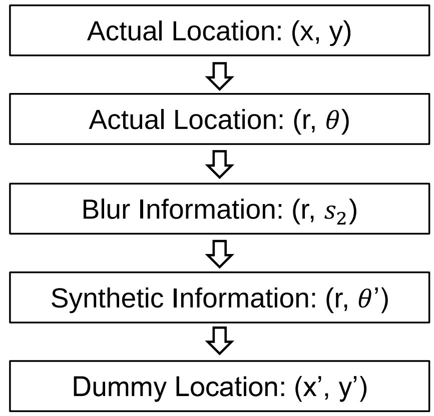
4 Bilevel Framework for Privacy Preserving in Supply chain collaboration

4.1 Preserving the real data in the Optimisation Process

This section we have answered Research Question:1 as shown in the Fig: 5 about how to preserve the privacy of the data in the collaboration of supply chain. In our Thesis, we are solving the supply chain collaboration where data sharing is vital to achieve the collaboration and coordination between the different parties in supply chain. Whereas, this leads to data or privacy breach of the customers data involved or this data could facilitate for knowing the company secrets. Thus, we have developed a novel approach which is cost effective, doesn't need a separate framework. Using our approach we can exchange the data by hiding the real data using the fuzzy data technique. Where the exchange of information can happen by securing the privacy thus the actual data is secured. This is a plug and play a simple method which could be used on any framework, just before exchanging the data we add the fuzziness to the data using our approach and thus the raw data can be secured.



(a) Coordination Conversion and Blur



(b) RS Strategy Data Flow

Figure 6: Radius-Sector Information Blur Strategy

4.1.1 Fuzzy Data Generation

Using this approach we have created a fuzzy data and used that data for the optimisation. In the situation, In order for the effective delivery with shortest time, where the suppliers need to collaborate and has to share their deliveries with the competent companies but they don't want to give them the exact customer locations before they agree to deliver. But without knowing the location details the competent company wouldn't be able to decide whether to accept or reject the request. Using our approach the receiver information of the deliveries, the actual information of the receiver locations are hidden all the way through the optimization process until the final decisions are made. To hide the actual locations, a Radius-Sector(RS) information blur strategy is proposed in Fig:6 shows the concept of how RS strategy works.

R-S converts any latitude-longitude pair into the $r - \theta$ coordination system using the j^{th} participant's hub as the zero point (see Fig:6a). The coordination system is divided into S predefined sectors and thus θ can be mapped into the corresponding sector. $S = 8$ is used in the given example in Fig.6a. When the coordinate is swapped from (r, θ) to (r, s_*), the actual location of the input coordinate is hidden with a fuzzy information because the reverse conversion output can be any point on the arc, whose radius is r in a sector. A randomly generated θ, $\theta \in s_*$ is used to synthesize a virtual location on the given arc for reverse converting the synthetic location back into the original coordination system. Fig.6b gives the data flow of an input latitude-longitude pair going through the R-S strategy and reverse converted into the fuzzy latitude-longitude pair.

4.2 Bilevel Framework using the R-S Strategy

In this section, we have proposed the framework using a Bilevel approach to answer our Research Question:2 as depicted in Fig: 8, where we tried to solve the supply chain collaboration problem in two levels of optimisation.We have detailed about the upper/lower layer optimization. As discussed in the Section: 3.3.1. Bilevel structure imitates the negotiation process between multiple participants. As a bilevel problem each layer has it's own objective and constraints and both the upper and lower simultaneously act together to obtain the optimised results. In the current framework, We are trying to solve multi party collaboration in supply chain with privacy preserving. Where we have the list of participants which are the Hubs having their own delivery jobs. Making a decision to give away the jobs or request the jobs based on the clustering and the shortest distance from the Hubs is a whole challenge we are trying to solve. Thus, by making this negotiation using the bilevel approach and preserving the privacy resulted in obtaining the optimised results which is discussed in below sections.

The upper layer process allows the participant to evaluate the game's big picture and know what situation it is facing. The upper layer optimization process helps the participant to decide which job to give away and which job it should request from other participants. All participants produce a give-away list and a request list. These lists are swapped to other participants along with the fuzzy data. The lower layer optimization process is triggered to decide whether to accept or reject the deliveries on the given lists. The concept of the proposed framework is depicted in Fig:7. In order to preserve the privacy of the customers data, we are not using the real data until the final list is been achieved, thus we have created a Fuzzy data to hide the real data using a R-S strategy in the Section: 4.1.1. This data is used as an input data into the below framework.

4.3 Upper Layer Optimization

The upper layer optimization process aims at creating the self-deliver, the request, and the push-away lists by evaluating the big picture of all existing jobs sent from senders to the participants, e.g. the participated service providers. In practice, the senders decide the initial delivery list because the senders can choose which service provider to use. Let $\mathcal{L}^i_{Init}$ denotes the set of the initial delivery of participant i, the number of service providers in the game is I. For any $\mathcal{L}^i_{Init}$ and $\mathcal{L}^j_{Init}$, where $i \neq j$ and $i, j \in I$, $|\mathcal{L}^i_{Init}|$ can be or may not be equal to $|\mathcal{L}^j_{Init}|$. The set of latitude-longitude (lat-lng) pairs corresponding to $\mathcal{L}^i_{Init}$ is denoted by $\mathcal{X}^i_i$. Since all participants know the number of participants in the game and the hub location of all participants are not a secret, every participant will prepare a customized fuzzy location set ($\mathcal{X}^j_i$) for protecting the actual location of the deliveries in $\mathcal{L}^i_{Init}$. $\mathcal{X}^j_i$ stands for the fuzzy

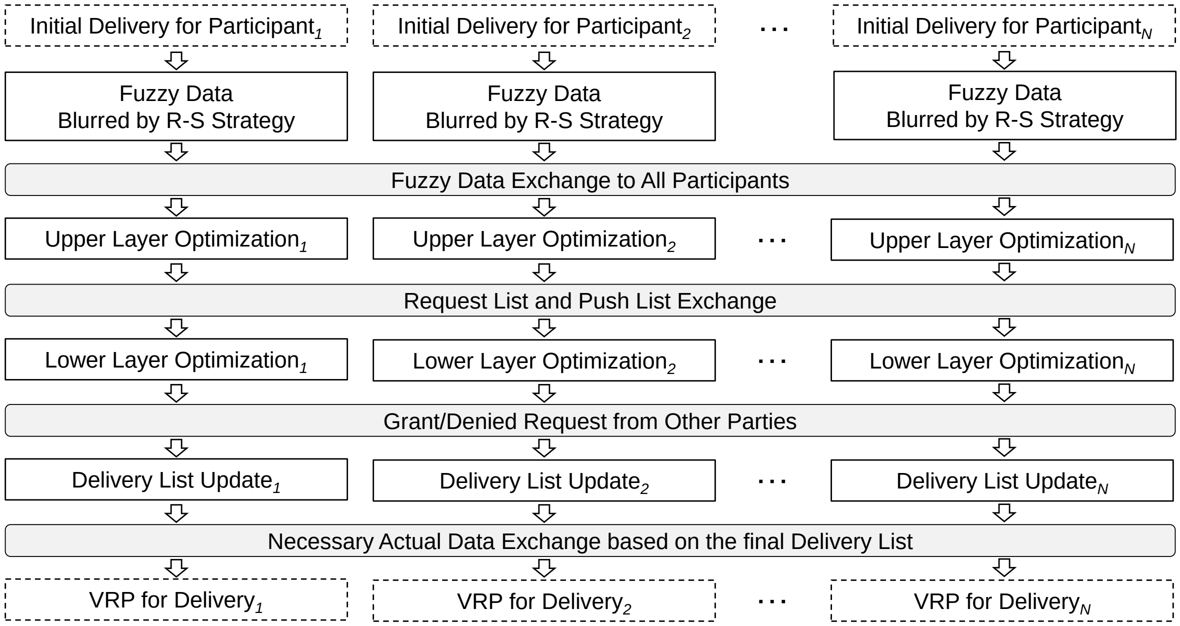


Figure 7: A Bilevel Optimization Framework

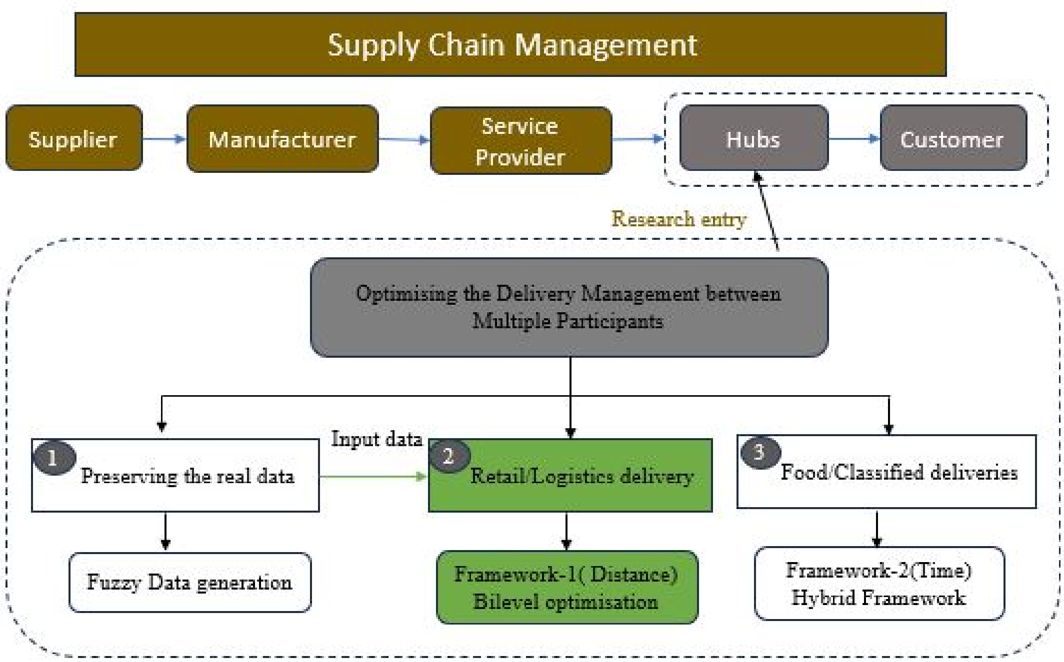


Figure 8: RQ-2 in Research Methodology

Index:	0	1	2	3	4	5	6	7	8	9	10	11
Solution Vector:	0.6	0.3	0.9	1.0	0.4	0.7	0.1	0.2	0.8	0.3	0.7	0.5
Rounded Vector:	1	0	1	1	0	1	0	0	1	0	1	1
	Clusters					Request				Push-away		

Figure 9: An Example Solver Vector in the Upper Layer Optimization

lat-lng pairs corresponding to the elements in $\mathcal{L}^i_{Init}$ customized for the j^{th} participant. The detail of how to generate $\mathcal{X}^j_i$ is given in the Section: 4.1.1. The solvers we choose in this work for clustering is DBSCAN and for optimisation we choose Particle Swarm Optimization (PSO). Depending upon the dataset used or the circumstances, both of these solvers can be replaced by any suitable solver in our proposed framework.

$\mathcal{X}^j_i$ is sent to the other participants for composing the elevated view of all deliveries across all participants. $X_i = \bigcup_{j=1}^{I} \mathcal{X}^j_i$ represents the union of all lat-lng pairs currently held by the i^{th} participant. Since all participants receive customized fuzzy data from others, every participant will not have identical but similar elevated view. Based on the given information, the i^{th} participant performs its own upper layer optimization by taking in X_i and $\mathcal{L}^i_{Init}$ as the input. DBSCAN is used for labelling elements in $\mathcal{X}_i$ into either members in the C^c_i cluster or outliers (O_i), where c in C^c_i indicating the c^{th} cluster. A total of $|C_i|$ clusters are created by DBSCAN for the i^{th} participant.

Based on the DBSCAN output, the candidates of the self-delivery ($\mathcal{L}^i_{Cad}$), the request to j ($\mathcal{R}^j_{Cad_i}$), and the push-away to j ($\mathcal{P}^j_{Cad_i}$) can be created. If an element e in $\mathcal{L}^i_{Init}$ is also included in C_i, it is included into $\mathcal{L}^i_{Cad}$. Otherwise, it is included into $\mathcal{P}^j_{Cad_i}$ according to the $\min^I_{j=1, j\neq i} \mathbf{D}(\mathcal{X}^i_i, H_j)$, where H_j is the hub location of participant j and $\mathbf{D}(\cdot,\cdot)$ is the distance calculation function. On the other hand, an element in C_i is included in $\mathcal{R}^j_{Cad_i}$ if it is not included in $\mathcal{L}^i_{Init}$. Elements in O_i are included in $\mathcal{P}^j_{Cad_i}$ if they are in $\mathcal{L}^i_{Init}$. Elements in O_i but not in $\mathcal{L}^i_{Init}$ are simply ignored.

To compose the solution vectors for the upper layer solver, a coding scheme targeting different representation scales is used. The classical PSO is used in our experiment and thus the solution vector contains floating points and needs to be rounded before decoding the solution. Fig:9 shows an example solution vector containing twelve elements.

As shown in Fig:9, we know that 5 clusters are identified by DBSCAN, 4 and 3 deliveries are put into $\mathcal{R}^j_{Cad_i}$ and $\mathcal{P}^j_{Cad_i}$, respectively. The rounded value "1" in the cluster part of the solution indicates that the participant will take up the deliveries inside that cluster. "1" in the request section of the vector implies that the delivery will be processed by the participant and "0" means otherwise. "1" in the push-away section indicates that the participant will give away the delivery and thus the corresponding delivery will be put into $\mathcal{P}^j_{Cad_i}$.

