SES UNIVERSITY OF TWENTE.

# Master Thesis

## Predicting satellite capacity usage by shipping vessels using Neural networks :

*A solution based on satellite beam classification, vessel trajectory prediction and demand forecasting*

**Faculty of Electrical Engineering ,Mathematics and Computer science (EEMCS)**
**In collaboration with**
**SES Satellites**

**Examination Committee :**
**Supervisor 1 : Prof.Faiza.A.Bukhsh**
**Supervisor 2: Prof. Maria Iacob**
**Supervisor 3: Stephen O'Shaughnessy**

**Written by –**
**Smrithi Menon**
**S2336413**
**MSc in Business and IT(Data science)**

# *Acknowledgments*

Two years ago, I came to this country with a dream to fulfil. It was not an easy decision to leave my country and come to a foreign land, but as they say, your dreams will take you places, and here I am. These two years have taught me a lot. Experience is your best teacher, and I realized the meaning of it now. From different cultures to different education system, I was challenged in every step of the way personally and professionally and I am proud of myself today to take these challenges head-on and be where I am. All of this would not have been possible without the support of my family and friends.

First, I would like to thank my company supervisor Stephen O'Shaughnessy for being the best ever there is. He made sure that I am guided and supported at every step of my way in my virtual internship. His leadership gave me a chance to grow and upskill myself the way I want. He also made sure that my opinions are heard and valued. Stephen, thank you for making my first job in Netherlands memorable. I will never forget my learnings, your inputs and will always keep your feedbacks in mind.

I would like to thank my university supervisors Prof Faiza and Prof Maria. They both have constantly moulded me with their constructive criticism. Maria, thank you for being there whenever I needed help with the BIT curriculum. Faiza, thank you for being my constant support through this thesis journey. Your timely feedback has taught me a lot and has proven to be really valuable. Thank you for your time and patience. I would like to extend my gratitude to Dakshina for being patient and kind with me on this journey. You were there every time I had a doubt and needed someone to bounce off my ideas. You are a significant part of my thesis.

Last but not least, thank you to the most important people in this journey without whom nothing of this would have seen the light of the day. To start with, Thank you, mom, dad, and sister, for everything. Your constant phone calls and motivation made me less homesick. Your motivational words pushed me to where I am today. From day 1, you believed in me, and you assured me that I

could do it. Thank you for being my best support system. My special love to my nephew Dhruvin,I used to look forward to seeing you on calls just to turn my bad day into a good one. Thank you, Suriya, Venus, Neetu, Jerro, and Bharath, for always giving me an extra dose of confidence.

My sincere thanks to my very special friend, Manav. You stood by me patiently whenever I needed a support to fall back on. You believed in me when I did not. You were there in this journey like a rock. With the pandemic, the last one year had been difficult for all of us and through the difficult times, your constant support helped me to keep going. This success is as much as yours as it is mine.

*Mom and Dad, this thesis is dedicated to you.*

# *Abstract*

Technological advancements and processes are rapidly changing and becoming increasingly integrated. As with the previous three industrial revolutions, ushering in the new industrial era, which is based on the ability to adapt and integrate digital and physical technology, brings with it a slew of opportunities and problems. At the heart of the Fourth Industrial Revolution is satellite technology's potential to give ubiquitous and increasingly fast communication to billions of people around the world. Unprecedented access to information, advances in processing power, and the integration of embedded devices will eventually lead to the creation of new narratives about how the satellite industry affects production, management, and distribution. The measures that the satellite business is doing today are laying the groundwork for future advances. At the heart of the Fourth Industrial Revolution is satellite technology's potential to give ubiquitous and increasingly fast communication to billions of people all over the world.

SES provides a vast and intelligent network comprising the world's only multi-orbit constellation of Geostationary (GEO) and Medium Earth Orbit (MEO) satellites and extensive ground infrastructure – to deliver video and data solutions that connect more people in more places with content that enriches their personal stories.. SES Networks' Skala Global Platform provides maritime service providers with a flexible and cost-effective way to deliver high-quality broadband services to ship owners, regardless of ship size or location, the research project hence focuses on this segment of SKALA

The main objective of this research is to is to produce a model with the help of neural network and machine learning algorithms that could predict how much capacity from a pool of allocated satellite beams with different overlapping coverages will be utilized by the vessels. With changing locations of the vessels, the model should be able to predict which satellite beam (Platform) will be utilized by the vessels as they move across the various beams. This model will eventually help in optimizing the sizing of capacity allocation to each Satellite-Beam coverage by deciding what approaches are best from a commercial perspective, which can include a trade-off of capacity allocation, solution design optimization, pricing of product packages and customer experience.

# Table of Contents


1.Introduction ..... 9

1.1 Problem Context ..... 9

1.2 Problem Identification ..... 11

1.3 Motivation ..... 12

1.4 Research Goal ..... 13

1.5 Research Approach and literature review method ..... 14

1.6 Chapter Summary : ..... 14

2 Related Work and Background ..... 16

2.1 Systematic Literature Review ..... 16

2.2 Trajectory Prediction: ..... 18

2.3 Demand Forecasting: ..... 21

2.4 Background ..... 25

2.4.1 Time Series Forecasting ..... 25

2.4.2 Components of Time Series ..... 26

2.4.3 Stationarity and Non-Stationarity ..... 27

2.4.4 Deep Learning and Time Series Forecasting ..... 27

2.4.5 Neural Network ..... 28

2.5 Chapter Summary ..... 33

2.5.1 Research Gap ..... 34

3 Methodology ..... 35

3.1 Business Understanding ..... 36

3.2 Solution Design ..... 37

3.3 Data Analytics ..... 37

3.4 Solution Implementation ..... 37

3.5 Result Analysis ..... 38

3.6 Chapter Summary : ..... 38

4.Dataset Preparation ..... 39

4.1 Data Collection ..... 39

4.1.1 Location Prediction Model and Platform Classification: ..... 39

4.1.2 Capacity Prediction Model: ..... 40

4.2 Data Exploration ..... 41

4.2.1 Location Prediction Model and Platform Classification Model: ..... 41

4.2.2 Capacity Prediction Model: ..... 42

4.3 Data Pre-processing....44
4.3.1 Location Prediction Model and Platform Classification Model:....44
4.3.2 Capacity Prediction Model:....46
4.4 Data Transformation....49
4.4.1 Location Prediction Model:....50
4.4.2 Platform Classification Model:....52
4.4.3 Capacity Prediction Model:....53
**4.5** Chapter Summary **:**....56
5 Modeling....58
5.1 Tools and Libraries....58
5.2 Experimental Setup and design:....59
*5.2.1* Location Prediction Model:....59
*5.2.2* Platform Classification Model:....61
*5.2.3* Capacity Prediction Model :....64
5.3 Chapter Summary....65
6 Result Analysis....67
*6.1* Location Prediction Model:....67
6.2 Platform Classification Model:....72
*6.3* Capacity Prediction Model :....74
6.4 Chapter Summary :....76
7 Discussion....77
**7.1** About the Datasets**:**....77
7.2 Data Scaling:....77
7.3 General:....77
7.4 Limitation and Future Work....78
7.5 Contribution to Science....79
7.6 Contribution to Practice....80
8 Conclusion....81
*References*....83
APPENDIX....85

# Table of Figures

*Figure 1: Skala global platform*----------10
*Figure 2:Overlapping beam coverages*.....12
*Figure 3:Mapping of research questions to their respective chapters*.....15
*Figure 4:Systematic Literature Review (SLR)[23]*.....18
*Figure 5:LSTM network for vehicle trajectory prediction on highway[10]*.....19
*Figure 6:Flowchart of the trajectory prediction model of vessels[11]*.....21
*Figure 7:Recurrent Neural network for demand forecasting[7]*.....22
*Figure 8:Methodology for sales forecasting[15]*.....23
*Figure 9:Proposed system architecture for energy consumption forecasting*.....24
*Figure 10:Components of time series[20]*.....26
*Figure 11:Stationarity and Non stationarity[19]*.....27
*Figure 12:A Neuron [22]*.....28
Figure 13:A dense layer of two linear units receiving two input and a bias[22].....29
Figure 14:Recurrent Neural network[17].....30
Figure 15:Understanding LSTM Networks[17].....31
Figure 16:Understanding Gated recurrent units[17].....32
Figure 17:Data science methodology[23].....35
Figure 18:Problem identification.....36
Figure 19:Code snippet of SAP query used to extract the location data.....45
Figure 20:Movement of terminal depicting randomness.....46
Figure 21:Code snippet of SAP query used to extract the capacity data.....48
Figure 22:Depiction of PERCENTILE_CONST function.....48
Figure 23:Convert a time series to a supervised learning problem.....49
Figure 24:Predicting latitude one-time step in future.....51
Figure 25:Predicting longitude one time-step in future.....51
Figure 26:Simultaneous prediction latitude and longitude.....52
Figure 27:Pair of coordinates mapped to a label encoded platform.....53
Figure 28:Predicting forward Mbps one-time step in future.....54
Figure 29:Predicting return Mbs one-time step in future.....55
Figure 30:Simulatenous prediction.....56
Figure 31:Final experiment setting of LSTM: Longitude.....61
Figure 32:Final experiment setting of LSTM: Latitude.....61
Figure 33:Neural network: LSTM.....63
Figure 34:Code snippet of XGBoost.....64
Figure 35:Experiment setting of LSTM.....65
Figure 36:Architectural overview of solution design.....67
Figure 37:Multi-Step prediction.....68
Figure 38:Single-step prediction.....68
Figure 39:Before hyperparameter tuning.....70
Figure 40:After hyperparameter tuning).....70
Figure 41:Actual vs predicted results for terminal A.....71
Figure 42:Actual vs predicted results terminal B.....71
Figure 43:Single terminal: XGboost(left) vs Neural networks(right).....73
Figure 44:Multiple terminals : XGboost(left) vs Neural networks(right).....74
Figure 45:Prediction results :Return Mbps.....75
Figure 46:Prediction results :Forward Mbps.....75

# 1.Introduction

The Fourth Industrial Revolution's influence and impact on the globe is based on the concept of digitizing everything. Machine learning, Artificial Intelligence (AI), the Internet of Things (IOT), and other advanced technologies are rapidly transforming and reshaping infrastructure, global-local economies, and future possibilities. The Fourth Industrial Revolution envisions a new hyper-connected paradigm that allows industry to react to changes in the environment in real time.
For the satellite sector, leveraging these new transitions and comprehending their disruption potential in terms of technology, altering demography, and global connection is critical. The satellite industry's current actions are laying the groundwork for future developments. At the heart of the Fourth Industrial Revolution is satellite technology's potential to give ubiquitous and increasingly fast communication to billions of people worldwide.

SES is an international satellite and terrestrial telecommunications network provider headquartered in Luxembourg, supplying video and data connectivity worldwide to broadcasters, content and internet service providers, mobile and fixed network operators, government institutions. The company provides a vast and intelligent network comprising the world’s only multi-orbit constellation of Geostationary (GEO) and Medium Earth Orbit (MEO) satellites and extensive ground infrastructure – to deliver video and data solutions that connect more people in more places with content that enriches their personal stories. As the leader in global content connectivity solutions, SES does the extraordinary in space to deliver amazing experiences everywhere on earth.

## 1.1 Problem Context

SES being a provider of global managed data services, enables high-performance networked communications virtually anywhere. The company enables a unique combination of ground system technology, advanced satellite capabilities and service lifecycle expertise that enables the creation and delivery of tailored services across a diverse range of use cases. SES Networks’ **Skala Global Platform** is a next-generation ground system optimized for the delivery of high-

quality broadband services. Skala Infrastructure is a distributed architecture provides unprecedented scalability, enabling the customer to increase capacity as required on a global basis—without additional hardware costs. This eliminates a major bottleneck to growth as customer's bandwidth requirements increase.

***Simple Connectivity Isn't Enough:***

***A consistent experience regardless of location***

The company's space assets provide global coverage using wide beam and high throughput satellites in geostationary orbit (GEO). Service providers, mobile network operators, enterprises, and vertical industry segments receive the best possible service to address their business requirements—wherever they are located.

Skala provides frictionless access to the SES next-generation satellite and ground network for customers looking to cover large geographies. Company's partners are able to receive megabit wholesale connectivity as a managed service via a turnkey combination of space segment and ground infrastructure while maintaining control of end user terminals.

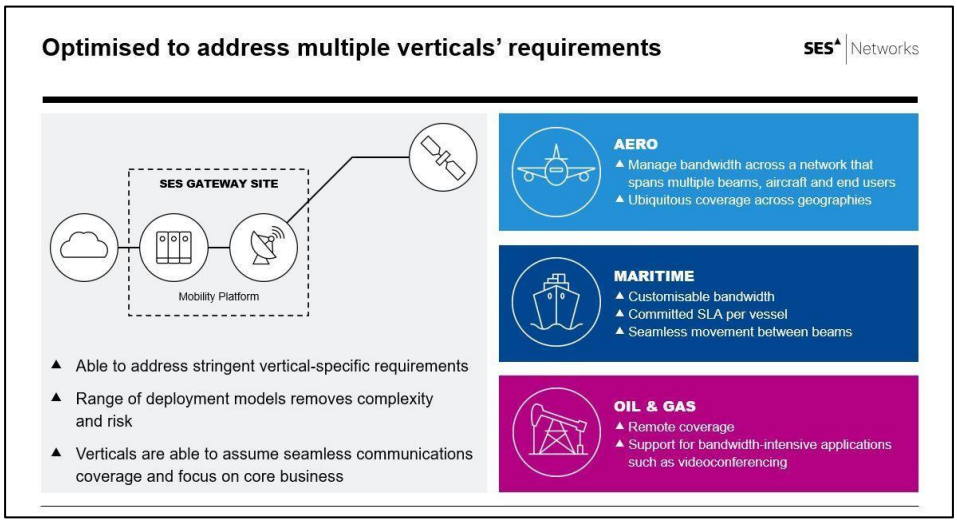


*Figure 1: Skala global platform*

***MEETING GROWING DEMAND***: The future is connected, and both businesses and consumers continue to capitalize on the benefits of digitalization. Their need for connectivity will only grow—regardless of where they are located. Combined with company's next-generation global space assets and service lifecycle expertise, Skala Global Platform ensures its customer's end users receive the right service, at the right time—without the complexity of stitching together regional connectivity, and the Capex costs associated with procuring equipment. Multiple service delivery models, including platform services and end-to-end managed network services, reduce operational costs and risk, while enabling a faster time to service.

Skala is a next-generation ground system optimised for the delivery of high throughput bandwidth that, when combined with SES's advanced satellite capabilities and service lifecycle expertise, enables the creation and delivery of tailored services. Skala's distributed architecture provides unprecedented scalability, enabling maritime service providers to increase capacity where and when it's needed without additional hardware costs.

## 1.2 Problem Identification

As mentioned in the previous section, SKALA's main segment is **Maritime**. SES Networks' Skala Global Platform provides maritime service providers with a flexible and cost-effective way to deliver high-quality broadband services to ship owners, regardless of ship size or location, the research project hence focuses on this segment of SKALA. There is little known about the vessels when the customer wants to sign up with SES. The customer can subscribe to a set of data packages or can go with a PAYG (pay as you go) model and may use the global coverage. Asset Management and technical teams tries to get some guidance from sales/solution engineers on where the vessels will roam, but there is no certainty how much time they will spend on each beam.

**Objective statement**: *How much capacity from a pool of allocated capacity across various satellite beams with different overlapping coverages will be utilized by the vessels?*

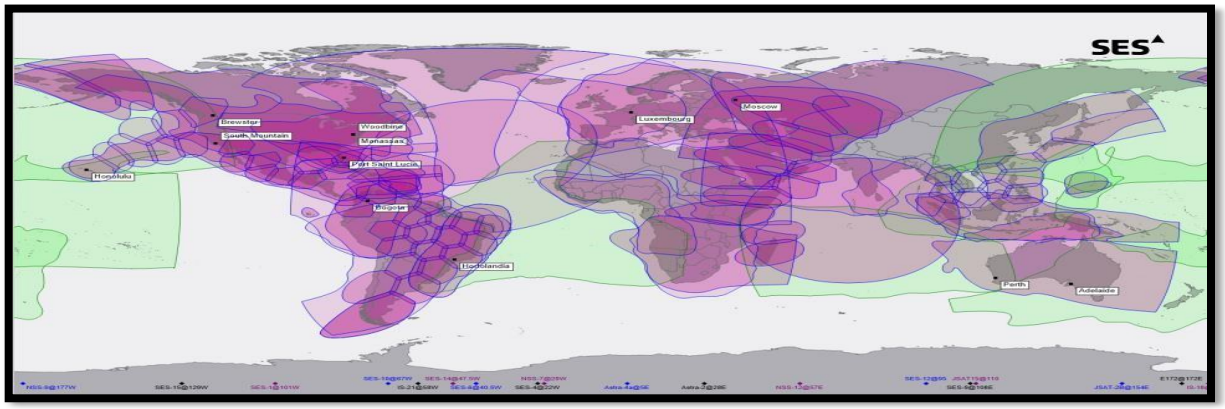


*Figure 2:Overlapping beam coverages*

## 1.3 Motivation

The Asset Management team assumes/guesstimate on the 'sectors' the vessel might use. With the customers over-utilizing / under-utilising the beam bandwidth, it has become necessary for the team to optimize the capacity allocation so that the customer needs are fully supported***. Future itineraries of the vessels are unknown, and it is essential to know the future locations of the ship***. This information will give a broader perspective about the platform the vessel will be using in the future. ***These locations have overlapping beam coverages, making it difficult to guess which beam the vessel will be utilized in the future***. This problem could be solved with business intelligence and machine learning skills. Various deep learning methods have been implemented in the critical areas of time series forecasting that are trajectory prediction and demand forecasting. Traditional time series forecasting is preferred for lesser complicated datasets; Deep learning methods are preferred with complex temporal dependencies. Most popular neural networks for time series forecasting because of their capability to remember information for long term dependencies re LSTMs. Complex and real-world data like this needs a smarter algorithm not only to address the problem but also for efficient data and result analysis .It is crucial to understand the significance of the "Big Data analytics" technology of **Industry 4.0** and harness it in the right way. It could be a step

forward in the alignment of Industry 4.0 with the satellite communication industry.

## 1.4 Research Goal

Based on the Problem Context and Problem Identification, the objectives and research questions were defined. The main goal is to produce a model that could predict how much capacity from a pool of allocated satellite beams with different overlapping coverages will be utilized by the vessels. With changing locations of the vessels, the model should be able to predict which satellite beam (Platform) will be utilized by the vessels as they move across the various beams. This model will eventually help in optimizing the sizing of capacity allocation to each Satellite-Beam coverage by deciding what approaches are best from a commercial perspective, which can include a trade-off of capacity allocation, solution design optimization, pricing of product packages and customer experience. The research questions are defined to integrate and explore both the Business and the IT side of the satellite communication industry.

**Main Research Question**: *Which Satellite Beam will each vessel use considering the changing locations of the vessels?*

**Sub-questions –**

***RQ1****-Can this problem be solved using traditional time series forecasting methods?*

***RQ2****-Is Neural network an optimal solution to this problem?*

***RQ3****-In this case, is Feature extraction required for LSTM algorithm for trajectory prediction and demand forecasting techniques?*

***RQ4****-Which part of Industry 4.0 is being harnessed by this solution with respect to satellite Industry?*

## 1.5 Literature review method and Research Approach

There are two approaches used in this study .One for Literature review and other for problem identification and solution implementation :

**Systematic Literature Review (SLR)**

SLR is method of review that involves collecting multiple studies and summarising them to explore the answers of determined research questions through step by step process. The literature review in this study has been done using SLR. Various articles were collected based on a determined search query. These articles were filtered based on the set of inclusion and exclusion criteria.

**Data Science methodology**

Data Science Methodology[22] is a methodology to address data science problems[Figure 17] . It is inspired by CRISP-DM[22]. It is a sequence of steps defined to guide business analysts, data analysts to find the solution to the problem. In this study right from business understanding to solution implementation, the data science methodology has been followed.

## 1.6 Chapter Summary :

This study uses the SLR method to conduct an extensive literature review of the previous work that has been done in the domain of Neural networks and Timeseries forecasting. It uses Data Science Methodology to design and implement the solution.
This thesis document is structured as follows: Chapter 2 is about the state-of the-art time series forecasting in trajectory prediction and demand forecasting. It also explains the essential concepts about Neural networks and Time-series forecasting utilized in this thesis. The methodology follows in chapter 3 that describes the Data science methodology in detail. Chapter 4 explains the data set preparation, data collection performed. It is followed by Modeling: Experiment design and set up in Chapter 5. In Chapter 6, an in-depth Result analysis has been performed . In this chapter, the essential insights are drawn from the result analysis. It is then followed by a discussion in Chapter 7, which

discusses the limitations and future work. It also emphasis on the scientific contribution of this thesis. Last but not least, the content of the document is concluded by answers to the research questions in chapter.

