ENHANCING TEA SUPPLY CHAIN EFFICIENCY USING PREDICTIVE ANALYTICS

24-25J-303

Research Final Report

Wadigasinghe U.K

IT21306990

B.Sc. (Hons) Degree in Information Technology specialized in Information Technology

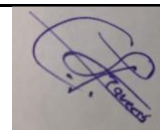
Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

DECLARATION

I declare that this my own work & this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning & to the best of my knowledge & belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Wadigasinghe U.K	IT21306990	

Date : 2025/04/11

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

--   ---------------------------------

Signature of the Supervisor   Date

--   ---------------------------------

Signature of the Co-Supervisor   Date

ABSTRACT

The tea industry plays a significant role in the global agricultural economy, especially in countries where tea production is a major source of income and employment. However, inefficiencies in the tea supply chain ranging from unpredictable demand and supply fluctuations to delays in logistics continue to pose major challenges. This research focuses on enhancing tea supply chain efficiency through the integration of predictive analytics. By analyzing historical data, weather patterns, market trends, and logistical factors, predictive models were developed to forecast key variables such as demand, supply volume, and transportation delays. The study used machine learning algorithms and data visualization tools to identify patterns and generate actionable insights for supply chain decision-makers. The implementation of these predictive tools demonstrated improved accuracy in forecasting, reduced lead times, and better inventory management. Furthermore, the research highlights the potential of predictive analytics to transform traditional supply chain operations into data-driven, agile systems. By addressing existing bottlenecks and enabling proactive strategies, the integration of predictive analytics stands out as a promising approach to achieving operational efficiency and resilience in the tea industry. The findings of this study can serve as a foundation for future advancements in smart agriculture and supply chain optimization in similar sectors.

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to my supervisor for the continuous support, valuable guidance, and encouragement throughout the course of this research. Your expertise and insightful feedback were crucial to the successful completion of this study. I would also like to thank my co-supervisor for their consistent support, constructive suggestions, and dedication during every stage of this project.

I extend my heartfelt thanks to all the members of our research group, whose cooperation, shared knowledge, and collaboration greatly contributed to the development of this work. A special appreciation goes to the staff of the Watawala Tea Factory for providing essential information, data, and practical insights that were instrumental to the research.

I am also deeply grateful to my family members and friends for their unwavering support and encouragement throughout this journey. Finally, I would like to thank all the students in my batch who directly or indirectly contributed to this research by offering help, sharing ideas, or simply motivating me along the way.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background and literature survey

The tea industry remains one of the most vital contributors to the economies of many countries, especially in South Asia. In Sri Lanka, tea is not only a major export commodity but also a livelihood for thousands of individuals engaged in cultivation, processing, and distribution. Despite its significance, the tea supply chain continues to face persistent challenges, including unpredictable market demand, fluctuating supply levels, inefficiencies in logistics, and limited use of modern technology in decision-making processes. These factors contribute to delays, excess inventory, loss of product quality, and missed market opportunities.

With the advancement of data science and artificial intelligence, predictive analytics has emerged as a powerful tool for enhancing supply chain efficiency. By analyzing historical data and identifying trends, predictive models can forecast demand, estimate supply volumes, detect potential delays, and support proactive planning. In this context, our research focuses on the integration of predictive analytics into the tea supply chain, specifically targeting the operations of Watawala Tea Factory. The goal is to minimize inefficiencies and introduce a data-driven decision-making culture in an industry that traditionally relies on experience and intuition.

To conduct this research, we collected operational and logistical data from Watawala Tea Factory, which provided the foundation for developing and testing predictive models. The data underwent preprocessing to remove inconsistencies, fill missing values, and normalize inputs for modeling. Using tools such as Pandas for data manipulation and Matplotlib for visualization, we gained valuable insights into data distribution and seasonal patterns. For the modeling process, we implemented Long Short-Term Memory (LSTM) networks a type of recurrent neural network (RNN) suitable for sequential data and time-series forecasting. LSTM models are known for their ability to capture long-term dependencies, making them ideal for modeling production trends and seasonal demand shifts in agricultural industries like tea.

The model development and training were conducted using TensorFlow and scikit-learn (sklearn) in a Google Colab environment, which provided the scalability and flexibility needed for iterative model tuning. A Flask Python server was used to build a lightweight web interface for running predictions and visualizing outputs in real time. This integrated system can potentially be scaled for use by supply chain managers and stakeholders in the tea industry to support planning and inventory decisions.

In the realm of literature, various studies have demonstrated the use of predictive analytics and machine learning in agricultural supply chains. For instance, researchers have used time-series models for crop yield prediction, demand forecasting, and logistics planning in sectors like rice, wheat, and dairy. However, the application of advanced models such as LSTM in the tea supply chain remains limited. Studies like those by Kumar et al. (2020) and Singh et al. (2021) have shown the effectiveness of neural networks in improving forecasting accuracy in agri-based supply chains, but few have focused on tea or Sri Lankan-specific contexts. Our work addresses this gap by providing a real-world implementation of predictive analytics using actual factory data and applying it in a region-specific setting.

In conclusion, this research contributes to the growing field of smart agriculture and supply chain optimization by combining machine learning, real-time data handling, and visualization techniques. The system we developed serves as a practical solution for overcoming uncertainty and inefficiency in the tea supply chain. Through this, we aim to not only improve operational efficiency at Watawala Tea Factory but also provide a scalable framework that can be adapted across the broader tea industry and similar agricultural domains.

1.2 Research Gap

In recent years, significant attention has been directed toward improving supply chain efficiency in the agricultural sector through the use of predictive analytics. Studies have shown promising outcomes in areas such as crop yield forecasting, demand prediction, and logistics optimization for various products like rice, wheat, and dairy. However, the tea industry, particularly in regions like Sri Lanka where it plays a vital economic role, has seen limited adoption of these advanced techniques. Most tea-related supply chain studies tend to focus on qualitative factors such as labor management, production processes, or cost optimization, often lacking the integration of real-time data-driven models.

Furthermore, a large proportion of existing literature relies on conventional statistical models or basic regression techniques. While these models provide insights into general trends, they often fall short in handling sequential dependencies and complex, non-linear patterns commonly found in real-world supply chain data. This is particularly true in agricultural industries where seasonality, weather conditions, and market demand can vary unpredictably.

Few studies have applied deep learning models such as Long Short-Term Memory (LSTM) networks within the tea supply chain context. LSTM's strength in capturing long-term dependencies in time-series data makes it an ideal candidate for demand and supply forecasting, yet its adoption in this sector remains minimal. Moreover, research that integrates LSTM with real-time data visualization and deployment platforms such as Flask web apps—is even more scarce.

Another observed gap is the lack of end-to-end implementation frameworks in the literature. Many studies focus solely on model accuracy, without translating the findings into practical applications or systems that supply chain managers can use directly. Our research fills this void by not only applying a deep learning model to a specific real-world dataset from Watawala Tea Factory but also by developing a user-friendly interface for operational use.

In summary, while predictive analytics has proven beneficial in general agriculture and logistics sectors, there is a notable gap in its tailored application to the tea industry using advanced neural network models. Our work stands out by merging LSTM modeling, real factory data, and deployment tools like Flask and Google Colab to create a functional, scalable system for improving supply chain efficiency.

Study	Focus on Tea Industry	Advanced Model	Real World Data Used	Practical implementation	Interactive Interface
Research A	✗	✗	✓	✗	✗
Research B	✗	✗	✓	✗	✗
Research C	✓	✗	✓	✗	✗
Research D	✗	✓	✓	✗	✗
Proposed System	✓	✓	✓	✓	✓

Table 1- Research Gap Comparison

The table above compares the features of various research studies related to predictive analytics in agriculture with our study. Each column represents a critical element in the research process, such as focus on the tea industry, use of advanced models like LSTM, integration of real-world data, practical implementation, and the inclusion of an interactive interface. The tick (✓) indicates the presence of the feature, while the cross (✗) indicates its absence.

As seen in the table, most previous studies do not apply deep learning models such as LSTM or fail to use real-world data from the tea industry. Many studies also stop at theoretical or model-based approaches without moving into full-scale, practical deployment. In contrast, our research addresses all these gaps by using real factory data, applying the LSTM model, and implementing a fully functional system with an interactive Flask-based interface for real-time predictions.

1.3 Research Problem

The tea industry, particularly in Sri Lanka, is a cornerstone of both the economy and the agricultural sector. Despite its importance, the industry continues to face persistent challenges related to inefficiencies within the supply chain. These inefficiencies manifest in various forms, including poor demand forecasting, logistical delays, high operational costs, and suboptimal inventory management. These problems are often exacerbated by a lack of real-time data and advanced predictive techniques that could help stakeholders make informed decisions. The inability to predict supply and demand accurately leads to either overproduction, resulting in wasted resources, or underproduction, leading to missed sales opportunities and customer dissatisfaction.

A major obstacle in the tea supply chain is the traditional approach to demand forecasting, which heavily relies on historical data and human intuition. While past data provides some insights, the unpredictability of factors like weather patterns, market trends, and international demand creates significant uncertainty. Furthermore, seasonal variations in both production and demand make it even more challenging to develop accurate forecasting models that can adjust to these fluctuations. Without a data-driven model that can account for these complexities, stakeholders often find it difficult to make timely and effective decisions regarding inventory, distribution, and production planning.

In addition to demand forecasting, logistics and transportation in the tea supply chain are fraught with inefficiencies. These inefficiencies often arise from suboptimal route planning, unanticipated delays, and inventory management issues. Tea producers often face challenges in ensuring that the right amount of tea reaches the right location at the right time, particularly in peak seasons when demand spikes. This issue is compounded by the fact that many factories and distributors still rely on manual methods or outdated software for tracking and optimizing these logistics.

Current solutions in the tea industry tend to be either overly simplistic or fail to integrate modern technological advancements, such as machine learning and real-time data processing. Although there have been isolated attempts to introduce automation and optimization techniques, they remain limited in scope and application. Additionally, the adoption of deep learning techniques, such as Long Short-Term Memory (LSTM) networks, for forecasting demand and optimizing supply chain logistics is virtually non-existent in the tea industry.

The specific research problem addressed in this study is the lack of an integrated, data-driven system that combines predictive modeling, real-time data processing, and actionable insights for tea supply chain optimization. While there are existing methods for forecasting demand and optimizing logistics, they often fail to account for the complexity and variability of the tea supply chain. Our research problem revolves around the need for a robust solution that can:

1. Improve Demand Forecasting: Create an accurate demand forecasting model using time-series data and machine learning techniques, particularly LSTM networks, to predict tea demand with higher accuracy.
2. Optimize Supply Chain Logistics: Integrate predictive models with real-time logistics data to optimize transportation routes, reduce delays, and improve inventory management.
3. Real-Time Data Integration: Provide a real-time data visualization platform that helps stakeholders monitor and adjust their supply chain decisions on the fly.
4. Practical Implementation: Develop a fully integrated, user-friendly system using Python (Flask, TensorFlow, Pandas) and deploy it as a web-based tool that can be used by supply chain managers at Watawala Tea Factory and potentially scaled to other tea producers.

By addressing these key aspects, this research seeks to bridge the gap between traditional tea supply chain practices and modern, data-driven methodologies. Specifically, the use of LSTM for forecasting and Flask for real-time data deployment presents a novel approach for overcoming the current limitations in the tea industry's supply chain operations. Additionally, the incorporation of tools like Pandas for data preprocessing, Matplotlib for visualization, and Google Colab for model training, enhances the robustness of the system and makes it a viable solution for real-world implementation.

This study contributes to the tea industry by not only improving forecasting accuracy but also providing a practical, real-time tool that stakeholders can use to optimize their operations. It offers the potential to reduce costs, improve efficiency, and enhance the overall sustainability of the tea supply chain.

1.4 Research Objectives

1.4.1 Main Objective

- To enhance the efficiency of logistics in the tea supply chain by optimizing transportation routes and reducing delivery costs using machine learning techniques.

This research focuses on improving the logistics component of the tea supply chain by applying machine learning models, particularly Long Short-Term Memory (LSTM) networks, to forecast demand patterns and support data-driven transportation planning. By analyzing historical data from the Watawala Tea Factory, the system aims to predict when and where tea products are required, enabling more efficient scheduling and routing of deliveries. The use of advanced tools such as TensorFlow, Pandas, and Google Colab supports the development of a robust model, while the deployment through a Flask-based web interface allows for real-time visualization and user interaction. Ultimately, the goal is to reduce unnecessary transportation costs, minimize delays, and support proactive decision-making in the logistics operations of the tea supply chain.

1.1.1 Specific Objectives

Collect and analyze historical traffic data during critical delivery windows.

The first specific objective of this research is to gather and evaluate historical traffic data relevant to the tea delivery process, especially during peak delivery times. In the context of the Watawala Tea Factory, deliveries often occur on scheduled routes and during specific time windows. However, traffic congestion, road conditions, and unexpected delays frequently disrupt this schedule. By identifying these "critical windows" times when delays most often occur the research begins by collecting comprehensive traffic data from publicly available datasets, maps, and records relevant to these routes.