The fitness function used in the upper layer optimization is given in Eq:(1), which contains components from Eqs:(2)-(5):

$$F_U = \frac{D_{cluster} + D_{request} + D_{push} + D_{h2h}}{NC + \sum_{j=1}^{I} NR_j + \sum_{j=1}^{I} NP_j + 1} \tag{1}$$

$$D_{cluster} = \sum_{c=1}^{NC} WCSS_c + \mathbf{D}(H_i, C_c) \tag{2}$$

$$D_{request} = \sum_{j=1}^{I} \sum_{r=1}^{NRj} \mathbf{D}(H_j, \mathcal{X}_i^{j,r})|_{j \neq i} \tag{3}$$

$$D_{push} = \sum_{j=1}^{I} \sum_{s=1}^{NPj} \mathbf{D}(H_j, \mathcal{X}_i^{j,s})|_{j \neq i} \tag{4}$$

$$D_{h2h} = \sum_{j=1}^{I} \mathcal{D}(H_i, H_j), \text{ if } NR_j > 0 \text{ or } NP_j > 0 \tag{5}$$

where F_U is the upper layer fitness function, $D_{cluster}$, $D_{request}$, D_{push}, and D_{h2h} stand for the accumulated distances for the cluster, the request candidate, the push-away candidate, and hub to hub (hub of participant i to hub of participant j), NC is the number of clusters identified by DBSCAN, NR_j and NP_j are the number of deliveries in the request candidate and the push candidate to participant j, respectively, $WCSS_c$ is the sum of distances between each point to the centroid within cluster c, C_c means the centroid of cluster c, H_i represents the location of the i^{th} participant's hub. $\mathcal{X}_i^{j,r}$ and $\mathcal{X}_i^{j,s}$ are the corresponding location of the fuzzy data for participant j for the request candidate and the push-away candidate lists, respectively.

When the upper layer optimization is completed, the near best solution is used to decode and decide the self-delivery ($\mathcal{L}_U^i$), request ($\mathcal{R}_{U_i}^j$), and push-away ($\mathcal{P}_{U_i}^j$) lists. Let C_c, $\mathcal{V}_*$, and $\mathcal{I}_*$ represent the set of deliveries gathered in cluster c, the set of rounded vector for a particular section, and the set of the solution index for that particular section, respectively. $\mathcal{L}_U^i = \bigcap_{c=1}^{NC} C_c \cap \mathcal{L}_{Init}^i$ when the rounded vector for cluster c is "1", $\mathcal{R}_{U_i}^j = \mathcal{R}_{Cad_i}^j \cap \mathcal{I}_R$ conditional on $\mathcal{V}_R = 1$, and $\mathcal{P}_{U_i}^j = \mathcal{P}_{Cad_i}^j \cap \mathcal{I}_P$ conditional on $\mathcal{V}_P = 0$. R and P represent the "request" and the "push-away" section, respectively. The goal in the upper layer optimization is to minimize the fitness value. $\mathcal{R}_{U_i}^j$ and $\mathcal{P}_{U_i}^j$ are exchanged to participant j along with $\mathcal{X}_i^j$ for the use in the lower layer optimization.

4.4 Lower Layer Optimization

After receiving $\mathcal{R}^i_{U_j}$, $\mathcal{P}^i_{U_j}$ and $\mathcal{X}^i_j$ from other participants, the lower layer optimization process is used to determine whether to accept or denied the requests from the others. To speed up the process, three filters are performed before organizing the lists for composing the solution vector for the optimization. First of all, the push-grant ($\mathcal{P}^j_{G_i}$), the request-grant ($\mathcal{R}^j_{G_i}$), and the request-reject ($\mathcal{R}^j_{R_i}$) lists for participant j are all set to $\emptyset$. Later on, $\mathcal{P}^j_{G_i} = \mathcal{R}^j_{U_i} \cap \mathcal{P}^i_{U_j}, \forall j \neq i$ includes deliveries appear in both the request list from i to j and the push-away list from participant j to i; $\mathcal{R}^j_{G_i} = \mathcal{P}^j_{U_i} \cap \mathcal{R}^i_{U_j}, \forall j \neq i$ includes deliveries contained in both the push list from i to j and the request list from j to i; $\mathcal{R}^j_{R_i} = \bigcup^I_{k=1,k\neq j,k\neq i} (\mathcal{R}^i_{U_j} \cap \mathcal{P}^k_{U_i}), \forall j \neq i$. After the filtering, the combined received push list ($\mathcal{P}_c$) is created by removing all $\mathcal{P}^j_{G_i}$ from $\mathcal{P}^i_{U_j}$ and the combined request list ($\mathcal{R}_c$) is formed by removing all $\mathcal{R}^j_{G_i}$ and $\mathcal{R}^j_{R_i}$ from $\mathcal{R}^i_{U_j}$. $\mathcal{L}^i_U$ is also updated by $\mathcal{L}^i_U = \mathcal{L}^i_U - \mathcal{R}^j_{G_i} \forall j$.

PSO is, again, chosen to be the solver in the lower layer optimization. The same coding scheme used in the upper layer but without the cluster section is used to form the solution vector. Since the lower layer optimization is used to decide whether to accept or reject the requests from other participants, only $\mathcal{P}_c$ and $\mathcal{R}_c$ need to be included in the solution vector. DBSCAN is used inside the fitness function for gathering the deliveries (composed of $\mathcal{L}^i_U$ and deliveries that has "1" in the rounded solution vector) in the clusters. The lower layer fitness function calculates the silhouette score of the identified clusters, which is defined in Eq:(6):

$$F_L = \begin{cases} \frac{2}{C\times(C+1)} \times \sum_{c=1}^{C} \sum_{d=c+1}^{C} \frac{\mathbf{D}(C_c,C_d)-\text{mean}(WCSS_c)}{\max(\text{mean}(WCSS_c),\mathbf{D}(C_c,C_d))}, & \text{if } C > 1 \\ -1, & \text{if } C = 1 \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where C_c and C_d represent the centroids of cluster c and d, respectively, mean($\cdot$) is the mean function, max($\cdot$) stands for the maximum function, $WCSS_c$ is the intra-cluster distance of cluster c, and C is the number of identified clusters.

Our goal is to minimize the fitness value, which implies that we prefer the clusters are highly overlapped, for reducing the total traveling distance when planning the route for deliveries. In the lower layer optimization, the optimizer looks at the density of the delivery distributions and try to identify the high density solutions. After retrieving the near best solution, the solution vector is used to decode the result lists. $\mathcal{L}^i_U$ has already been settled before the optimization process is applied. $\mathcal{R}^j_{G_i}$ and $\mathcal{R}^j_{R_i}$ are updated by $\mathcal{R}^j_{G_i} = \mathcal{R}^i_{U_j} \cap \mathcal{I}_R$ conditional on $\mathcal{V}_R$ is "0" and $\mathcal{R}^j_{R_i} = \mathcal{R}^i_{U_j} \cap \mathcal{I}_R$ conditional on $\mathcal{V}_R$ equals to "1", respectively. $\mathcal{P}^j_{G_i}$ is updated by $\mathcal{P}^j_{G_i} = \mathcal{P}^i_{U_j} \cap \mathcal{V}_P$ conditional on $\mathcal{V}_P$ equals to "1" while the push-reject

$(\mathcal{P}^j_{R_i})$ list is updated by $\mathcal{P}^j_{R_i} = \mathcal{P}^i_{U_j} \cap \mathcal{V}_P$ conditional on $\mathcal{V}_P$ equals to "0".

4.5 Information Exchange for List Update and Solution Deployment

When receiving $\mathcal{R}^i_{G_j}$, $\mathcal{R}^i_{R_j}$, $\mathcal{P}^i_{G_j}$, and $\mathcal{P}^i_{R_j}$ from other participants, we know whether $\mathcal{R}^j_{U_i}$ and $\mathcal{P}^j_{U_i}$ are granted or rejected. This information can be used for updating $\mathcal{L}^i_U$ by Eq:(7):

$$\mathcal{L}^i = \mathcal{L}^i_U \cup \mathcal{R}^i_{G_j} \cup \mathcal{P}^i_{R_j} \cup \mathcal{P}^j_{G_i} \cup (\mathcal{R}^i_{R_j} - \mathcal{R}^k_{G_i}), i \neq j, \text{ and } j \neq k \tag{7}$$

where $\mathcal{L}^i$ is the final decision of the self-delivery list.

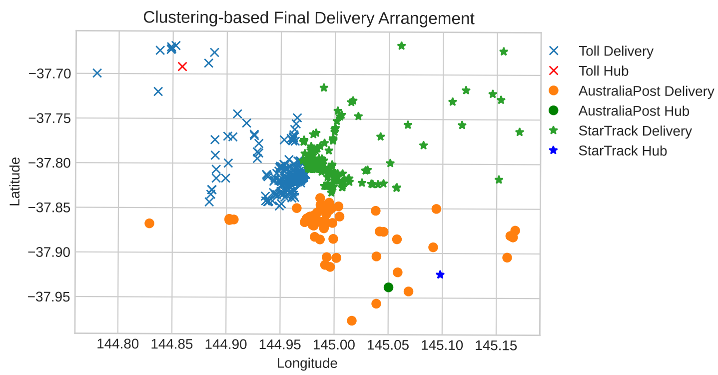
After all participants have updated their $\mathcal{L}^i$, the real lat-lng pairs corresponding to the swapped deliveries will be handed over to the participants. In such a process, the revealing of the real information is minimized to only the necessary deliveries. The related information is not leaked out during the whole process because only the fuzzy data is involved.

4.6 Experimental Evaluations

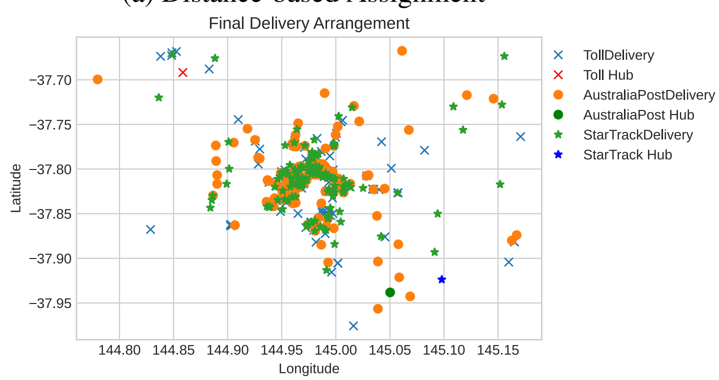
From the proposed fitness functions in our design, it is obvious that we prefer to form the delivery destinations into clusters rather than assigning the deliveries based on the shortest distance between the destination and the service provider's hub. Fig:10 provides an evidence showing that using the shortest distance to assign the deliveries is not always the best. The cluster-based approach we proposed in our design may have a better result.

The black dot lines shown in Fig:10 are the connection of equal distances between points A and B. The number on the solid line is the distance between two points. Fig:10a shows the result of assigning deliveries to the closest hub while Fig:10b presents the result of assigning deliveries based on the centroids to the closest hub. Since the WCSS in a cluster is relatively short and won't change much, it is ignore in the example for simplicity. Thus, the total travelling distances for completing all deliveries in Fig:10a and Fig:10b are $(4+4)+(3+6+5) = 22$ and $(5+5)+[(2+2)+(3+3)] = 20$, respectively. The cluster-based assignment achieves the equilibrium for reducing the overall traveling distance for completing the deliveries. Moreover, using the closest hub for arranging the deliveries is not practical because the initial delivery is decided by the senders, which is not controllable. The chance to get a delivery distribution across the participants to be similar to the shortest distance-based assignment is nearly impossible.

To verify whether the proposed bilevel-optimization framework design can work with the proposed privacy preserving strategy, a simulation is carried out with Yelp_500_PUBS.csv dataset from Pub Hopping in Melbourne project on Kaggle[88]. This dataset contains many



(a) Distance-based Assignment



(b) Cluster-based Assignment

Figure 10: Comparison on Shortest Distance based and Cluster based Strategies

Parameter	Value	Parameter	Value	Parameter	Value
Population	60	Max Velocity	1	Weight Range	$0.9 \rightarrow 0.4$
Max Iteration (Upper)	50	Max Iteration (Lower)	100	c_1 and c_2	2.0

Table 4: Parameter Settings for the Experiments

pub locations in Melbourne, Australia. We are using these locations to be the delivery destinations. Moreover, since our proposed framework is capable of accommodating multiple participants, three delivery service provides including Toll, Australia Post Office, and StarTrack in Melbourne are selected and their warehouses are used as the hub of the participants. The final output of our framework is the customized delivery list for all participants. The produced list can be used as the input for any Vehicle Routing Problem. The WCSS value using the participant's hub as the centroid and its delivery as the in cluster data is adopted to be the evaluation matrix. The sum of WCSS over all participants is used to calculate the total travel distance. The proposed method is compared with k-means with the initial delivery lists as the initial clusters. To eliminate the random effect, the experiments are carried out 5 times and the average result is reported as the final outcome. The great-circle distance is used to calculate the travel distances. The initial delivery allocation to all participants is equally distributed. DB-SCAN parameters Pts and ϵ are set to 2 and one-tenth of the data boundary, respectively. The remaining parameters used in the experiments are listed in Table:4.

Fig.11 reveals the total travel distances obtained by different methods and strategies.