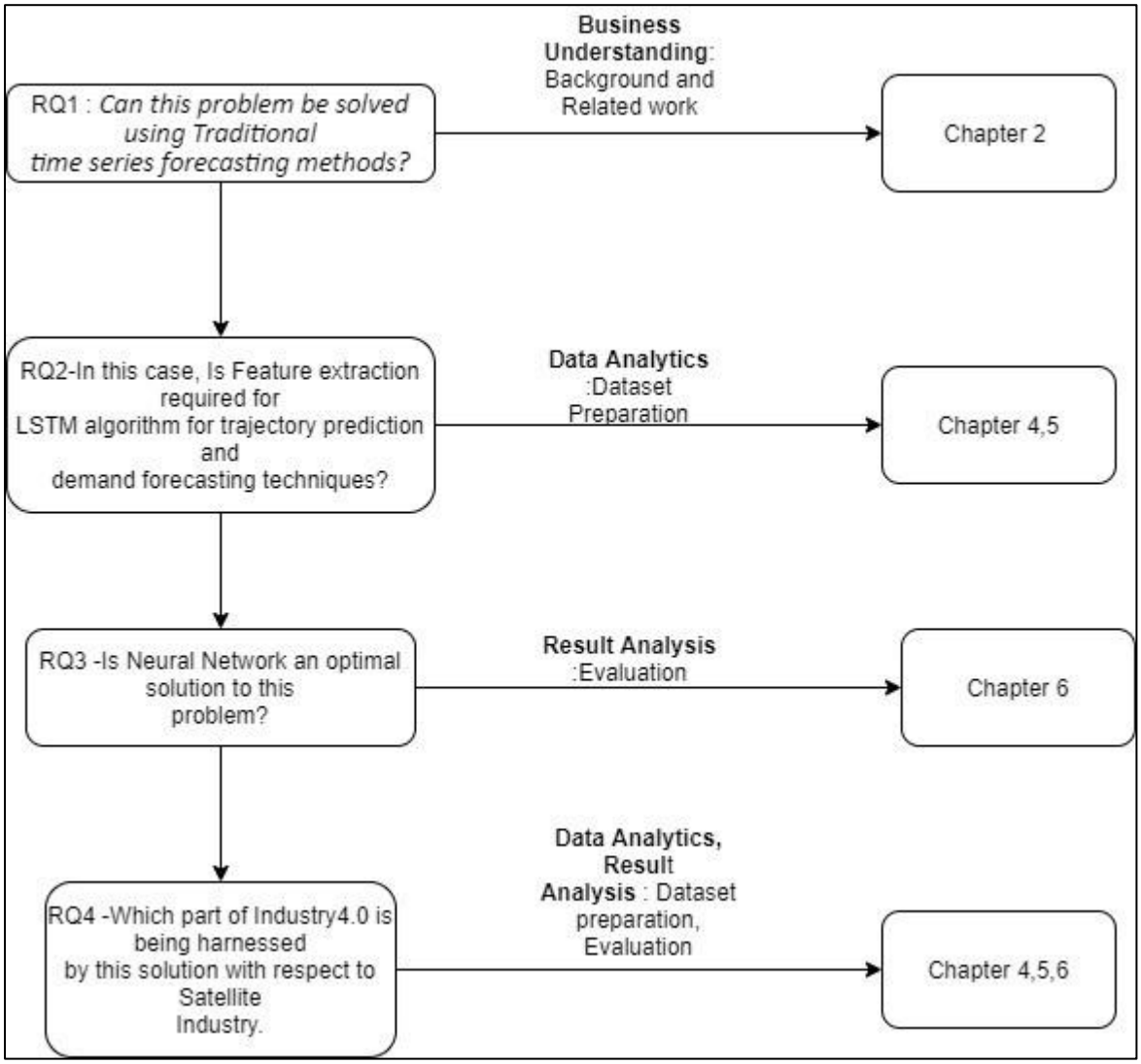


*Figure 3:Mapping of research questions to their respective chapters keeping in view the methodology*

# 2 Related Work and Background

This chapter is about the current state-of-the-art of Time Series forecasting in trajectory prediction and demand forecasting . First, the literature review is presented. Then ,popular works done in this domain has been presented. It continues with theoretical explanation of time series forecasting and neural network.

## 2.1 Systematic Literature Review

For in depth analysis of the work and literature in the field of Neural Networks and Time series forecasting ,a systematic Literature review has been performed. It is a protocol proposed by Kitchenham (2004), used to identify and critically appraise the research to answer the formulated research questions

- **Research Question 1**: What are the Deep learning methodologies used for Trajectory prediction? ***[section 2.2.1]***
- **Research Question 2**: What are the Deep learning methodologies used for Demand Forecasting? ***[section 2.2.2]***
- **Research Question 3** :Why are Deep learning methods preferred over traditional time series forecasting algorithms?***[section 2.3]***
- **Research Question 4**: What is the kind of data the existing studies have dealt with?***[section 2.2,2.3]***

The literature review is done in following phases :

**1 - Planning :**

In this phase, primarily the databases are determined for selection process by running a search query,

Following databases were selected for this process:

- Scopus (http://www.scopus.com/)

- Google Scholar (https://scholar.google.com/)
- IEEE (http://ieeexplore.ieee.org/)
- ScienceDirect (http://www.sciencedirect.com/)
- Web of Science (https://www.webofknowledge.com/)

The search query was defined using following terms

{“Neural Network” AND “Time Series forecasting “}
OR
{“Neural Network” AND “Location prediction” }
OR
{“Neural Network” AND “Trajectory prediction “}
OR
{“Neural Network” AND “Demand forecasting “}

**2- Selection**

Based on the following inclusion and exclusion criteria the results obtained from the search query were filtered. Figure 4 depicts the whole SLR process followed to perform the literature review .

**Inclusion Criteria :**

1-The study includes at least one of the determined keywords

2- The study is published in English

3-The study gives insights into at least one of the research questions

4-The study is in its latest version

5-Full length text of the study is available to access online.

**Exclusion Criteria :**

1-The study does not answer any of the research question

2-The study does not have relevant keywords

3-Duplicate studies

4-The study does not meet the inclusion criteria

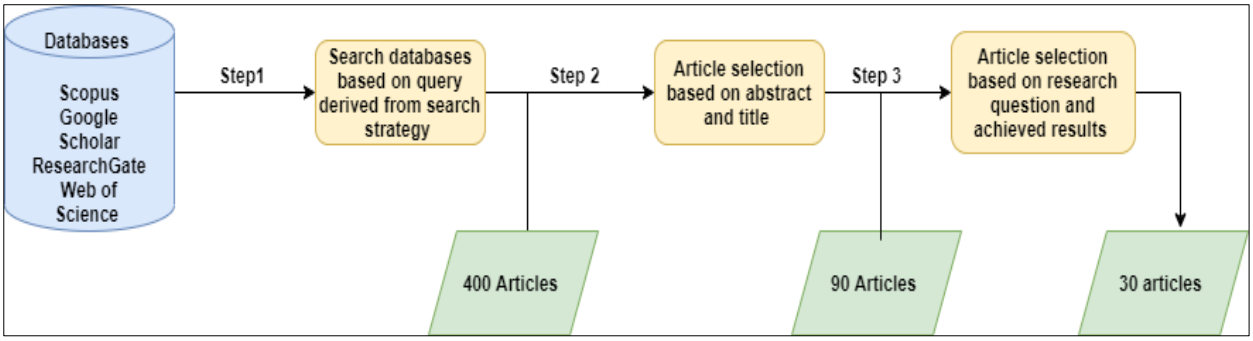


*Figure 4:Systematic Literature Review (SLR)[23]*

## 2.2 Findings

### 2.2.1 Trajectory Prediction:

Arnaud de La Fortelle et.al [10] introduces an LSTM network to accurately predict future longitudinal and latitudinal trajectories for vehicles on the highway. They have used the NGSIM US101 dataset (Next-generation simulation data) collected by the United States Federal Highway Administration. The dataset consists of 45 minutes of trajectories for vehicles on the US101 highway. It has a record of 6000 individual vehicle trajectories. The dataset contains trajectories in the form of (X, Y) coordinates, which represent coordinates of the front centre of the vehicle in the global frame and the form of (x,y)coordinates of the same point (road aligned frame).In this article, the authors have used the

Savitzy-Golay filter with a window length of 11 to smooth the longitudinal and lateral positions. As part of feature selection, they have defined a few features: the target vehicle like local lateral position, local longitudinal position, lateral and longitudinal velocities, and vehicle type. Other vehicles apart from the target vehicle have defined features like lateral velocity, longitudinal velocity, lateral distance, longitudinal distance from the target vehicle, signed time to collision, and type of the vehicle.

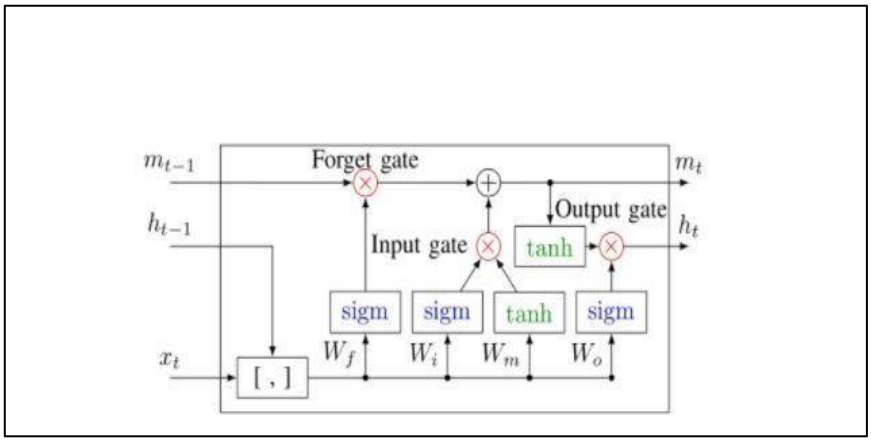


*Figure 5:LSTM network for vehicle trajectory prediction on highway[10]*

The authors have selected 80 % of the trajectories as their training set and 20% of the remaining trajectories as their testing set. On average, their RMS error was roughly 70 cm for the lateral position 10 s in the future and lower than 3ms1 for the longitudinal velocity**.**

**Limitation,** of this paper, as also mentioned by the author, has not considered the stochasticity or confidence levels which is also an important information to be considered in a Time Series Problem.

Chujie Wang et.al [9], The baseline problem which the paper addresses are human mobility prediction. This paper has focused on the Multi-user-Multi step trajectory Prediction by initially designing a basic deep learning model for prediction framework. This model applies the LSTM network directly in order to learn user-specific mobility patterns and then later on incorporating seq2seq learning for multi-user multi-step trajectory prediction framework. The authors

have highlighted how prediction accuracy decreases with an increase in the candidate location under high discretization granularity of coordinates and hence has investigated the approaches to predict trajectories that are composed of continuous coordinates. The paper has focused on both single and multi-user perspective. For a single-user perspective, the authors have proposed an LSTM network. They have used a model-based dataset (generated by referring to Self-similar least action human walk and the SMOOTH model). From a Multi-user perspective, the authors have extended the work to the Seq2Seq framework.
Zhongyi Zheng et.al [11] use machine learning and data mining techniques for the trajectory prediction of vessels. It mentions about two methods that can be used in future trajectory prediction of a moving object. One that uses motion function in order to predict the next location of the object. This method has good accuracy in cases where future time is close to the current time, and there are other factors like velocity, weather conditions, etc. are known. Second, that uses historical information of the object's motion to predict the future trajectory. This method is applicable when there are sufficient historical records.

In the method proposed by the authors, the historical trajectories are clustered using clustering parameters. These parameters classify a new vessel's cluster. The authors' prediction model consists of two parts -historical trajectory prediction and classification of new vessels. SVM (support vector machine) is used in order to classify the new vessels. The total number of trajectories taken is 2862.The trajectories are input to the model in chronological order. **The limitation** of this model is that this model's performance depends on a lot of information related to vessels. The question is, will this model be equally good where there isn't much information about the vessels.

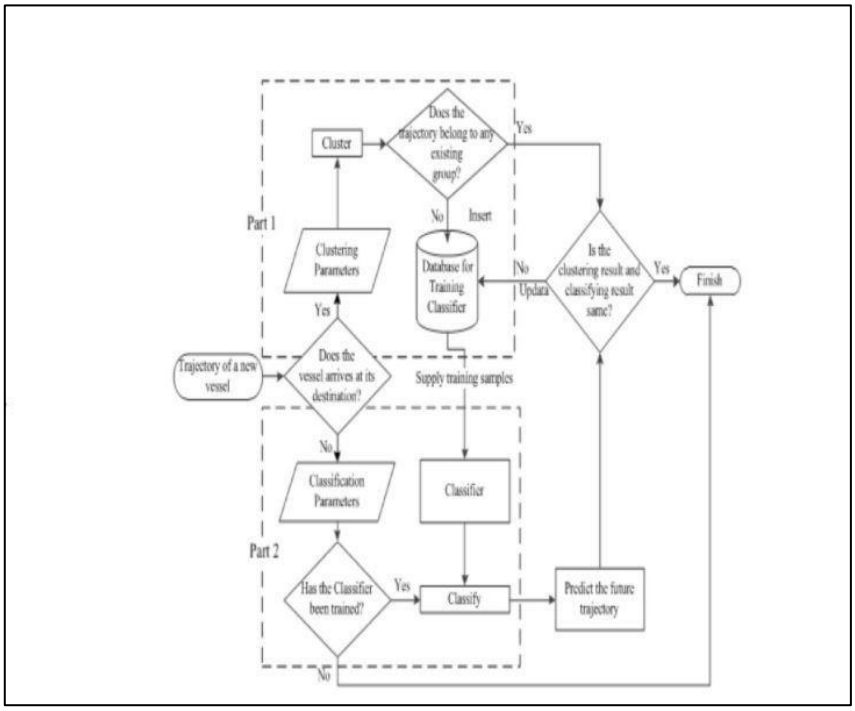


*Figure 6:Flowchart of the trajectory prediction model of vessels[11]*

### 2.2.2 Demand Forecasting:

Real Carbonneau et.al [7], aims to analyze different machine learning methods to forecast distorted demand at the end of a supply chain, the term for which is coined as the bullwhip effect. The Authors have used 2 datasets for their analysis. Both are distorted demand data. One is based on simulation of the extended supply chain. The second one is extracted from the estimated value of new orders received by Canadian Foundries. The forecasting techniques the authors have used in their analysis are, Naïve forecast, Average, Moving average, Trend, Multiple Linear Regression, Neural networks, Recurrent neural networks, Support Vector Machines. The Authors have used one of the most common neural networks; Feed-Forward error back-propagation type neural nets.

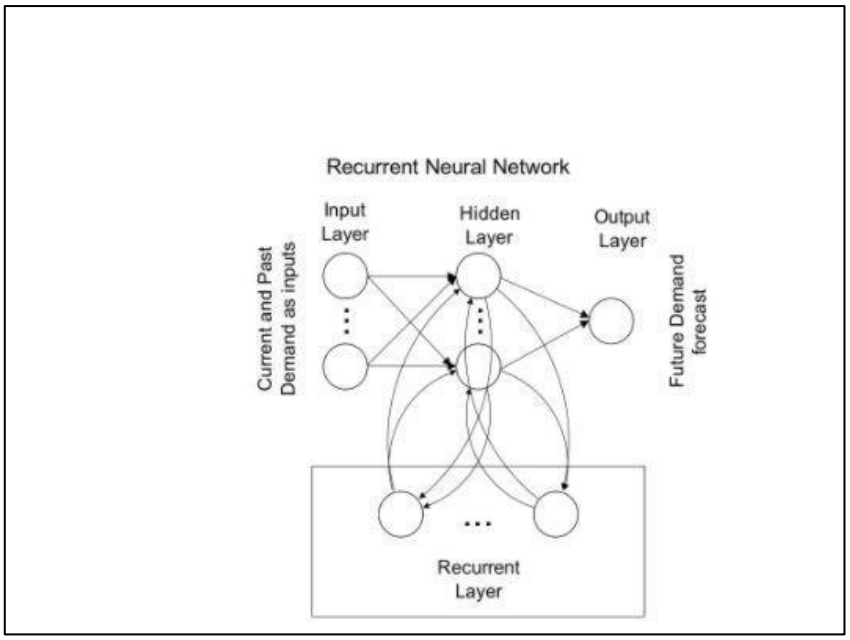


*Figure 7:Recurrent Neural network for demand forecasting[7]*

The training and testing data is the same across all the forecasting methods to aid valid comparisons. The models' input variables were the change in demand in the past, and the output variable was total change over the next three periods. According to the experiments' analysis and the results, the authors conclude that the Neural Networks and SVM were the most accurate forecasting techniques. In contrast, Trend and Native are the worst types of demand signal processing with the highest error level.

Hossein Abbasimehra et.al [15], aims to forecast fluctuating demand data using multi-layer LSTM networks. The Authors use a gird search method in order to select the best model by considering different combinations of LSTM hyperparameters.

According to the authors, the proposed method's main advantage is capturing non-linear parameters in time series data. This method searches for optimal hyperparameters of the LSTM network using the grid search approach.

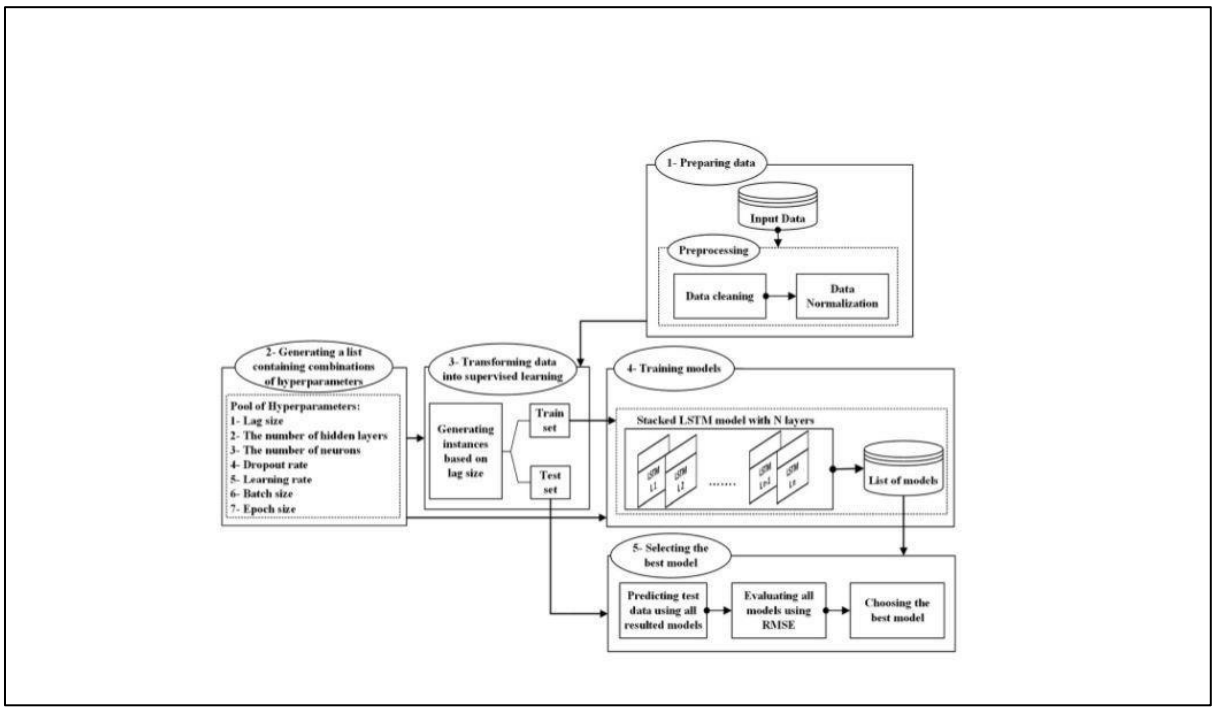


*Figure 8:Methodology for sales forecasting[15]*

The hyperparameter used for the gird search purpose is Lag size, The number of hidden layers, the number of neurons dropout rate, learning rate, batch size, and epoch size. The demand data of a furniture company has been used for the experiments. The data consists of the monthly sale quantity of a product from 2007-2017.he time series consists of a total of 132 months. The authors have compared their method with statistical methods like ETS and ARIMA. Based on performance measures, the proposed method is better than both the statistical methods. Authors also compare their method with computational intelligence methods like KNN, RNN, SVM, Single layer LSTM, and ANN. The proposed method's overall performance versus these methods based on metrics like SMAPE and RMSE is better.

Rodrigo F. Berriel et.al [6] aims to forecast monthly energy consumption using a deep learning-based model. The authors have evaluated three architectures: fully connected (FC), convolutional neural network (CNN), and Long short term memory(LSTM) Networks. They propose a system that works on the temporal sequence of energy consumption and some custom attributes.

The system takes the energy consumption of a single customer over a span of 12 customers. The input comprises of 12 months energy consumption, the month to be predicted, the number of phases of electric power, and the class of the customer. The input data is continuous and categorical data of each customer. The time series for the 12months of energy consumption is normalized and then fed to the model. Another continuous input data is average energy consumption over the period of the 12months. The data set used for this experiment is extracted from a Brazillian power company.