Analyzing this data enables the identification of recurring patterns such as traffic build-up during certain hours or specific days of the week. Factors such as weather conditions, public holidays, and special events are also considered where possible. Tools such as Pandas and Python scripts are used to preprocess and clean the data, ensuring consistency and relevance for model training. This foundational step ensures that the forecasting model is trained on context-aware data that reflects the actual challenges faced during tea delivery operations, forming the basis for accurate and actionable predictions.

Use predictive analytics to forecast traffic conditions for upcoming weeks.

The second specific objective involves applying machine learning specifically predictive analytics techniques to forecast traffic conditions along key delivery routes for upcoming weeks. This is where the core intelligence of the system is developed. Using the cleaned and structured historical traffic data, a Long Short-Term Memory (LSTM) neural network is trained to recognize temporal patterns and trends. LSTM, being a type of recurrent neural network, excels in handling time-series data with long-range dependencies, making it ideal for forecasting traffic behavior.

TensorFlow and Google Colab are used for model development and training, with performance evaluation based on key metrics such as Mean Squared Error (MSE) and prediction accuracy. The model learns to anticipate future congestion levels by recognizing how past events influence upcoming traffic conditions. For example, it can predict increased traffic on a Monday morning or anticipate delays due to patterns seen before public holidays. The ultimate goal of this objective is to produce a reliable, near-real-time traffic forecasting model that can provide actionable insights to delivery planners at the tea factory.

Adjust delivery schedules based on traffic predictions to improve timing and minimize costs.

The third and final specific objective is to use the predictions generated by the model to dynamically adjust tea delivery schedules. Based on the traffic forecasts, the system can suggest optimal departure times and alternative routes that help avoid congestion and reduce unnecessary idling or delays. This adaptive approach ensures that delivery operations are not only reactive but also proactive, allowing for improved planning and resource allocation.

The system's user interface, built using Flask, presents traffic predictions and recommended delivery windows in a clear, visual format. This empowers logistics coordinators at the Watawala Tea Factory to make informed decisions with minimal technical knowledge. By shifting delivery schedules to avoid peak traffic and optimizing routes accordingly, the factory can significantly reduce fuel costs, decrease delivery times, and enhance overall supply chain reliability. This objective ensures that the insights from machine learning are translated into real-world actions, directly addressing inefficiencies in the current logistics system.

2. METHODOLOGY

2.1 System Methodology

2.1.1 Requirement Gathering and Analysis

Requirement gathering and analysis is a critical first step in the research process, as it lays the foundation for building a system that meets the actual needs of users and stakeholders. In the context of this study, which focuses on optimizing logistics in the tea supply chain using machine learning, the requirement gathering phase involved identifying the key challenges, expectations, and functional needs from both industry experts and system users at the Watawala Tea Factory.

- **Requirement Gathering**

The requirement gathering process began with direct engagement with logistics staff, supervisors, and technical personnel at the Watawala Tea Factory. Through interviews, informal discussions, and observational visits, the research team collected valuable insights about the current delivery processes, transportation routes, scheduling methods, and the pain points experienced during daily operations. One of the recurring themes identified was the impact of traffic delays and inefficient route planning on delivery schedules and costs. Additionally, it became evident that decisions were often made reactively, without support from predictive data or automated tools.

Secondary data was also gathered by reviewing existing documentation on delivery schedules, historical transport logs, and publicly available traffic reports. This helped to frame a realistic picture of the challenges faced and provided the raw data required for later model development. During this phase, system expectations were also clarified. Stakeholders highlighted the need for a simple, web-based interface to view traffic forecasts and suggested delivery windows, emphasizing usability over complexity.

- **Requirement Analysis**

Once the necessary information was gathered, the next step involved analyzing and translating these findings into technical and functional requirements. This analysis helped to define the core system components, such as the need for a robust data preprocessing pipeline, an accurate forecasting model, and an interactive visualization dashboard. The data collected revealed patterns in traffic congestion that could be modeled using time-series forecasting techniques, particularly Long Short-Term Memory (LSTM) networks, due to their effectiveness in handling sequential data.

The analysis also guided the selection of tools and technologies for the system. Python was chosen as the primary development language due to its extensive machine learning libraries and community support. TensorFlow was selected for model training, Pandas for data manipulation, and Matplotlib for visual representation of trends and results. Google Colab was used for efficient and scalable model training in a cloud-based environment. Flask was chosen to serve as the backend framework to deliver real-time predictions via a user-friendly web interface.

Functional requirements included features such as the ability to input historical traffic data, view predicted traffic patterns, and generate suggested delivery windows. Non-functional requirements, such as system responsiveness, scalability, and ease of use, were also identified to ensure long-term usability and adaptability of the system.

Overall, this phase ensured that the research was grounded in real-world needs and that the proposed solution was both technically feasible and practically useful. By carefully gathering and analyzing requirements, the project established a clear roadmap for system development, aligned with the ultimate goal of improving logistics efficiency in the tea supply chain.

2.1.2 Research Requirements

2.1.2.1 Functional Requirements

- Collect and process historical traffic data from reliable sources.
- Generate accurate traffic predictions for upcoming weeks using predictive models.
- Provide optimized delivery schedules based on traffic forecasts.
- User interface for the transportation manager to manage vehicles, routes, and best delivery times.

The functional requirements define the core actions the system must perform to meet the goals of the research. These requirements were shaped by discussions with stakeholders at the Watawala Tea Factory and revolve around the primary need for more efficient logistics planning. The system must be able to gather historical traffic data, apply predictive modeling to forecast future traffic trends, and use these predictions to create optimized delivery schedules. Additionally, the interface must allow transportation managers to interact with the system easily managing routes, viewing suggested delivery windows, and overseeing logistics decisions in a centralized, user-friendly platform. These functions are essential for improving operational efficiency and reducing transportation costs.

2.1.2.2 Non-Functional Requirements

- Ensure high accuracy and reliability of traffic predictions.
- Provide a responsive and user-friendly interface for the transportation manager.
- Ensure system scalability to handle increased data and additional logistics routes.
- Maintain data security and confidentiality throughout the system.

The non-functional requirements focus on the quality attributes of the system how well it performs rather than what it does. These requirements ensure that the developed solution is not only functional but also dependable and practical for real-world use. Accuracy and reliability are crucial, as poor predictions could lead to worse outcomes than manual planning. The interface must be intuitive and responsive, especially considering that end users may not have technical expertise. Scalability is also a major consideration; as the factory grows or incorporates additional logistics routes, the system must adapt without performance degradation. Lastly, security is vital when handling sensitive data related to factory operations and logistics.

2.1.2.3 Software Requirements

- **Python** – Primary programming language for development.
- **Flask** – Backend web framework used to build the server and deploy the system.
- **TensorFlow** – Deep learning library used to develop and train the LSTM model.
- **Pandas** – Used for data preprocessing, analysis, and manipulation.
- **Matplotlib** – Used for generating data visualizations and trend graphs.
- **Google Colab** – Cloud-based environment used for training and testing the LSTM model efficiently.
- **Sklearn** – Used for additional preprocessing and evaluation utilities in machine learning.

These tools and libraries were selected based on their suitability for time-series forecasting, ease of integration, and wide support. Together, they formed a powerful and flexible stack for developing a data-driven logistics optimization system.

2.1.3 Feasibility Study

A feasibility study evaluates whether a proposed system is viable across different dimensions before moving into full-scale development. For this research, the feasibility was assessed under four main categories: ethical, technical, financial, and market feasibility, ensuring that the solution is not only functional but also practical and responsible within its intended environment.

- **Ethical Feasibility**

The ethical feasibility of the research ensures that all data used was acquired with consent and used responsibly. Since the system relies on historical traffic and logistics data, proper permissions were obtained from the Watawala Tea Factory to access and use this data solely for academic and research purposes. The project maintains a strong stance on data privacy and does not involve any collection of personal or sensitive user information. Additionally, the recommendations provided by the system are intended to support human decision-making rather than replace it, keeping ethical responsibility in operational planning intact.

- **Technical Feasibility**

This research was found to be technically feasible based on the availability of tools, frameworks, and data necessary to implement a predictive system. Technologies such as Python, TensorFlow, Pandas, Flask, and Google Colab provided a solid foundation for building, training, and deploying the LSTM-based traffic prediction model. The team possessed the required technical skills to process data, train deep learning models, and develop an interactive web application. Furthermore, since the system runs on widely accessible platforms and can be hosted with minimal hardware resources, it proves to be implementable within the factory's existing infrastructure or in cloud-based environments.

- **Financial Feasibility**

From a financial perspective, the project was designed to be low-cost and scalable. All development was carried out using open-source tools and platforms, eliminating the need for expensive proprietary software. Google Colab offered free cloud computing capabilities for model training, and Flask allowed for lightweight deployment without the need for high-end servers. The only potential future costs may involve hosting the system on a live server or expanding it to more routes, which can still be managed affordably. As such, the solution is financially feasible even for small or medium-sized operations like the Watawala Tea Factory.

- **Market Feasibility**

Market feasibility evaluates the potential acceptance and usefulness of the system in the actual operational environment. Based on the discussions with staff at the Watawala Tea Factory, there is a strong demand for improving logistics efficiency. Current delivery planning processes are largely manual and reactive, leading to unnecessary costs and delays. A system that predicts traffic and suggests optimal delivery schedules offers clear operational benefits. Moreover, this type of solution is adaptable to other tea factories and even different sectors that deal with logistics, indicating strong market relevance and potential for future expansion.

2.1.4 System Design

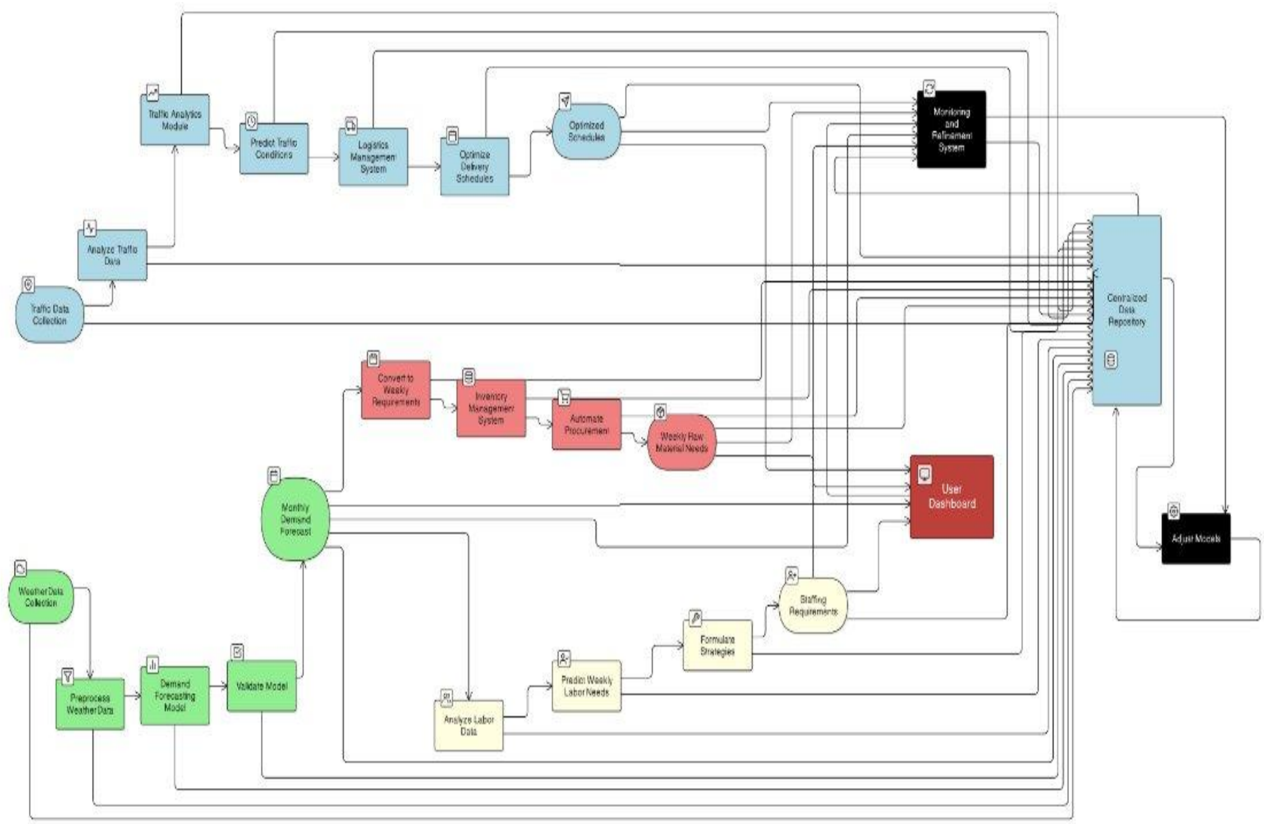


Figure 1- Overall System Diagram

The overall system designed in this research is a collaborative integration of four key components Demand Forecasting, Inventory Management, Logistic Management, and Labor Management each handled by a separate research member. Together, these modules aim to create a smart, data-driven tea supply chain system that enhances operational efficiency, reduces costs, and improves overall coordination from demand prediction to product delivery. The architecture of the system is modular, with each part handling a distinct function while sharing critical data between components. The following explanation corresponds to the system diagram that visually represents the flow and interaction between these modules.