- A-BO stands : Bilevel optimization framework fed with actual data
- RS-BO : Bilevel optimization framework with the RS information blur strategy
- A-CL : Clustering method with the actual data
- RS-CL : Clustering method with the RS information blur strategy
- RAND : Random assignment

The simulation results indicate that our proposed framework reduces the total travel distance about 65.22% comparing to the clustering method while the RS information blur strategy is deployed. The main reason is that the bilevel optimization framework allows different decision makers to negotiate with others and avoid making misjudged decisions through a zero-knowledge proof like process. The differences between RS-BO, A-BO, A-CL, and RAND with fully reveal the actual data in the process are not significant but A-BO

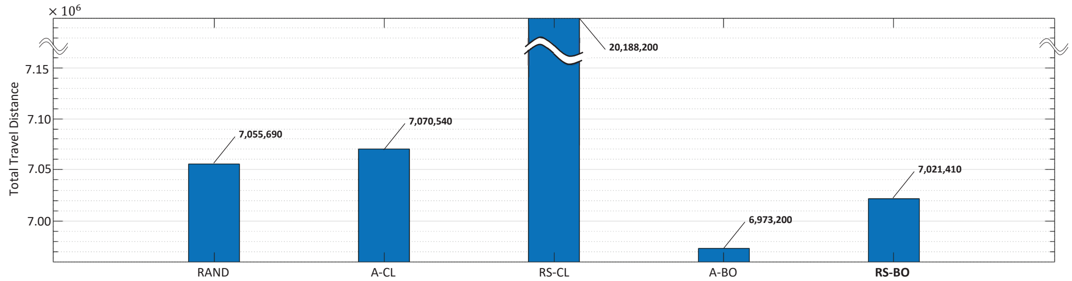
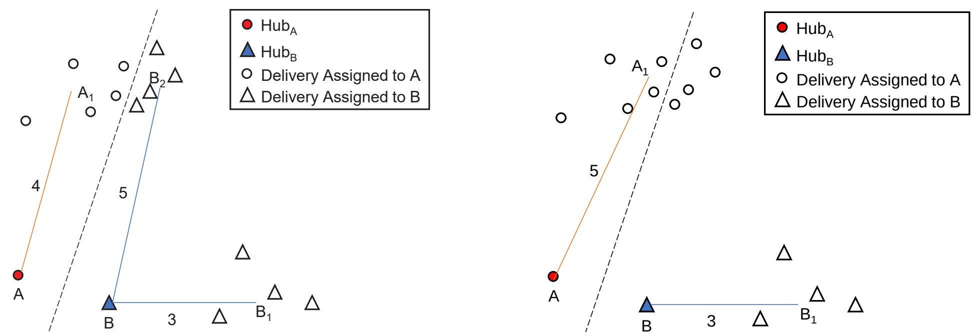


Figure 11: Simulation Result Comparison



(a) Cluster-based Assignment (b) Assignment from Our Method

Figure 12: Sample Delivery Assignment Results

shows the most improvements. RS-BO presents nearly the same result than A-BO with only 0.69% greater distance in total. However, it still outperforms other baselines of the clustering methods. This means that our proposed framework works with the fuzzy data and can still present results close to those obtained by actual data. Other methods such as clustering can be significantly impacted when the data is fuzzy.

Fig.12 reveals the representative delivery arrangement for all participants from our method and the baseline.

From Fig.12a, it is obvious that the result is basically driven by the initial deliveries and thus, the hubs can sometimes not be inside the cluster. Fig.12b is the result from our method. The fitness function used in the lower layer optimization is minimizing the silhouette value and it implies that we are looking for clusters that have overlapping on each other. This is beneficial for designing the delivery route using the Vehicle routing approach solutions.

4.6.1 Vehicle Routing

Vehicle routing involves in determining the optimal route and schedules for the fleet of vehicles to deliver their goods for a set of customers. With a primary objective of reducing the total time taken, transportation costs and maximise the resource utilisation. Vehicle routing problem is been one of the serious issues in supply chain for over a while, Obtaining the

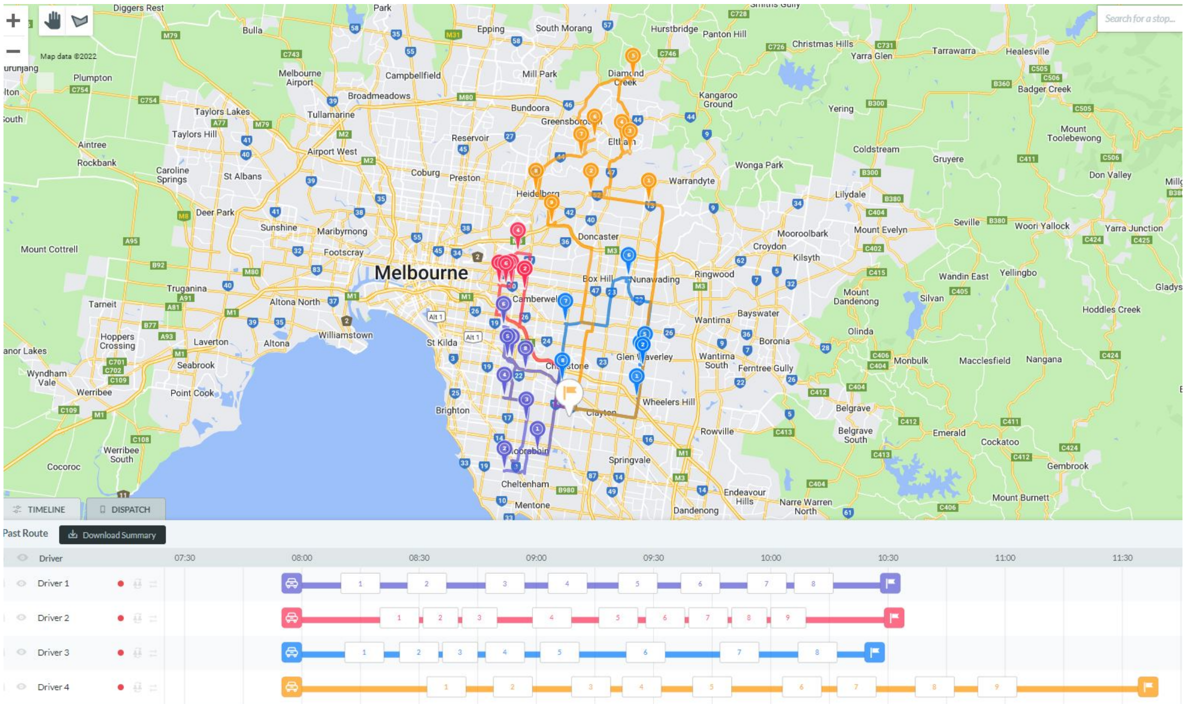


Figure 13: Deliveries assigned on maps

minimum distance and time taken to travel and complete the trips is really important to have a better optimisation. As there are tons of research conducted on Vehicle routing problems and various algorithms are developed to solve the problem. There are exact algorithms, Heuristic, Meta-heuristic and Hybrid algorithms [126].

As a result of Bilevel Framework in Section: 4.2, we obtain a final delivery list after swapping the jobs between the multiple participants. Each delivery list contains a job locations and the Hub details from where the delivery should start. For us to verify the results obtained, we have implemented the final list on the VRP tool called Routific- Route planning software for deliveries [127]. We used the final list obtained for 4 different delivery hubs and did a simulation on the Routific to verify the results. The Fig: 13 demonstrated the visualisation of the same, where every hub has a different colour code to differentiate and Job_IDs are used as the delivery job index.

4.7 Conclusion

Our research has demonstrated the efficacy of Bilevel optimization as a powerful technique for addressing complex problems, particularly those categorized as NP-hard. The intricate nature of supply chain collaboration among multiple parties presents a formidable challenge, and in response, we have developed a robust Bilevel framework. This framework employs a two-tiered optimization approach, encompassing both upper and lower optimization levels, all with the overarching objective of minimizing the total distance traveled in the context of last-mile delivery services. Our framework's performance has yielded favorable outcomes when compared to baseline models, showcasing its potential as an impactful solution in the realm of supply chain logistics. Moreover, we have introduced a pioneering dimension by incorporating privacy-preserving features into our optimization process, aligning with the evolving landscape of data-driven environments. Notably, our approach has achieved a remarkable reduction in the overall distance traveled, achieving a reduction of approximately 65.2%, all while safeguarding the privacy of customer locations.

This research not only contributes to the advancement of optimization techniques but also addresses real-world challenges in supply chain logistics, where efficiency, privacy, and sustainability are paramount. As we look to the future, there is a wealth of opportunity for further refinement and application of our framework, ultimately enhancing the efficiency and effectiveness of last-mile delivery services in a data-conscious world.

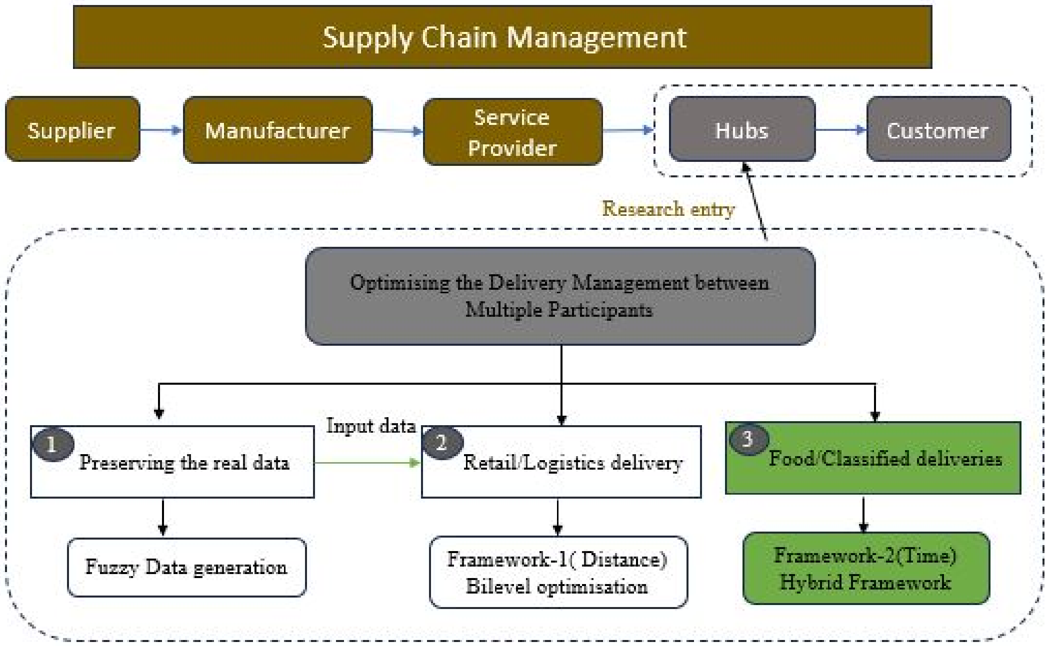


Figure 14: RQ-3 in Research Methodology

5 Hybrid Framework for Optimising the Multi Party Deliveries of Supply Chain Collaboration

In the previous Section:4.2, We developed a novel Bilevel framework tailored for multi-party collaboration, employing advanced Bilevel optimization techniques. This framework operates under the premise that all deliveries fall within a general category, obviating the need for intricate classification into distinct categories. This approach aligns seamlessly with conventional logistics and parcel delivery services, which predominantly handle generic deliveries.

However, the landscape of delivery services is undergoing rapid transformation. Gone are the days when deliveries were primarily confined to parcels and logistics. Today, we encounter a diverse array of products, including everyday essentials like food products, fresh produce, and more. The evolving nature of deliveries necessitates a fundamental shift in last-mile delivery methods to accommodate these dynamic changes effectively. To illustrate this point, consider the scenario of supermarket deliveries. Such deliveries encompass a wide range of items, from food and fresh juice to meat, seafood, frozen goods, pesticides, toilet liquids, and garden tools. The combined delivery of items like frozen food, hot baked chicken, and rodent killer sprays raises critical concerns regarding cross-contamination, hygiene, and temperature control—particularly for food products, which require specialized handling. It becomes evident that delivering all these products together in a single shipment is untenable. The classification of these products before delivery becomes imperative.

In light of these challenges, our research addresses the pressing need for effective delivery methods that optimize routes, minimize travel time, and reduce distances in scenarios where diverse and short-shelf-life products are involved. The significance of this research is underscored by the heightened importance of customer feedback and complaints in the contemporary delivery sector. Tight delivery deadlines, often requiring same-day or even within-hours delivery, necessitate meticulous planning of delivery routes, selection of delivery partners, and the seamless operation of last-mile delivery services. While numerous studies have explored efficient product delivery by minimizing time and distance, there remains a scarcity of research dedicated to the handling of short-shelf-life and categorized products.

Within this context, our framework emerges as a solution to these challenges by advocating for the classification of deliveries before dispatch. In this section, we discuss on our hybrid framework and provide comprehensive insights into its design and functionality, as elucidated in Algorithm:1.

5.1 Data preprocessing

For this Framework, we have obtained the dataset with location details containing a lattitude and longitude to deliver to the destinations. The steps involved in this process is been shown in Fig: 15. The original dataset D_o, Yelp_500_PUBS.csv is downloaded from Kaggle website [88]. Which contained 425 records of Pub locations across Melbourne, Australia. Most of the classifiers, whose goals is to achieve higher accuracy, incline to having the larger samples. In many cases imbalanced data, minority samples are wrongly classified as noise samples [98]. Commonly used ensemble learning algorithms are Bagging [99] and Boosting [100]. Bagging is an ensemble method of a machine learning which is used to increase the size of dataset by creating the multiple subsets of the original dataset through resampling. Whereas, Boosting focuses on different samples iterative by increasing weight for wrongly classified ones by base classifier.