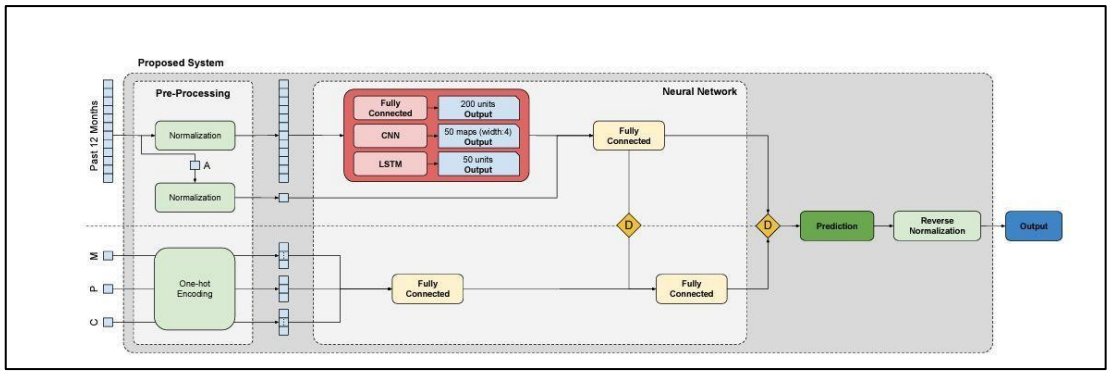


*Figure 9:Proposed system architecture for energy consumption forecasting*

Total 21,978,437 measurements from 934,945 different customers are collected. The Authors also conclude that LSTM-Based model outperforms other architecture in terms of Absolute and relative errors. Sol Kim 1 et.al [8] paper aims to present an energy consumption model based on a Gaussian process regression by training a support vector machine. It implements a EUP (Energy usage pattern) predictive model with a prediction rate of 95%. The authors emphasize the importance of occupants' behaviour and the physical and environmental factors that affect energy consumption. This paper focuses on deriving energy usage patterns (EUPs) using K-modes clustering. The energy usage patterns are specifically based on single-person households. The dataset used for this purpose was EUP datasets built by redefining 5193 Korean time use survey data (KTUS) points. Household characteristics and energy consumption of each of the 7 derived EUP types were analyzed.

The limitation of this study, also mentioned by the authors, is that this study is limited to single-person households, and the consumption data was created through energy simulations.

## 2.3 Background

This section explains different concepts of Time series forecasting, components of time series and in the later subsections various concepts of Deep learning has been covered.

### 2.3.1 Time Series Forecasting

With the increasing demand for machine learning algorithms for solving complex business problems, one important area is now widely being adopted as a solution to solving these problems. This area is Time Series Forecasting. Many real-world problems have time component and cannot be efficiently solved by traditional machine learning algorithms, where the various time series forecast methods come into the picture. Right from Sales forecasting, stock market analysis, workload projections to process and quality control, and weather forecasting, there are a plethora of applications of time series analysis and forecasting [2]. Time series is a series of observations of the values that a variable takes at different time intervals. With time series forecasting, we typically predict the future by analyzing the historical data based on time. Time series forecasting is when these data points, i.e., the past values, are analyzed and studied to forecast the future to aid important business decisions. Based on the number of variables, a time series could be divided into two types. One is the Univariate time series, where only a single variable takes values over a period of time. The second one is the Multivariate time series, where multiple variables take on the values over a period of time. Some popular classical time series algorithms are implemented as statistical models in many time series forecasting problems, including Autoregressive Moving Average (ARMA), ARIMA, and SARIMA.ARIMA stands for Autoregressive Integrated Moving Average. It is one of the traditional statistical models used for analyzing and forecasting time series data. The extension of the ARIMA model is the SARIMA model. It compensates for the inefficiency of the ARIMA model to handle seasonal components. SARIMA has the ability to model the seasonal component

of the series directly and hence stands for "Seasonal Autoregressive Integrated Moving Average. "Machine learning algorithms (deep Learning to be specific) are preferred in various time series forecasting applications, from being robust to noise to do multi-variate, multi-step forecasting efficiently

### 2.3.2 Components of Time Series

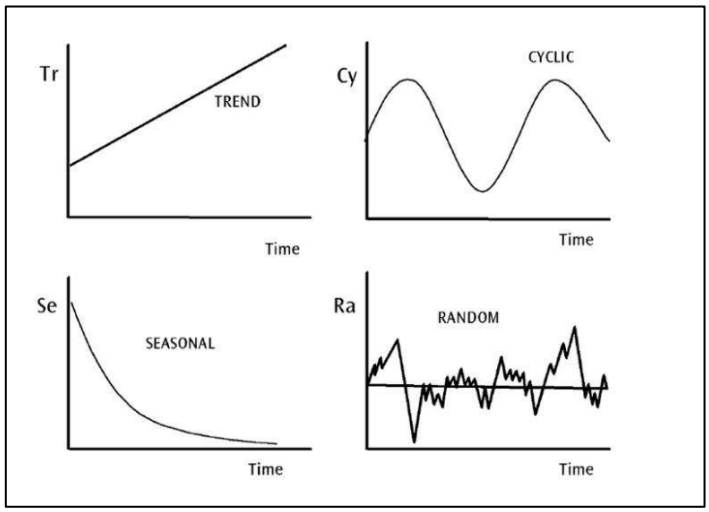


*Figure 10:Components of time series[20]*

Time Series has 4 components:

- Trends: Increase or decrease in the series over a period of time. It persists over a long period of time.
- Seasonality: Regular Pattern of up and down fluctuations. It is a short time variation occurring due to seasonal factors
- Cyclicity: A medium-term variation caused by circum-stances that repeat in irregular intervals. In this variation, the duration is not fixed, and the length between the two cycles could be longer.
- Irregularity: Refers to variation which occurs due to un-predictable factors and also does not repeat in particular patterns

### 2.3.3 Stationarity and Non-Stationarity

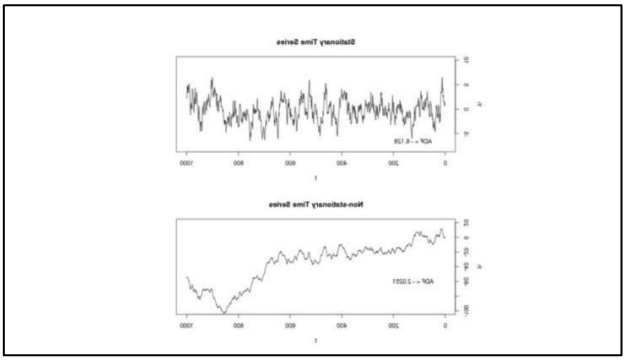

*Figure 11:Stationarity and Non stationarity[19]*

Any time series in which the mean and variance of the data remain constant over the period of time is known as stationary time series. We don't see any trend or fluctuations in such a time series, and the auto-co variance does not depend on time. Most of the time series data observed in real-world problems are non-stationary data as they see some trend, seasonality, or irregularities, as mentioned in the previous section.

### 2.3.4 Deep Learning and Time Series Forecasting

Although traditional time series algorithms have proven to be effective, there are few reasons why deep learning techniques like Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM), Gated recurrent units (GRUs) are preferred over these traditional algorithms.

- Time series forecasting is considered to be one of the complex problems in forecasting domains. There are a lot of complex temporal dependencies [20]
- There are non-linear components in most of the real-time series forecasting problems for which statistical methods are not enough as these algorithms rely on linear relationships [15]
- Traditional time series methods cannot work with datasets with missing values as they need complete data [20]

- Many times, effectiveness of these traditional algorithms are proved only in univariate data, and there are many real-world problems out there that are multivariate.

### 2.3.5 Neural Network

In order to understand Neural Network, it is essential to know what goes on at the unit level. The below-given diagram is pictorial representation of a linear unit (one neuron)

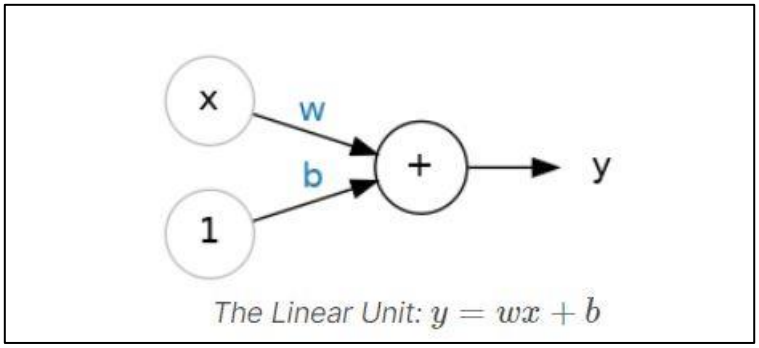


*Figure 12:A Neuron [22]*

In the above given figure x is the input, w is the weight of the connection of the input with the neuron, and b is the bias which enables the neuron to modify the output independently and y is the output calculated by the neuron by adding up all the values it has received through the connection which in this case would be y=wx+b[22]

**Layers** are nothing but what the neural network organizes the neurons into. It is a collection of the linear units with a common set of inputs resulting in a dense layer as shown in the figure given below:

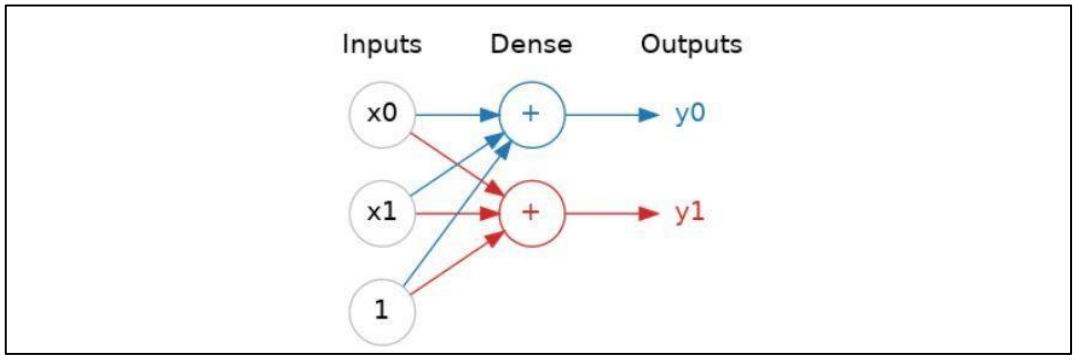


*Figure 13:A dense layer of two linear units receiving two input and a bias[22]*

**Activation function** is applied to every output of every layer of the neural network in order to fit the curves; in other words, to make the network learn the non-linear relationships. Many activation functions are used, including Sigmoid/Logistic, Tanh, ReLU, Softmax, etc.

As part of training the neural network that is nothing but optimizing the weights to transform the features into the outputs, there are some concepts to be aware of:

- **Loss function**, in order to measure the accuracy of the network predictions. It measures the difference between the actual output and the predicted output. An example of the Loss function is Mean absolute error, mean squared error.
- **Optimizer: Stochastic Gradient Descent**, guides the network on how to change the weights. Optimizer helps the network to adjust the weights in order to minimize the loss. These optimization algorithms are iterative algorithms. These algorithms train the model again and again until the loss is small or cannot be minimized anymore.
- **Hyperparameters and how to tune them?** The main hyperparameter related to the training algorithm is:

    - Epoch: This is the number of times the work will see each training data sets. Example - Two epochs means that the data set has been given twice to the model.

- Momentum: It indicates the direction of the subsequent step based on the previous steps.
- Batch/Minibatch: In the case of huge training datasets, the dataset is divided into smaller datasets, which are called batches.

#### *2.3.5.1 Recurrent Neural Network*

Recurrent Neural network, unlike feed-forward neural network tasks decision based on what it has learnt from the past. While the Feed Forward network remembers the information from only training, RNN remembers the information from training and the previous input while giving an output. In a feedforward network, only weights influence the output, while in RNN, the outputs are influenced by weights and by a hidden state vector.

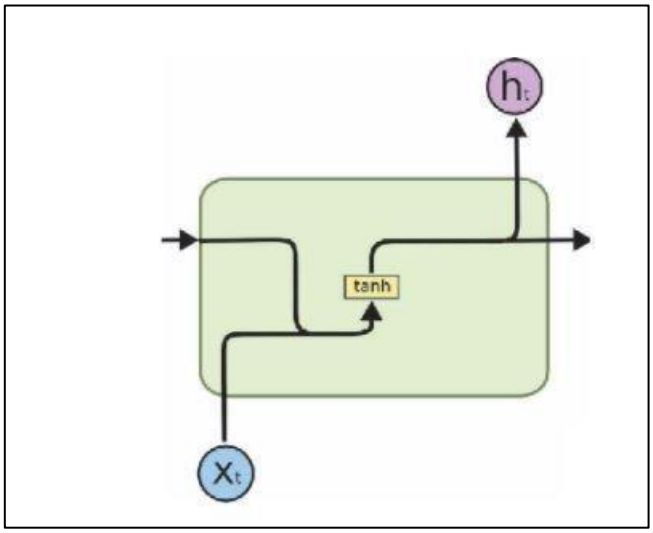


*Figure 14:Recurrent Neural network[17]*

There are couple of limitations of regular RNN which lead to the popularity of other specialised forms of RNN. The limitations are:

- **Exploding Gradient:** When a high importance /value is assigned to weights by the algorithm, it is called Exploding Gradient. Hence when large error Gradients accumulate, leading to large updates to Neural network, making it unstable.

- **Vanishing Gradient:** While moving backward in the network to calculate the loss, the gradient concerning the weights tends to get smaller and smaller, resulting in Neurons in the earlier layer learning slowly compared to the neurons in the last layer. Earlier neurons are essential to learn the patterns; hence this vanishing gradient problem makes the process slow.

#### *2.3.5.2 LSTM and GRU to overcome the Limitations*

LSTMs and GRUs could be used to overcome the limitations of the regular RNN. These have gates responsible for regulating the flow of the information. These gates decide which information can pass them hence making sure only essential and relevant information can pass.

**Long Short-Term Memory Networks**: LSTM stand or Long short-term Memory and is designed to overcome the limitations of RNN as mentioned above. LSTM has the capability to remember information for an extended period of time i.e it has the capacity to retain an information for longer sequences and hence are preferred for such problems.

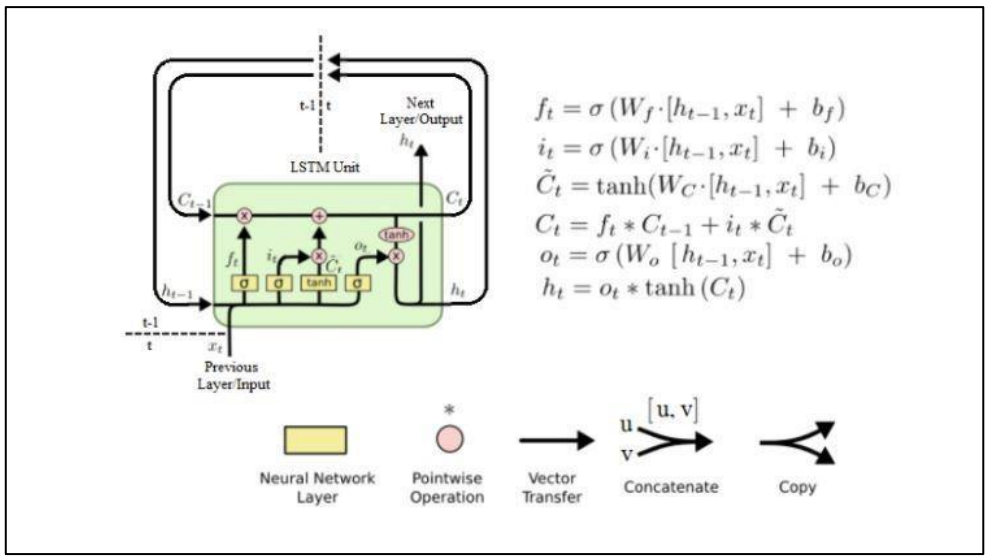


*Figure 15:Understanding LSTM Networks[17]*

- The Sigmoid Function: The sigmoid outputs from 0 to 1 and can be used to forget or remember the information.0: Forget the information and 1: Remember the information
- tanh: This function overcomes the vanishing gradient problem. tanh's second derivative can sustain for a long range before going to zero
- Forget Gate: How much of the past to forget is indicated by this gate
- Input Gate: Signifies what new info will be stored in the cell state
- Output Gate: Sigmoid layer decides which part of cell state is selected for output and tanh layer gives weights to the values(-1 to1)

**Gated Recurrent Units:** GRU is similar to LSTM, method but has only two gates

- Reset gate decides how much information from the past needs to be forgotten or eliminated
- Update gate decides which information from the past needs to be passed to the next steps

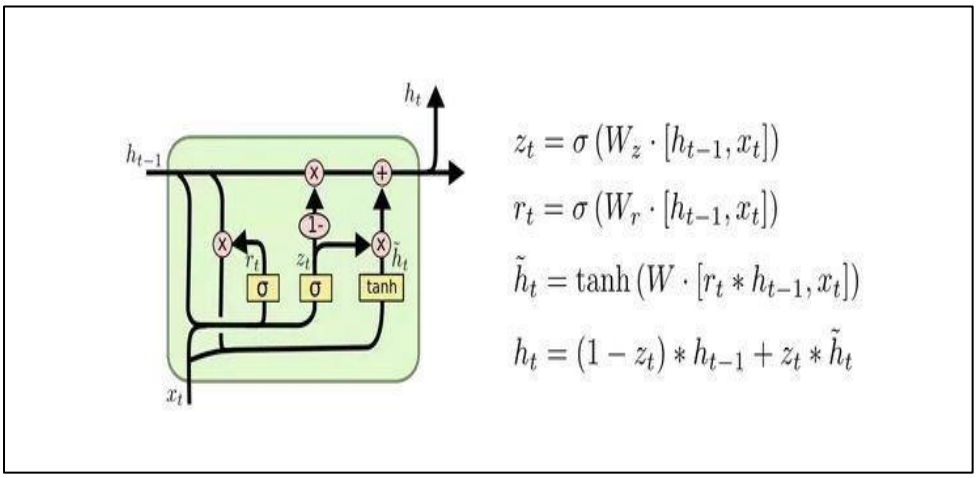


*Figure 16:Understanding Gated recurrent units[17]*

### *2.4.5.3 Multi-Class Classification*

Any classification task could be of two types: Binary Classification and Multiclass classification. The binary class classification consists of two classes, and the Multi-Class classification consists of multiple courses. This research project deals with the Multi-Class classification problem.

From classic machine learning algorithms like support vector machines, deep learning algorithms like LSTMs to Gradient Boosting algorithms like XGboost, all have been tested in the field of Multi-Class classification. Among all the algorithms, XGboost is the most preferred algorithm because of its simplicity to implement and also the efficiency with which it performs.

Most of the real-world Multi-Class classification problems need to be performed on Imbalanced Dataset. An imbalanced dataset is a dataset where the distribution of classes is unequal. Due to this, when these datasets are fed to the Models for classification, there are chances that the results are highly inaccurate because of the class imbalance. Even though the accuracy could be 98 %, that could also result from the fact that 98% of the instances in the datasets belong to one class. Hence it is very crucial to balance the dataset before it to the model.

In the Later section, A thorough comparison of XGboost and LSTMs has been made in terms of its performance in the Multi-Class classification Problem.

## 2.4 Chapter Summary

There are various applications of time series analysis and forecasting, as mentioned previously. This literature review especially focuses on two-time series forecasting applications: Trajectory prediction and Demand forecasting. The reason for choosing these applications in this review, in particular, was their wide applicability across various fields. Trajectory predictions have been implemented widely with the help of traditional algorithms. The boom of machine learning and deep learning methodologies has recently proved to be more efficient in complex trajectory prediction problems like human mobility, vehicles on the highway, vessel trajectory, aircraft trajectory, etc. Demand forecasting is another application of time series forecasting whose benefits have been realized broadly in various fields like sales, supply chain, cloud resources, and many more. The methods to forecast a particular unit's demand and utilization of resources ,make up for popular demand forecasting algorithms .

The review indicates that LSTM(Long short-term memory) networks are among the most popular neural networks for time series forecasting because of their capability to remember information for long term dependencies.

### 2.4.1 Research Gap

**The Limitations** of this review are that it does not give an in-depth overview of these deep learning methods for single step and multi-step forecasting or univariate and multivariate forecasting which, when combined with this study, will require a longer timeline.

**Future work** could focus on surveying methods used for univariate multi-step time series forecasting and multivariate multi-step time series forecasting in demand and trajectory prediction. These methods could be compared, and a conclusion could be drawn upon what methods are preferred in such problems concerning these two applications.