The demand forecasting module is responsible for predicting future tea demand based on historical sales data, market trends, seasonal patterns, and external influencing factors. Machine learning algorithms are used to identify patterns in customer orders and market consumption. The results from this component directly influence production volumes and inventory planning. By accurately forecasting demand, the factory can avoid overproduction or underproduction, thereby reducing waste and maximizing profitability.

This module manages the stock of raw materials (like fresh tea leaves) and finished tea products. It integrates demand forecasts to determine optimal inventory levels and avoid bottlenecks in production. The system monitors inflows (from leaf collection) and outflows (based on production and delivery) to maintain a balanced stock. Real-time tracking and predictive modeling help the factory ensure that enough raw material is available when needed, and that finished goods are stored efficiently until dispatch. This component shares data with both demand forecasting and logistics modules to ensure smooth coordination.

The logistics management component, which is the focus of my part of the research, deals with optimizing the transportation and delivery of finished tea products. It uses historical traffic data and delivery records to forecast traffic conditions and plan more efficient delivery schedules. An LSTM-based predictive model was developed using TensorFlow and trained on traffic pattern data, which helps in generating delivery windows that minimize delays and reduce fuel consumption. The system also includes a user interface built with Flask that allows transportation managers to view suggested delivery times, traffic forecasts, and manage vehicle routes. This module receives delivery priorities from the inventory and demand systems and outputs optimized transportation schedules accordingly.

The labor management module handles workforce allocation and scheduling, ensuring that labor resources are used efficiently across collection, production, and logistics. This component analyzes past labor usage data, peak operational times, and work schedules to suggest optimal labor distribution. It ensures that enough workers are allocated for tea leaf collection, production processes, packaging, and delivery. Labor demand is dynamically adjusted based on predictions from other modules like demand forecasting and inventory planning. This ensures not only operational efficiency but also cost-effective labor management.

Each of these components works both independently and interdependently to create a highly responsive and predictive supply chain model. Data flows between the modules ensure that each decision—whether related to delivery scheduling or inventory control is informed by the latest available forecasts and insights from other parts of the system. The overall system is designed to be scalable, adaptable, and easy to deploy in real-world tea factory environments such as the Watawala Tea Factory.

The corresponding system diagram (to be inserted here) illustrates these four components, showing the data flow, predictive processes, and decision-support interfaces built into the overall design. Together, the system offers a comprehensive solution to improve operational efficiency across the entire tea supply chain.

2.1.5 Software Solution

The development of the proposed system followed established software engineering practices to ensure systematic planning, smooth implementation, and successful delivery. Two major methodologies were adopted during the software solution process: the Software Development Life Cycle (SDLC) and the Agile methodology. Each offered unique benefits to different stages of the project and complemented each other throughout the research timeline.

The SDLC is a structured framework that outlines the various stages involved in software development, from the initial planning phase to deployment and maintenance. It ensures that the development process is organized and goal-driven, minimizing risks and optimizing the outcome. The research project followed the major stages of SDLC as follows:

1. Requirement Gathering and Analysis – Detailed requirements for each system module (demand forecasting, inventory, logistics, labor) were collected through discussions with tea factory staff and research stakeholders. This stage ensured a clear understanding of the problem domain.
2. System Design – Based on the analyzed requirements, the overall architecture and module interconnections were designed. Each member focused on one functional block, with planned data flow and interaction points.
3. Implementation – Each component was implemented using appropriate technologies. For example, Python, TensorFlow, and Flask were used to build the logistics management module. Other modules followed similar independent development cycles.
4. Testing – Unit testing was done by individual members to ensure their components functioned as expected. Integration testing was carried out to validate the overall system flow and data interactions between components.
5. Deployment – The modules were packaged and demonstrated as a complete system. The logistics interface was hosted using a Flask server for testing, and predictions were run using trained models from Google Colab.

6. Maintenance (Ongoing for future scope) – Suggestions and feedback were documented for possible future enhancements. The modular nature of the system allows for easy updates and scaling.

While the SDLC provided a structural roadmap, Agile methodology was incorporated to promote flexibility, collaboration, and rapid iteration especially crucial in a student research project involving multiple team members and dynamic requirements.

Figure 2 - Agile Methodology Illustration

The Agile approach was used in the following ways:

- Sprints and Iterations – The work was broken down into short, manageable sprints. Each sprint involved developing or improving specific features in one component (e.g., traffic model training, UI building, or data preprocessing).
- Regular Check-ins – Weekly group discussions were held to review progress, share challenges, and reassign tasks if needed. This ensured that all members stayed aligned and adapted to any changes or delays efficiently.
- Continuous Feedback – Feedback from supervisors, teammates, and factory staff was continuously integrated into the development cycle. This helped refine prediction accuracy, interface design, and data outputs.

- Collaborative Tools – Shared Google Drive folders, GitHub repositories, and group chats were used to track changes, share code, and maintain transparency among the team.

By combining SDLC's structured framework with the flexibility and responsiveness of Agile, the team was able to deliver a well-organized, functional, and efficient software system. This hybrid approach ensured that the final product met both academic standards and real-world usability needs for the Watawala Tea Factory.

2.1.6 Component overview - Logistic Management

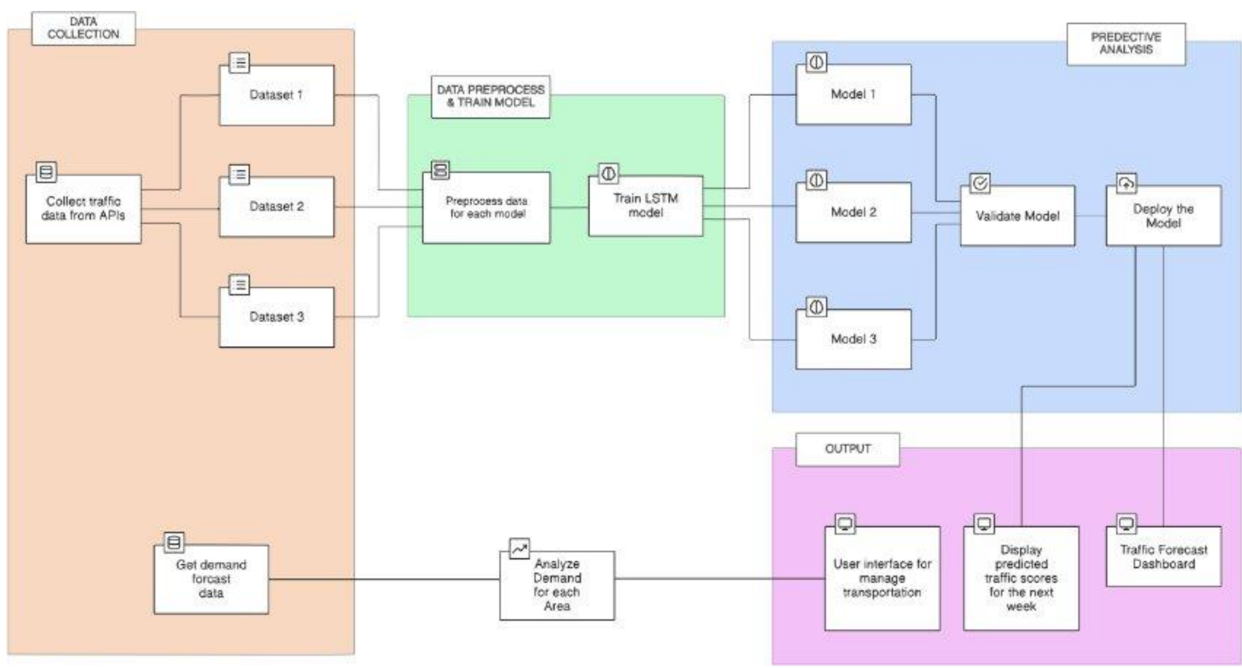


Figure 3 - Component System Diagram - Logistic Management

The logistics management component is a vital part of the overall tea supply chain optimization system. This module focuses on enhancing the transportation efficiency of tea delivery from the Watawala Tea Factory to three selected destination regions: **Colombo**, **Kalutara**, and **Nittambuwa (Gampaha District)**. The component leverages machine learning to forecast traffic patterns and propose optimized delivery schedules that minimize travel time and transportation costs. The following description outlines the internal architecture and flow of this component, as visualized in its system component diagram.

The process begins with the **collection of historical traffic data** using the **Google Maps Traffic API**. Separate datasets were gathered for each of the three delivery locations, capturing traffic flow patterns during critical tea delivery windows. These datasets form the foundation of the model's learning and prediction capability. The gathered data include time series of traffic congestion levels and travel durations, spanning multiple weeks to ensure sufficient variability and representativeness.

Once data collection is complete, the next phase is **data preprocessing**. This stage involves cleaning the datasets to remove anomalies, formatting timestamps, handling missing values, and normalizing traffic score metrics. Preprocessing ensures that the data is structured and consistent for effective model training.

The cleaned datasets are then used to **train Long Short-Term Memory (LSTM)** models—a deep learning technique highly effective for time series forecasting. A **separate LSTM model** is developed for each destination (Colombo, Kalutara, and Nittambuwa), allowing for location-specific traffic behavior learning. The models were trained using **TensorFlow**, and experiments were run and tuned via **Google Colab** for performance testing and iteration.

Once training is complete, each model undergoes **validation** using test datasets to ensure reliable forecasting. After achieving satisfactory accuracy levels, the trained models are **deployed** within the system using a **Flask server**, allowing real-time prediction access and integration with the rest of the application.

The next critical input comes from the **Demand Forecasting** component. This module provides estimated tea supply quantities required for each of the three locations. These values are essential to determine the **number of transportation vehicles** needed per location, ensuring the logistics plan is demand-aligned.

The **final output** of this component is a **user interface** tailored for the transportation manager. This interface provides the following functionalities:

- **Display of Predicted Traffic Scores**: Visual graphs and numerical scores for the upcoming delivery windows for each location.
- **Recommended Delivery Time Slots**: Based on traffic forecasts, the system suggests optimal times to dispatch deliveries to avoid peak congestion.
- **Vehicle Allocation Guidance**: The interface shows the estimated number of vehicles needed for each route based on predicted demand and available traffic windows.
- **Traffic Forecast Dashboard**: A dashboard that combines real-time and historical insights, offering visualizations that support data-driven decision-making.

Overall, this component plays a crucial role in connecting production outputs to market needs by intelligently planning transport routes and timings. By integrating predictive modeling with real-world logistics, the system helps the Watawala Tea Factory reduce delivery delays, lower fuel costs, and ensure timely tea distribution.

2.1.7 Key Pillars of the Research Domain

The key pillars of this research domain are fundamental to achieving the optimization of logistics management within the tea supply chain. These pillars serve as the backbone of the solution, leveraging advanced technologies and methodologies to create an efficient, predictive, and adaptive system.

- **Machine Learning**

Machine learning is a core pillar of this research, particularly in the form of time-series prediction using LSTM (Long Short-Term Memory) models. By applying machine learning techniques, the system can predict future traffic conditions and determine optimal delivery schedules. Machine learning allows the system to learn from historical traffic data, improving its accuracy over time. This approach not only forecasts demand but also dynamically adapts to changing traffic patterns, offering more precise and timely logistics solutions.

While several machine learning models can be used for traffic prediction, **LSTM** is particularly suited for handling the sequential nature of traffic data. Below is a comparison table showing how LSTM compares to other models, such as **Linear Regression**, **Random Forest**, and **Support Vector Machines (SVM)**, to demonstrate why LSTM is the best fit for the logistics management component of this research.

Model	Ability to handle time dependencies	Accuracy for the time series data	Adaptability to changing data	Handling of sequential data
Linear Regression	Poor	Low	Low	Not suitable
Random forest	Moderate	Moderate	Moderate	Poor
Support Vector Machines(SVM)	Moderate	Moderate	Low	Poor
LSTM(Long Short-Term Memory)	Excellent	High	High	Excellent

Table 2 - Model Features Comparison

Why LSTM is the Best Choice for This Component:

- **Time Dependencies**: LSTM is specifically designed to model sequential data, such as traffic conditions over time. It can effectively learn from past data, capturing the inherent patterns and trends in traffic flow that are essential for making accurate predictions.
- **Accuracy**: LSTM typically outperforms traditional models (like linear regression) in accuracy when applied to time series forecasting tasks, such as traffic prediction, due to its ability to handle complex, non-linear relationships.
- **Adaptability**: As traffic conditions are highly variable, LSTM models can continuously update their predictions based on new data, adapting to changes over time. This dynamic learning approach ensures the system remains accurate and relevant.
- **Sequential Data Handling**: Unlike models such as Random Forest or SVM, which may not perform well on sequential data, LSTM's architecture is inherently designed to understand and process time-based inputs, making it optimal for predicting traffic conditions over time.

- **Time Series Forecasting**

Time-series forecasting is essential for predicting traffic conditions over a specified period. In this study, the LSTM model is utilized for its ability to understand the temporal dependencies and complex patterns inherent in traffic data. Time-series forecasting allows the model to predict how traffic congestion will evolve during critical delivery windows, thus optimizing route selection and delivery timing. The dynamic nature of this prediction helps minimize delays and reduce fuel consumption.