We are using one of the Bagging techniques to expand the size of sample for Yelp_500_-PUBS dataset. We have increased the size by almost 5x times. For this extended dataset(D_N) we also add a Job_ID column with unique ID numbers to identify every job. In order to classify the samples based on the Classification metrics defined in Section: 5.2. We randomly create the numbers defined in range of classification metrics, and add an extra column for the

Algorithm 1 Design of the Hybrid Framework

INPUT: Original Location Dataset D_o
OUTPUT: Total_time taken to deliver the deliveries upon classification

Data Preparation for the model as per the Fig:15
Divide the D_N between the n
Implement the Classification matrix defined in Fig:16
D_A & D_B will be divided into different buckets

# DBSCAN
for bucket in D_A **do**
    calculate the total_time taken from Hub_A using the Algorithm:2
    compute the DBSCAN using Algorithm:3
    obtain Clusters_A and Noise_points_A
end for
for bucket in D_B **do**
    calculate the total_time taken from Hub_B using the Algorithm:2
    compute the DBSCAN using Algorithm:3
    obtain clusters_B and Noise_points_B
end for

#Noise_Handling
$all\_noise = Noise\_points_A + Noise\_points_B$
for noise in all_noise **do**
    for clusters in clusters_A **do**
        calculate the shortest_dist from clusters to noise as per Section: 5.4.1
        assign the noise to closest cluster
    end for
    for clusters in clusters_B **do**
        calculate the shortest_dist from clusters to noise as per Section: 5.4.1
        assign the noise to closest cluster
    end for
end for

# Cluster_Handling
for cluster in clusters_A **do**
    calculate total_distance using Algorithm: 2 for clusters from Hub_A
    calculate total_distance using Algorithm: 2 for clusters from Hub_B
    if (cluster_dist from Hub_A) < (cluster_dist from Hub_B) **then**
        Leave the cluster to Hub_A
    else(cluster_dist from Hub_A) > (cluster_dist from Hub_B)
        Assign the cluster to Hub_B
    end if
end for
We obtain final optimised clusters & total_time taken using Algorithm: 2

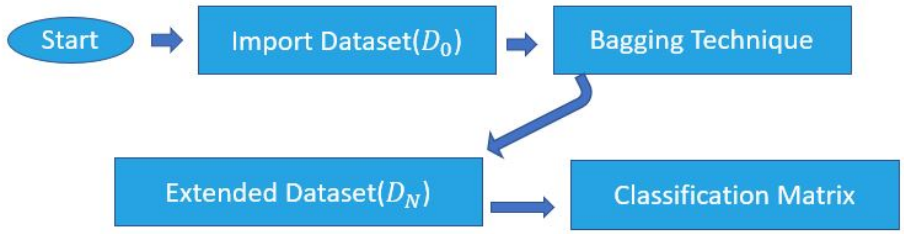


Figure 15: Data preparation for the model

D_N with numbers in $class_{value}$ column.

$$Original\_Dataset, D_o = \{\text{lat, long, name}_{\text{loc}}\} \tag{8}$$

$$Updated\_dataset, D_N = \{Job_{id}, \text{lat, long, name}_{\text{loc}}, \text{class}_{\text{value}}\} \tag{9}$$

We are solving the multi party collaboration, which contains multiple parties involved in single delivery. If n is the number of delivery partners. We will divide the D_N on equal distribution between the delivery partners D_N/n. Assuming every delivery partner would have their hub location from where the delivery starts. For example, If we consider two delivery partners as Partner_A and Partner_B. Their Hub locations would be $Hub_A \& Hub_B$. Dataset for Hub_A and Hub_B would be randomly divided as $D_N/2$ and delivery jobs would be equally shared.

5.2 Classification of the deliveries

As discussed in earlier sections, classifying the deliveries into various buckets makes the delivery very efficient. This classification could be based on various factors. Size, volume, capacity, sensitive environment, temperature controlled etc. Whereas, in order to explain our framework we have used the supermarket delivery products based on their categories and sub categories. We have investigated the supermarkets of Australia product category classification and developed a confusion matrix to use for our framework. The confusion matrix is shown

Class	1	2	3	4	5	6	7	8
1	1	1	1	1	0	0	0	0
2	1	1	1	1	0	0	0	0
3	1	1	1	1	0	0	0	0
4	1	1	1	1	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	1	0	1
7	0	0	0	0	0	0	1	0
8	0	0	0	0	0	1	0	1

Table 5: Confusion Matrix

below in Tab:5. The Class is the $class_{value}$ which we defined in the Section: 5.1.

$$\text{Class-}\left\{\begin{array}{l} 1: \text{Bakery} \\ 2: \text{Dairy, Eggs, Fridge} \\ 3: \text{Drinks} \\ 4: \text{Fruits \& vegetables} \\ 5: \text{Meats \& seafoods} \\ 6: \text{Pantry} \\ 7: \text{Chemicals} \\ 8: \text{Utensils} \end{array}\right\}$$

Once the dataset is been divided equally between the n number of Hubs. We implement the confusion matrix to separate the dataset into various buckets.Based on the rules defined for the classification in Confusion Matrix,we have four different buckets as per the Fig: 16. Depending upon the number of Hubs, each Hub will have four different buckets to be delivered. Each bucket has to be delivered separately and cannot be combined together.

For this framework we have assumed the $n = 2$. It has two different delivery partners. Partner_A and Partner_B having their Hub locations as Hub_A and Hub_B. Thus, the dataset D_N would be split into D_A& D_B. The below equations defines the buckets at each Hub. B_{A1} = Bucket_1 at Hub_A, similarly $B_{A2,..A4}$ = Bucket_2, Bucket_3 and Bucket_4 at Hub_A. For the Dataset D_B containing four different Buckets, they are also classified as B_{B1} = Bucket_1 at Hub_B and so on.

$$D_A = B_{A1} \cup B_{A2} \cup B_{A3} \cup B_{A4} \tag{10}$$

$$D_B = B_{B1} \cup B_{B2} \cup B_{B3} \cup B_{B4} \tag{11}$$

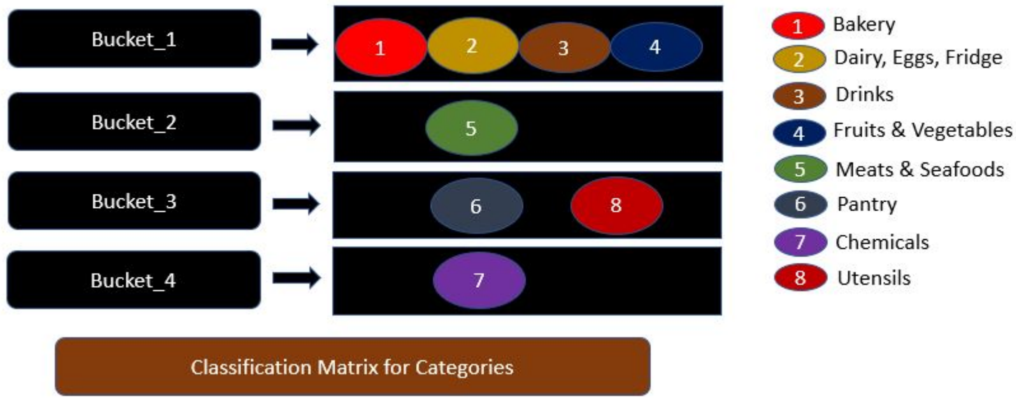


Figure 16: Classification Matrix

Once the classified buckets are obtained.For the current framework,If the buckets at Hub_A then it's initial_bucket_A1,initial_bucket_A2, initial_bucket_A3, initial_bucket_A4. Similarly at Hub_B it is initial_bucket_B1,initial_bucket_B2, initial_bucket_B3,initial_bucket_B4.

We also calculate the total time taken to travel from each corresponding Hub to every bucket and tabulate it using the Algorithm: 2.

5.3 Distance calculation

In order to obtain the shortest time taken for the deliveries, We have used the OpenStreetMaps to calculate the time taken from every point to every other points. In OpenStreetMaps, the distance is calculated using the geographical coordinates using the Haversine formula and similarly the time taken based on the distance. Which calculates the great circle distance between the two points on the Earth's surface given their latitude and longitude coordinates [101].

The extended dataset D_N which has the location coordinates [latitude,longitude] pair is used as an input for the Open street Maps.The Hub locations used for the current dataset are Hub_A = [-37.81530493483055, 144.75121104863356] which is the Australian Toll. Hub_B = [-37.829853434842384, 145.04287303731252] Australian Post location. Using the OpenStreetMap API we have calculated the time taken from Hubs to all the locations in the D_N, and from Every location to all the locations. The entire time_matrix is built with $m * n$ where m,n is the number of location points in the dataset. We have added the Job_ID for the row and column header at index0, to identify each Job in the D_N matches to the exact Job_Id of the distance_matrix. During the time calculation for the framework, we load the pre computed Time_matrix into the setup and do a lookup for the time taken between the points.The sample for the time_matrix is shown in Tab:6

Once we obtain the time_matrix, we could use that in the Time Calculation Algorithm: 2.

Job_ID	*Hub_A*	*Hub_B*	0	1	2	3	.....	1999
Hub_A	0	1751.5	1559.0	1698.8	1231.2	198.5	.....	1098.1
Hub_B	234.1	0	653.1	987.3	124.1	987.3	....	198.5
0	764.1	145.3	0	874.1	543.6	653.9	.....	764.3
1	321.8	487.6	983.1	0	432.5	176.2	.....	874.2
2	432.1	543.2	235.6	654.3	0	654.9	.....	874.2
3	875.4	123.7	542.6	987.4	764.2	0	.....	654.9
.	.	.	.	.	.	.	.....	.
.	.	.	.	.	.	.	.....	.
.	.	.	.	.	.	.	.....	.
.	.	.	.	.	.	.	.....	.
1999	665.2	123.8	987.3	764.3	543.8	154.2	.....	0

Table 6: Time Matrix

We have used the Brute force approach in the algorithm to compute the shortest time where we have explored all possible solutions for all possible paths between the source point to each other points using the Time_matrix which we created in Tab: 6. Then, we compare their time values and find the shortest among them. As we didn't have very large dataset, we were able to obtain the calculations with an average time of 809.8724177seconds to run the framework. Whereas, if the dataset is huge then Brute force would take really longer time and computationally expensive. So, we can replace the Distance Calculation function with other efficient algorithms like Dijkstra's algorithm or the Bellman-Ford algorithm which are used to find the shortest distance effectively.

5.4 Clustering

In this section, we are discussing the clustering of the deliveries in each bucket after the classification of the deliveries into their corresponding buckets as discussed in Section: 5.2. Based on the literature review discussed in Section: 3.4.1. Clustering the delivery locations into smaller groups before delivering increases the effectiveness and thus reduces the time and distance taken to deliver to the customer locations. For last mile logistics with food delivery services, the effectiveness is higher by implementing the Clustering for customer locations [102]. In our Framework, we are using the DBSCAN algorithm for clustering. Clustering is applied on each bucket from every hub to form the clusters and Noises_points. DBSCAN used in this framework is been demonstrated in Algorithm: 3. Upon applying the clustering using this algorithm on each bucket obtained after classification, we obtain their corresponding clusters and noise_points. For example, If Bucket_1 from Hub_A has obtained

Algorithm 2 Time Calculation

Load $time\_matrix_{sub}$ precomputed
for each cluster, jobs in $cluster_{jobs}$ **do**
  shortest_time = ∞
  for i in **range**(len(jobs)) **do**
    current_time = 0
    current_time += $time\_matrix_{sub}$.loc[Hub_x, jobs[i]]
    for j in **range**(i, len(jobs)-1) **do**
      current_time += $time\_matrix_{sub}$.loc[Hub_x, jobs[j], jobs[j + 1]]
    end for
    current_time += $time\_matrix_{sub}$.loc[jobs[-1], Hub_x]
    if $current\_time \leq$ shortest_time **then**
      shortest_time = current_time
    end if
  end for
end for

around three clusters($C_{A1_0}, C_{A1_1}, C_{A1_2}$) and Noise_points($NP_A1$), as it's represented in Equation: 12, Similarly if Bucket_B1 from Hub_B has clusters and Noise_Points, as represented in Equation: 13

$$B_{A1} = (C_{A1_0}, C_{A1_1}, C_{A1_2}, NP_A1[., ., ., .]) \tag{12}$$

$$B_{B1} = (C_{B1_0}, C_{B1_1}, C_{B1_2}, C_{B1_3}, NP_B1[., ., .]) \tag{13}$$

Once the clusters are formed , Noise_points are the delivery jobs which is not in the range of clusters. Now the decision has to be made about how do we deliver to noisy_jobs.

Algorithm 3 DBSCAN

Load the time matrix pre computed
for each bucket in Hub_X **do**
  $jobs = bucket_x[Job_{id}]$
  $unique_{id} = jobs[Job_{id}]$
  Extract the $time\_matrix_{sub}$ for $unique_{id}$
  $myeps = c1$
  $min\_pts = c2$
  $my\_disc = Precomputed$
  $dbscan = DBSCAN(myeps, min\_pts, my\_disc, dist\_matrix_{sub})$
  $clusters = dbscan.fit\_predict(dist\_matrix_{sub})$
  for each cluster in clusters **do**
    get cluster_label, cluster_jobs,noise_points
  end for
end for

5.4.1 Handling the noise

In this section we define the decision making rules for handling the noisy_jobs which are Noise_points(NP_A1). While the clustering happened based on the $myeps$ and min_pts defined in the Algorithm: 3. Certain jobs which are not in the range of this distance and if jobs cannot be merged together into the minimum points, they are left behind as the noise in DBSCAN. Whereas in real world we have to deliver to every location irrespective of they are closer or farther. We have tried to include these jobs into the existing clusters for efficient delivery.