# 3 Methodology

In this research, the methodology used to approach the solution is inspired by "The Data Science Methodology "[22]. In order to convert a business problem into a data science solution, this is one of the most popular and essential methodologies, the methodology in this thesis project has been divided into concrete phases: **Business understanding, Solution design, Data analytics, Solution implementation and Result analysis**

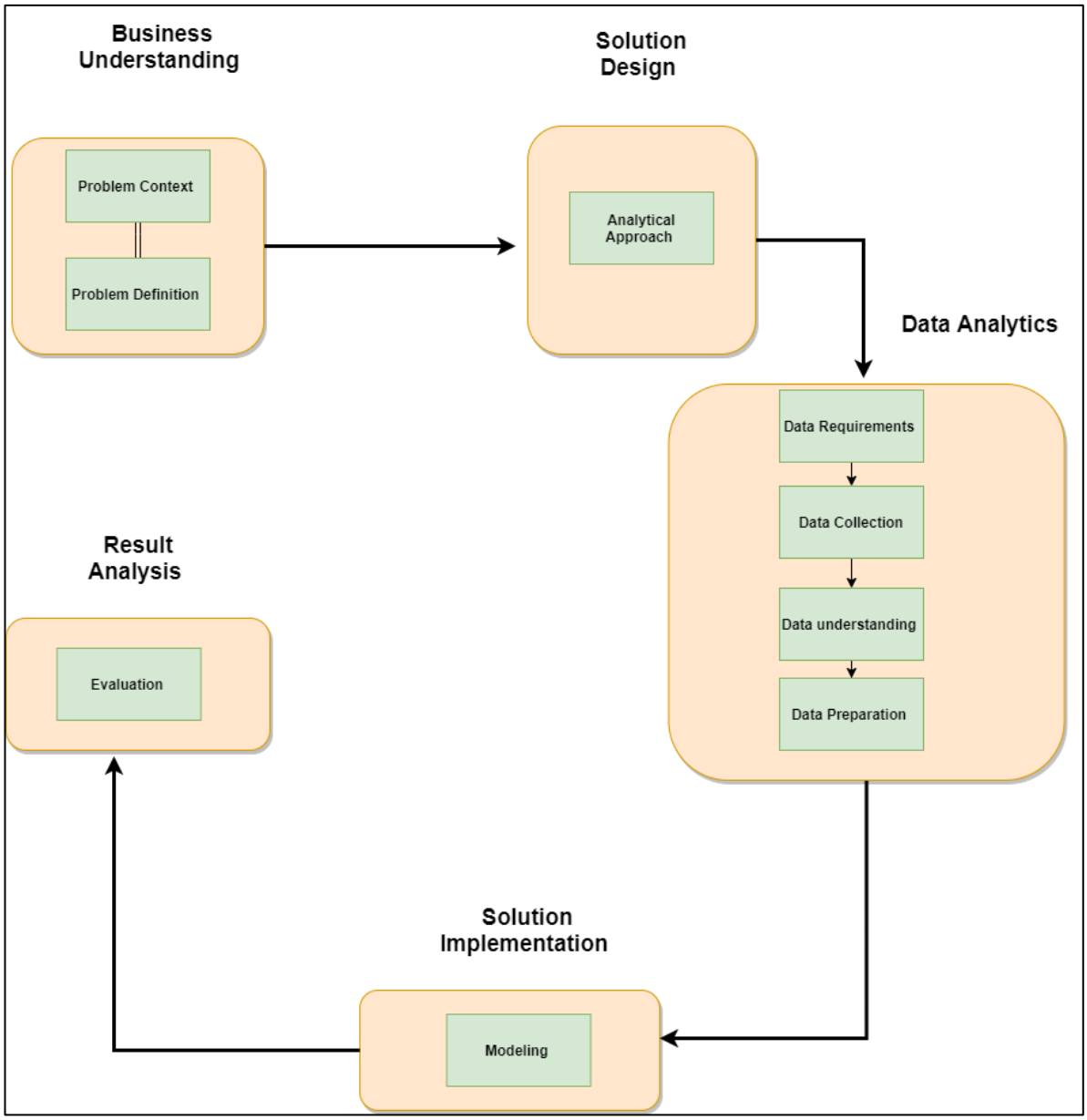


*Figure 17:Data science methodology[23]*

## 3.1 Business Understanding

This phase is about defining the problem to be solved. In order to form a solid foundation of an impactful solution, it is imperative to know the answers to the following questions.

- What is the context of the problem? What is the background business knowledge required in order to understand the problem?
- How to identify and refine the problem statement?

This stage of the methodology comprises of ***Problem context*** and ***Problem definition.*** This research involves breaking down the main objective into multiple sequential questions. Hence it was vital to understand the flow in which these research questions will be answered. Below, the diagram depicts how questions are related to each other and in what order these questions were answered

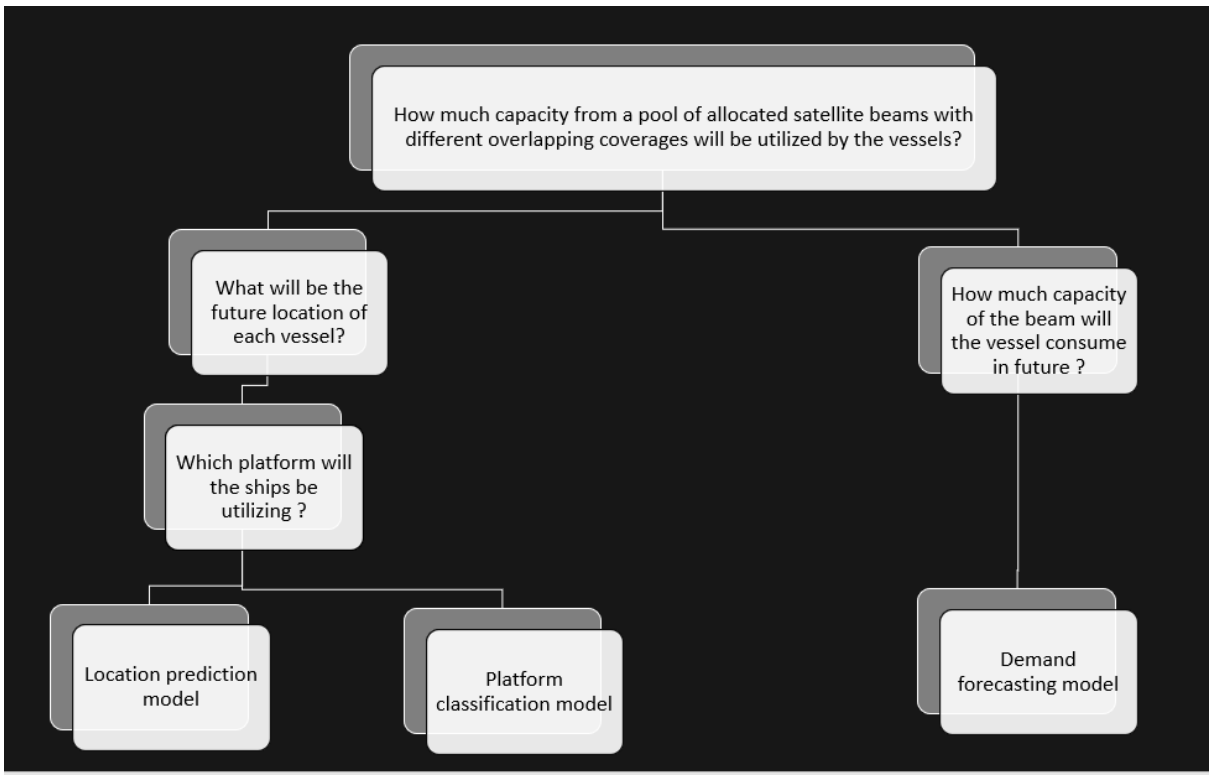


*Figure 18:Problem identification*

## 3.2 Solution Design

After defining the problem statement, as part of solution design, **an analytical approach** is defined. This stage involves decisions like which machine learning or statistical approaches should be used to approach the problem. Different methods based on the Literature survey, past work, type of data was decided in this stage. In the modelling section, different methods used to approach the solution are discussed in detail.

## 3.3 Data Analytics

Any typical Data science problem is 60% of Data analytics and 20% of Modelling and evaluation[20][22]. Data analytics stage consists of different data preparation steps, Right from Data requirements and collection to Data transformation. The next section of the report explains each and every step involved in Data analytics phase of this research project.

- **Data Requirements**: What Data is required to answer the research questions?
- **Data Collection:** What is the source of Data? How is the data retrieved from these sources?
- **Data Understanding**: Is the Data structured or Semi-structured? What are the characteristics of each variable?
- **Data Preparation:** In what ways could the data be transformed to be fed into a predictive model?

## 3.4 Solution Implementation

Once the analytical approach and data preparation is done, it's time to put these into use. This is where modeling comes into picture. Modeling is that stage of Methodology where a data scientist designs and develops different predictive or descriptive models. These models are designed and improved by methods like

hyperparameter tuning. This study involves making predictions ;hence it focuses on predictive modeling.

## 3.5 Result Analysis

**Evaluation**: This stage involves comparison of the results of different models. This comparison is based on different evaluation techniques. These techniques could vary from model to model and since this research project involves three different predictive models and each model is different in nature, evaluation technique for each model also differ accordingly.

## 3.6 Chapter Summary :

This chapter summarised each phase of approaching the objective solution. The following table maps the research questions to the chapters where they have been answered.

**TABLE 1: Research questions and chapters**

| Research Question | Phase :Chapter Name | Chapters |
|---|---|---|
| *1-Can this problem be solved using Traditional time series forecasting methods?* | Business Understanding: Background and Related work | Chapter 2 |
| *2-In this case, Is Feature extraction required for LSTM algorithm for trajectory prediction and demand forecasting techniques?* | Data Analytics :Dataset Preparation | Chapter 4,5 |
| *3-Is Neural Network an optimal solution to this problem?* | Result Analysis :Evaluation | Chapter 6 |
| *4-Which part of Industry4.0 is being harnessed by this solution with respect to Satellite Industry.* | Data Analytics, Result Analysis : Dataset preparation, Evaluation | Chapter 4,5,6 |

# 4.Dataset Preparation

This section describes in detail how the data is collected and what are the transformations applied on the data to make it ready for the prediction model. Since this research project consists of designing three different machine learning models, the data collection and preparation for these models differ accordingly.

**Location Prediction Model**: Machine learning model designed to predict the future location of the ships

**Platform Classification Model**: Machine learning model designed to classify which platform the ship would be using in future

**Capacity Prediction**: Machine learning model designed to forecast future bandwidth utilisation by the ship.

## 4.1 Data Collection

### Database Source:

SAP HANA stands for SAP High-Performance Analytic Appliance. The company uses SAP HANA as one of its databases and the datasets for this project are all stored in its raw form in SAP HANA. It is an in-Memory database. Deployment of this database can take place on premise and cloud. The technology of SAP HANA is column-based, row-based, and object-based".

#### 4.1.1 Location Prediction Model and Platform Classification:

**Table name: TERMINAL_EM_GPS:** Gives all the details of historical location of each terminal captured at the frequency of 5mins

The structure of the table:

| TERMINAL_ID | TIMESTAMP | NETOWRK_ID | LATITUDE | LONGITUDE |
|---|---|---|---|---|

- **TERMINAL_ID** - Unique Identifier of the fixed and mobile terminals
- **TIMESTAMP-** Time at which the data is recorded (Recorded at Five minutes frequency)
- **NETWORK_ID** - Unique id to identify the platform or in other words platform id. This id identifies the unique combination of satellite beam band.
- **LATITUDE** - Latitudinal position of the terminal at the given point of time
- **LONGITUDE**- Longitudinal position of the terminal at the given point of time

### 4.1.2 Capacity Prediction Model:

**Table Name: TERMINAL_IP_VOLUME:** Gives all the details of Satellite beam bandwidth utilisation per platform per Terminal.

| TERMINAL_ID | TIMESETAMP | NETOWRK_ID | RX_TOTAL_KBYTE | TX_TOTAL_KBYE |
|---|---|---|---|---|

- **TERMINAL_ID** - Unique Identifier of the fixed and mobile terminals
- **TIMESTAMP**- Time at which the data is recorded (Recorded at Five minutes frequency)
- **NETWORK_ ID** - Unique id to identify the platform or in other words platform id. This id identifies the unique combination of satellite beam band.
- **RX_TOTAL_KBYTE** – Kilobytes utilised in the return link
- **TX_TOTAL_KBYTE** -Kilobytes utilised in the forward link

## 4.2 Data Exploration

### 4.2.1 Location Prediction Model and Platform Classification Model:

The historical location data of each ship is a time series of its own. Hence it is important to understand that each terminal will have its individual forecasting model. For the location prediction model, few terminals were randomly selected to avoid any bias and to test the efficiency of the algorithm irrespective of the location the terminal is covering.

Below given is the screenshot from the SAP HANA database. It is the historical location of a terminal captured at the frequency of 5 mins for about a year.

**TABLE 2 : Historical location data of each terminal in SAP HANA**

| TIMESTAMP | NETWORK_ID | LATITUDE | LONGITUDE |
|---|---|---|---|
| 20201005120000 | 0000008548 | 51.0043 | 2.1776 |
| 20201005120500 | 0000008548 | 51.0043 | 2.1776 |
| 20201005121000 | 0000008548 | 51.0043 | 2.1776 |
| 20201005121500 | 0000008548 | 51.0043 | 2.1776 |
| 20201005122000 | 0000008548 | 51.0043 | 2.1776 |
| 20201005122500 | 0000008548 | 51.0043 | 2.1776 |
| 20201005123000 | 0000008548 | 51.0043 | 2.1776 |
| 20201005125500 | 0000008548 | 51.0043 | 2.1776 |
| 20201005130500 | 0000008548 | 51.0043 | 2.1776 |
| 20201005131000 | 0000008548 | 51.0043 | 2.1776 |
| 20201005131500 | 0000008548 | 51.0043 | 2.1776 |
| 20201005132000 | 0000008548 | 51.0043 | 2.1776 |
| 20201005132500 | 0000008548 | 51.0043 | 2.1776 |
| 20201005133000 | 0000008548 | 51.0043 | 2.1776 |

Following are the observations made before exporting the data in the required format for further pre-processing. Below given, observations were made for all the terminals picked for the research project.

*TABLE 3:SQL format of the columns(Location prediction model)*

| | View Column | Table Column | SQL Data Type |
|---|---|---|---|
| 1 | MANDT | MANDT | NVARCHAR |
| 2 | TERMINAL_ID | TERMINAL_ID | NVARCHAR |
| 3 | TIMESTAMP | TIMESTAMP | DECIMAL |
| 4 | NETWORK_ID | NETWORK_ID | NVARCHAR |
| 5 | LATITUDE | LATITUDE | NVARCHAR |
| 6 | LONGITUDE | LONGITUDE | NVARCHAR |

**TIMESTAMP** – The timestamp in SAP HANA is in decimal format. The timestamp column given in the database needs is converted into a typical ***DateTime*** type in data extraction phase so as to be processed further in the data transformation phase.

**LATITUDE AND LONGITUDE**: The format of is NVARCHAR, i.e. variable size string data. This is converted to a decimal format in data extraction phase.

### 4.2.2 Capacity Prediction Model:

Just like location data of a ship, historical bandwidth utilisation by every ship is a time series of its own the terminals picked for this model is the same as the terminals picked for location prediction and platform classification model. This is to make sure, that a complete solution is formulated for each terminal.

Below given is the screenshot from the SAP HANA database. It is the historical bandwidth capacity utilisation data captured at the frequency of 5 mins for about a year.

*Table 4 :Historical data of capacity utilisation per terminal*

| TERMINAL_ID | TIMESTAMP | T_INTERVAL | NETWORK_ID | RX_TOTAL_KB… | TX_TOTAL_KB… |
|---|---|---|---|---|---|
| 0000003903 | 20210223074000 | 0000000300 | 0000008548 | 0000000045 | 0000000124 |
| 0000003903 | 20210222232000 | 0000000300 | 0000008548 | 0000001015 | 0000008792 |
| 0000003903 | 20210222234000 | 0000000300 | 0000008548 | 0000000521 | 0000011155 |
| 0000003903 | 20210222225000 | 0000000300 | 0000008548 | 0000005034 | 0000010637 |
| 0000003903 | 20210223000000 | 0000000300 | 0000008548 | 0000005505 | 0000006808 |
| 0000003903 | 20210222232500 | 0000000300 | 0000008548 | 0000000584 | 0000011880 |
| 0000003903 | 20210222225500 | 0000000300 | 0000008548 | 0000002315 | 0000011546 |
| 0000003903 | 20210222235500 | 0000000300 | 0000008548 | 0000003263 | 0000007802 |
| 0000003903 | 20210222234500 | 0000000300 | 0000008548 | 0000002393 | 0000011077 |
| 0000003903 | 20210222230000 | 0000000300 | 0000008548 | 0000000582 | 0000011063 |
| 0000003903 | 20210222235000 | 0000000300 | 0000008548 | 0000005148 | 0000010902 |
| 0000003903 | 20210222224500 | 0000000300 | 0000008548 | 0000002123 | 0000011259 |
| 0000003903 | 20210222231500 | 0000000300 | 0000008548 | 0000002649 | 0000009036 |
| 0000003903 | 20210222233500 | 0000000300 | 0000008548 | 0000000610 | 0000010895 |
| 0000003903 | 20210222231000 | 0000000300 | 0000008548 | 0000000560 | 0000009924 |

Following are the observations made before exporting the data in the required format for further pre-processing. Below given, observations were made for all the terminals picked for the research project.

*TABLE 5:SQL format of the columns (Capacity prediction model)*

| | View Column | Table Column | SQL Data Type |
|---|---|---|---|
| 1 | MANDT | MANDT | NVARCHAR |
| 2 | TERMINAL_ID | TERMINAL_ID | NVARCHAR |
| 3 | TIMESTAMP | TIMESTAMP | DECIMAL |
| 4 | T_INTERVAL | T_INTERVAL | NVARCHAR |
| 5 | NETWORK_ID | NETWORK_ID | NVARCHAR |
| 6 | RX_TOTAL_KBYTE | RX_TOTAL_KBYTE | NVARCHAR |
| 7 | TX_TOTAL_KBYTE | TX_TOTAL_KBYTE | NVARCHAR |

**TIMESTAMP** – The timestamp in SAP HANA is in decimal format. The timestamp column given in the database needs is converted into a typical

**DateTime** type in data extraction phase so as to be processed further in the data transformation phase.

**RX_TOTAL_KB:** Kilobytes utilised by the vessel in return link communication. It is in NVARCHAR format which is converted to decimal type during data extraction phase.

**TX_TOTAL_KB: :** Kilobytes utilised by the vessel by the vessel to the satellite in uplink communication**.** It is in NVARCHAR format which is converted to decimal type during data extraction phase

## 4.3 Data Pre-processing

### 4.3.1 Location Prediction Model and Platform Classification Model:

Historical location data of each terminal has been captured at every 5 mins for about a year. This leads to repetition of geographical points (Latitude and longitude). In other words, the location of a terminal is pretty much constant for the entire day, hence it would be efficient to pick single pair of latitude and longitude representative of each day. This will significantly reduce the burden of unnecessary records and give a clearer picture of the different location points covered by the terminal.

Before moving to data transformation in python, it was important to extract data from SAP HANA and do a part of data pre-processing while extracting the data. Data extraction from SAP HANA was done in SAP HANA SQL as shown in the following picture.

```
SELECT
*
FROM
(
    SELECT

    temp2.TERMINAL_ID
    ,temp2."DATE"
    ,temp2."Latitude"
    ,temp2."Longitude"
    ,temp2."NETWORK_ID"
    ,ROW_NUMBER() OVER (PARTITION BY  temp2.TERMINAL_ID, temp2."DATE") "Rnk"

    FROM
    (
        SELECT DISTINCT

         GPS.TERMINAL_ID
         ,GPS.NETWORK_ID
        ,TO_DATE     (GPS.TIMESTAMP) "DATE"
        ,TO_DECIMAL(GPS.LATITUDE) "Latitude"
        ,TO_DECIMAL(GPS.LONGITUDE) "Longitude"


        FROM

        "ZBW_HANA"."ses.zbw.MRC::TERMINAL_EM_GPS" GPS

    ) temp2
) temp3

WHERE temp3."Rnk" =1 AND temp3.TERMINAL_ID =3903
```

*Figure 19:Code snippet of  SAP query used to extract the location data*

The ***Timestamp column*** is converted to a Date format (SAP HANA Date Format). The Latitude and Longitude data are converted to SQL Decimal format. Millions of records in the dataset are queried to extract Terminal Location Data of each day. This significantly reduced the amount of data and aggregated the data on a "Daily Frequency".

Below given picture is the Location Data of a Terminal Mapped on ***a POWER BI VISUAL***. It could be seen that there is lot of randomness in the patterns of locations visited by the terminal. Hence this rule out the possibility of pattern mining from the Terminal Trajectory. This adds to the complexity of the problem and the only way to deal with this is discussed in detail in the" Modelling Section"

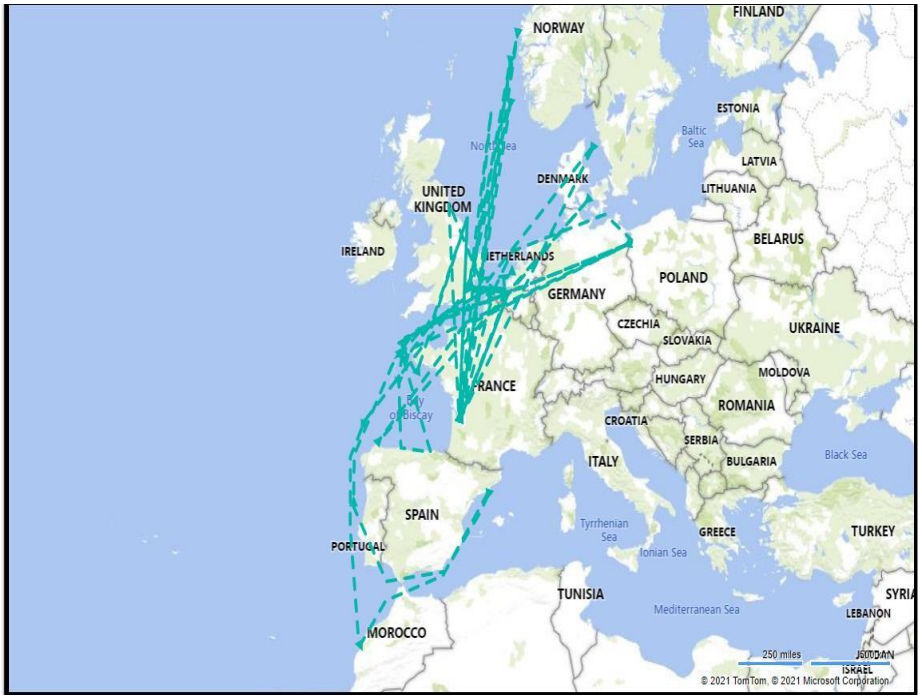


*Figure 20:Movement of terminal depicting randomness*

### 4.3.2 Capacity Prediction Model:

For capacity prediction, instead of extracting individual records of every 5 minutes of Forward and Return Rate, it was decided to perform aggregations on the data which could significantly reduce the amount of data but still retain useful information.