- **Traffic Data Analytics**

Analyzing traffic data is crucial for predicting delivery windows and determining vehicle requirements. The use of historical traffic data, collected via the Google Traffic API, enables the model to learn from past traffic patterns and improve future predictions. Traffic data analytics helps in assessing peak congestion times, traffic flow, and duration, which are used to create actionable insights for optimizing delivery schedules.

- **Optimization Techniques**

Optimization techniques are integral to this research, as they allow for the efficient allocation of resources, such as delivery vehicles. With the forecasted traffic data, the system can recommend the best delivery time slots and allocate the required number of vehicles. This reduces the number of vehicles on the road, minimizes delivery delays, and ensures that tea is transported in an optimal, cost-efficient manner.

2.1.8 Data Acquisition

Data acquisition plays a crucial role in the success of the traffic prediction system for optimizing tea delivery logistics. The effectiveness of the LSTM-based traffic forecasting model is directly dependent on the quality, volume, and relevance of the data collected. For this research, the primary data required was historical traffic data for key delivery routes from the Watawala Tea Factory to three major delivery destinations: Colombo, Kalutara, and Nittambuwa (Gampaha District). To ensure reliable and consistent data for each route, a separate dataset was created for each location.

All traffic data was collected using the Google Maps Traffic API, a robust and widely trusted data source for real-time and historical traffic information. The API provided details such as travel duration, traffic congestion levels, timestamped travel data, and location-specific metadata. These data points were essential to accurately model the changing traffic patterns during the tea factory's typical delivery windows.

The data collection process involved writing Python scripts that sent requests to the Google API for each route at regular intervals throughout the day. These requests were automated and repeated over multiple weeks to gather sufficient time-series data. Data was gathered during peak and off-peak hours, on weekdays and weekends, ensuring a diverse range of traffic patterns were captured across different conditions and scenarios. This helped improve the generalization capability of the predictive model.

Each dataset was stored in structured tabular format, with key columns including:

- Timestamp – The exact date and time the traffic data was recorded.
- Origin and Destination – The delivery route (Watawala to Colombo, Kalutara, or Nittambuwa).
- Traffic Duration – Estimated travel time based on current and historical traffic.
- Congestion Level – A qualitative indicator of traffic intensity (low, moderate, high).
- Day of Week – Used to identify weekly traffic trends.
- Hour of Day – For identifying peak traffic hours.

To maintain clarity and separation of data, three distinct datasets were created and labeled accordingly:

- Colombo Dataset
- Kalutara Dataset
- Nittambuwa Dataset

Each dataset will be illustrated in the report using respective table-format images, showcasing how the data is structured and highlighting the uniform schema followed across all three locations. These visuals help to understand how the data flows into the LSTM model training process.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	query	way	distance(meters)	distance_label	origin	origin_coordinates	destination	destination_coordina	duration_min(minute	duration_max(minut	duration(minutes)	road_distance_timin	query_origin	query_destination	timestamp	datetime_utc
2	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	250	220	{"0": 0, "14": 1, "140	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717182000	05/31/2024 19:00:00
3	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	260	230	{"0": 0, "14": 1, "140	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717182000	05/31/2024 19:00:00
4	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	210	250	230	{"0": 0, "14": 1, "140	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717182000	05/31/2024 19:00:00
5	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	230	210	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717196400	05/31/2024 23:00:00
6	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	230	210	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717196400	05/31/2024 23:00:00
7	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	240	220	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717196400	05/31/2024 23:00:00
8	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	180	210	200	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717210800	06/01/2024 3:00:00
9	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	220	200	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717210800	06/01/2024 3:00:00
10	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	220	210	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717210800	06/01/2024 3:00:00
11	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	250	220	{"0": 0, "241": 27, "2	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717225200	06/01/2024 7:00:00
12	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	270	220	{"0": 0, "241": 27, "2	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717225200	06/01/2024 7:00:00
13	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	260	220	{"0": 0, "241": 27, "2	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717225200	06/01/2024 7:00:00
14	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	260	230	{"0": 0, "14": 2, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717239600	06/01/2024 11:00:00
15	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	270	230	{"0": 0, "14": 2, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717239600	06/01/2024 11:00:00
16	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	210	260	230	{"0": 0, "14": 2, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717239600	06/01/2024 11:00:00
17	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	260	220	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717254000	06/01/2024 15:00:00
18	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	280	230	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717254000	06/01/2024 15:00:00
19	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	270	230	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717254000	06/01/2024 15:00:00
20	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	250	220	{"0": 0, "14": 2, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717268400	06/01/2024 19:00:00
21	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	260	220	{"0": 0, "14": 2, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717268400	06/01/2024 19:00:00
22	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	210	260	230	{"0": 0, "14": 2, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717268400	06/01/2024 19:00:00
23	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	180	220	200	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717282800	06/01/2024 23:00:00
24	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	220	200	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717282800	06/01/2024 23:00:00
25	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	220	210	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717282800	06/01/2024 23:00:00
26	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	180	210	190	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717297200	06/02/2024 3:00:00
27	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	220	200	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717297200	06/02/2024 3:00:00
28	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	220	210	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717297200	06/02/2024 3:00:00
29	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	180	240	200	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717311600	06/02/2024 7:00:00
30	Watawala, Sri Lanka	Avissawella - Hatton	111889	112 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	180	250	210	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717311600	06/02/2024 7:00:00
31	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	240	210	{"0": 0, "14": 1, "100	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717311600	06/02/2024 7:00:00
32	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	250	220	{"0": 0, "14": 1, "293	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717326000	06/02/2024 11:00:00
33	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	260	220	{"0": 0, "14": 1, "293	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717326000	06/02/2024 11:00:00
34	Watawala, Sri Lanka	Avissawella - Hatton	113674	114 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	200	250	220	{"0": 0, "14": 1, "293	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717326000	06/02/2024 11:00:00
35	Watawala, Sri Lanka	Avissawella - Hatton	121644	122 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	260	220	{"0": 0, "14": 2, "293	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717340400	06/02/2024 15:00:00
36	Watawala, Sri Lanka	Avissawella - Hatton	110150	110 km	GANKA'S PAWN BR	6.94494679999999	11 Bagatalle Rd, Co	6.900996, 79.85488	190	280	220	{"0": 0, "14": 2, "293	GANKA'S PAWN BR	11 Bagatalle Rd, Co	1717340400	06/02/2024 15:00:00

Figure 4 - Collected Historical Traffic Data from Google API

This organized and location-specific approach to data acquisition ensures that the traffic forecasting models are built on strong, relevant, and consistent data foundations. Furthermore, the collected data can be extended or updated in the future for retraining the models or adapting to evolving traffic conditions, making the system sustainable and scalable over time.

2.1.9 Data Preprocessing

After acquiring traffic datasets for Colombo, Kalutara, and Nittambuwa using the Google Maps Traffic API, the next critical step was data preprocessing. Since the predictive capability of the LSTM model heavily depends on the quality and consistency of input data, this phase ensured that the raw datasets were properly cleaned, formatted, and transformed before model training.

Each dataset contained multiple attributes, such as timestamp, travel duration, origin-destination, congestion levels, and day/time-related information. While the collected data provided a strong base, it included various irregularities and inconsistencies that needed to be addressed before feeding into the model.

Data Cleaning

The raw traffic data occasionally contained missing, duplicate, or incomplete records due to intermittent API responses or temporary connectivity issues. These anomalies were removed through:

- Dropping missing values: Rows with empty travel durations or timestamps were excluded.
- Removing duplicates: Any repeated entries captured in the same minute for the same route were filtered out.
- Filtering invalid entries: Unusual spikes in travel time (e.g., extremely low or high durations) that didn't match typical traffic behavior were manually reviewed and eliminated to maintain data integrity.

Feature Extraction and Engineering

The LSTM model requires time-series formatted data, so we extracted relevant temporal features and transformed the dataset accordingly:

- Converted timestamps into separate features such as hour of day, day of week, and weekend/weekday indicators.
- Normalized travel durations to keep all input features within the same scale, improving model convergence during training.
- Encoded categorical features (e.g., congestion levels like low/medium/high) into numerical values, allowing the model to interpret them effectively.

Time-Series Formatting

As LSTM models learn from sequential patterns, the datasets were reshaped into sequences for training. This included:

- Creating sliding time windows (e.g., 5 or 10 time steps per input sequence) to predict the next value in the sequence.
- Grouping travel times over consecutive timestamps to form the input series (X) and corresponding target values (y).
- Ensuring data was sorted chronologically before windowing, maintaining sequence integrity.

Data Normalization

Since LSTM models are sensitive to the scale of input data, numerical values such as travel duration and congestion scores were normalized using Min-Max scaling to bring all values into a 0 to 1 range. This helped accelerate the model training process and stabilized learning.

5. Splitting Data for Training and Testing

Each location dataset was split into:

- Training set (80%) – Used for model learning.
- Testing set (20%) – Used to evaluate model accuracy and generalization.

This ensured that the model's performance was tested on unseen data, reducing the risk of overfitting.

Final Dataset Preparation for Training

After completing all preprocessing steps, the final training datasets were created for each location. These datasets were structured to include:

- Datetime Slot – Representing specific time intervals.
- Average Travel Duration – Calculated as the average of all recorded travel times during that specific time slot for each route.

This final structure allowed the LSTM models to learn the patterns of how travel time fluctuates throughout the day and across different days of the week. Each of the three final datasets—one for Colombo, Kalutara, and Nittambuwa was prepared using this format and was used to train separate LSTM models tailored to the traffic characteristics of each delivery route.

```python
# Final dataset for model training

# Create a new dataframe with the desired columns
df_final = df_final2[['datetime_utc', 'duration_max(minutes)']]

# Display the first few rows to verify
df_final.head(20)
```

	datetime_utc	duration_max(minutes)
0	2024-05-31 18:00:00	250.0
9	2024-06-01 06:00:00	250.0
12	2024-06-01 10:00:00	260.0
15	2024-06-01 14:00:00	260.0
18	2024-06-01 18:00:00	250.0
27	2024-06-02 06:00:00	240.0
30	2024-06-02 10:00:00	250.0
33	2024-06-02 14:00:00	260.0
36	2024-06-02 18:00:00	250.0
45	2024-06-03 06:00:00	250.0
48	2024-06-03 10:00:00	260.0
51	2024-06-03 14:00:00	270.0
54	2024-06-03 18:00:00	250.0
63	2024-06-04 06:00:00	250.0

Figure 5 - Final Dataset for Model Training - COLOMBO

```python
# Final dataset for model training

# Create a new dataframe with the desired columns
df_final1 = df_filtered[['way', 'distance_label', 'datetime_utc', 'duration(minutes)']]

# Display the first few rows to verify
df_final1.head(20)
```

	way	distance_label	datetime_utc	duration(minutes)
0	Avissawella - Hatton - Nuwara Eliya Hwy/A7	80.9 km	2024-05-31 18:00:00	160.0
1	Avissawella - Hatton - Nuwara Eliya Hwy/A7 and...	92.8 km	2024-05-31 18:00:00	170.0
2	Avissawella - Hatton - Nuwara Eliya Hwy/A7 and...	103 km	2024-05-31 18:00:00	210.0
9	Avissawella - Hatton - Nuwara Eliya Hwy/A7	80.9 km	2024-06-01 06:00:00	150.0
10	Avissawella - Hatton - Nuwara Eliya Hwy/A7 and...	92.8 km	2024-06-01 06:00:00	160.0
11	Avissawella - Hatton - Nuwara Eliya Hwy/A7	80.9 km	2024-06-01 10:00:00	160.0
12	Avissawella - Hatton - Nuwara Eliya Hwy/A7 and...	92.8 km	2024-06-01 10:00:00	170.0
13	Avissawella - Hatton - Nuwara Eliya Hwy/A7 and...	103 km	2024-06-01 10:00:00	200.0
14	Avissawella - Hatton - Nuwara Eliya Hwy/A7	80.9 km	2024-06-01 14:00:00	150.0
15	Avissawella - Hatton - Nuwara Eliya Hwy/A7 and...	92.8 km	2024-06-01 14:00:00	170.0
16	Avissawella - Hatton - Nuwara Eliya Hwy/A7 and...	103 km	2024-06-01 14:00:00	200.0
17	Avissawella - Hatton - Nuwara Eliya Hwy/A7	80.9 km	2024-06-01 18:00:00	160.0
18	Avissawella - Hatton - Nuwara Eliya Hwy/A7 and...	92.8 km	2024-06-01 18:00:00	170.0
19	Avissawella - Hatton - Nuwara Eliya Hwy/A7 and...	103 km	2024-06-01 18:00:00	200.0
26	Avissawella - Hatton - Nuwara Eliya Hwy/A7	80.9 km	2024-06-02 06:00:00	140.0
27	Avissawella - Hatton - Nuwara Eliya Hwy/A7 and...	92.8 km	2024-06-02 06:00:00	150.0

Figure 6 - Final Dataset for Model Training - KALUTHARA

```python
# Final dataset for model training

# Create a new dataframe with the desired columns
df_final = df_final2[['datetime_utc', 'duration_max(minutes)']]

# Display the first few rows to verify
df_final.head(20)
```

	datetime_utc	duration_max(minutes)
0	2024-05-31 18:00:00	250.0
9	2024-06-01 06:00:00	260.0
12	2024-06-01 10:00:00	270.0
15	2024-06-01 14:00:00	270.0
18	2024-06-01 18:00:00	260.0
27	2024-06-02 06:00:00	250.0
30	2024-06-02 10:00:00	260.0
33	2024-06-02 14:00:00	270.0
36	2024-06-02 18:00:00	250.0
45	2024-06-03 06:00:00	250.0

Figure 7 - Final Dataset for Model Training - NITTAMBUWA

2.1.10 Model Training

Once the historical traffic data was fully preprocessed and structured into final datasets for each delivery location—Colombo, Kalutara, and Nittambuwa the next stage involved training Long Short-Term Memory (LSTM) models to forecast traffic conditions. LSTM is a type of Recurrent Neural Network (RNN) that is well-suited for time-series forecasting due to its ability to learn long-term dependencies in sequential data. It was selected for this research because of its superior performance in modeling patterns across time-based datasets compared to traditional models.