While including the noise points, classification rules defined to divide the dataset into buckets in Fig: 16 should be followed. As per the classification rules, jobs from different buckets cannot be merged together. so possible ways of merging the jobs is between the same bucket clusters from different Hub locations. For example: B_{A1} has clusters $(C_{A1_0}, C_{A1_1}, C_{A1_2})$ and noise NP_{A1}. B_{B1} has clusters (C_{B1_0}, C_{B1_1}) and noise NP_{B1} can be merged together. similarly clusters and noises from (B_{A2}, B_{B2}) , (B_{A3}, B_{B3}), (B_{A4}, B_{B4}) can be merged together.

- Merge all the noise_points together into one list from the possible combination as explained above. $NP = [NP_\mathbf{A} + NP_\mathbf{B}]$
- Using the defined time_calculation in Algorithm: 2. Calculate the time taken from each Noise_Point in the list to every point in the all the clusters of Hub_A of allowed buckets.
- Repeat the Step:2 with all the clusters of Hub_B.
- Compare the minimum time taken obtained for the Noise_Point from both the Hubs clusters and assign to the closest cluster.
- Record the swapped jobs and assigned points.
- At this state, we don't have any more unassigned Noise_points. We have the clusters with all the delivery jobs included.
- For the total time travelled, we also should include the time taken to travel in between the Hub_A and Hub_B, to exchange the items of the exchanged jobs.
- We calculate the total distance for all the buckets and it's clusters of Hub_A from Hub_A and Hub_B, similarly we do the same for buckets and it's clusters of Hub_B.
- Upon comparing the time taken between both. If cluster_0 of bucket_A1 from Hub_A (C_{A1_0} which originally needs to be delivered from Hub_A has the time calculated from Hub_B is minimum. Then we assign the (C_{A1_0} to bucket_B1 from Hub_B.

- We need to repeat the same for every clusters and thus we finally obtain the final clusters after all the jobs swaps.

- Now we can send out the actual delivery from corresponding Hubs for their clusters and tabulate the total time taken, including the twice the distance between the Hubs travelled to exchange their jobs.

$$Delivery\_time = Total\_time\_taken + 2 * (time\_taken\_between\_Hubs) \tag{14}$$

5.5 Experimental results

In order to evaluate our Framework, We have used the Pub dataset Yelp_500_PUBS.csv, downloaded from Kaggle website[88]. The data is been processed and extended using the bagging techniques for the efficient results and obtained the new dataset called Yelp_500_-PUBS1_ex2000.csv. The Pre_computed_time matrix which has the time taken to deliver from one point is built based on the new dataset using the OpenStreetMap API. The DBSCAN parameters used are tabulated below.

The Framework was written using language Python version 3.1. The experiments were conducted on the Windows 64bit operating system, 8GB RAM, Intel(R) i5 processor.

$$\text{Parameters\_DBSCAN-} \begin{Bmatrix} \epsilon(c1) : 600 \\ Min\_Pts(c2) : 4 \\ Distance : Pre\_computed \\ Input : Each\_Bucket \\ Output : Clusters, Noise\_points \end{Bmatrix}$$

The experiment was conducted using two different delivery partners, Partner_A & Partner_-B and the experiment was repeated at ten different rounds by randomly separating the data between them with a random seed. For the comparison of the results, we have tabulated the total time taken results before the clustering and after clustering with assigning noises into two different tables. In Tab: 7 which shows the initial delivery time taken for each Partner to deliver the different buckets, As part of this experiment each Partner would have four different buckets as shown in Eq: 15,& Eq: 16. Where $time\_B\_A1$ refers to time taken to deliver all the jobs in the Bucket_A1, correspondingly the time taken to deliver all the Buckets $A1, A2, A3, A4$ are calculated and summed up together to calculate total time taken for both Partner_A and Partner_B. The Initial_total_time refers to sum of Partner_A & Partner_B as represented in Eq: 17.

In Tab: 8 which shows the final delivery time taken for each Partner to deliver the different buckets, The final time taken is calculated after the clustering and assigning of the noises to the corresponding clusters, the calculation of time is explained in Algorithm: 2 and noise assignment is explained in Section:5.4.1. Final delivery time for Partner_A would be calculated by summing up all the time taken by individual buckets of Partner_A as per the Eq: 15 and similarly for Partner_B in Eq: 16. The Final_total_time would be the sum of Partner_A time taken and Partner_B time taken as per the Eq: 17.

$$Partner\_A = time\_B\_A1 + time\_B\_A2 + time\_B\_A3 + time\_B\_A4 \tag{15}$$

$$Partner\_B = time\_B\_B1 + time\_B\_B2 + time\_B\_B3 + time\_B\_B4 \tag{16}$$

$$Initial\_delivery\_time = Partner\_A + Partner\_B \tag{17}$$

Initial Delivery Time		
Partner_A	Partner_B	Initial_total_time
12,948.70	8,405.00	21,353.70
12,100.20	9,735.00	21,835.20
14,211.30	7,324.60	21,535.90
14,123.10	7,890.60	22,013.70
13,009.00	9,301.90	22,310.90
11,111.70	9,247.20	20,358.90
13,011.40	9,612.60	22,624.00
13,885.60	9,611.60	23,497.20
11,257.70	10,863.00	22,120.70
13,974.40	9,423.00	23,397.40

Table 7: Results - Initial time taken for delivery

The Table: 9 shows the complete results for the 10 different rounds on using the different random seeds from 40-49. The corresponding processing time is also tabulated. The initial delivery time shows the Hub_A and Hub_B which is the sum of all the buckets from Hub_A and Hub_B respectively, Initial_time is the total of Hub_A and Hub_B. Similarly with the Final delivery time, Partner_A and Partner_B includes the sum of their individual buckets. Final_Time is the sum of Partner_A and Partner_B. Delivery_Time is the Final_Time with the time taken to travel between the Hub_A to Hub_B and Hub_B to Hub_A as per the Eq: 14. This is mainly for the exchanged jobs between the Hubs, so one travel from the Hub to deliver their jobs to other and in the same way accept the jobs which needs to be taken from them.

Final Delivery Time		
Partner_A	Partner_B	Final_total_time
2,524.70	9,662.90	12,187.60
5,480.60	10,557.20	16,037.80
2,249.00	7,324.60	9,573.60
959.10	9,620.20	10,579.30
835.80	9,976.50	10,812.30
5,061.00	9,629.50	14,690.50
5,061.00	12,311.30	17,372.30
8,959.30	9,197.90	18,157.20
2,523.40	9,153.30	11,676.70
6,005.40	8,424.80	14,430.20

Table 8: Results - Final time taken for delivery

5.6 Baseline comparisons

The Fig: 17 acts as a baseline comparison to demonstrate the difference between the initial_-time taken to do the deliveries before the clustering of delivery locations and after the complete implementation of our model to do the clustering of the locations and assigning the deliveries between the closest hubs, the final_time obtained. In the Fig:17 the graph is drawn for over ten different rounds of running the experiment with random values where the initial deliveries were equally divided between both the hubs. The difference of final_time been reduced in almost every round of experiment is evident. Thus we were able to reduce the total time taken to do the delivery upon classifying the deliveries by almost 24% compared to the initial_time.

5.6.1 Visualisation

For the visualisation of the results, we have plotted a Bucket_4 from Hub_B, Bucket_B4_initial jobs and Bucket_B4_Final jobs on the Goggle maps. Initially Bucket_B4 had 117 delivery jobs to be delivered from Hub_B, but after finishing all the steps in the framework and exchanging jobs, Bucket_B4 has 249 jobs close to Hub_B thus the jobs have been exchanged from Hub_A to Hub_B in the Bucket_4. The little home icon in the map represents the Hub and all the location icons represents the delivery job locations. In Fig: 18 the location icons in pink represents the initial delivery jobs from Hub_B which was destined to be delivered before exchanging the jobs. In Fig: 19 the location icons in purple are the final delivery jobs to be delivered from Hub_B. By looking at the visualisation on the maps, it's practically true for the delivery locations close to the Hub_B to be delivered from Hub_B which saves the total distance and time and in turn increases the customer satisfaction results. Thus our

Table 9: Complete Results

# No_Runs	Seed	Processing_time	Initial delivery Time			Final delivery Time			
			Hub_A	Hub_B	Initial_Time	Partner_A	Partner_B	Final_Time	Delivery_Time
1	40	788.95	12,948.70	8,405.00	21,353.70	2,524.70	9,662.90	12,187.60	15,738.20
2	41	798.63	12,100.20	9,735.00	21,835.20	5,480.60	10,557.20	16,037.80	19,588.40
3	42	799.54	14,211.30	7,324.60	21,535.90	2,249.00	7,324.60	9,573.60	13,124.20
4	43	946.58	14,123.10	7,890.60	22,013.70	959.10	9,620.20	10,579.30	14,129.90
5	44	978.23	13,009.00	9,301.90	22,310.90	835.80	9,976.50	10,812.30	14,362.90
6	45	697.52	11,111.70	9,247.20	20,358.90	5,061.00	9,629.50	14,690.50	18,241.10
7	46	793.17	13,011.40	9,612.60	22,624.00	5,061.00	12,311.30	17,372.30	20,922.90
8	47	783.25	13,885.60	9,611.60	23,497.20	8,959.30	9,197.90	18,157.20	21,707.80
9	48	784.65	11,257.70	10,863.00	22,120.70	2,523.40	9,153.30	11,676.70	15,227.30
10	49	728.20	13,974.40	9,423.00	23,397.40	6,005.40	8,424.80	14,430.20	17,980.80

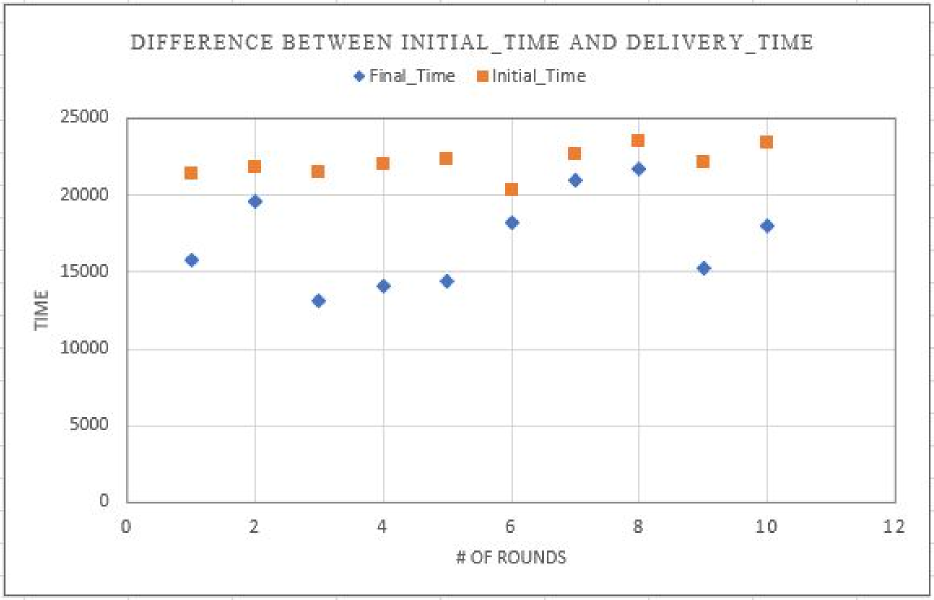


Figure 17: Benchmark

model have obtained the satisfactory results in classifying and clustering the jobs based on the classification matrix and the delivery time.

5.7 Synthesis of Results

In the experimental evaluations of our framework, it became apparent that effectiveness and reliability of our solution were closely tied to the quantity and diversity of the dataset. The framework performance benefits from a more extensive and varied dataset, which enables it to explore a broader solution space effectively. To provide the initial assessment, we conducted experiments using a relatively modest dataset with 400 records. The dataset served as a foundational basis for our experimental setup, and a distance matrix was constructed accordingly. The results of these initial experiments are detailed in Tab:10.