There are many possibilities of aggregating data as listed below:

- Every day for each terminal: 97 Percentile of (TX_TOTAL_KBYTE) and 97 Percentile of (RX_TOTAL_KBYTE) for the whole day.

- Every day for each terminal: 97 Percentile of (TX_TOTAL_KBYTE) and 97 percentile of (RX_TOTAL_KBYTE) in the Morning(Up until 12pm) , 97 Percentile of (TX_TOTAL_KBYTE) and 97 percentile of (RX_TOTAL_KBYTE) in the evening (Till 12am midnight)

- Every day for each terminal: Maximum of (TX_TOTAL_KBYTE) in the Morning (Up until 12pm), Maximum of (RX_TOTAL_KBYTE) in the evening (Till 12am midnight)

- Every day for each terminal: Average of (TX_TOTAL_KBYTE) in the Morning (Up until 12pm), Average of (RX_TOTAL_KBYTE) in the evening (Till 12am midnight)

After careful consideration and research, it was decided to perform the first aggregation i.e. : 97 Percentile of (TX_TOTAL_KBYTE) and 97 percentile of (RX_TOTAL_KBYTE) in the Morning(Up until 12pm) , 97 Percentile of (TX_TOTAL_KBYTE) and 97 percentile of (RX_TOTAL_KBYTE) in the evening (Till 12am midnight) .The other possibilities were ruled because of the following reasons –

Aggregating based on the “MAXIMUM” will not give a proper idea of the downtimes or the times when capacity utilization was minimum.

Similarly, the “AVERAGE” amount could be misleading considering the peaks, lows, and regular data usage by the terminals. Hence two options were left:

97 Percentile of (TX_TOTAL_KBYTE) and 97 percentile of (RX_TOTAL_KBYTE) for the whole day and 97 Percentile of (TX_TOTAL_KBYTE) and 97 percentile of (RX_TOTAL_KBYTE) in the Morning(Up until noon), 97 Percentile of (TX_TOTAL_KBYTE) 97 percentile of (RX_TOTAL_KBYTE) in the evening (Till midnight).

The Data was extracted in both ways. It was observed that in the second case, the evening data was accounting for very low usage. Therefore, most of the outliers from the dataset were the records pertaining to the evening data. However, instead of removing these outliers, it was decided to go with the first option. The first option will consider the whole distribution of the data across the day and take data in the evening hours into account and then calculate 97 percentiles. This seemed to be a win-win situation.

Final Aggregation was done in SAP HANA in data extraction phase in the following way.

```
SELECT DISTINCT

TO_DATE("TIMESTAMP") "Date"
,HOUR(TO_TIMESTAMP("TIMESTAMP")) "Hour"
,TERMINAL_ID
,ROUND(PERCENTILE_CONT(0.97) WITHIN GROUP (ORDER BY 0.008*TO_DECIMAL(TX_TOTAL_KBYTE)/300)
OVER (PARTITION BY TO_DATE("TIMESTAMP"), TERMINAL_ID),4) "FWD, 97 %ile"
,ROUND(PERCENTILE_CONT(0.97) WITHIN GROUP (ORDER BY 0.008*TO_DECIMAL(RX_TOTAL_KBYTE)/300)
OVER (PARTITION BY TO_DATE("TIMESTAMP"), TERMINAL_ID),4) "RTN, 97 %ile"

FROM
"ZBW_HANA"."ses.zbw.MRC::TERMINAL_IP_VOLUME"
WHERE TERMINAL_ID =3903
```

*Figure 21:Code snippet of SAP query used to extract the capacity data*

In the above query, TX_TOTAL_KBYTE and RX_TOTAL_KBYTE is first converted from Kilobytes to Megabits per sec(Mbps) and 97 percentiles is calculated using PERCENTIE_CONT function

As stated in SAP HANA docs - Returns an interpolated value using the percentile value of *<constant_literal>*. A continuous distribution of the values of *<expression>* is assumed.

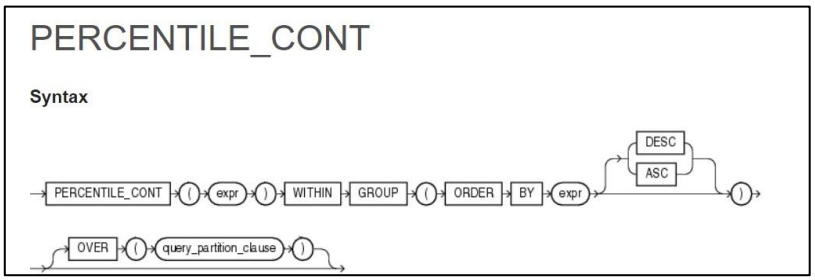


*Figure 22:Depiction of PERCENTILE_CONST function*

## 4.4 Data Transformation

The Data transformation in Location prediction model and Capacity prediction model is done in such a way that it enables the Neural network to map the input sequence to the output sequence. Hence both the Time series forecasting problems in this project have been solved by converting them to supervised learning problem. The below given picture depicts a general idea of how the time series forecasting problem is connected to a supervised learning problem.

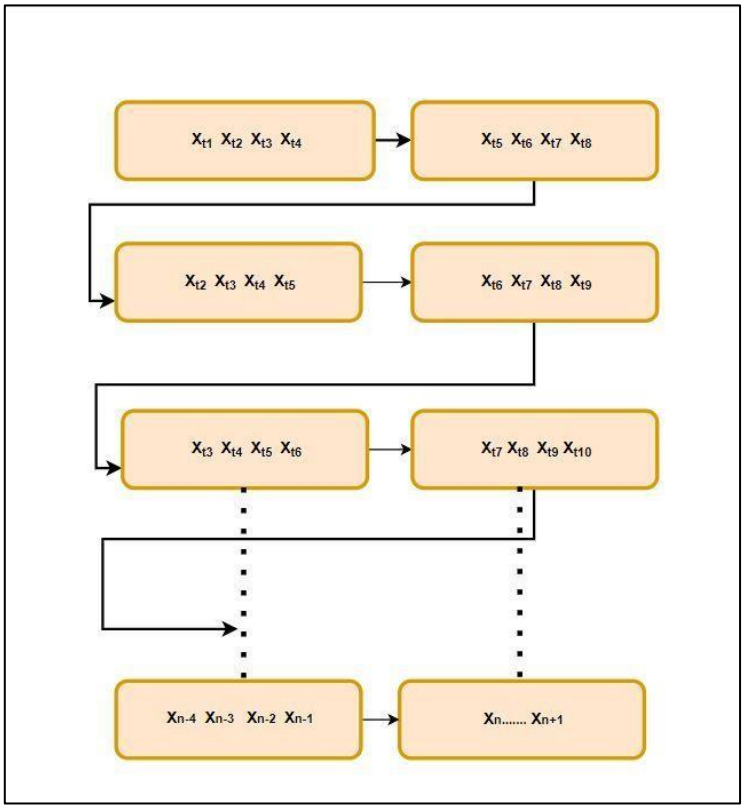


*Figure 23:Convert a time series to a supervised learning problem*

X1- - - - - - - - - - - -Xn, Xn+1 symbolise the observations recorded at timestep where n. In order to convert time series to a supervised learning problem, the observations are shifted by **a Sliding window** of determined length. This length is optimized based on the problem and the desired results. For example, in the above-given picture, the window length is 1; the function shifts the observations by a timestep of 1. Each shift results in a new set of input and output sequences,

enabling the neural network to map the historical observation with the future observation while training.

As mentioned in the Literature sections, a Time series forecasting could be either Univariate or Multivariate. In this Research Project, we are dealing with Multivariate time series prediction, which also adds to the complexity of the problem. The predictions need to be made conjointly with other variables in such a problem as there is more than one variable with temporal dependency. These variables are dependent on their past values and can also be dependent on past values of other variables. These Variables are called **Endogenous Variables.**

### 4.4.1 Location Prediction Model:

In order to build a machine learning model for predicting future location of the ship, there are two possible ways to transform the data.

- Multivariate and Multiple Input Series.
- Multivariate and Multiple Parallel Series.

**Multivariate and Multiple Input Series**: In this transformation. The data is transformed into a supervised learning problem in such a way that it enables the model to predict future Latitude based on historical Values of latitude and longitude. Similarly, the dataset is transformed again to enable the model to predict the Longitude based on historical values of latitude and longitude.

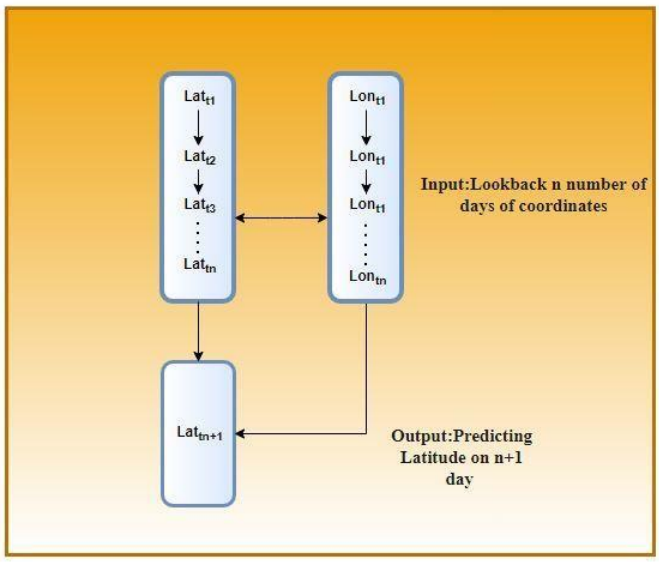


*Figure 24:Predicting latitude one-time step in future*

In Figure 22, latitude, and longitude up to time $\mathbf{t_n}$ are fed into the model as input sequence (A multivariate sequence) and latitude is predicted one-time step into the future ($\mathbf{t_{n+1}}$)(Univariate Output)

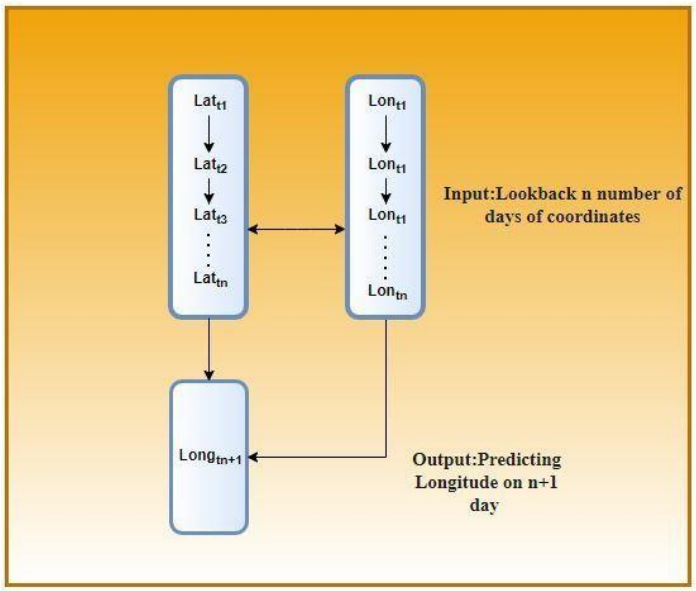


*Figure 25:Predicting longitude one time-step in future*

In Figure 23, Latitude and Longitude up to time Tn are fed into the model as input sequence (A multivariate sequence) and Latitude is predicted one-time step into the future, $\mathbf{t_{n+1}}$(Univariate Output)

**Multivariate and Multiple Parallel Series:** In this transformation. The data is transformed into a supervised learning problem in such a way that it enables the model to predict future Latitude and Longitude parallelly based on historical Values of Latitude and Longitude.

In Figure 24, Latitude and Longitude up to time Tn are fed into the model as input sequence (A multivariate sequence) and Latitude and Longitude are predicted simultaneously one-time step into the future, $\mathbf{t_{n+1}}$(Multivariate Output)

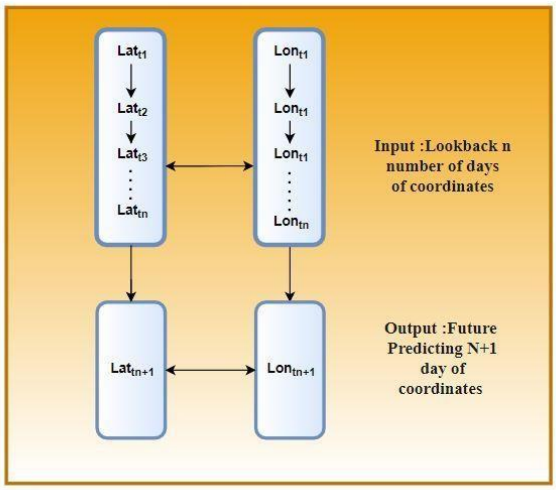


*Figure 26:Simultaneous prediction latitude and longitude*

### 4.4.2 Platform Classification Model:

For this model, data transformation was independent of time component. The reason being the beam utilised at each set of coordinates was dependent on the strength and availability of beam at the particular location. Hence the focus

was to feed the model the set of observed latitude and longitude as input and the beam utilised (Platform) by the vessel at that location as output.

Figure 23 depicts how the data was prepared for this model. As shown in the figure the platform ids are unique ids and are categorical type of data. In order to feed any categorical data to a machine learning model it has to be Label encoded.

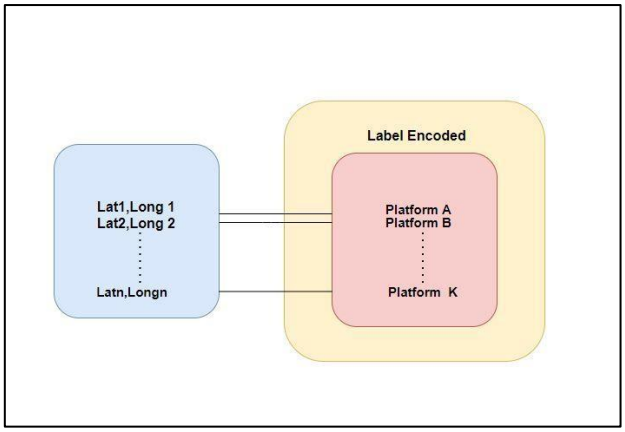


*Figure 27:Pair of coordinates mapped to a label encoded platform*

**Label Encoding**

| X1 | X2 | Y1 |
|---|---|---|
| 51.3695 | 2.7298 | 23703 |
| 50.7824 | 0.9101 | 8548 |
| 51.4237 | 1.4427 | 7770 |

***{"23703" →0***

***"8548" →1***

***"7770" →2}***

| X1 | X2 | Y1 |
|---|---|---|
| 51.3695 | 2.7298 | 0 |
| 50.7824 | 0.9101 | 1 |
| 51.4237 | 1.4427 | 2 |

### 4.4.3 Capacity Prediction Model:

Data transformation in this model is similar to the Location prediction model. In order to build a machine learning model for predicting capacity of the beam utilisation by the ship in future, there are two possible ways to transform the data.

- Multivariate and Multiple Input Series.
- Multivariate and Multiple Parallel Series

**Multivariate and Multiple Input Series**:

In this transformation. The data is transformed into a supervised learning problem in such a way that it enables the model to predict future forward/Uplink Megabits based on historical Values forward/Uplink and Downward/Return Megabits. Similarly, the dataset is transformed again to enable the model to predict the Downward/Return Megabits based on historical values of forward/Uplink and Downward/Return Megabits.

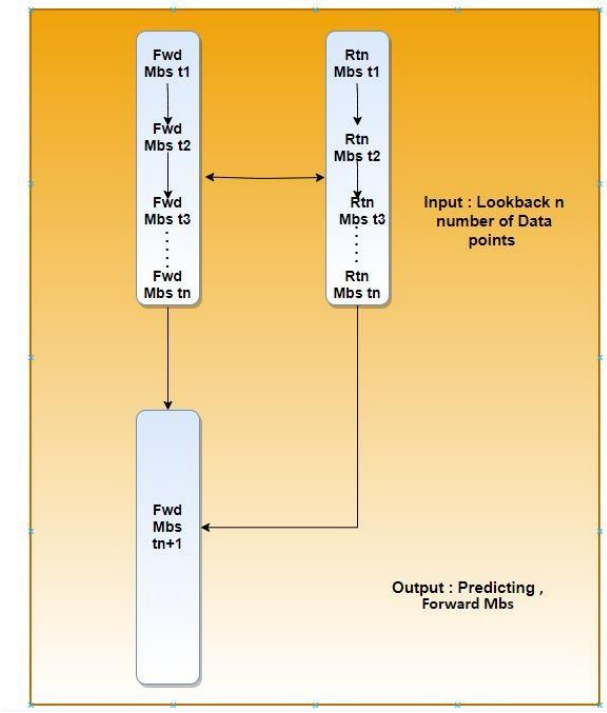


*Figure 28:Predicting forward Mbps one-time step in future*

In Figure 26, Forward and Return Mbps up to time $\mathbf{t_n}$ are fed into the model as input sequence (A multivariate sequence) and Forward Mbps are predicted onetime step into the future ($\mathbf{t_{n+1}}$)(Univariate Output)

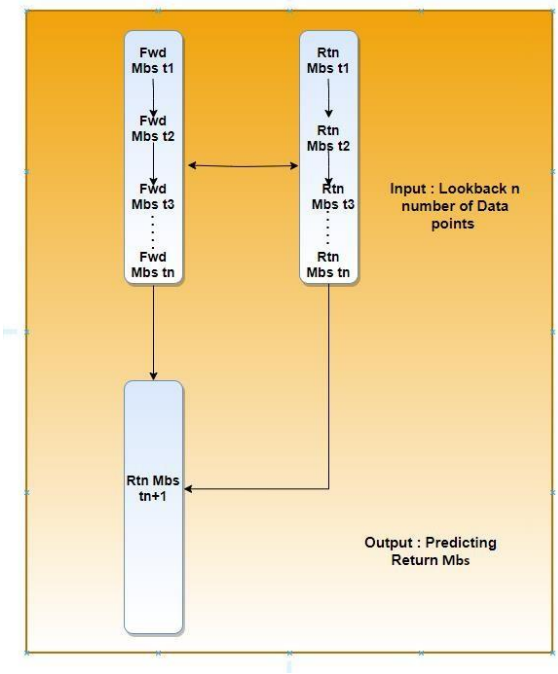


*Figure 29:Predicting return Mbs one-time step in future*

In Figure 27, Forward and Return Mbps up to time $\mathbf{t_n}$ are fed into the model as input sequence (A multivariate sequence) and Return Mbps is predicted one-time step into the future, $\mathbf{t_{n+1}}$(Univariate Output).

**Multivariate and Multiple Parallel Series:**

In this transformation. The data is transformed into a supervised learning problem in such a way that it enables the model to predict of forward/Uplink and Downward/Return Megabits parallelly based on historical Values of forward/Uplink and Downward/Return Megabits.

In the figure 28, Forward/Uplink and Downward/Return Megabits up to time Tn are fed into the model as input sequence (A multivariate sequence) and of forward/Uplink and Downward/Return Megabits are predicted simultaneously one-time step into the future, tn+1(Multivariate Output)

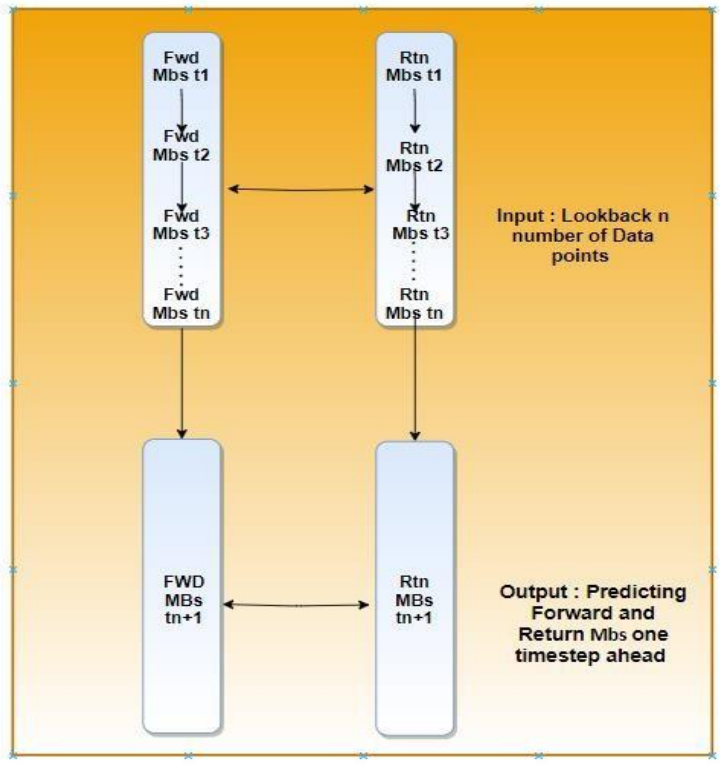


*Figure 30:Simulatenous prediction*

## 4.5 Chapter Summary :

This chapter demonstrate the longest phase of the study : The Data Analytics .It involved data collection ,data exploration and data Transformation Data was collected from SAP HANA database .Data transformation for both Capacity prediction model and Location prediction model are done in similar ways .