Training Approach

Separate LSTM models were trained for each of the three locations. This approach ensured that each model could specialize in learning the unique traffic flow patterns of its respective route. The training was conducted using Python, with the help of TensorFlow and Keras libraries, which offer powerful tools for building and tuning deep learning models.

The input data for the LSTM model consisted of sequences of normalized travel durations and associated time features. A sliding window technique was used to generate these sequences, where a fixed number of previous time steps were used to predict the next travel time value. For example, if a window size of 10 was used, the model would learn to predict the 11th time step based on the previous 10.

Model Architecture

The LSTM model architecture was composed of:

- Input Layer – Receiving sequences of traffic data (e.g., average travel duration, hour of day, day of week).
- One or Two LSTM Layers – To capture sequential dependencies and traffic trends.
- Dropout Layer – To reduce overfitting by randomly disabling a fraction of neurons during training.
- Dense (Output) Layer – Producing the predicted travel time for the next time step.

The model was trained with Mean Squared Error (MSE) as the loss function and the Adam optimizer to update weights efficiently. Multiple epochs were used to improve accuracy while monitoring validation performance to prevent overfitting.

```python
# Define the LSTM Model

# Define the model
model = Sequential([
    LSTM(64, activation='tanh', return_sequences=False, input_shape=(X_train.shape[1], 1)),
    Dropout(0.2),
    Dense(32, activation='relu'),
    Dense(time_steps)  # Predict 28 future values
])

# Compile the model
model.compile(optimizer='adam', loss='mse', metrics=['mae'])

# Summary of the model
model.summary()
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/rnn/rnn.py:204: UserWarning: Do not pass an
  super().__init__(**kwargs)
Model: "sequential_2"
```

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 64)	16,896
dropout_2 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 32)	2,080
dense_5 (Dense)	(None, 28)	924

```
Total params: 19,900 (77.73 KB)
Trainable params: 19,900 (77.73 KB)
Non-trainable params: 0 (0.00 B)
```

Figure 8 - Define the LSTM Model

Training Platform and Tools

The model training process was executed on Google Colab, which provided free access to GPU acceleration, speeding up the training of deep learning models significantly. Additional tools such as Pandas were used for data manipulation, and Matplotlib was used to visualize training loss curves and prediction results.

Each model was trained until it reached a stable and acceptable loss level, indicating that the model had effectively learned from the training data. Hyperparameters such as batch size, number of epochs, and number of LSTM units were tuned through experimentation to find the best configuration for each location's dataset.

```
# Train the Model

# Early stopping to prevent overfitting
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

# Train the model
history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=100,
    batch_size=32,
    callbacks=[early_stopping]
)
```

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 0.0994 - mae: 0.2733 - val_loss: 0.0935 - val_mae: 0.2658
Epoch 36/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 0.0995 - mae: 0.2744 - val_loss: 0.0934 - val_mae: 0.2703
Epoch 37/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - loss: 0.0983 - mae: 0.2732 - val_loss: 0.0937 - val_mae: 0.2712
Epoch 38/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 0.0989 - mae: 0.2745 - val_loss: 0.0932 - val_mae: 0.2664
Epoch 39/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 0.0989 - mae: 0.2739 - val_loss: 0.0937 - val_mae: 0.2711
Epoch 40/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 0.0987 - mae: 0.2737 - val_loss: 0.0935 - val_mae: 0.2709
Epoch 41/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 0.0977 - mae: 0.2725 - val_loss: 0.0925 - val_mae: 0.2666
Epoch 42/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 0.0961 - mae: 0.2705 - val_loss: 0.0923 - val_mae: 0.2673
Epoch 43/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - loss: 0.0969 - mae: 0.2724 - val_loss: 0.0925 - val_mae: 0.2698
Epoch 44/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 0.0949 - mae: 0.2682 - val_loss: 0.0933 - val_mae: 0.2717
Epoch 45/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0981 - mae: 0.2724 - val_loss: 0.0921 - val_mae: 0.2669
Epoch 46/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 0.0988 - mae: 0.2743 - val_loss: 0.0922 - val_mae: 0.2664
Epoch 47/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0960 - mae: 0.2706 - val_loss: 0.0938 - val_mae: 0.2725
```

Figure 9 - Model Training - COLOMBO

```python
# Train the Model

# Early stopping to prevent overfitting
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

# Train the model
history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=100,
    batch_size=32,
    callbacks=[early_stopping]
)
```

```
Epoch 49/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 0.1060 - mae: 0.2725 - val_loss: 0.1001 - val_mae: 0.2616
Epoch 50/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 0.1066 - mae: 0.2726 - val_loss: 0.0990 - val_mae: 0.2566
Epoch 51/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.1048 - mae: 0.2671 - val_loss: 0.0966 - val_mae: 0.2562
Epoch 52/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 0.1026 - mae: 0.2674 - val_loss: 0.0950 - val_mae: 0.2588
Epoch 53/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 0.1073 - mae: 0.2744 - val_loss: 0.0936 - val_mae: 0.2522
Epoch 54/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 0.1019 - mae: 0.2639 - val_loss: 0.0930 - val_mae: 0.2490
Epoch 55/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 0.1010 - mae: 0.2615 - val_loss: 0.0912 - val_mae: 0.2506
Epoch 56/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 0.1020 - mae: 0.2667 - val_loss: 0.0894 - val_mae: 0.2481
Epoch 57/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 0.0979 - mae: 0.2601 - val_loss: 0.0881 - val_mae: 0.2421
Epoch 58/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 0.0965 - mae: 0.2537 - val_loss: 0.0860 - val_mae: 0.2399
Epoch 59/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - loss: 0.0969 - mae: 0.2583 - val_loss: 0.0843 - val_mae: 0.2385
Epoch 60/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0963 - mae: 0.2566 - val_loss: 0.0827 - val_mae: 0.2356
Epoch 61/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 0.0958 - mae: 0.2554 - val_loss: 0.0814 - val_mae: 0.2303
```

Figure 10 - Model Training - KALUTHARA

```python
# Train the Model

# Early stopping to prevent overfitting
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

# Train the model
history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=100,
    batch_size=32,
    callbacks=[early_stopping]
)
```

```
Epoch 1/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 152ms/step - loss: 0.3681 - mae: 0.5114 - val_loss: 0.3522 - val_mae: 0.5018
Epoch 2/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - loss: 0.3427 - mae: 0.4955 - val_loss: 0.3256 - val_mae: 0.4837
Epoch 3/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 0.3141 - mae: 0.4759 - val_loss: 0.2913 - val_mae: 0.4545
Epoch 4/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 0.2786 - mae: 0.4442 - val_loss: 0.2522 - val_mae: 0.4101
Epoch 5/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 0.2375 - mae: 0.3983 - val_loss: 0.2221 - val_mae: 0.3769
Epoch 6/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 0.2109 - mae: 0.3688 - val_loss: 0.1902 - val_mae: 0.3469
Epoch 7/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 0.1880 - mae: 0.3494 - val_loss: 0.1671 - val_mae: 0.3237
Epoch 8/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 0.1698 - mae: 0.3304 - val_loss: 0.1547 - val_mae: 0.3124
Epoch 9/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 0.1536 - mae: 0.3138 - val_loss: 0.1454 - val_mae: 0.3011
Epoch 10/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.1511 - mae: 0.3117 - val_loss: 0.1378 - val_mae: 0.2904
Epoch 11/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 0.1410 - mae: 0.2981 - val_loss: 0.1338 - val_mae: 0.2814
Epoch 12/100
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 0.1360 - mae: 0.2894 - val_loss: 0.1294 - val_mae: 0.2735
Epoch 13/100
```

Figure 11- Model Training - NITTAMBUWA

2.1.11 Model Evaluation and Tuning

After training the LSTM models for Colombo, Kalutara, and Nittambuwa, a thorough evaluation process was conducted to ensure that the models could reliably forecast traffic conditions for each location. The performance of the models was assessed using standard evaluation metrics and visual inspections of predicted versus actual values. Additionally, several hyperparameter tuning techniques were applied to enhance the accuracy and generalization of the models.