Upon closer examination of the results, mainly by comparing between the "Initial_TOTAL" and "Final_TOTAL" columns in Tab:11. We have observed that the differences between the initial and final were not consistently in favor of the model. In some instances, the model's performance did not align with our expectations, leading to suboptimal outcomes. Recognising that the limited dataset size may have influenced these results, we decided to address this limitation by significantly increasing the dataset sample size 2000 datapoints. Subsequently, we have conducted an extensive round of experiments with this expanded dataset, yielding the outcomes presented in the Tab:9. The results from this larger dataset exhibited a marked improvement in the framework's performance. Notably we observed

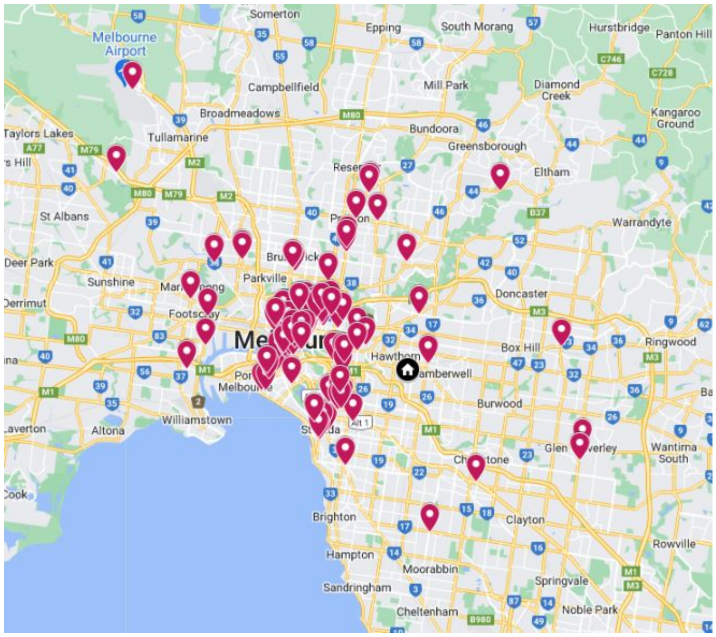


Figure 18: Initial Delivery jobs for Bucket_B4

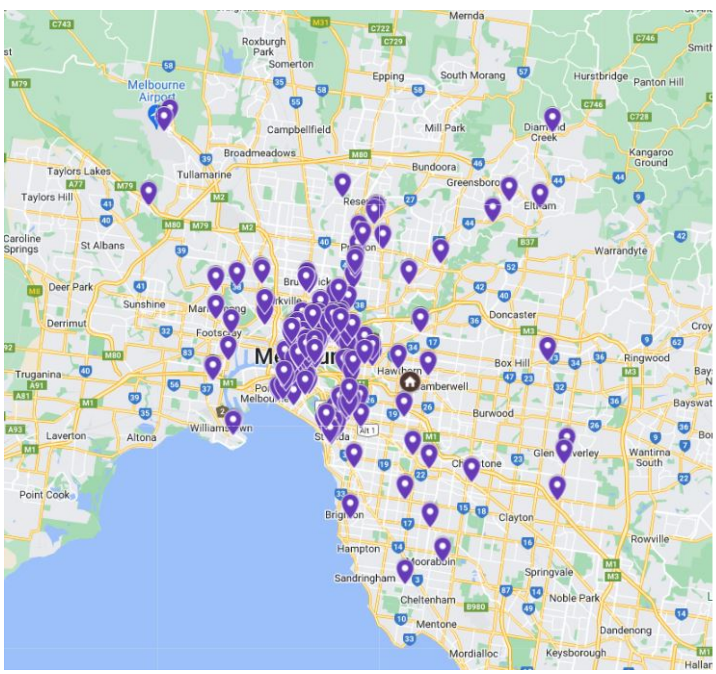


Figure 19: Final Delivery jobs for Bucket_B4

the substantial reduction in the time required to traverse clusters and complete the assigned delivery tasks, significantly the framework also enhanced the efficiency and effectiveness.

This transition from smaller dataset to the more substantial and diverse dataset underscores the significance of data quality and quantity in the context of our research. It highlights the framework's adaptability and scalability to accommodate the broader range of scenarios and challenges. The experiments conducted with the expanded dataset reaffirm the framework's potential and lay the groundwork for future refinement and optimisation. The experimental evaluations emphasize the critical role of data in shaping the performance of our framework. By increasing the shape of dataset size and diversity, we were able to achieve more satisfactory results and enhance the frameworks overall efficiency. These findings provide valuable insights for the continued development and application of our solution, positioning it as a robust and adaptable tool for addressing real world delivery optimisation challenges.

5.8 Conclusion

Our research has addressed the real challenge of last mile delivery service, particularly delivering short-life and diverse supermarket products. The products which require separate handling due to the varying delivery constraints, represent a distinct set of challenges that cannot be seamlessly integrated into traditional logistics delivery models.

In response to this challenge, our hybrid framework leverages the classification and clustering techniques. This innovative approach equips last-mile delivery services with tools require to effectively manage and optimise the delivery of diverse and time sensitive products. Our framework designated to enhance the customer satisfaction by ensuring the timely and efficient delivery of products.Through rigorous testing and experimentation using two distinct datasets, we have gained valuable insights into the framework's performance. It is evident from our findings that the framework's effectiveness is contingent on the diversity and variety of the dataset it operates on. While our initial experiments demonstrated promising results, we recognize the need for a more extensive and diverse dataset to further validate and refine the framework's capabilities.

Notably, our experimental evaluations have shown that the utilization of our framework leads to a substantial reduction in the total travel time by up to 24% compared to scenarios where our model is not employed. This outcome underscores the practical utility and tangible benefits of our solution in improving the efficiency and effectiveness of last-mile delivery operations. Thus, our research represents a significant step forward in addressing the unique challenges posed by the delivery of short life and diverse products.

Table 10: Simulation with 400 datapoints

# No_Runs	Seed	Processing_time	Initial delivery			Final delivery		
			Hub_A	Hub_B	Initial_TOTAL	Partner_A	Partner_B	Final_TOTAL
1	40	47.59	182,554.09	60,014.80	242,568.89	48,533.40	58,195.40	188,032.70
2	41	62.77	188,327.50	53,778.10	242,105.60	17,810.80	67,092.50	166,207.20
3	42	57.29	209,157.30	61,764.70	270,922.00	43,973.10	44,012.30	169,289.30
4	43	41.59	166,501.20	61,985.90	228,487.10	15,854.10	95,129.50	192,287.50
5	44	44.69	192,094.60	55,309.60	247,404.20	14,674.20	75,599.60	171,577.70
6	45	49.64	172,564.10	65,425.70	237,989.80	44,836.50	82,675.80	208,816.20
7	46	43.14	151,779.20	59,152.50	210,931.70	38,956.80	85,567.50	205,828.20
8	47	42.44	109,774.60	48,394.70	158,169.30	40,972.60	56,634.70	178,911.20
9	48	40.78	172,423.90	47,777.90	220,201.80	27,634.90	69,723.00	178,661.80
10	49	40.62	178,512.79	68,101.10	246,613.89	0.00	111,684.70	192,988.60

Table 11: Comparison of Initial and Final Totals

Initial_TOTAL	Final_TOTAL	Difference
242,568.89	188,032.70	54,536.19
242,105.60	166,207.20	75,898.40
270,922.00	169,289.30	101,632.70
228,487.10	192,287.50	36,199.60
247,404.20	171,577.70	75,826.50
237,989.80	208,816.20	29,173.60
210,931.70	205,828.20	5,103.50
158,169.30	178,911.20	-20,741.90
220,201.80	178,661.80	41,540.00
246,613.89	192,988.60	53,625.29

6 Publications List

As part of my research, i could contribute to three papers as tabulated in Table: 12. My first Bilevel framework contributing to my first paper. My second Hybrid framework contributing to my second paper. My literature review would contribute to my review paper.

Sl.No	Paper Title	**Publication**	**Status**
1	Bilevel Framework for Privacy Preserving in Supply Chain	PAKDD 2024	Ready-to-submit
2	Hybrid Framework for Optimizing Multi-Party Deliveries in Supply Chain Collaboration	Complex & Intelligent Systems	Ready-to-submit
3	Last Mile Delivery Services: Optimization and Privacy Preservation - A Review	Advanced Data Mining Techniques	Ready-to-submit

Table 12: Publication Status of Research Papers

7 Conclusion

This study has undertaken a multifaceted exploration aimed at addressing critical challenges in the optimization of last-mile delivery services within collaborative supply chain environments. Three distinct research questions guided this investigation, each contributing to the overarching goal of improving supply chain operations, enhancing privacy preservation, and optimizing delivery processes. In response to the first research question, a novel method called Radius Sector strategy has been developed to conceal sensitive real data during the optimization process. This method stands as a testament to the fusion of privacy preservation and optimization, ensuring the confidentiality of critical information while maintaining the efficacy of optimization algorithms. Its emergence represents a noteworthy contribution to the intersection of optimization and data privacy. The second research question sought to establish a comprehensive framework for optimizing last-mile delivery services in multi-party collaboration scenarios within the supply chain. We have developed a Bilevel framework using bilevel optimisation using the Radius Sector strategy to hide the real data. This framework has successfully reduced the overall distance travelled by 65.2% compared to other baselines while maintaining the privacy preserving. This framework can harness diverse optimization techniques and collaborative strategies to bolster the efficiency and effectiveness of last-mile deliveries. The findings underscore its potential to revolutionize supply chain operations, reduce costs, and elevate customer satisfaction levels. This collaborative framework promises to be instrumental in reshaping the logistics and supply chain landscape. Turning to the third research question, the study pioneered a hybrid framework that combines clustering and classification techniques to address category limitations in last-mile delivery optimization. This innovative approach has demonstrated its adaptability and resilience, effectively overcoming category constraints and demonstrated reduction in 24% of total travel time, while advancing the precision and efficiency of last-mile deliveries. It signifies a crucial step forward in the realm of hybrid methodologies, particularly in the face of intricate operational challenges within logistics and supply chain contexts. Collectively, these research endeavors culminate in a holistic perspective on the optimization of last-mile delivery services, striking a harmonious balance between data privacy, collaborative excellence, and adaptability in the face of constraints. The amalgamation of these contributions not only advances the academic discourse but also offers tangible benefits to industry stakeholders, positioning them to navigate the complexities of contemporary supply chain landscapes with heightened efficiency, security, and operational finesse. As such, this research represents a noteworthy stride toward the ongoing refinement and optimization of logistics and supply chain practices.

7.1 Future Works

Based on the comprehensive research presented in this Thesis, several promising avenues for future explorations and developments emerge. Future research can focus on refining and enhancing the frameworks we have built by using more advanced optimisation algorithms and clustering algorithms and larger datasets with more diverse data. Additional datasets exploration would solidify its role in the evolving landscape of last mile delivery. The implementation of R-S strategy can be extended to various other frameworks and applications which can utilise datasets containing latitude ad longitude information. Which would facilitate the adoption of privacy preserving technique in broader range of applications. In conclusion, the work in this thesis serves as a solid foundation for future research endeavors.

8 References

References

[1] Chang, Y.C. and Lee, C.Y., 2004. Machine scheduling with job delivery coordination. European Journal of Operational Research, 158(2), pp.470-487.

[2] Dablanc, L., 2007. Goods transport in large European cities: Difficult to organize, difficult to modernize. Transportation Research Part A: Policy and Practice, 41(3), pp.280-285.

[3] Quak, H., 2008. Sustainability of urban freight transport: Retail distribution and local regulations in cities (No. EPS-2008-124-LIS).

[4] Arvidsson, N., 2013. The milk run revisited: A load factor paradox with economic and environmental implications for urban freight transport. Transportation Research Part A: Policy and Practice, 51, pp.56-62.

[5] Fernie, J., Sparks, L. and McKinnon, A.C., 2010. Retail logistics in the UK: past, present and future. International Journal of Retail & Distribution Management, 38(11/12), pp.894-914.

[6] Kin, B., Verlinde, S. and Macharis, C., 2017. Sustainable urban freight transport in megacities in emerging markets. Sustainable cities and society, 32, pp.31-41.

[7] Jiang, W.E.I. and Siddiqui, S., 2020. Hyper-parameter optimization for support vector machines using stochastic gradient descent and dual coordinate descent. EURO Journal on Computational Optimization, 8(1), pp.85-101.

[8] McFarlane, D., Giannikas, V. and Lu, W., 2016. Intelligent logistics: Involving the customer. Computers in Industry, 81, pp.105-115.

[9] Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J. and Zhang, W., 2023. A survey on federated learning: challenges and applications. International Journal of Machine Learning and Cybernetics, 14(2), pp.513-535.

[10] Zegordi, S.H., Abadi, I.K. and Nia, M.B., 2010. A novel genetic algorithm for solving production and transportation scheduling in a two-stage supply chain. Computers & industrial engineering, 58(3), pp.373-381.

[11] Thomas, D.J. and Griffin, P.M., 1996. Coordinated supply chain management. European journal of operational research, 94(1), pp.1-15.

[12] Dempe, S., 2018. Bilevel optimization: theory, algorithms and applications (Vol. 3). Freiberg, Germany: TU Bergakademie Freiberg, Fakultät für Mathematik und Informatik.

[13] Chamikara, M.A.P., Bertok, P., Khalil, I., Liu, D. and Camtepe, S., 2021. Privacy preserving distributed machine learning with federated learning. Computer Communications, 171, pp.112-125.

[14] Konecny, J., McMahan, B. and Ramage, D., 2015. Federated optimization: Distributed optimization beyond the datacenter. arXiv preprint arXiv:1511.03575.

[15] Liang,Liang, Y., Mao, C. and Bao, X., 2020, February. Online Variant of Parcel Allocation in Last-Mile Delivery. In 2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) (pp. 900-904). IEEE.

[16] Praet, S. and Martens, D., 2020. Efficient parcel delivery by predicting customers' locations. Decision Sciences, 51(5), pp.1202-1231.

[17] Perboli, G., Rosano, M., Saint-Guillain, M. and Rizzo, P., 2018. Simulation–optimisation framework for City Logistics: an application on multimodal last-mile delivery. IET Intelligent Transport Systems, 12(4), pp.262-269.

[18] Mangiaracina, R., Perego, A., Seghezzi, A. and Tumino, A., 2019. Innovative solutions to increase last-mile delivery efficiency in B2C e-commerce: a literature review. International Journal of Physical Distribution & Logistics Management, 49(9), pp.901-920.

[19] Clark, P.A. and Westerberg, A.W., 1983. Optimization for design problems having more than one objective. Computers & Chemical Engineering, 7(4), pp.259-278.

[20] Bracken, J. and McGill, J.T., 1973. Mathematical programs with optimization problems in the constraints. Operations research, 21(1), pp.37-44.

[21] Google AI Blog. (2022). Federated Learning: Collaborative Machine Learning without Centralized Training Data. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html. Accessed February 8, 2022.