- Multivariate and Multiple Input Series.
- Multivariate and Multiple Parallel Series

For Platform classification model data was transformed by label encoding the categorical data .The next chapter ,Modeling will demonstrate various experiments done with these transformations on all three models .

Figures 22 and 23 from Location prediction model and figure 26 and 27 from capacity prediction model follow the same concept of data transformation : Multivariate and Multiple Input series .The only difference in these figures are the

variables. In figure 22 and 23 the variables are latitude and longitude that are provided to location prediction model while in the figures 26 and 27 forward and return Mbps.

Figure 24 from Location prediction model and figure 28 from capacity prediction model follow the same concept of data transformation: Multivariate and Multiple Parallel series. As mentioned above the difference is the variables, figure 24 is from location prediction model hence variables are latitude and longitude while figure 28 is from capacity prediction model hence variables are forward and return Mbps.

# 5 Modeling

This section is dedicated to the Modeling decision taken and experimental Setup done for all the three Models designed in this research. These Decisions were taken by referring to past work in a similar domain and answering the sub research questions mentioned at the beginning of this report.

Neural networks are attracting a lot of attention of researchers in the field of time series forecasting and classification .This research focuses on taking the work done in this domain a step ahead by implementing one of the most popular Neural networks in the field of time series forecasting in Location prediction ,Capacity prediction and Platform classification in satellite communication industry.

## 5.1 Tools and Libraries

**Data Processing and Modeling:** Below given Libraries are the libraries used for Modeling and Data processing

- **Pandas:** Pandas is one of the most popular and widely used libraries of python used for Data analysis. From data wrangling to Data mining, Pandas has to offer many functions, making it easier and quicker to execute various data analysis operations. Most of the data analysis operations performed in the data pre-processing, and modeling stage in this research has been through Pandas.
- **Keras:** It is an open-source library in python used for Developing, Implementing, and evaluating Deep Learning models. Since this research project revolves around Neural networks, this is one of the essential libraries.
- **Sklearn:** The evaluation metrics used for result analysis in this research project, for example, mean squared error, have been imported from Sklearn.

**Visualisation:** Below given tools are the tools used for the visualisation of data and prediction results.

- **Power BI:** It is a dashboarding tool by Microsoft. This tool was used to visualize the location prediction model results on a global map

- **Matplotlib:** Matplotlib is one of the most preferred libraries for performing basic visualizations. This, too, is used to visualize the actual results vs. predicted results using graphs.

## 5.2 Experimental Setup and design:

The experiments are carried out individually on a vessel data. The same vessel has been considered for all the three models for definite reasons and in order to answer the main objective research question.

### *5.2.1* Location Prediction Model:

**Data Scaling**:

Neural networks are sensitive to the values being fed to the model. The difference in the scale of the values belonging to different features may lead to discrepancies in the weight values thus causing instability. Due to this reason, it is recommended to scale the values of all the features in such a way that the scales do not differ a lot from each other. There are a couple of ways to do that. These ways are discussed in the "Capacity Prediction Model". In the" Location Prediction Model," the scaling has been done differently. Instead of following the traditional ways of scaling the data, the Geographical coordinates in the dataset have been scaled by converting them to "Radians "unit. This conversion not only helps in avoiding the large or small weight values in the Neural network but also helps in holding the relevance of location data intact.

**Train and Test split**:

In any machine learning problem, a dataset is split into Train and Test data set to train the model with the training dataset and evaluate its performance on the test dataset. Generally, this split is done randomly, but temporal dependency is

an essential factor in the case of a time series problem. Splitting a dataset randomly into train and test set in time series problems will not give beneficial results as the predictions could be biased. The reason could be that test points might be closer to a point from the training data set temporally; hence the prediction will be quickly learned by the Model. In order to simulate real-world forecasting into the future, the train test split should be done in such a way that all the points in the train set are in the past ordered by time, and all the points in the test set should be in the future in accordance with the train set, ordered by time. The Dataset is divided into three datasets in the Location Model: Train, Validation, and Test. This split has been done manually without the use of any function. The Dataset for all the tested vessels comprises about 450 days of data after data prepossessing and aggregating the data. As mentioned in the "Data Pre-processing" section, the frequency of the raw data was 5 mins, and after aggregation, the data has been reduced to a frequency of "A Day". The Train Validation-Test split has been done so that the train test has approximately 280 days of data. The Validation set has approximately 70 days of data, and the Test set has approximately 90 days of data. The reason for this split is to make sure that the Model has just enough data to train and validate the results during training. Also, 90 days of data is a good amount of data to test the predictions get a good idea about the Model's performance.

Before feeding the Datasets to the model for training and testing, they are reshaped.

**Hyperparameter Tuning:**

Hyperparameter tuning is one of the most critical aspects of machine learning algorithms. When dealing with a Neural network, a few hyperparameters need to be fine-tuned to make the model as efficient as possible. These parameters are tuned by setting them to different values in as many combinations as possible. There are three ways to do it, Grid Search, Random Search, and Manual tuning. Keeping the computational power and time complexity in mind, the best choice in terms of efficiency was Random Search.

The values for parameters in Random search were as following.

**Learning Rate: {**0.1,0.01,0.001,0.0001}

**Number of Layers:** {1,2,34}

**Number of Neurons in each layer: {**32,50,64,100,224,512**}**

```
model1 = Sequential()
model1.add (LSTM(100, input_shape=(train_X.shape[1],
train_X.shape[2]),return_sequences=True))
model1.add (LSTM(50,return_sequences=True))
model1.add((LSTM(50)))
model1.add (Dense(1))
model1.compile(Adam(lr =0.0001),loss='mae',metrics=['mean_squared_error',
'mean_absolute_error', 'mean_absolute_percentage_error', 'cosine proximity'])
```

*Figure 31:Final experiment setting of LSTM: Longitude*

```
model11 = Sequential()
model11.add(LSTM(200, input_shape=(train_X.shape[1],
train_X.shape[2]),return_sequences=True))
model11.add(LSTM(100,return_sequences=True))
model11.add(LSTM(100,return_sequences=True))
model11.add((LSTM(50)))
model11.add(Dense(1))
model11.compile(Adam(lr =0.0001),loss='mae',metrics=['mean_squared_error',
'mean_absolute_error', 'mean_absolute_percentage_error', 'cosine_proximity'])
history = model11.fit(train_X, train_y, epochs=200, batch_size=10, validation_data=(validation_X,
validation_y), verbose=2, shuffle=False)
```

*Figure 32:Final experiment setting of LSTM: Latitude*

### *5.2.2* Platform Classification Model:

**Imbalanced Dataset**: As mentioned in the literature section real-world multiclass classification problems tend to be imbalanced and there are various ways to balance the data. The dataset for the platform classification model has been balanced using the parameter class_weight='balanced'

**Train and Test split:**

As mentioned in the previous section, this model did not have a temporal dependency, which also means the train test split here could be a random split. This split helps the model to understand the classification pattern in depth by enabling the model to learn as many mapping instances of latitude and longitude pairs with their platform id.

In this Model, apart from neural networks, we have also implemented A XGBoost classifier based on the algorithm's success and popularity with most of the classification problems in the world of machine learning. Since the focus of the research is to measure the efficiency of LSTM, XGBoost seemed to be the best competitor for this classification model. As mentioned in the literature section, LSTMs were chosen to be tested in this project because they have been one of the most preferred in time series prediction problems. Since this particular model is devoid of temporal dependency, measuring its efficiency versus a gradient boosting network like XGBoost is essential.

The Train test split for this model was done using the train_test_split function by keeping a random state of 42. A random state could be of any value and was given to the function to shuffle the data identically, enabling the results to be identical. This is important to execute an insightful result analysis.

**Parameter Setting for XGBoost** : The parameter setting for XGboost Classifier is a default setting .The only parameter value that deviates from default is objective. The value for Objective parameter is multi: softprob in order to predict probability of each class rather than the class itself.

**Why Probabilistic approach: Probabilistic Vs Deterministic classification?**

Deterministic Classification here means that only one class(Platform) is predicted for a given set of coordinates. In contrast, Probabilistic Classification means the probability for each Platform is predicted for a given set of coordinates. This research focuses on the latter—the reason being various factors involved in the platform being chosen at a given location at a given point

of time. There are different overlapping beam coverages at a given location, and the platform used by a ship could depend on various factors like beam strength, availability, etc. Hence to account for these uncertainties and multiple factors, it is essential to consider the probabilities of a platform that will be used at a given location. The algorithm will calculate these probabilities by looking into thousands of previous records. Based on these probabilities a well-informed decision could be made

**Final Experiment Setting :**

```
model = Sequential ()
model.add (LSTM(200, input_shape=(Trainc_x.shape[1], Trainc_x.shape[2]),return_sequences=True))
model.add (LSTM(100, return_sequences=True))
model.add (LSTM (50,return_sequences=True))
model.add((LSTM(50)))
model.add (Dense(15,activation ='softmax'))
model.compile(Adam(lr =0.0001),loss='categorical_crossentropy',metrics=['accuracy'])
```

*Figure 33:Neural network: LSTM*

```
model =XGBClassifier(base_score=0.5, booster='gbtree', class_weight='balanced',
        colsample_bylevel=1, colsample_bynode=1, colsample_bytree=0.5,
        gamma=0.0, gpu_id=-1, importance_type='gain',
        interaction_constraints='', learning_rate=0.25, max_delta_step=0,
        max_depth=10, min_child_weight=1,
        monotone_constraints='()', n_estimators=100, n_jobs=12,
        num_parallel_tree=1, objective='multi:softprob', random_state=0,
        reg_alpha=0, reg_lambda=1, scale_pos_weight=None, subsample=1,
        tree_method='exact', validate_parameters=1, verbosity=None)
```

*Figure 34:Code snippet of XGBoost*

### *5.2.3* Capacity Prediction Model :

**Data Scaling :**

For the reason similar to what mentioned in the “Platform Classification Model” it is important to scale the input values in the Capacity prediction model as well. Although the scaling is done differently here. The Values are scaled using MinMaxscaler () function where all the values are scaled to a minimum value of 0 and maximum value of 1 .

**Train and Test split**:

Similar to the location model the dataset is divided into three datasets: Train, Validation and Test. This split has been done manually without the use of any function. The Dataset for all the tested vessels comprises about 450 days of data after data prepossessing and aggregating the data. As mentioned in the "DataPre-processing" section, the frequency of the raw data was 5 mins, and after aggregation, the data has been reduced to a frequency of "A Day". The Train Validation-Test split has been done so that the train test has approximately 280 days of data. The Validation set has approximately 70 days of data, and the Test set has approximately 90 days of data. The reason for this split is to make sure that the model has just enough data to train and validate the results during

training. Also, 90 days of data is a good amount of data to test the predictions get a good idea about the model's performance. Before feeding the Datasets to the model for training and testing, they are reshaped.

**Hyperparameter Tuning:**

Hyperparameters for this model are tuned by setting them to different values in as many combinations as possible. There are three ways to do it, Grid search, Random search, and Manual tuning. Keeping the computational power and time complexity in mind, the best choice in terms of efficiency was Random search.

The values for parameters in Random search were as following.

**Learning Rate: {**0.1,0.01,0.001,0.0001}

**Number of Layers:** {1,2,34}

**Number of Neurons in each layer: {**32,50,64,100,224,512**}**

```
model2 = Sequential()
model2.add (LSTM (100, input_shape=(train_X.shape[1], train_X.shape[2]),return_sequences=True))
model2.add (LSTM(50,return_sequences=True))
model2.add((LSTM(50)))
model2.add(Dense(1))
model2.compile(Adam(lr =0.0001),loss='mae',metrics=['mean_squared_error'])
history = model2.fit(train_X, train_y, epochs=1000, batch_size=64, validation_data=(validation_X,
validation_y), verbose=2, shuffle=False)
```

*Figure 35:Experiment setting of LSTM*

## 5.3 Chapter Summary

This Chapter demonstrated various experiments performed to arrive at the final solution for all three models.

1- **Location Prediction Model:** LSTM model was built to predict the latitude and longitude individually and simultaneously. The final experiment setting that was chosen was based on the best results obtained. These results were obtained after performing hyperparameter tuning on the model.

2-**Platform Classification Model:** Xgboost and LSTM both were implemented to predict which platform class belongs to the given set of latitude and longitude. The final settings of both the algorithm were determined based on the results.

3- **Capacity Prediction Model:** LSTM model was built to predict the forward and Return Mbps individually and simultaneously. The final experiment setting that was chosen was based on the best result obtained. These results were obtained after performing hyperparameter tuning on the model.

# 6 Result Analysis

This chapter summarises the results obtained from various settings of each of the model designed ,to answer each research question. This chapter also explains the motivation behind the final version of each model.

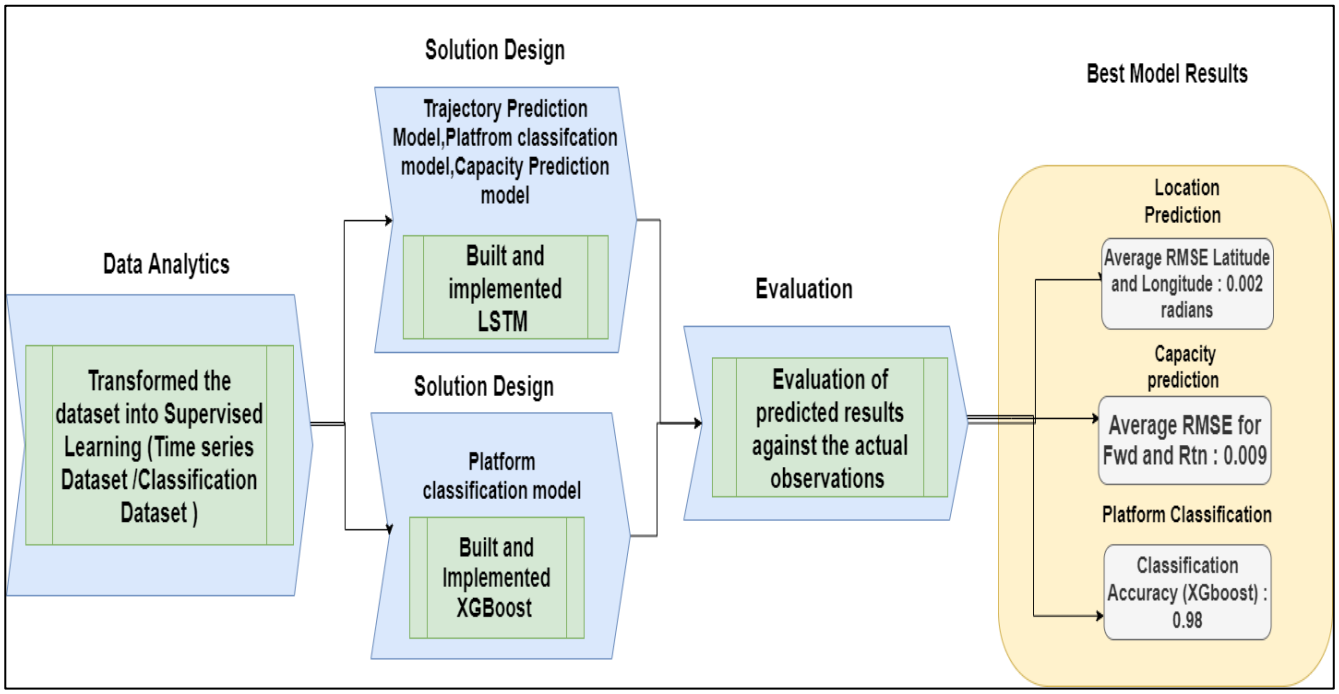


*Figure 36:Architectural overview of solution design*

## *6.1* Location Prediction Model:

As mentioned in the "Data Transformation" section, there were two possible ways to transform data for this model, namely:

- Multivariate and Multiple Input Series (Single Step Forecast)
- Multivariate and Multiple Parallel Series (Multi-Step Forecast)

It was observed that "***Multivariate and Multiple Input Series (Single Step Forecast)***" transformation performed significantly better as compared to

***Multivariate and Multiple Parallel Series (Multi-Step Forecast)***

From, Figure 37 and Figure 38 the difference between the prediction results obtained from **Multivariate and Multiple Parallel Series** and **Multivariate and Multiple Input Series** for a particular terminal (id.163634) respectively.

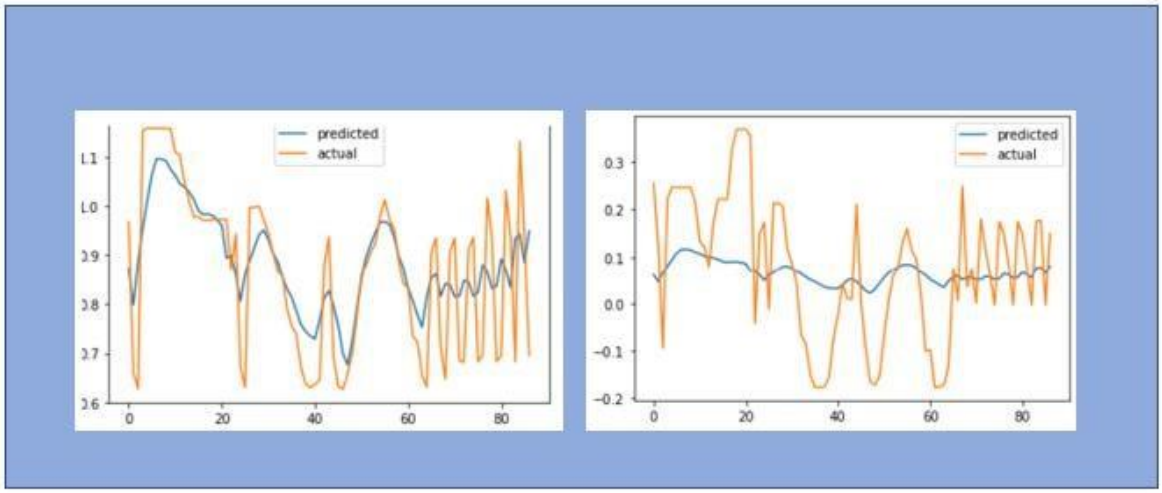


*Figure 37:Multi-Step longitude prediction(left)and multi-step latitude prediction(right)*

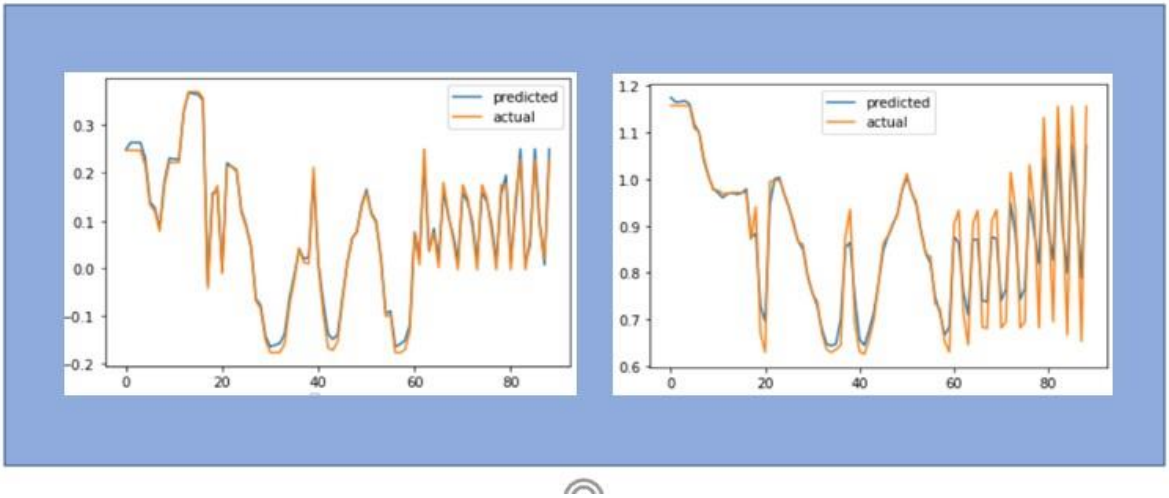


*Figure 38:Single-step longitude prediction(left)and single-step latitude prediction(right)*

**Importance of Hyperparameter Tuning:**

As mentioned in the experiment design and setup section, Hyperparameter tuning plays a significant role in designing a machine learning model. Therefore, the results could be improved by tuning the hyperparameter of a neural network. The following figures will show the difference between results obtained before and after hyperparameter tuning.

| Hyperparameter Tuning | Latitude RMSE | Longitude RMSE |
|---|---|---|
| **Before** | 0.028 | 0.017 |
| **After** | 0.002 | 0.003 |

Although RMSE could be one of the most important evaluation metrics but when it comes to location coordinates ,it is always good to also refer to the coordinates visualisation on a graph and map in order to analyse the results more efficiently. Since here although the location is a numerical figure ,it does not refer to a quantity and hence should be better judged based on visualisation. Following graphs give a better understanding of how hyper parameter tuning improves the predictions .