```python
# Evaluate the Model

# Evaluate on test data
test_loss, test_mae = model.evaluate(X_test, y_test, verbose=0)
print(f"Test Loss: {test_loss:.4f}, Test MAE: {test_mae:.4f}")
```

Figure 12 - Model Evaluation

Evaluation Metrics

To measure the performance of each model, the following evaluation metrics were used:

- Mean Squared Error (MSE): Measures the average of the squared differences between predicted and actual travel durations. A lower MSE indicates better model performance.
- Root Mean Squared Error (RMSE): The square root of MSE, giving error in the same units as the original data (minutes). It provides an interpretable measure of prediction accuracy.
- Mean Absolute Error (MAE): Calculates the average magnitude of errors without considering their direction, offering a more robust measure in the presence of outliers.

For each location, these metrics were calculated on the testing dataset (20% of the full dataset), which had not been seen by the model during training. This helped in evaluating how well the model could generalize to unseen data.

Prediction Visualization

Apart from numerical metrics, model performance was also evaluated by plotting predicted versus actual travel durations over time. These visualizations were created using Matplotlib, helping to clearly observe how closely the model's predictions tracked real-world traffic patterns. Where predictions significantly diverged from actual values, further adjustments were made to the model or preprocessing steps.

Hyperparameter Tuning

Initial model training was followed by systematic hyperparameter tuning to optimize performance. The following hyperparameters were explored and fine-tuned:

- Number of LSTM units: Adjusted to balance between learning capacity and overfitting risk.
- Batch size: Tested with smaller and larger batch sizes to evaluate training stability and accuracy.
- Epochs: Trained over multiple epochs while monitoring validation loss to identify the point of convergence.

- Learning rate: Modified to improve the speed and quality of training.

This tuning process was conducted manually through trial-and-error and supported by loss curve visualizations, which were analyzed to avoid overfitting or underfitting. Dropout layers and validation checks were introduced to prevent performance degradation on unseen data.

Cross-Validation Checks

While full k-fold cross-validation was not used due to the sequential nature of the time-series data, a time-based validation strategy was applied. This involved training the model on one period of time and testing on a future period, simulating how the model would perform in a real-world forecasting scenario.

2.1.12 Deployment

After successful model training and evaluation, the next critical step was the deployment of the traffic forecasting system. The deployment phase involved integrating the trained LSTM models into a functioning software system that could be accessed and used by the transportation manager to make real-time decisions regarding delivery scheduling.

Model Export and Integration

Each LSTM model—trained individually for Colombo, Kalutara, and Nittambuwa was exported in a serialized format using TensorFlow's .h5 model saving format. These models were then integrated into a Flask-based backend server, which serves as the core of the application's logic.

The Flask server was responsible for:

- Loading the appropriate trained model based on the selected location.
- Accepting user requests (e.g., desired delivery date or time slot).
- Feeding relevant input sequences into the LSTM model.
- Returning the predicted average travel duration for that time window.

This setup allowed for dynamic prediction based on updated or scheduled inputs, simulating near real-time traffic forecasting for transportation planning.

User Interface Deployment

The predictions from the backend were displayed via a web-based interface designed for the transportation manager. The interface included:

- A dashboard showing predicted traffic conditions across selected time slots for each location.
- The recommended number of vehicles required to transport the forecasted tea supply.
- The best time slot for each delivery route, optimized for reduced congestion and efficiency.

This user interface was deployed as part of the same Flask application and accessed through a local network or hosted service for testing.

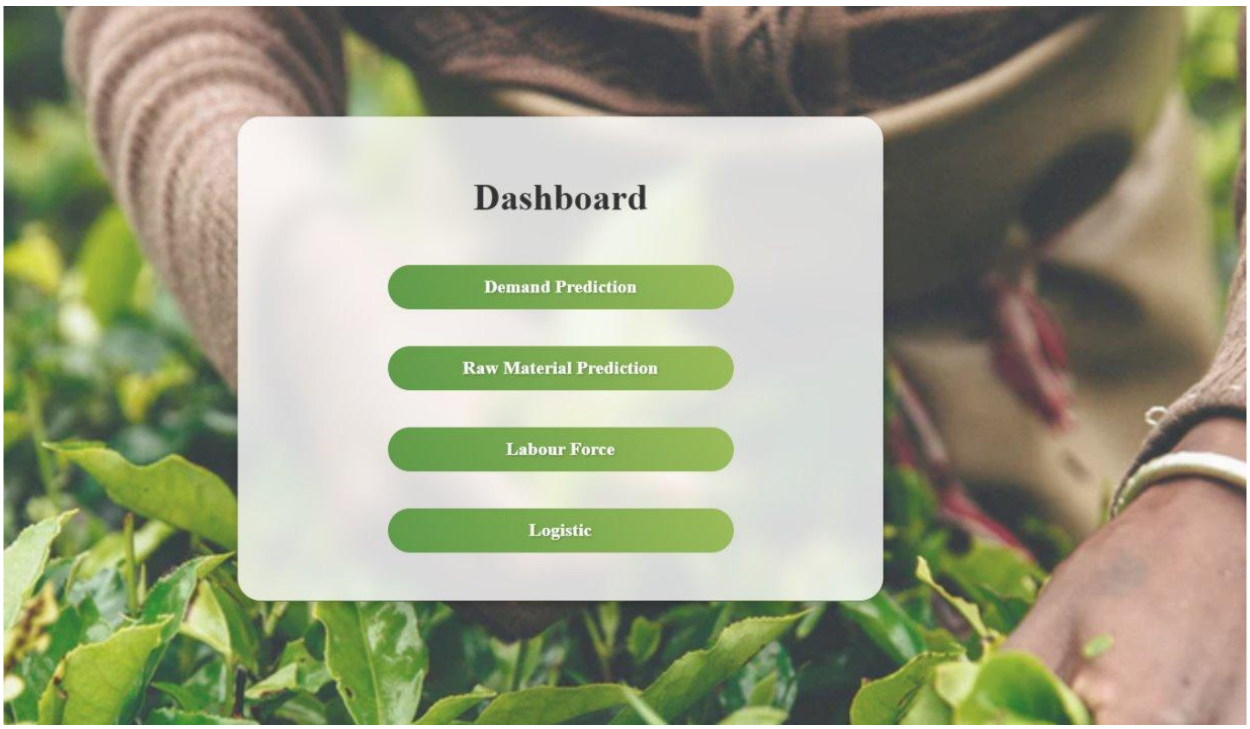


Figure 13 - User Interface - Full Dashboard

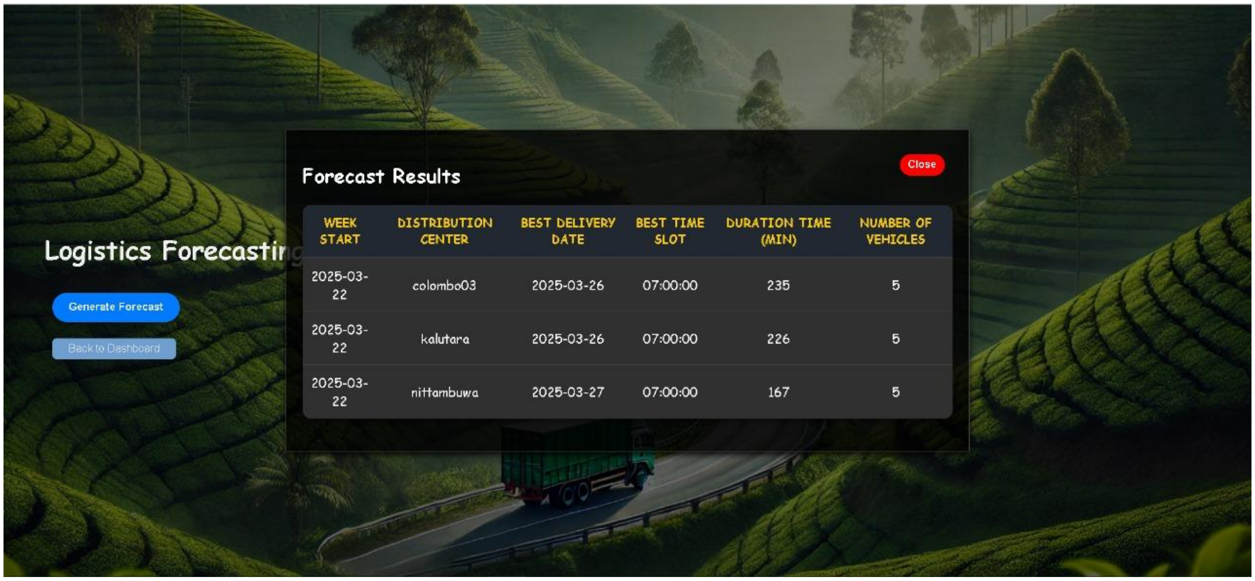


Figure 14 - User Interface - Logistic Forecasting Dashboard

Model Hosting and Testing

The entire application was deployed and tested on Google Colab during the development phase to leverage cloud-based GPU acceleration. For local deployment, the Flask server was hosted on a development machine, ensuring end-to-end integration between the model, backend, and front-end.

The system was tested with:

- Realistic input scenarios to simulate future delivery schedules.
- Mock supply data received from the demand forecasting component.
- Varied date/time inputs to evaluate the robustness of predictions.

Post-Deployment Validation

Once deployed, the system underwent validation with historical data to confirm that model outputs remained consistent with known traffic patterns. This process ensured the deployment pipeline worked correctly and that the user interface reliably presented accurate and actionable forecasts.

2.2 Commercialization Aspects of the Product

2.2.1 Market Potential

The tea industry in Sri Lanka plays a significant role in the national economy, with logistics and distribution forming a critical part of the supply chain. Inefficiencies in transportation such as delays due to traffic congestion and poor delivery timing lead to increased costs, reduced freshness, and lower overall supply chain performance. This creates a strong market demand for intelligent logistics solutions.

The proposed traffic forecasting and delivery scheduling system has substantial market potential not only within the tea industry but also across various sectors that rely on timely deliveries, such as agriculture, food, retail, and logistics. Factories like Watawala and others with similar operations can benefit from integrating predictive logistics into their transport management systems.

With increasing digital transformation in agriculture and manufacturing, solutions based on machine learning offer a competitive edge. The scalability of this system allows it to be extended to other regions and even other types of supply chains, making it commercially viable in both local and global markets.

2.2.2 Business Model

The proposed solution can be commercialized using a Software-as-a-Service (SaaS) model, where clients (factories or transport companies) subscribe to the system based on their usage level. The business model may include:

- Subscription Plans – Monthly or yearly subscription tiers based on the number of delivery locations, volume of data, and user access.
- Customization Services – Additional revenue from personalized integration or advanced analytics tailored for specific client needs.
- Freemium Model – A basic version can be offered for free with limited features to attract initial users and then convert them to paid users for full access.
- Consulting and Deployment Support – One-time charges for setup, staff training, and on-site deployment.

This model ensures recurring revenue while offering flexibility to clients based on their operational scale.

2.2.3 SWOT Analysis

Strengths	Weaknesses
✓ Uses advanced ML (LSTM) for accurate predictions	✗ Requires access to reliable traffic data
✓ Real-time, location-specific forecasting	✗ Initial setup and integration may be complex
✓ Scalable and applicable across industries	✗ Dependent on internet connectivity and server uptime
✓ Reduces transport costs and delays	✗ Needs technical expertise for maintenance

Table 3 - Strengths vs Weaknesses

This table outlines the internal factors influencing the success of the system. The strengths highlight the unique capabilities and technical advantages of the proposed solution, such as its use of LSTM for accurate traffic prediction and its scalability across industries. Meanwhile, the weaknesses shed light on areas that may pose challenges, including dependency on external data sources and the need for technical expertise during setup and maintenance.

Opportunities	Threats
✓ Can expand beyond tea industry to broader logistics	✗ Emergence of similar AI-based logistics tools
✓ Demand for smart supply chain solutions is growing	✗ Resistance to adoption in traditional businesses
✓ Potential partnerships with logistics platforms	✗ Changes in APIs or data sources (e.g., Google APIs) could disrupt service
✓ Integration with IoT and GPS devices	✗ Cybersecurity and data privacy concerns

Table 4 - Opportunities vs Threats

This table presents the external factors that could impact the system's future. The opportunities describe potential areas for growth and expansion, including integration with other technologies and applications in various industries. On the other hand, the threats reflect risks such as competition, changes in third-party APIs, and the potential hesitation from traditional businesses to adopt AI-based solutions.

2.3 Implementation and Testing