[22] Attaran, M. and Attaran, S., 2007. Collaborative supply chain management: the most promising practice for building efficient and sustainable supply chains. Business process management journal, 13(3), pp.390-404.

[23] Hao, M., Li, H., Luo, X., Xu, G., Yang, H. and Liu, S., 2019. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. IEEE Transactions on Industrial Informatics, 16(10), pp.6532-6542.

[24] Vaidya, J., Yu, H. and Jiang, X., 2008. Privacy-preserving SVM classification. Knowledge and Information Systems, 14, pp.161-178.

[25] Kantarcıoglu, M., Vaidya, J. and Clifton, C., 2003, November. Privacy preserving naive bayes classifier for horizontally partitioned data. In IEEE ICDM workshop on privacy preserving data mining (pp. 3-9).

[26] Agrawal, R. and Srikant, R., 2000, May. Privacy-preserving data mining. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 439-450).

[27] Rudolph, S., Tserendorj, T. and Hitzler, P., 2008, October. What is approximate reasoning?. In International Conference on Web Reasoning and Rule Systems (pp. 150-164). Berlin, Heidelberg: Springer Berlin Heidelberg.

[28] Jiang, W., Li, H., Xu, G., Wen, M., Dong, G. and Lin, X., 2019. PTAS: Privacy-preserving thin-client authentication scheme in blockchain-based PKI. Future Generation Computer Systems, 96, pp.185-195.

[29] Zhang, Y., Xu, C., Ni, J., Li, H. and Shen, X.S., 2019. Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage. IEEE Transactions on Cloud Computing, 9(4), pp.1335-1348.

[30] Zhao, C., Zhao, S., Zhao, M., Chen, Z., Gao, C.Z., Li, H. and Tan, Y.A., 2019. Secure multi-party computation: theory, practice and applications. Information Sciences, 476, pp.357-372.

[31] Xu, G., Li, H., Liu, S., Wen, M. and Lu, R., 2019. Efficient and privacy-preserving truth discovery in mobile crowd sensing systems. IEEE Transactions on Vehicular Technology, 68(4), pp.3854-3865.

[32] Mao, Y., You, C., Zhang, J., Huang, K. and Letaief, K.B., 2017. A survey on mobile edge computing: The communication perspective. IEEE communications surveys & tutorials, 19(4), pp.2322-2358.

[33] McMahan, B., Moore, E., Ramage, D., Hampson, S. and y Arcas, B.A., 2017, April. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.

[34] Albaseer, A., Abdallah, M., Al-Fuqaha, A. and Erbad, A., 2021, December. Client selection approach in support of clustered federated learning over wireless edge networks. In 2021 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.

[35] Sattler, F., Müller, K.R. and Samek, W., 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. IEEE transactions on neural networks and learning systems, 32(8), pp.3710-3722.

[36] Sattler, F., Müller, K.R., Wiegand, T. and Samek, W., 2020, May. On the byzantine robustness of clustered federated learning. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 8861-8865). IEEE.

[37] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," arXiv preprint arXiv:2012.08565, 2020.

[38] Zhang, M., Sapra, K., Fidler, S., Yeung, S. and Alvarez, J.M., 2020. Personalized federated learning with first order model optimization. arXiv preprint arXiv:2012.08565.

[39] Briggs, C., Fan, Z. and Andras, P., 2020, July. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-9). IEEE.

[40] Irfan, D., Xiaofei, X., Shengchun, D. and Khan, I.A., 2007, December. Clustering Framework for Supply Chain Management (SCM) System. In Second Workshop on Digital Media and its Application in Museum & Heritages (DMAMH 2007) (pp. 422-426). IEEE.

[41] Doring, A., Dangelmaier, W. and Danne, C., 2007, August. Using k-means for clustering in complex automotive production systems to support a Q-learning-system. In 6th IEEE International Conference on Cognitive Informatics (pp. 487-497). IEEE.

[42] Hu, J., Hua, E.T., Fei, Y.L. and Chen, D.Q., 2009, September. Research of neural network based on fuzzy clustering in supply chain quality affecting elements data mining. In 2009 International Conference on Management and Service Science (pp. 1-5). IEEE.

[43] Yang, Q., Liu, Y., Chen, T. and Tong, Y., 2019. Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), pp.1-19.

[44] AbdulRahman, S., Tout, H., Ould-Slimane, H., Mourad, A., Talhi, C. and Guizani, M., 2020. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. IEEE Internet of Things Journal, 8(7), pp.5476-5497.

[45] Swinhoe, D. (2020, April 17). The 15 biggest data breaches of the 21st century. CSO Online. https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html. Accessed April 20, 2020.

[46] McMahan, B., Moore, E., Ramage, D., Hampson, S. and y Arcas, B.A., 2017, April. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.

[47] Islam, M.R., Mahmud, M.R. and Pritom, R.M., 2020. Transportation scheduling optimization by a collaborative strategy in supply chain management with TPL using chemical reaction optimization. Neural Computing and Applications, 32, pp.3649-3674.

[48] Konečný, J., McMahan, H.B., Ramage, D. and Richtárik, P., 2016. Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527.

[49] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A. and Seth, K., 2017, October. Practical secure aggregation for privacy-preserving machine learning. In proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1175-1191).

[50] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, B. and Van Overveldt, T., 2019. Towards federated learning at scale: System design. Proceedings of machine learning and systems, 1, pp.374-388.

[51] Li, T., Sahu, A.K., Talwalkar, A. and Smith, V., 2020. Federated learning: Challenges, methods, and future directions. IEEE signal processing magazine, 37(3), pp.50-60.

[52] Caruana, R., 1997. Multitask learning. Machine learning, 28, pp.41-75.

[53] Jacob, L., Vert, J.P. and Bach, F., 2008. Clustered multi-task learning: A convex formulation. Advances in neural information processing systems, 21.

[54] Kumar, A. and Daume III, H., 2012. Learning task grouping and overlap in multi-task learning. arXiv preprint arXiv:1206.6417.

[55] Smith, V., Chiang, C.K., Sanjabi, M. and Talwalkar, A.S., 2017. Federated multi-task learning. Advances in neural information processing systems, 30.

[56] Corinzia, L., Beuret, A. and Buhmann, J.M., 2019. Variational federated multi-task learning. arXiv preprint arXiv:1906.06268.

[57] Ghosh, A., Hong, J., Yin, D. and Ramchandran, K., 2019. Robust federated learning in a heterogeneous environment. arXiv preprint arXiv:1906.06629.

[58] Bhowmick, A., Duchi, J., Freudiger, J., Kapoor, G. and Rogers, R., 2018. Protection against reconstruction and its applications in private federated learning. arXiv preprint arXiv:1812.00984.

[59] Hitaj, B., Ateniese, G. and Perez-Cruz, F., 2017, October. Deep models under the GAN: information leakage from collaborative deep learning. In Proceedings of the 2017 ACM SIGSAC conference on computer and communications security (pp. 603-618).

[60] Winkenbach, M. and Janjevic, M., 2018. Classification of last-mile delivery models for e-commerce distribution: A global perspective. City Logistics 1: New Opportunities and Challenges, pp.209-229.

[61] Farahani, R.Z., Rezapour, S., Drezner, T. and Fallah, S., 2014. Competitive supply chain network design: An overview of classifications, models, solution techniques and applications. Omega, 45, pp.92-118.

[62] Dalal, V. and Sharma, S., 2018. Synthesis of a New Service Classification Matrix in the Digital Era for Innovation and Delivery Based on Current Service Taxonomy. IUP Journal of Supply Chain Management, 15(1).

[63] Konstantakopoulos, G.D., Gayialis, S.P. and Kechagias, E.P., 2020. Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. Operational research, pp.1-30.

[64] Fredrikson, M., Jha, S. and Ristenpart, T., 2015, October. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security (pp. 1322-1333).

[65] Carlini, N., Liu, C., Erlingsson, Ú., Kos, J. and Song, D., 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In 28th USENIX Security Symposium (USENIX Security 19) (pp. 267-284).

[66] Melis, L., Song, C., De Cristofaro, E. and Shmatikov, V., 2019, May. Exploiting unintended feature leakage in collaborative learning. In 2019 IEEE symposium on security and privacy (SP) (pp. 691-706). IEEE.

[67] Xie, M., Long, G., Shen, T., Zhou, T., Wang, X., Jiang, J. and Zhang, C., 2021. Multi-center federated learning. arXiv preprint arXiv:2108.08647.

[68] Sahin, K.H. and Ciric, A.R., 1998. A dual temperature simulated annealing approach for solving bilevel programming problems. Computers & chemical engineering, 23(1), pp.11-25.

[69] Machine Learning Mastery. (Year). Types of Classification in Machine Learning. [https://machinelearningmastery.com/types-of-classification-in-machine-learning/](Accessed: 17 July 2023).

[70] Analytics Vidhya. (Year). Complete Guide to Understand Classification in Machine Learning. [https://www.analyticsvidhya.com/blog/2021/09/a-complete-guide-to-understand-classification-in-machine-learning/](Accessed: 17 July 2023).

[71] Mansour, Y., Mohri, M., Ro, J. and Suresh, A.T., 2020. Three approaches for personalization with applications to federated learning. arXiv preprint arXiv:2002.10619.

[72] Barcena, J.L.C., Ducange, P., Ercolani, A., Marcelloni, F. and Renda, A., 2022, July. An approach to federated learning of explainable fuzzy regression models. In 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (pp. 1-8). IEEE.

[73] Ruspini, E.H., Bezdek, J.C. and Keller, J.M., 2019. Fuzzy clustering: A historical perspective. IEEE Computational Intelligence Magazine, 14(1), pp.45-55.

[74] Ruspini, E.H., 1969. A new approach to clustering. Information and control, 15(1), pp.22-32.

[75] Stallmann, M. and Wilbik, A., 2022. Towards Federated Clustering: A Federated Fuzzy c-Means Algorithm (FFCM). arXiv preprint arXiv:2201.07316.

[76] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R. and D'Oliveira, R.G., 2021. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1–2), pp.1-210.

[77] Yin, X., Zhu, Y. and Hu, J., 2021. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. ACM Computing Surveys (CSUR), 54(6), pp.1-36.

[78] Khan, L.U., Saad, W., Han, Z., Hossain, E. and Hong, C.S., 2021. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. IEEE Communications Surveys & Tutorials, 23(3), pp.1759-1799.

[79] Kaissis, G.A., Makowski, M.R., Rückert, D. and Braren, R.F., 2020. Secure, privacy-preserving and federated machine learning in medical imaging. Nature Machine Intelligence, 2(6), pp.305-311.

[80] Rastrigin, L.A., 1963. The convergence of the random search method in the extremal control of a many parameter system. Automaton & Remote Control, 24, pp.1337-1342.

[81] Das, S.K., Kumar, A., Das, B. and Burnwal, A.P., 2013. On soft computing techniques in various areas. Comput. Sci. Inf. Technol, 3(59), p.166.

[82] Gao, Y., Zhang, G., Lu, J. and Wee, H.M., 2011. Particle swarm optimization for bi-level pricing problems in supply chains. Journal of Global Optimization, 51, pp.245-254.

[83] Medium. (2022). DBSCAN Clustering - Explained. https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556. Accessed May 8, 2022.

[84] Kuo, R.J. and Huang, C.C., 2009. Application of particle swarm optimization algorithm for solving bi-level linear programming problem. Computers & Mathematics with Applications, 58(4), pp.678-685.

[85] Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996, August. A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

[86] Li, X., Tian, P. and Min, X., 2006, June. A hierarchical particle swarm optimization for solving bilevel programming problems. In International Conference on Artificial Intelligence and Soft Computing (pp. 1169-1178). Berlin, Heidelberg: Springer Berlin Heidelberg.

[87] Shih, H.S., Wen, U.P., Lee, S., Lan, K.M. and Hsiao, H.C., 2004. A neural network approach to multiobjective and multilevel programming problems. Computers & Mathematics with Applications, 48(1-2), pp.95-108.

[88] Kaggle Dataset: Pub Hopping in Melbourne - EDA on Pubs in Melbourne + Logistic Regression. (n.d.). Available at: https://www.kaggle.com/shashankarao96207/pub-hopping-in-melbourne-yelp (Last accessed 28 Oct 2021).

[89] Lee, H.L., Padmanabhan, V. and Whang, S., 1997. Information distortion in a supply chain: The bullwhip effect. Management science, 43(4), pp.546-558.

[90] Lan, K.M., Wen, U.P., Shih, H.S. and Lee, E.S., 2007. A hybrid neural network approach to bilevel programming problems. Applied Mathematics Letters, 20(8), pp.880-884.

[91] Ahluwalia, M., Chen, Z., Gangopadhyay, A. and Guo, Z., 2007. Preserving Privacy in Supply Chain Management: a Challenge for Next Generation Data Mining.

[92] Schmidt, G. and Wilhelm, W.E., 2000. Strategic, tactical and operational decisions in multi-national logistics networks: a review and discussion of modelling issues. International Journal of Production Research, 38(7), pp.1501-1523.

[93] Cleophas, C., Cottrill, C., Ehmke, J.F. and Tierney, K., 2019. Collaborative urban transportation: Recent advances in theory and practice. European Journal of Operational Research, 273(3), pp.801-816.

[94] Sternberg, H. and Norrman, A., 2017. The Physical Internet–review, analysis and future research agenda. International Journal of Physical Distribution & Logistics Management, 47(8), pp.736-762.

[95] Dahl, S. and Derigs, U., 2011. Cooperative planning in express carrier networks—An empirical study on the effectiveness of a real-time Decision Support System. Decision Support Systems, 51(3), pp.620-626.