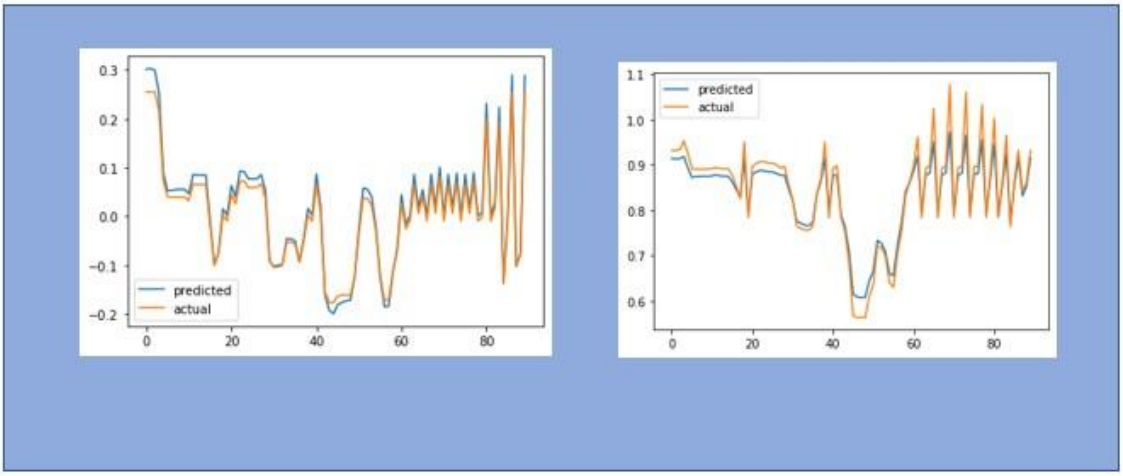


*Figure 39:Before hyperparameter tuning longitude(left) latitude(right)*

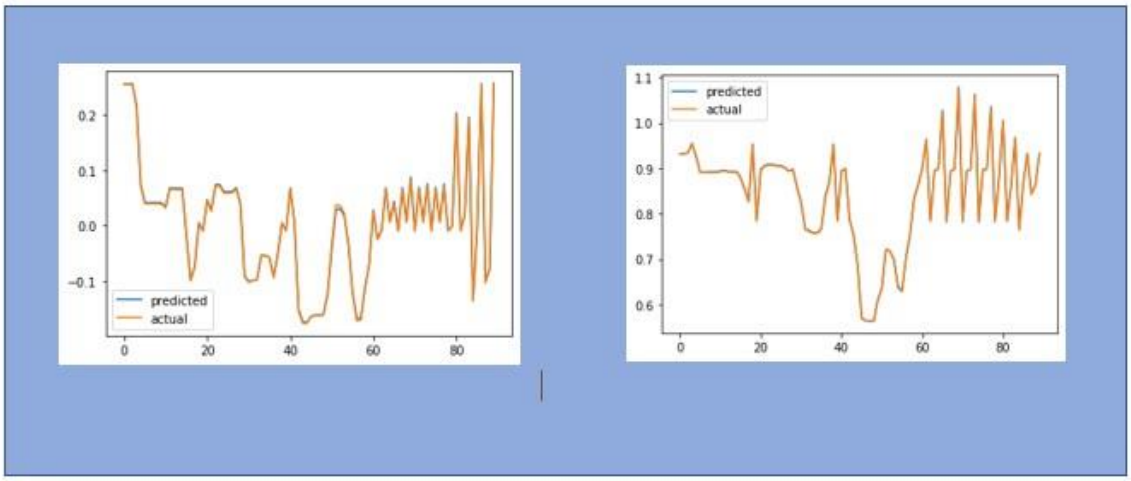


*Figure 40:After hyperparameter tuning longitude(left) latitude(right)*

From the above discussion, it is clear the LSTM has a great potential in Location Prediction Model in a "Multivariate and Multiple Input Series" Setting. It is important to take these coordinates out of the Matplotlib world and see if the predictions replicate the actual points on a real-world map. Hence ,As mentioned in the tools section, following visuals have been made in Power BI for the same.

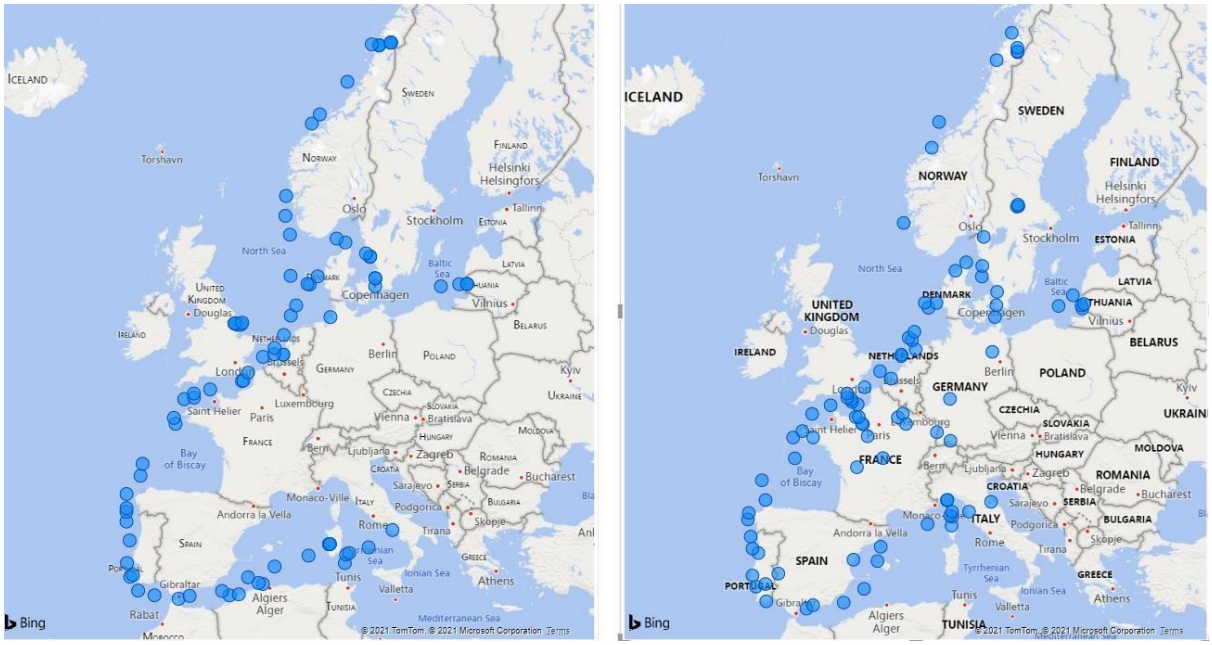


*Figure 41:Actual vs predicted results for terminal A*

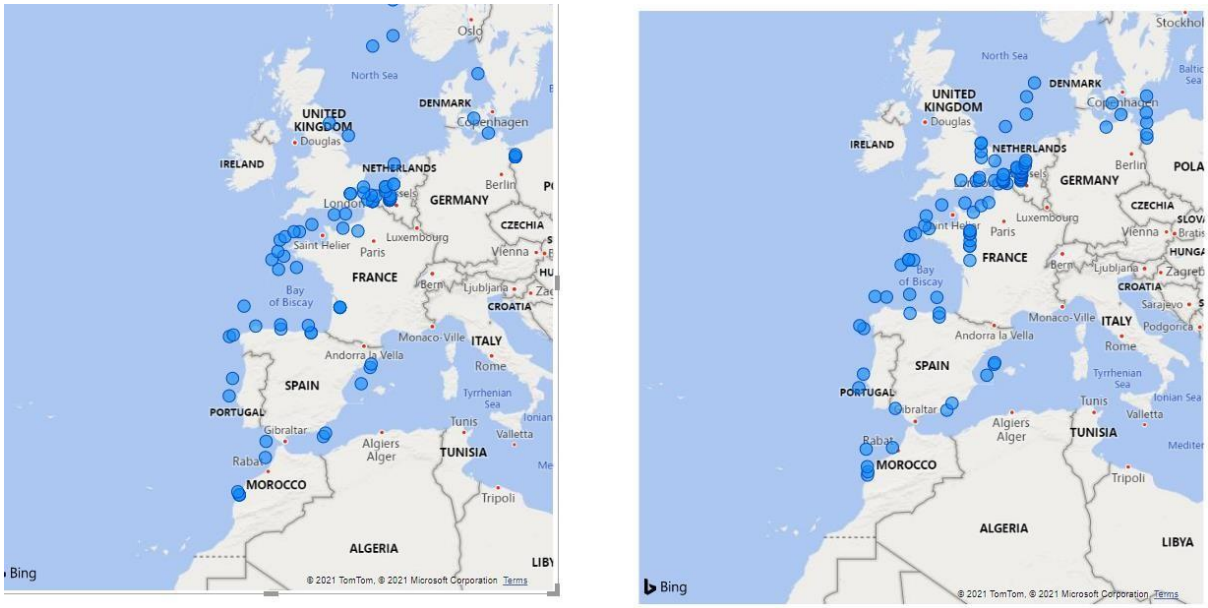


*Figure 42:Actual vs predicted results terminal B*

From figure 41 and 42 it could be seen that LSTMs were able to perform well in predicted future locations .Although there are some coordinates which fall on the land while the location is supposed to be on Sea. This was solved by writing

a code snippet to push the points on the land to the nearest sea point .This will not have major impact on the end result as the coverage of a geo stationary satellite is pretty broad .

## 6.2 Platform Classification Model:

As mentioned in the experiment design setup section ,two different models were designed to perform platform classification : XGBoost and LSTM neural Network. As mentioned in the previous section objective focus here is to predict the probability of potential classes(Platforms), the particular coordinate set will be mapped to .Unlike traditional classification problems where the focus is to predict single class ,the aim of both the algorithms here is to predict probability of each class being mapped to the given set of coordinates.

The evaluation is done based on how many times the actual class falls in the predicted set of classes in other words how many times the actual platform of the said coordinates falls in the n number of platforms predicted by the model. For example, if there are a total of 8 platforms. Based on the highest probabilities predicted by the model for each platform, the top 3(0r any number) are picked. Evaluation is based on how many times the actual platform falls in the top 3 predicted platforms. The following result analysis will explain how each algorithm performed based on classification accuracy on a dataset of Single terminal and on dataset of multiple terminals

**Single Terminal Dataset :**

| | Classification accuracy | | | |
|---|---|---|---|---|
| Number of classes Predicted | 1 | 2 | 3 | 4 |
| **Neural Network** | **72/150=48%** | **127/150=84%** | **144/150=96%** | **147/150=98%** |
| **XG Boost** | **85/150=56.66** | **125/150=83.33%** | **141/150 =94%** | **146/150=97.33%** |

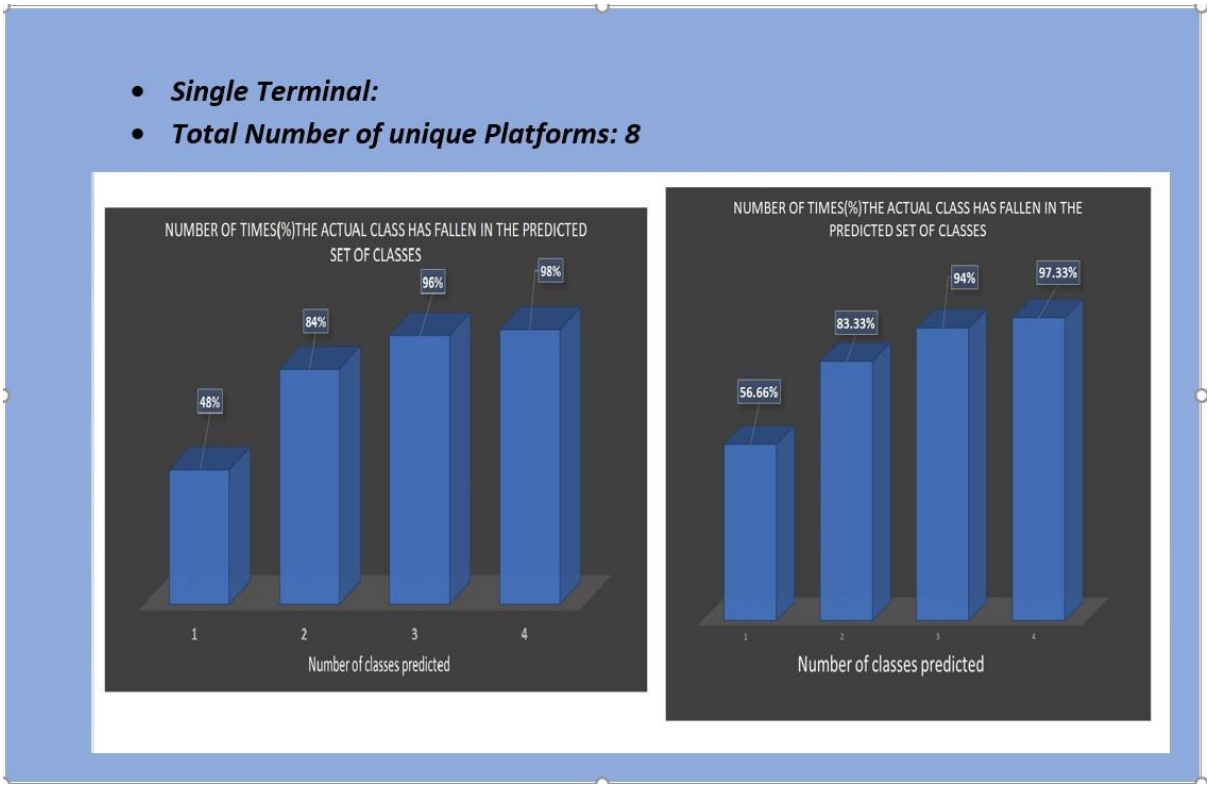


*Figure 43:Single terminal: XGboost(left) vs Neural networks(right)*

## Multiple Terminals Dataset :

| | Classification Accuracy | | | |
|---|---|---|---|---|
| Number of classes Predicted | 1 | 2 | 3 | 4 |
| **Neural Network** | **281/580=48%** | **471/580=81.2%** | **547/580=94.3%** | **567/580=97.75%** |
| **XG Boost** | **372/580=64.13%** | **489/580=84.3** | **551/580=95%** | **569/580=98.1%** |

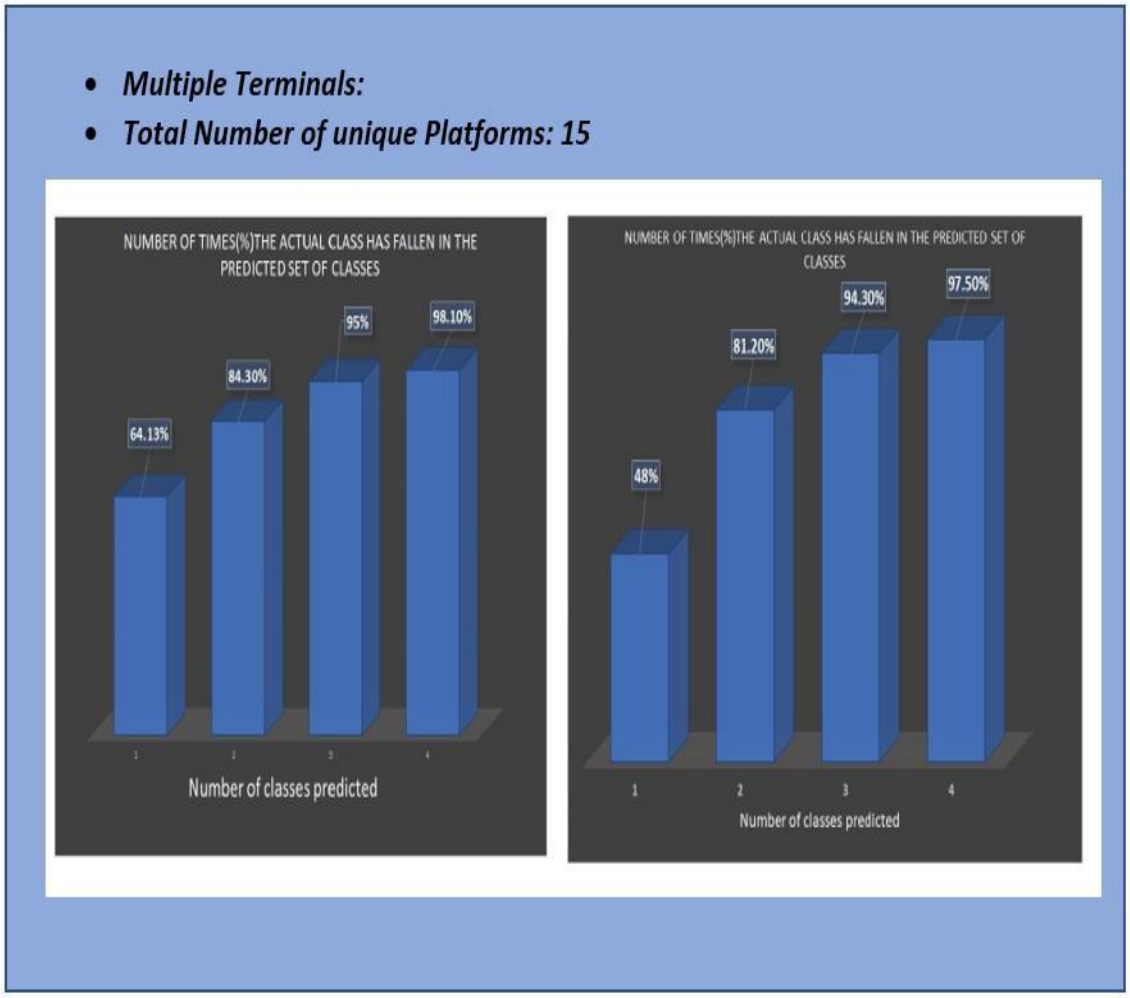


*Figure 44:Multiple terminals : XGboost(left) vs Neural networks(right)*

## *6.3* Capacity Prediction Model :

As mentioned in the "Data Transformation" section, there were two possible ways to transform data for this model, namely:

- Multivariate and Multiple Input Series (Single Step Forecast)
- Multivariate and Multiple Parallel Series (Multi-Step Forecast)

It was observed that "***Multivariate and Multiple Input Series (Single Step Forecast)***" transformation performed significantly better as compared to ***Multivariate and Multiple Parallel Series (Multi-Step Forecast)***

Below given graphs will explain the difference between the prediction results obtained from **Multivariate and Multiple Parallel Series** and **Multivariate and Multiple Input Series** for a particular terminal id respectively.

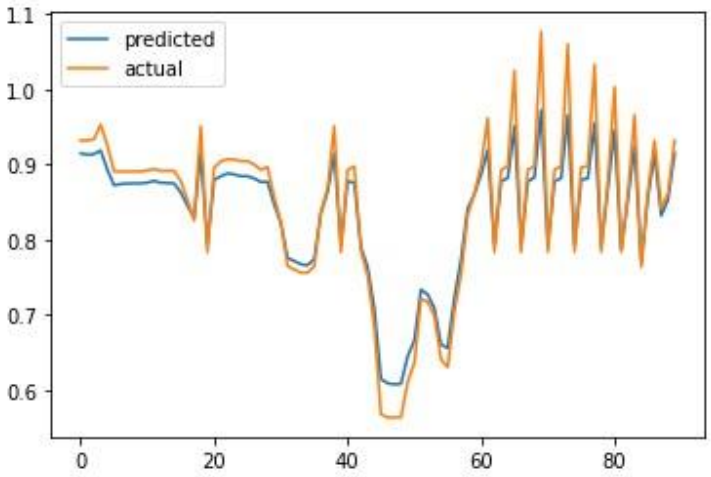


*Figure 45:Prediction results :Return Mbps*

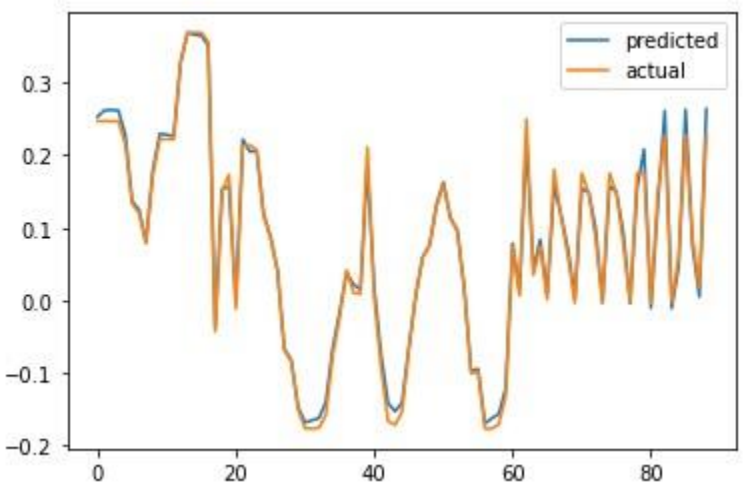


*Figure 46:Prediction results :Forward Mbps*

## 6.4 Chapter Summary :

The following summary will give a quick overview of the best results from all three models. It is then followed by an architectural overview of the solution design.

**Best Results : Platform Classification**

- Algorithm : XGboost
- Number of classes predicted : 4/15
- Accuracy : 98.1%

**Best Results : Location prediction**

- Algorithm : LSTM
- Latitude RMSE : 0.002
- Longitude RMSE : 0.003

**Best Results : Capacity Prediction**

- Algorithm : LSTM
- Fwd RMSE : 0.002
- RTN RMSE : 0.0

From the above result analysis, we could see that LSTM is a powerful and efficient algorithm. In case of platform classification ,when the data from all the terminals were merged ,XGboost performed slightly better than LSTM.In capacity and location prediction model., multivariate and multiple inputs settings seemed to perform way better than multi variate and parallel input.

# 7 Discussion

This section will discuss some observations about the model followed by the future work in this domain and how this solution could be made scalable at a bigger level. It will also discuss the limitations and novelty of the solution and what it has to provide to the science .