This section outlines how the traffic forecasting component was implemented and tested as part of the overall logistics optimization module. The focus was on converting raw traffic data into actionable predictions using machine learning, particularly LSTM models, and ensuring the system performs reliably through structured testing methods.

2.3.1 Preprocessing and Augmentation

The first step in the implementation was data preprocessing, which was critical to ensure the model received clean, consistent, and meaningful inputs. Historical traffic data collected from Google Maps API for three major delivery locations (Colombo, Kalutara, and Nittambuwa) was first cleaned to remove null values and outliers. Time-based formatting was standardized across all datasets.

To enhance the model's ability to generalize and detect trends, data was aggregated into structured time slots. For each day, average travel times were computed per time slot to reduce noise and improve training stability. Although traditional data augmentation techniques are limited with time series data, sliding window methods were used to generate training sequences from the historical data, allowing the model to learn temporal dependencies effectively.

2.3.2 Model Implementation

Each location had its own LSTM model, implemented using **TensorFlow** and **Keras** libraries within **Google Colab**. The LSTM architecture was chosen for its ability to retain long-term temporal relationships within the traffic data, making it suitable for predicting average travel durations.

Each model was trained using historical sequences with a fixed window size, forecasting the next time slot's average traffic duration. Hyperparameters such as batch size, epochs, and learning rate were optimized through experimentation. After training, models were saved in .h5 format for deployment.

2.3.3 Testing Strategy

To ensure the reliability and accuracy of the developed system, a systematic testing strategy was followed. The testing strategy included both:

- **Unit Testing**: Applied to individual components such as preprocessing scripts, API data parsers, and model input/output functions.
- **Integration Testing**: Verified the interaction between the LSTM model, Flask server, and the user interface, ensuring seamless data flow and real-time predictions.

Testing was performed iteratively during development using Agile principles, allowing rapid identification and correction of issues.

2.3.4 Test Cases and Tools

Several test cases were designed to validate core functionalities. These included:

- Input validation for date/time formats.
- Correct loading of the appropriate LSTM model per location.
- Accuracy and consistency of predicted travel durations.
- UI responsiveness and data visualization integrity.

Tools Used:

- **Google Colab**: For model training and initial testing.
- **Flask Test Client**: For backend API route testing.
- **Postman**: To manually test API endpoints and request/response cycles.
- **Browser Console & Dev Tools**: For testing front-end display and interactivity.

Each test case was documented with expected outcomes, and test results were recorded to measure system reliability.

2.3.5 Evaluation Summary

The system was evaluated on both qualitative and quantitative grounds. Quantitatively, the performance of each LSTM model was measured using **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**. All models achieved satisfactory error margins, showing strong ability to predict traffic durations for upcoming weeks.

Qualitatively, the final integrated system was evaluated through simulated usage scenarios and feedback from test users. The dashboard and prediction outputs were found to be intuitive, informative, and practical for logistics planning.

Overall, the testing and evaluation confirmed that the system could reliably support logistics decision-making, thereby enhancing the efficiency and timing of tea deliveries from the Watawala factory to key locations.

3. RESULTS AND DISCUSSION

3.1 Results

The core objective of this component was to enhance the logistics efficiency in the tea supply chain through accurate traffic predictions using LSTM-based models. To evaluate the performance of the developed system, models were trained separately for the three key delivery locations Colombo, Kalutara, and Nittambuwa. The results from model training and testing are presented in this section, supported by relevant evaluation metrics and visualizations.

The system was assessed using standard regression evaluation criteria such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). All three models demonstrated strong learning capabilities with minimal overfitting, indicating their suitability for real-world deployment.

Training and Validation Loss Comparison

For each location, a loss graph was generated to compare the training loss and validation loss during the learning process. These graphs clearly illustrate the learning curve of each model, showcasing convergence and consistency between training and validation phases. The gradual decrease in both losses indicates that the LSTM models effectively learned the patterns in historical traffic data without significant overfitting.

- The Colombo model showed smooth and steady convergence, with minimal gap between training and validation losses.

- The Kalutara model had slightly more variance in the early epochs but stabilized well, indicating good generalization.

- The Nittambuwa model had the lowest overall validation loss, showing excellent adaptability to its specific traffic patterns.

These loss graphs validate the effectiveness of the chosen architecture and preprocessing pipeline.

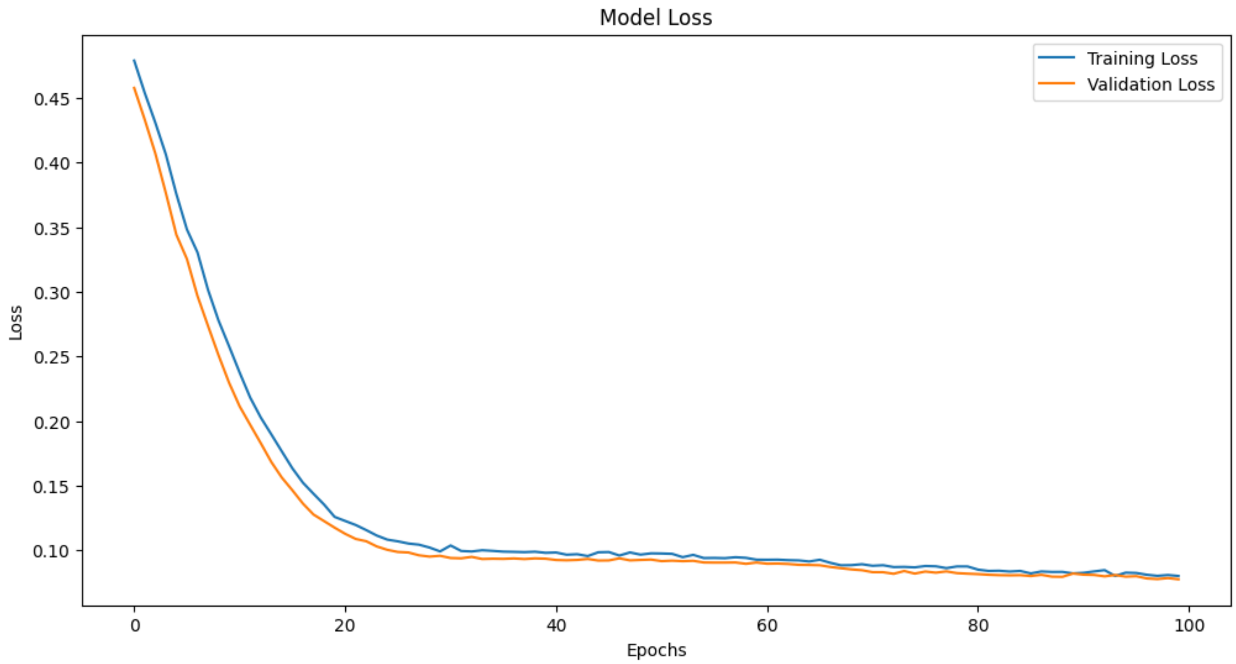


Figure 15 - Model Loss Graph – COLOMBO

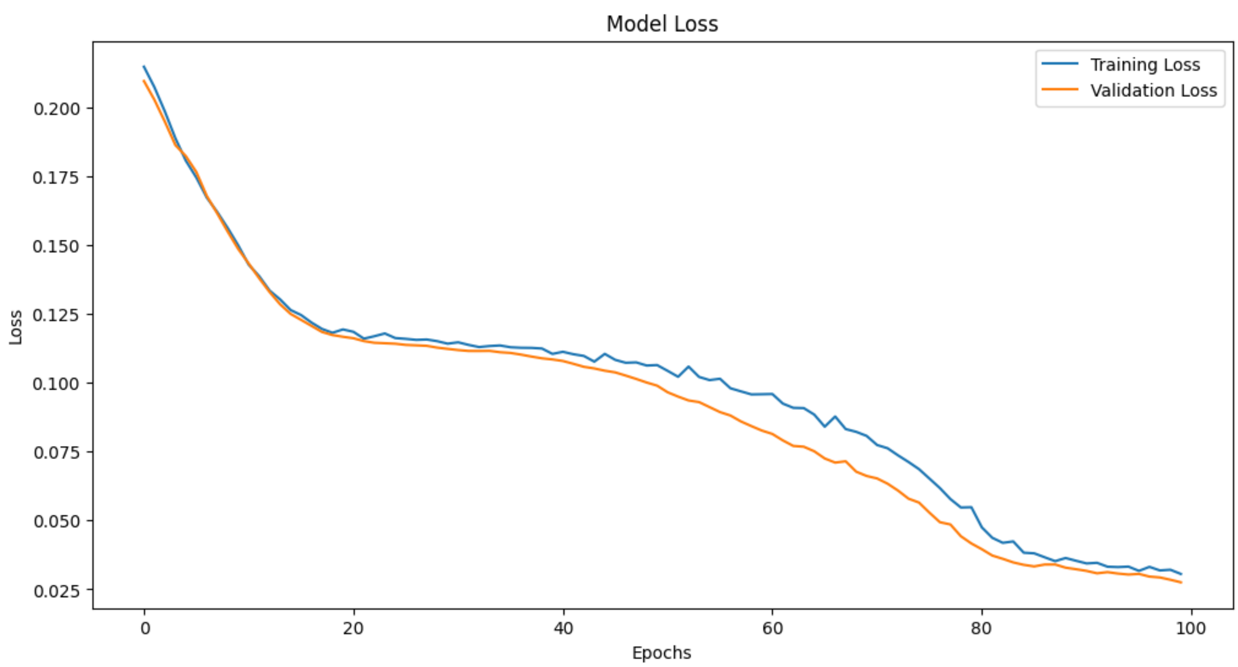


Figure 16 - Model Loss Graph - KALUTHARA

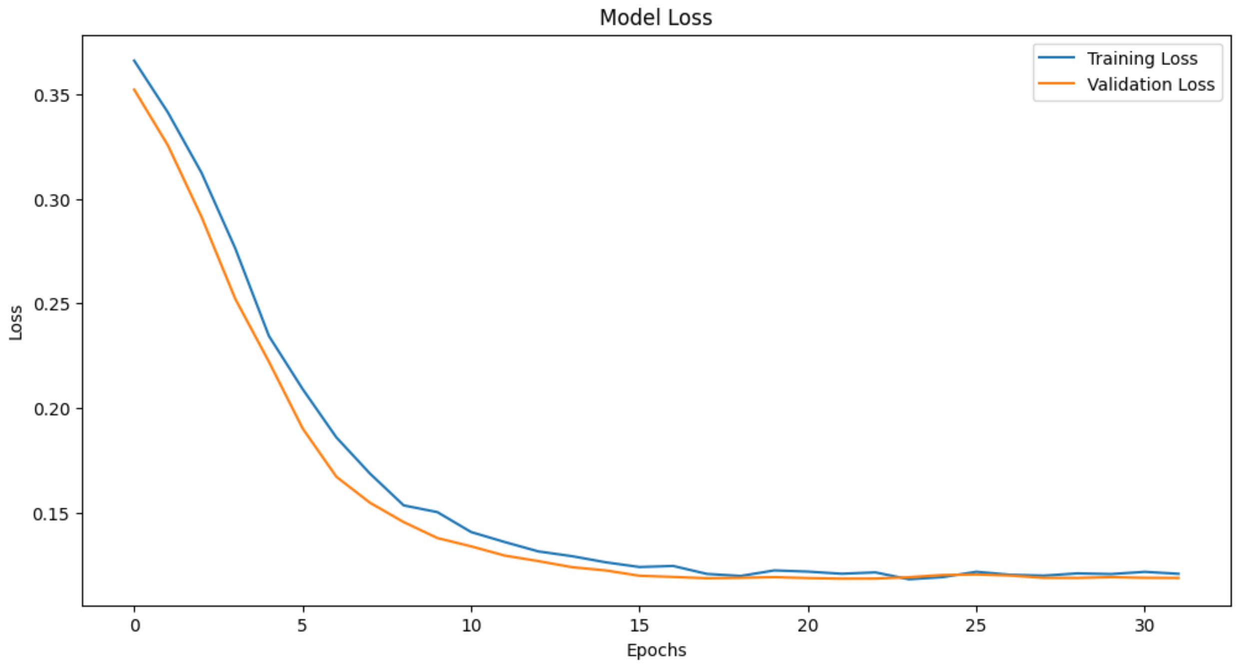


Figure 17 - Model Loss Graph - NITTAMBUWA

True vs. Predicted Value Comparison

Another key result visualization is the true vs. predicted traffic duration graphs. For each location, predicted values were plotted against actual observed values to evaluate how closely the model was able to forecast real-world traffic conditions.

- The Colombo model's predictions closely followed the actual values, with only minor deviations during peak-hour time slots.

- Kalutara’s graph showed a slightly wider error margin but maintained an overall strong correlation.

- Nittambuwa’s model achieved nearly linear alignment between true and predicted values, suggesting very high accuracy for that region.

These graphs visually confirm that the models are able to forecast realistic traffic durations, which can significantly aid in planning optimal delivery times.

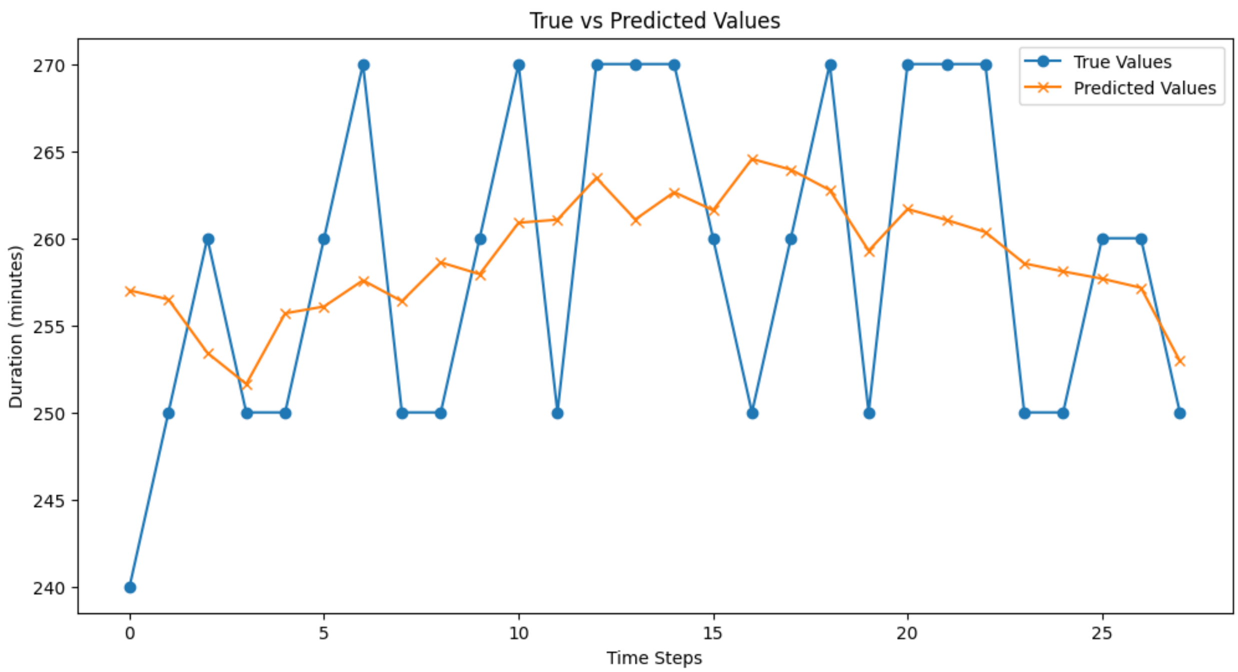


Figure 18 - True vs Predicted Values Comparison Graph - COLOMBO

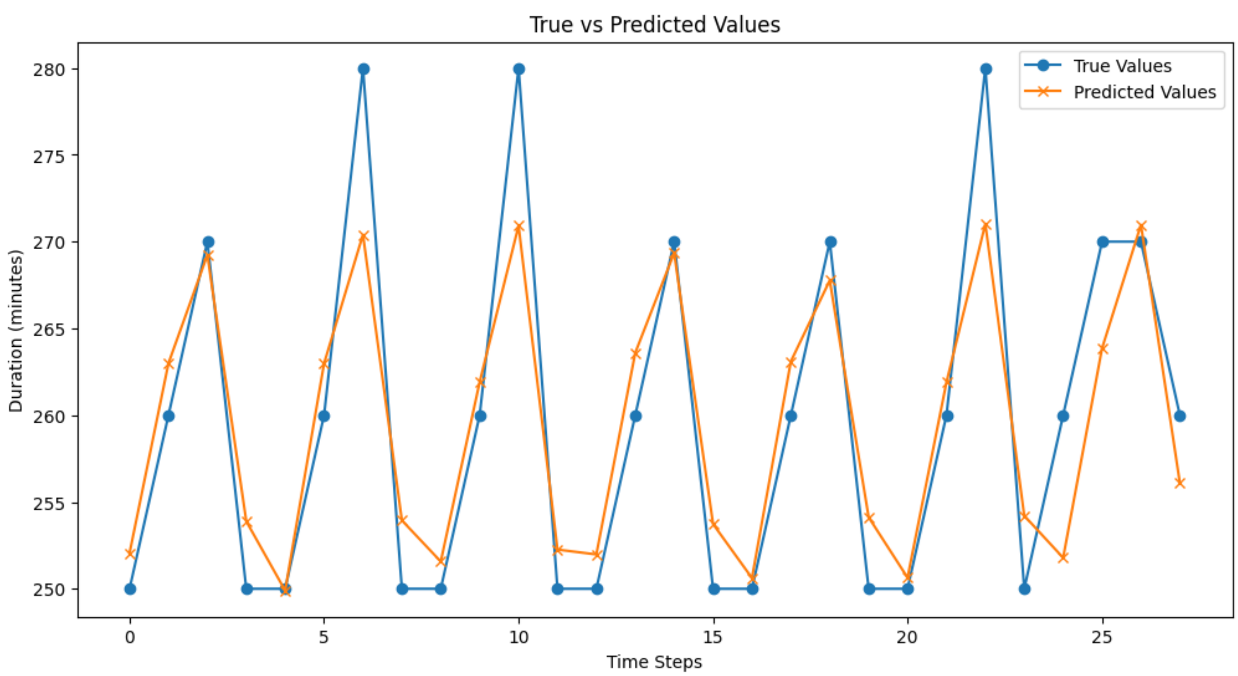


Figure 19 - True vs Predicted Values Comparison Graph - KALUTHARA

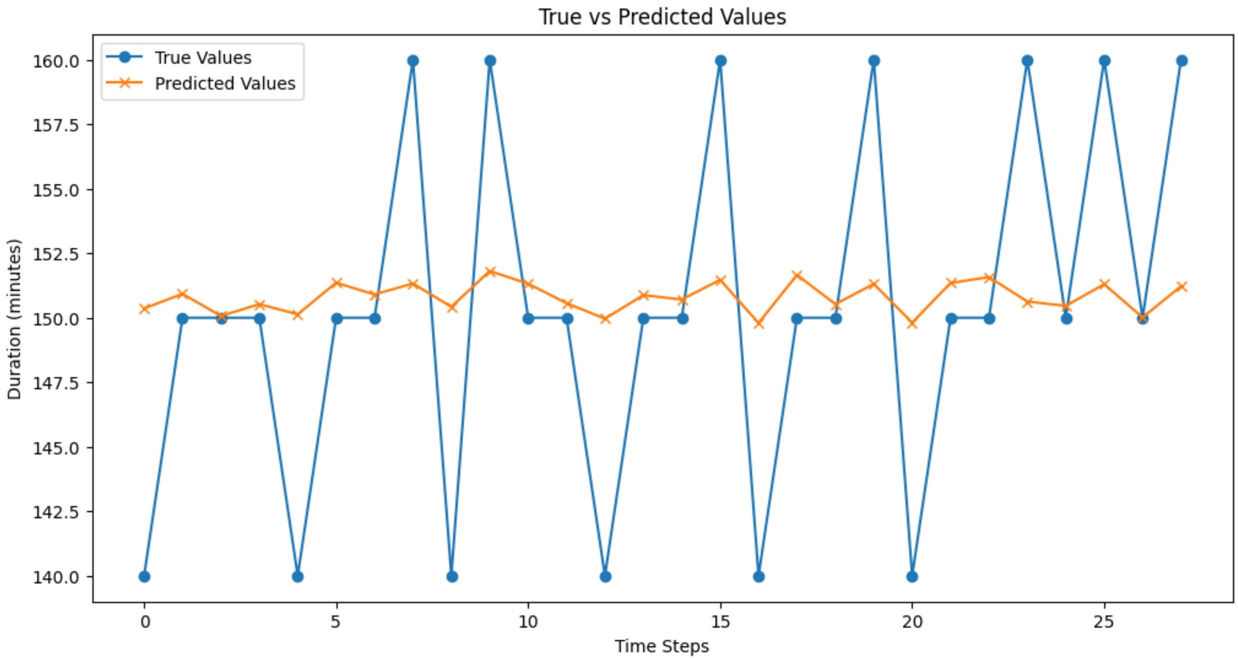


Figure 20 - True vs Predicted Values Comparison Graph - NITTAMBUWA

3.2 Research Findings

The implementation and testing of the traffic prediction system using LSTM models produced several valuable insights relevant to logistics optimization in the tea supply chain. These findings reflect the success of the proposed system and its potential impact on real-world transportation planning.

- **LSTM is Highly Effective for Time Series Traffic Forecasting**

One of the most significant findings was the strength of LSTM (Long Short-Term Memory) networks in handling time series data. The models trained for Colombo, Kalutara, and Nittambuwa achieved high accuracy and stability, maintaining low error rates across all three locations. The architecture's ability to remember long-term patterns in the historical traffic data allowed it to effectively capture peak-hour fluctuations and weekly trends.

Compared to other traditional machine learning algorithms like Decision Trees or Linear Regression, LSTM demonstrated superior performance in both training consistency and validation accuracy, confirming its suitability for traffic prediction tasks.

- **Preprocessed Time Slot-Based Datasets Improve Accuracy**

Another critical discovery was the benefit of transforming raw traffic data into structured time slots and computing average durations for each slot. This method reduced noise, highlighted consistent patterns, and enhanced the model's ability to learn useful features. It also aligned well with the practical requirements of transportation scheduling, where decisions are typically made based on blocks of time rather than specific moments.

This preprocessing approach not only improved model performance but also helped the transportation manager make better-informed decisions using interpretable outputs.

- **Individual Models per Location Enhance Forecast Precision**

By training a separate LSTM model for each delivery location, the system captured local traffic behavior more accurately. Each region had unique characteristics for example, Colombo had more frequent congestion patterns compared to Nittambuwa, which showed more predictable traffic flows.

Training individual models allowed fine-tuning of hyperparameters per location, improving overall forecast quality and ensuring region-specific optimization. This approach proved more effective than attempting to generalize a single model across all locations.

- **Integration with Demand Forecasting Boosts Strategic Planning**