[96] Truong, N., Sun, K., Wang, S., Guitton, F. and Guo, Y., 2021. Privacy preservation in federated learning: An insightful survey from the GDPR perspective. Computers & Security, 110, p.102402.

[97] Harikrishnakumar, R., Dand, A., Nannapaneni, S. and Krishnan, K., 2019, December. Supervised machine learning approach for effective supplier classification. In 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA) (pp. 240-245). IEEE.

[98] Roshan, S.E. and Asadi, S., 2020. Improvement of Bagging performance for classification of imbalanced datasets using evolutionary multi-objective optimization. Engineering Applications of Artificial Intelligence, 87, p.103319.

[99] Breiman, L., 1996. Bagging predictors. Machine learning, 24, pp.123-140.

[100] Freund, Y. and Schapire, R.E., 1996, July. Experiments with a new boosting algorithm. In icml (Vol. 96, pp. 148-156).

[101] Wikipedia contributors. (2023). ”OpenStreetMap.” In: Wikipedia. Retrieved from https://en.wikipedia.org/wiki/OpenStreetMap

[102] Prajapati, D., Harish, A.R., Daultani, Y., Singh, H. and Pratap, S., 2023. A clustering based routing heuristic for last-mile logistics in fresh food E-commerce. Global Business Review, 24(1), pp.7-20.

[103] Comert, S.E., Yazgan, H.R., Kır, S. and Yener, F., 2018. A cluster first-route second approach for a capacitated vehicle routing problem: a case study. International Journal of Procurement Management, 11(4), pp.399-419.

[104] Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), pp.651-666.

[105] de Almeida, M.M.K., Marins, F.A.S., Salgado, A.M.P., Santos, F.C.A. and da Silva, S.L., 2015. Mitigation of the bullwhip effect considering trust and collaboration in supply chain management: a literature review. The International Journal of Advanced Manufacturing Technology, 77, pp.495-513.

[106] de Almeida, M.M.K., Marins, F.A.S., Salgado, A.M.P., Santos, F.C.A. and da Silva, S.L., 2017. The importance of trust and collaboration between companies to mitigate the bullwhip effect in supply chain management. Acta Scientiarum. Technology, 39(2), pp.201-210.

[107] Paulraj, A., Lado, A.A. and Chen, I.J., 2008. Inter-organizational communication as a relational competency: Antecedents and performance outcomes in collaborative buyer–supplier relationships. Journal of operations management, 26(1), pp.45-64.

[108] Nyaga, G.N., Whipple, J.M. and Lynch, D.F., 2010. Examining supply chain relationships: do buyer and supplier perspectives on collaborative relationships differ?. Journal of operations management, 28(2), pp.101-114.

[109] Madenas, N., Tiwari, A., Turner, C.J. and Woodward, J., 2014. Information flow in supply chain management: A review across the product lifecycle. CIRP Journal of Manufacturing Science and Technology, 7(4), pp.335-346.

[110] Boon-itt, S. and Wong, C.Y., 2011. The moderating effects of technological and demand uncertainties on the relationship between supply chain integration and customer delivery performance. International Journal of Physical Distribution & Logistics Management, 41(3), pp.253-276.

[111] Ouhader, H. and El Kyal, M., 2017. Combining facility location and routing decisions in sustainable urban freight distribution under horizontal collaboration: how can shippers be benefited?. Mathematical Problems in Engineering, 2017.

[112] Regazzoni, F., Bhasin, S., Pour, A.A., Alshaer, I., Aydin, F., Aysu, A., Beroulle, V., Di Natale, G., Franzon, P., Hely, D. and Homma, N., 2020, November. Machine learning and hardware security: Challenges and opportunities. In Proceedings of the 39th International Conference on Computer-Aided Design (pp. 1-6).

[113] Fischer-Hübner, S., Hansen, M., Hoepman, J.H. and Jensen, M., 2022. Privacy-Enhancing Technologies and Anonymisation in Light of GDPR and Machine Learning. In IFIP International Summer School on Privacy and Identity Management (pp. 11-20). Cham: Springer Nature Switzerland.

[114] Eke, D., Aasebø, I.E., Akintoye, S., Knight, W., Karakasidis, A., Mikulan, E., Ochang, P., Ogoh, G., Oostenveld, R., Pigorini, A. and Stahl, B.C., 2021. Pseudonymisation of neuroimages and data protection: Increasing access to data while retaining scientific utility. Neuroimage: Reports, 1(4), p.100053.

[115] Georgopoulos, L. and Hasler, M., 2014. Distributed machine learning in networks by consensus. Neurocomputing, 124, pp.2-12.

[116] Wang, J., Kolar, M., Srebro, N. and Zhang, T., 2017, July. Efficient distributed learning with sparsity. In International conference on machine learning (pp. 3636-3645). PMLR.

[117] Lindell, Y., 2005. Secure multiparty computation for privacy preserving data mining. In Encyclopedia of Data Warehousing and Mining (pp. 1005-1009). IGI global.

[118] Du, W. and Atallah, M.J., 2001, September. Secure multi-party computation problems and their applications: a review and open problems. In Proceedings of the 2001 workshop on New security paradigms (pp. 13-22).

[119] Xie, C., Zhong, W. and Zhang, Y., 2008, October. A study of privacy preserving joint-ordering policy. In 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing (pp. 1-4). IEEE.

[120] Kerschbaum, F., Schröpfer, A., Zilli, A., Pibernik, R., Catrina, O., de Hoogh, S., Schoenmakers, B., Cimato, S. and Damiani, E., 2011. Secure collaborative supply-chain management. Computer, 44(9), pp.38-43.

[121] Ji, Z., Lipton, Z.C. and Elkan, C., 2014. Differential privacy and machine learning: a survey and review. arXiv preprint arXiv:1412.7584.

[122] Xiong, P., Zhu, T. and Wang, X.F., 2014. A survey on differential privacy and applications.

[123] Nandakumar, L., Ferrari, R. and Keviczky, T., 2019. Privacy-preserving of system model with perturbed state trajectories using differential privacy: with application to a supply chain network. IFAC-PapersOnLine, 52(20), pp.309-314.

[124] Nandakumar, L., 2018. Privacy-Aware State Estimation based on Obfuscated Transformation and Differential Privacy: With applications to smart grids and supply chain economics.

[125] Geismar, H.N., Laporte, G., Lei, L. and Sriskandarajah, C., 2008. The integrated production and transportation scheduling problem for a product with a short lifespan. INFORMS Journal on Computing, 20(1), pp.21-33.

[126] Zirour, M., 2008. Vehicle routing problem: models and solutions. Journal of Quality Measurement and Analysis JQMA, 4(1), pp.205-218.

[127] Routific. (2012). Routific: Route Optimization Software. Retrieved 04/09/2023, from https://routific.com/

[128] Liu, Z., Wang, H., Chen, W., Yu, J. and Chen, J., 2016. An incidental delivery based method for resolving multirobot pairwised transportation problems. IEEE Transactions on Intelligent Transportation Systems, 17(7), pp.1852-1866.

[129] Naeem, M. and Ombuki-Berman, B., 2010, July. An efficient genetic algorithm for the uncapacitated single allocation hub location problem. In IEEE Congress on Evolutionary Computation (pp. 1-8). IEEE.

[130] Zheng, W., Shen, Y. and Xiao, T., 2020, September. Asocial: Adaptive Task Re-Allocation in Distributed Computing Systems with Node Failures. In 2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS) (pp. 179-184). IEEE.

[131] James, J.Q. and Lam, A.Y., 2017. Autonomous vehicle logistic system: Joint routing and charging strategy. IEEE Transactions on Intelligent Transportation Systems, 19(7), pp.2175-2187.

[132] Marinakis, Y. and Marinaki, M., 2013. A bilevel particle swarm optimization algorithm for supply chain management problems. In Metaheuristics for Bi-level Optimization (pp. 69-93). Berlin, Heidelberg: Springer Berlin Heidelberg.

[133] Bidgoli, H., 2010. The handbook of technology management, supply chain management, marketing and advertising, and global management (Vol. 2). John Wiley & Sons.

[134] Ma, W. and Wang, M., 2013. Particle swarm optimization-based algorithm for bilevel joint pricing and lot-sizing decisions in a supply chain. Applied Artificial Intelligence, 27(6), pp.441-460.

[135] Hejazi, S.R., Memariani, A., Jahanshahloo, G. and Sepehri, M.M., 2002. Linear bilevel programming solution by genetic algorithm. Computers & Operations Research, 29(13), pp.1913-1925.

[136] Calvete, H. I., C. Gale´, and P. M. Mateo. 2008. A new approach for solving linear bilevel problems using genetic algorithms. European Journal of Operational Research 188:4–28.

[137] Li, M. Q., D. Lin, and S. Y. Wang. 2010. Solving a type of biobjective bilevel programming problem using NSGA-II. Computers and Mathematics with Applications 59:706–715.

[138] Wen, U. P., and A. D. Huang. 1996. A simple tabu search method to solve the mixed-integer problem bi-level programming problem. European Journal of Operational Research 88:563–571.

[139] Rajesh, J., Gupta, K., Kusumakar, H.S., Jayaraman, V.K. and Kulkarni, B.D., 2003. A tabu search based approach for solving a class of bilevel programming problems in chemical engineering. Journal of Heuristics, 9, pp.307-319.

[140] Ertenlice, O. and Kalayci, C.B., 2018. A survey of swarm intelligence for portfolio optimization: Algorithms and applications. Swarm and evolutionary computation, 39, pp.36-52.

[141] Kennedy, J. and Eberhart, R., 1995, November. Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.

[142] Dorigo, M., Maniezzo, V. and Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. IEEE transactions on systems, man, and cybernetics, part b (cybernetics), 26(1), pp.29-41.

[143] Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. IEEE control systems magazine, 22(3), pp.52-67.

[144] Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

[145] Chu, S.C., Tsai, P.W. and Pan, J.S., 2006. Cat swarm optimization. In PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence Guilin, China, August 7-11, 2006 Proceedings 9 (pp. 854-858). Springer Berlin Heidelberg.

[146] Yang, X.S., 2009, October. Firefly algorithms for multimodal optimization. In International symposium on stochastic algorithms (pp. 169-178). Berlin, Heidelberg: Springer Berlin Heidelberg.

[147] Karimkashi, S. and Kishk, A.A., 2010. Invasive weed optimization and its features in electromagnetics. IEEE transactions on antennas and propagation, 58(4), pp.1269-1278.

[148] Yang, X.S., 2010. A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Berlin, Heidelberg: Springer Berlin Heidelberg.

[149] Tan, Y. and Zhu, Y., 2010. Fireworks algorithm for optimization. In Advances in Swarm Intelligence: First International Conference, ICSI 2010, Beijing, China, June 12-15, 2010, Proceedings, Part I 1 (pp. 355-364). Springer Berlin Heidelberg.

[150] Derrouiche, R., Holimchayachotikul, P. and Leksakul, K., 2011, May. Predictive performance model in collaborative supply chain using decision tree and clustering technique. In 2011 4th International Conference on Logistics (pp. 412-417). IEEE.

[151] Beni, G. and Wang, J., 1993. Swarm intelligence in cellular robotic systems. In Robots and biological systems: towards a new bionics? (pp. 703-712). Berlin, Heidelberg: Springer Berlin Heidelberg.

[152] Mishra, S.K., Panda, G. and Meher, S., 2009, December. Multi-objective particle swarm optimization approach to portfolio optimization. In 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (pp. 1612-1615). IEEE.

[153] Wang, B., & Watada, J. (2013). Multiobjective particle swarm optimization for a novel fuzzy portfolio selection problem. IEEJ transactions on electrical and electronic engineering, 8(2), 146-154.

[154] Madani, B. and Ndiaye, M., 2019, April. Autonomous vehicles delivery systems classification: introducing a TSP with a moving depot. In 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO) (pp. 1-5). IEEE.

[155] Gevaers, R., Van de Voorde, E. and Vanelslander, T., 2009. Characteristics of innovations in last-mile logistics-using best practices, case studies and making the link with green and sustainable logistics. Association for European Transport and contributors, 1, p.21.

[156] Zhu, S., Hu, X., Huang, K. and Yuan, Y., 2021. Optimization of product category allocation in multiple warehouses to minimize splitting of online supermarket customer orders. European journal of operational research, 290(2), pp.556-571.

[157] Brownlee, J. (2016). ”A Gentle Introduction to XGBoost for Applied Machine Learning.” Machine Learning Mastery. Available at: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/ (Accessed: 17 July 2023)

[158] Trung, N. T. (n.d.). ”Story and Lessons Behind the Evolution of XGBoost.” [Online]. Available at: https://sites.google.com/site/nttrungmtwiki/home/it/data-science—python/xgboost/story-and-lessons-behind-the-evolution-of-xgboost (Accessed: 15 March 2023)

[159] J. Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics, pp.1189-1232.

[160] Neptune.ai. (n.d.). ”XGBoost: Everything You Need to Know.” [Online]. Available at: https://neptune.ai/blog/xgboost-everything-you-need-to-know (Accessed: 15 April 2023)

[161] Chen, T. and Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[162] Yang, M., Song, L., Xu, J., Li, C. and Tan, G., 2019. The tradeoff between privacy and accuracy in anomaly detection using federated xgboost. arXiv preprint arXiv:1907.07157.

[163] Yang, H., Pasupa, K., Leung, A.C.S., Kwok, J.T., Chan, J.H. and King, I. eds., 2020. Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part II (Vol. 12533). Springer Nature.