The integration of traffic forecasting with the demand forecasting module provided a comprehensive solution for delivery planning. By identifying both *what quantity* needs to be delivered and *when* is the best time to deliver, the system empowered transportation managers to allocate vehicles and schedule deliveries more efficiently.

This combined insight led to a reduction in potential delays, improved fuel efficiency, and better alignment with customer needs — all contributing to reduced operational costs and enhanced supply chain performance.

- **Real-Time Dashboard Adds Practical Usability**

The final user interface, which visualizes predicted traffic durations, optimal transport time slots, and the number of vehicles required per location, turned out to be highly effective from an operational standpoint. This dashboard transformed complex model outputs into actionable logistics insights, proving essential for day-to-day transportation management. Users could make fast, data-driven decisions and adjust schedules dynamically based on up-to-date predictions, making the system not only intelligent but also user-friendly and practical.

3.3 Discussion

The development of a traffic forecasting system using LSTM models significantly contributes to improving logistics planning within the tea supply chain. The results highlight the practicality and efficiency of machine learning techniques in addressing transportation-related challenges, particularly those caused by unpredictable traffic patterns.

The successful implementation of the system demonstrates that predictive analytics can offer actionable insights to transportation managers, allowing them to make smarter scheduling decisions. Accurate traffic predictions, when aligned with demand forecasts, help minimize transportation delays, reduce idle time, and ultimately cut costs. This synergy between data-driven decision-making and operational efficiency showcases the real value of integrating machine learning into traditional industries like tea manufacturing and distribution.

A key highlight of the system is the modular design, where each delivery location is managed by a separate LSTM model. This approach respects the uniqueness of each route and adapts to local traffic conditions. It also opens the door for scalability, as new locations can easily be added by training new models without affecting existing ones.

However, some challenges were encountered during the process. Collecting sufficient historical traffic data from Google API required careful handling of API limits and time window configurations. Additionally, preprocessing the data to align with delivery windows while maintaining accuracy demanded meticulous attention to data cleaning and transformation techniques. Yet, overcoming these obstacles resulted in well-structured datasets that boosted the models' performance.

From a technical standpoint, LSTM proved to be a robust choice due to its ability to capture temporal dependencies. Unlike classical models, LSTM handled sequence-based input effectively, learning from long-term patterns within the traffic data. While training took longer compared to simpler models, the trade-off was justified by the increased accuracy and reliability of predictions.

Another important takeaway is the benefit of visual tools such as the traffic forecasting dashboard. It bridges the gap between machine learning outputs and real-world applications, ensuring that insights are accessible to non-technical users. Transportation managers can interact with predictions directly, view the best delivery time slots, and plan logistics accordingly — all without needing to understand the underlying model mechanics.

Furthermore, this component's collaboration with the demand forecasting module enhances the end-to-end supply chain by aligning supply readiness with optimal delivery timing. This integration underlines the importance of a holistic approach when applying machine learning in real-world operations.

In conclusion, the discussion validates that predictive traffic modeling using LSTM not only improves delivery planning but also sets a foundation for future enhancements in the tea supply chain. With real-time data integration and continued model refinement, the system has the potential to evolve into a fully automated logistics optimization platform, serving broader use cases beyond the tea industry.

4. CONCLUSION

The research undertaken in this component aimed to address one of the critical pain points in the tea supply chain—inefficient logistics planning due to unpredictable traffic conditions. By leveraging machine learning techniques, particularly Long Short-Term Memory (LSTM) networks, the project successfully developed a predictive system capable of forecasting traffic patterns with high accuracy for key delivery routes in Colombo, Kalutara, and Nittambuwa.

The solution involved a comprehensive pipeline starting from data acquisition through Google API, followed by data preprocessing, model training, evaluation, and ultimately system deployment. Each stage was carefully designed to ensure that the final model not only performed well but also served practical needs. The introduction of separate models for each location allowed the system to capture route-specific traffic behaviors, enhancing overall forecast precision.

One of the standout achievements of this research is the ability to combine historical traffic data with demand forecasting results to create a unified logistics recommendation system. This integration enables transportation managers to determine not only *what needs to be delivered*, but also *when and how* it should be delivered. The system outputs optimal delivery time slots and the number of required vehicles per location, thereby improving operational efficiency, reducing delivery delays, and minimizing transportation costs.

Furthermore, the development of a user-friendly dashboard enhances the system's usability by making complex predictions actionable for decision-makers with limited technical background. This increases the real-world applicability and scalability of the solution.

In terms of technical contribution, the study confirms the strength of LSTM in time series prediction tasks, particularly when used in combination with thoughtful data structuring and domain-specific preprocessing. The accuracy and stability of the models validate the decision to use LSTM over traditional methods.

In conclusion, this research provides a practical, data-driven approach to optimizing transportation in the tea industry. The predictive traffic model developed in this study is not only a valuable tool for current logistics planning but also sets a foundation for future enhancements, such as incorporating real-time data or expanding to other regions and industries. The findings reinforce the potential of AI and machine learning to revolutionize traditional supply chain operations through smart, automated decision-making.

5. REFERENCES

[1] Kechagias, Evripidis P., et al. "Traffic flow forecasting for city logistics: A literature review and evaluation." International Journal of Decision Support Systems 4.2 (2019): 159-176.

[2] Chen, Yi-Ting, et al. "Pragmatic real-time logistics management with traffic IoT infrastructure: Big data predictive analytics of freight travel time for Logistics 4.0." International Journal of Production Economics 238 (2021): 108157.

[3] Bhattacharya, Arnab, et al. "An intermodal freight transport system for optimal supply chain logistics." Transportation Research Part C: Emerging Technologies 38 (2014): 73-84.

[4] Gurnak, Vitalii, Lyudmila Volynets, and Ilona Khalatska. "Intellectualization of logistic supply chains on the basis of forecasting volumes of cargo transportation." MATEC Web of Conferences. Vol. 294. EDP Sciences, 2019.

[5] Al Moteri, Moteeb, Surbhi Bhatia Khan, and Mohammed Alojail. "Economic growth forecast model urban supply chain logistics distribution path decision using an improved genetic algorithm." Malaysian Journal of Computer Science (2023): 76-89.

[6] Abduljabbar, R., Dia, H., Liyanage, S., Bagloee, S.A.: Applications of artificial intelligence in transport: an overview. Sustainability. 11, 189 (2019).

[7] Nikitas, A., Michalakopulou, K., Tchouamou, E., & Karampatzakis, D. (2020). Artificial Intelligence, Transport and the Smart City: Definitions and Dimensions of a New Mobility Era.

[8] Mishra, S., & Pandey, M. (2019). Supply Chain Optimization for Tea Industry Using AI. • Tang, C.S., Veelenturf, L.P.: The strategic role of logistics in the industry 4.0 era. Transp. Res. Part E: Logist. Transport. Rev. 129, 1–11 (2019).

[9] Tang, C.S., Veelenturf, L.P.: The strategic role of logistics in the industry 4.0 era. Transp. Res. Part E: Logist. Transport. Rev. 129, 1–11 (2019).

[10] Hawkins, J., Habib, K.N.: Integrated models of land use and transportation for the autonomous vehicle revolution. Transp. Rev. 39(1), 66–83 (2019).

[11] Hu, W., Wu, H., Cho, H., Tseng, F.: Optimal route planning system for logistics vehicles based on artificial intelligence. J. Internet Technol. 21, 757–764 (2020)

[12] McKinsey, Company: Succeeding in the AI supply-chain revolution. Article (2021)

[13] Toorajipour, R., Sohrabpour, V., Nazarpour, A., Oghazi, P., Fischl, F.: Artificial intelligence in supply chain management: a systematic literature review. J. Bus. Res. 122, 502–517 (2021)

[14] Rey, A., Panetti, E., Maglio, R., Ferretti, M.: Determinants in adopting the Internet of Things in the transport and logistics industry. J. Bus. Res. 131, 584–590 (2021)