## 7.1 About the Datasets:

In the Location prediction model, it was observed that even after extracting daytime features, there was hardly any difference in the model performance. This could be because of the random movement of vessels and the inability of neural networks to map the vessel's movement to a particular day. All the past work that has been done in location or trajectory prediction of a mobile unit using neural networks work with data that observe some kind of pattern in the vehicle movement. This makes it easier for the model to predict future locations. In this study, there was no pattern in vessel movement, which added to the complexity of the problem.

## 7.2 Data Scaling:

It is always advised to scale or normalize the data before feeding it to the neural networks. For the Capacity prediction model, the data were normalized and then provided to the model. But in the Location prediction model, instead of scaling or normalizing the data, the latitude and longitude were converted to radians. This reduced the scale of the data from substantial decimal numerical to smaller numbers, and the model learned and performed well. This proves that with the help of domain knowledge, other ways could be explored, to reduce and align the scale of the features.

## 7.3 General:

Some of the general observations during the study: Hyperparameter tuning could significantly improve the model's performance. Given the time and computational power, A grid search could be a better hyperparameter tuning method as it is the most exhaustive search. The multiple input and multiple output model produced the worst results, and it could stem from the fact that

there were not enough features for the model to learn. If the company could get more information about the vessels and their capacity usage in the future, multiple-input and multiple-output model could be considered.

## 7.4 Limitation and Future Work

***Global Model For all Terminals:*** The current solution is focused on finding a solution based on terminal data. Granularity helps make more accurate results but, at the same time, could be a tedious task when scaled to a more extensive level. The future work would be to implement the same solution so that all three models could be designed universally for all the terminals. This generalisation will help the stakeholders in making decisions at a contractual level as and when needed.

***Multi-Step Forecasting farther into the Future***: The current research focuses on extending the work through direct- recursive forecasting. In the Future, if the time and computational power allow, the same models could be improved upon to make more accurate multi-step forecasting farther into the future.

***Lack of Features***: The Lack of features is one of the downsides of the problem. Although it also contributes to the solution's novelty and represents how most of the real-world issues look, it is essential to understand the significance of valuable features. In the Future, more work could be done on effective feature extraction, which might also help in Multi-Variate and Multi-Step forecasting as mentioned above.

***Type of Vessel***: Future work could determine the type of shipping vessels by their usage and movement pattern. Insights could be drawn from historical data. This could also help in giving more context and information to the existing models.

***Too many Models*** : Current solution focuses on predicting each feature separately by treating all other variables as endogenous variables. Although it adds to the accuracy of the model ,it accounts for too many models for one solution. This could be improved by working on an algorithm which could predict both the variables in one model .This will make the solution less complex .

***Hyperparameter Tuning****:* The hyperparameter tuning is done through a Random search. This search is efficient when the time and resources are limited. There were too many models involved in this solution, and hyperparameter tuning for each model was a tedious task in itself, which is why Random Search was chosen to find the best hyperparameters. This method was tested on limited terminals, but there are chances that this search might not work well with all terminals because of limited search space. Grid search will be a better option for the solution that could be implemented on a larger scale, on a cloud, as it will perform the most exhaustive search for hyperparameters, thus making the solution scalable.

## 7.5 Contribution to Science

1- ***Integration of solutions from three different domains*** : According to the literature survey performed , there was no evidence found of a capacity optimisation problem addressed in a satellite or telecommunication industry by integrating three highly different domains: Location prediction (Geographical data), satellite beam classification, and capacity prediction(Demand forecasting in bandwidth data) using the machine learning algorithms. This research has taken a step in that direction.

2- **Probabilistic Classification***:* In the Literature survey, no evidence was found for any work done for a satellite beam classification using probabilistic classification. This study accounts for all the uncertainties and factors that are responsible for a beam being chosen at a given point by introducing probabilistic classification instead of deterministic classification. This classification provides the top 3-4 platforms based on

the predicted probability of each platform. This method enables the business intelligence team to take the right decision based on domain knowledge and model results.

## 7.6 Contribution to Practice

1- ***Location Prediction of a vessel with just the use of coordinates:*** All the past work done on determining a vessel's future trajectory depends on way more features than geographic coordinates. One of the main contributions of this work is implementing a neural network in a Location prediction problem with just the use of coordinates and no extra features. This adds to the complexity of the problem and hence makes the solution a significant one. It addresses real-world issues faced by businesses – lack of Significant Features in Data.

2- ***Satellite Beam classification from the User perspective***: This research project also contributes to the satellite industry by proposing a machine learning solution for beam classification from a user level. It helps in determining which beam from overlapping beam coverages a user will be using. The user, in this case, is a shipping vessel, but this could be scaled to other domains as well.

# 8 Conclusion

In this research, an in-depth analysis and implementation of chosen machine learning methods have been done for a demand forecasting problem. This research revolved around a few research questions, and answers to those were found through different analytical experiments. Answers to those research questions have been summarised below .

**1- Which Satellite Beam will each vessel use considering the changing locations of the vessels?**

This research question was answered by designing a multi-class classification model. This was implemented using XGBoost and LSTM. Both the algorithms were successful enough in determining the probability of a satellite beam being chosen at a given point of time at the given location

**2- Can this problem be solved using Traditional time series forecasting methods?**

Traditional time series methods are well suited for Uni-Variate data. However, techniques like Vector Autoregression claim to be designed for multivariate forecasting problems. The problems addressed in this research project were Multivariate-Nonlinear problems with a lot of seasonality. These problems needed algorithms that learned better and memorized the information for a longer time. Hence the traditional time series methods fail to fulfil the objective of the project.

**3- Is LSTM an optimal solution to this problem?**

From the experiments performed on all the three problems: Location Prediction, Platform classification, and Capacity Prediction, it could be concluded that LSTMs performed really well and depicts promising results. Although with a larger amount of data in the Platform classification Model, XGboost performed slightly better than LSTM. But overall, LSTMs did a great job in fulfilling the objective of this project.

**4- In this case, Is Feature extraction required for the LSTM algorithm for trajectory prediction and demand forecasting techniques?**

Relevant Features are essential in making a machine learning model better. In this case, there were not many features available in the dataset. Dataset for Location prediction problem was limited to two features: latitude and longitude, and LSTM performed reasonably well. Although extra date time features were extracted, they did not add any value to the model. In Capacity prediction model. feature extraction significantly improved the model. Hence it is safe to say that feature extraction always helps make the model better, but it also depends on the problem. Like, in this case, the Location prediction model did well without any feature extraction, while capacity prediction problem did require a few extra features to be extracted to perform well.

**5-Which part of Industry4.0 is being harnessed by this solution with respect to Satellite Industry.**

This research project is completely data-driven ,that is all the informed decisions made through this project has been based upon the insights drawn from the data. One of the technologies of Industry 4.0 is Big Data analytics. The objective of this research project was to align the principles of Industry 4.0 with a satellite company. This research project has harnessed the “Big Data Analytics” technology of Industry 4.0 and has taken a step forward in making a satellite communications company a data-driven company.

# *References*


[1] G. Chen, L. Hu, Q. Zhang, Z. Ren, X. Gao and J. Cheng, "ST-LSTM: Spatio-Temporal Graph Based Long Short-Term Memory Network For Vehicle Trajectory Prediction," 2020 IEEE International Conference on Image Processing (ICIP), 2020, pp. 608-612, doi: 10.1109/ICIP40778.2020.9191332.

[2] S. Siami-Namini, N. Tavakoli and A. Siami Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 1394-1401, doi: 10.1109/ICMLA.2018.00227.

[3] González Vidal, Aurora & Jiménez, Fernando & Skarmeta, Antonio. (2019). A Methodology for Energy Multivariate Time Series Forecasting in Smart Buildings Based on Feature Selection. Energy and Buildings. 196. 10.1016/j.enbuild.2019.05.021.

[4] Chen, Yi-Ting & Sun, Edward & Lin, Yi-Bing. (2020). Machine learning with parallel neural networks for analyzing and forecasting electricity demand. Computational Economics. 56. 10.1007/s10614-019-09960-5.

[5] Kumar P. Forecasting Cloud Resource Utilization Using Time Series Methods [Internet] [Dissertation]. 2018. (TRITA-EECS-EX)

[6] R. F. Berriel, A. T. Lopes, A. Rodrigues, F. M. Varejão and T. Oliveira-Santos, "Monthly energy consumption forecast: A deep learning approach," 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 4283-4290, doi: 10.1109/IJCNN.2017.7966398.

[7] Carbonneau, Réal & Laframboise, Kevin & Vahidov, Rustam. (2008). Application of machine learning techniques for supply chain demand forecasting. European Journal of Operational Research. 184. 1140-1154. 10.1016/j.ejor.2006.12.004.

[8] Kim, Sol & Jung, Sungwon & Baek, Seung-Man. (2019). A Model for Predicting Energy Usage Pattern Types with Energy Consumption Information According to the Behaviors of Single-Person Households in South Korea. Sustainability. 11. 245. 10.3390/su11010245.

[9] C. Wang, L. Ma, R. Li, T. S. Durrani and H. Zhang, "Exploring Trajectory Prediction Through Machine Learning Methods," in IEEE Access, vol. 7, pp. 101441-101452, 2019, doi: 10.1109/ACCESS.2019.2929430.

[10] Altché, Florent & de La Fortelle, Arnaud. (2017). An LSTM network for highway trajectory prediction. 353-359. 10.1109/ITSC.2017.8317913.

[11] Bontempi, Gianluca & Ben Taieb, Souhaib & Le Borgne, Yann-Aël. (2013). Machine Learning Strategies for Time Series Forecasting. 10.1007/978-3-642-36318-4_3.

[12] Fildes, Robert & Kumar, V.. (2002). Telecommunications demand forecasting - A review. International Journal of Forecasting. 18. 489-522. 10.1016/S0169-2070(02)00064-X.

[13] Wu, Fan & Fu, Kun & Wang, Yang & Xiao, Zhibin & Fu, Xingyu. (2017). A Spatial-Temporal-Semantic Neural Network Algorithm for Location Prediction on Moving Objects. Algorithms. 10. 37. 10.3390/a10020037.

[14] Cheng, Yao & Xu, Chang & Mashima, Daisuke & Thing, Vrizlynn & Wu, Yongdong. (2017). PowerLSTM: Power Demand Forecasting Using Long Short-Term Memory Neural Network. 10.1007/978-3-319-69179-4_51.

[15] Che, Zhengping & Purushotham, Sanjay & Cho, Kyunghyun & Sontag, David & Liu, Yan. (2018). Recurrent Neural Networks for Multivariate Time Series with Missing Values. Scientific Reports. 8. 10.1038/s41598-018-24271-9.

[16] Saurabh Rathor(June 2,2018):Simple RNN vs GRU vs LSTM :- Difference lies in More Flexible control: https://medium.com/@saurabh.rathor092/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control-5f33e07b1e57

[17] Kaushik Mani(Feb 17,2019) GRU's and LSTM's: https://towardsdatascience.com/grus-and-lstm-s-741709a9b9b1

[18] Shay Palachy(April 8,2019):Stationarity in time series analysis: https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322

[19] Manikanta Varaganti(July 22,2020): Introduction to Time Series Forecasting: https://medium.com/@varagantimanikanta/introduction-to-time-series-forecasting-34890ac66a4b

[20] Christophe Pere (September 21,2020):How to use Deep Learning for Time Series Forecasting: https://towardsdatascience.com/how-to-use-deep-learning-for-time-series-forecasting-3f8a399cf205

[21] Kaggle: Welcome to Deep Learning!: https://www.kaggle.com/ryanholbrook/a-single-neuron

[22] Cornellius Yudha Wijaya(April 26,2020): CRISP-DM Methodology For Your First Data Science Project: https://towardsdatascience.com/crisp-dm-methodology-for-your-first-data-science-project-769f35e0346c

[23] Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J. and Linkman, S., 2009. Systematic literature reviews in software engineering–a systematic literature review. Information and software technology, 51(1), pp.7-15.

## VII. APPENDIX

### Literature Review

| Ref | Contribution | Methods and techniques | Domain/Type of data | Source of the data |
|---|---|---|---|---|
| Rodrigo Ferreira Berriel.et.al | Propose a system to predict monthly energy consumption using deep learning techniques | Three deep learning models were studied: Deep Fully Connected, Convolutional and Long Short-Term Memory Neural Networks. Due to the sensitivity of these models to the input range, normalization techniques were also investigated. The proposed system was validated with real data of almost a million customers (resulting in over 9 million samples) | The energy consumption data come from a Brazilian power company and is collected every month by the company employees. In total, there were 21,978,437 measurements from 934,945 different customers. Most of the measurements (21,978,255) were obtained in a 2-year interval between March 2014 and February 2016. | Collected from Brazilian power company, person goes to every customer and reads the month energy consumption from the electrical meter. |
| Guangxi Chen.et.al | Trajectory Prediction of Autonomous Vehicles | A vehicle trajectory prediction model based on LSTM encoder-decoder is proposed. The encoder is mainly a layered LSTM module based on ST graph, which captures information of spatial, temporal and trajectory history data respectively, concatenated into a whole context vector, and finally made trajectory prediction through the decoder. Experimental results show that the proposed model achieves better prediction results compared with other existing models. | Collected at 10Hz over a 45-minute period and divided into15 minutes of light, moderate and heavy traffic. The data set provides the coordinates of the vehicles in a local coordinate system. A quarter of the trajectories from each of the three subsets of the data sets were tested. Each trajectory is divided into 8s segments, consisting of a 3s trajectory history and a 5s prediction range. | US-101and I-80 in NGSIM dataset were selected for training and testing |
| Gianluca Bontempi et.al | Proposed a modification of the local learning technique to take into account the temporal behavior of the multi-step forecasting problem and consequently improve the results of the recursive strategies | Presents an overview of machine learning techniques in time series forecasting by focusing on three aspects: the formalization of one-step forecasting problems as supervised learning tasks, the discussion of local learning techniques as an effective tool for dealing with temporal data and the role of the forecasting strategy when we move from one-step to multiple-step forecasting The authors proposed a modification of the local learning technique to take into account the temporal behavior of the multi-step forecasting problem and consequently improve the results of the recursive strategies" | Representative historical data | Not mentioned. |
| Sima Siami-Namini et al | Research question investigated in this article is that whether and how the newly developed deep learning based algorithms for forecasting time series data, such as Long Short-Term Memory (LSTM) , are superior to the traditional algorithms | ¨ This paper compares ARIMA and LSTM models with respect to their performance in reducing error rates. As a representative of traditional forecast modeling, ARIMA is chosen due to the non-stationary property of the data collected and modeled. In an analogous way and as a representative of deep learning-based algorithms. LSTM method is used due to its use in preserving and training the features of given data for a longer period of time. The paper provides an in-depth guidance on data processing and training of LSTM models for a set of economic and financial time series data. Investigate the influence of the number of training times. The study shows that the number of training performed on data, known as epoch in deep learning, has no effect on the performance of the trained forecast model and it exhibits a truly random behavior. Advocates the benefits of applying deep learning-based algorithms and techniques to the economics and financial data.¨ | Historical monthly financial time series from Jan 1985 to Aug 2018 from the Yahoo finance Website1. The monthly data included Nikkei 225 index (N225), NASDAQ composite index (IXIC), Hang Seng Index (HIS), S and P 500 commodity price index (GSPC), and Dow Jones industrial average index (DJ) Monthly economics time series for different time periods from the Federal Reserve Bank of St. Louis2, and the International Monetary Fund (IMF) Website¨ | Yahoo finance Website Federal Reserve Bank of St. Louis<br>and the International Monetary Fund (IMF) Website" |
| Sanjay Purushotham1 et.al | Develop a novel deep learning model based on GRU, namely GRU-D, to effectively exploit two representations of informative missingness patterns, i.e., masking and time interval. | The model captures the observations and their dependencies by applying masking and time interval (using a decay term) to the inputs and network states of GRU, and jointly train all model components using back-propagation. the model not only captures the long-term temporal dependencies of time series observations but also utilizes the missing patterns to improve the prediction results The proposed method is suitable for many time series classification problems with missing data, and in particular is readily applicable to the predictive tasks in emerging health care applications. Proposed GRU-D model with trainable decays has similar running time and space complexity to original RNN models, and are shown to provide promising performance and pull significantly ahead of non-deep learning methods on synthetic and real-world healthcare datasets. " | One synthetic and two real-world health-care datasets | Gesture phase segmentation dataset (Gesture): UCI dataset4PhysioNet Challenge 2012 dataset (PhysioNet). |
| Chujie Wang.et.al | Multi-user multi-step trajectory prediction of mobile users. | 1.Comprehensive investigation for the approaches to predict trajectories composed of continuous coordinates 2. This paper makes a detailed exploration of the trajectory prediction problem from both the single-user perspective and multiuser perspective. 3. An LSTM-based single-user prediction framework and evaluate its performance on a model based dataset. 4. The user-specific prediction scheme is further extended to a region-oriented prediction scheme and a multi-user multi-step trajectory prediction framework based on the Seq2Seq learning is put forward" | MODEL-BASED DATASET :Authors refer to the Self-Similar Least-Action Human Walk (SLAW) [20] and the SMOOTH model [29] to generate our mobility data. REALISTIC DATASET :. The dataset was collected by 182 users, containing 18,670 trajectories with various sampling rates. Each trajectory is represented by a series of timestamp points with latitude and longitude coordinates recorded by GPS-functioned phone" | Geolife project of Microsoft Research Asia |
| Le QI.et.al | This study proposes an intelligent model to solve the issue of the trajectory prediction of vessels based on data mining and machine learning methods. | Model based on data mining and machine learning methods | Trajectory data of vessels, other parameter of vessels (Not clearly defined ) | Not mentioned |
| Florent Altché.et.al | Prediction of Longitudinal and latitudinal trajectories for vehicles on highway | Usage of long -short-term memory(LSTM) neural Networks and Train a predictor for the future trajectory of a single "target" vehicle .As part of Data Preparation, usage of Savitzky-Golay filter which performs well for signal differentiation with window length 11 to smooth the longitudinal and lateral positions and compute the corresponding velocities . The choice of features was made to replicate the information a human driver is likely to base its decisions upon: the features from surrounding vehicles are all relative to the target vehicle, as the drivers were expected to usually make decisions based on perceived distances and relative speeds rather than their values in an absolute frame | "• Vehicle trajectories in the form of (X,Y) coordinates of the front center of the vehicle in a global frame, and of local (x,y) coordinates of the same point on a road-aligned frame. • The local coordinates (dataset columns 5 and 6), where x is the lateral position of the vehicle relative to the leftmost edge of the road, and y its longitudinal position. • Moreover, the dataset contains each vehicle's lane identifier at every time step, as well as information on vehicle dimensions and type (motorcycle, car or truck). • Finally, the data also contains the identifier of the preceding vehicle for every element in the set (when applicable)" | NGSIM US-101 dataset |
| Hossein Abbasimehra et.al | Demand Forecasting especially for fluctuating data. | A Forecasting method is proposed, which has a strong capability of predicting highly fluctuating demand data. A demand forecasting method based on multi-layer LSTM networks is proposed. The proposed method automatically selects the best forecasting model by considering different combinations of LSTM hyperparameters for a given time series using the grid search method. It has the ability to capture nonlinear patterns in time series data, while considering the inherent characteristics of non-stationary time series data. | To perform statistical tests, four extra time series were create from the original time series .The four time series datasets are created as follows: for obtaining D1 the last h points of the original timeseries are selected as test set and the remaining part is considered as training set. | Demand data of a furniture company |

| | | | | |
|---|---|---|---|---|
| Prashant Kumar et.al | Review the perfromance of forecasting methods ,experiment with ensemble methods and assses the accuracy of the forecast model on the given dataset using different techniques.The Modle is used to forecat the resource utilisation. | Propose a combined forecasting model .Usage of ARIMA,ENSEMBLE and ANN methods.A comparison has also been provided | CPU Utilisation metrics spaced at 5 min time interval | Amazon Cloudwatch ec2 data |
| Yi-Ting Chen.et.al | Dynamic monitoring and forecasting of power demand | Quantitative method of parallel neural networks (PNNs) for robust monitoring and forecasting power demand.Authors generalize the implementation by using simulated data and real data from Australia.A conclsuon is drawn that a PNN is a robust algorithm that enriches the class of big data mining methods applied for forecasting power demand | Power Demand Series | Generate through simulation |
| Fan Wu.et.al | Location prediction of Moving objects using LSTM based model | "Each of the points in a trajectory is first preprocessed in order to convert the real continuous values associated to the geospatial coordinates of latitude and longitude, into discrete codes associated to specific regions.A LSTM-based prediction model is proposed to handle long sequences.In a nutshell the the algorithm consists of two steps, the spatial-temporal-semantic (STS) feature extraction and the LSTM-based model building." | "Two real world trajectory dataset .ethod. The courier dataset is a self-collected data from a courier of a food delivery platform. The MTA Bus dataset.The dataste contains trajectories of 5725 buses from 315 bus routes acorss the newyork region " | The MTA Bus dataset is provided by the Metropolitan Transportation Authority (MTA) |
| Sol Kim.et.al | Derives types of "energy usage patterns" (EUPs) for single-person households in South Korea | " By identifying which application was used according to the occupants' behavior, the behaviors were classified into the behavior codes of: Leisure (L1), Laundry (L2), Cooking (C1), Grooming (G1), and Dish,Six SVM models were trained through the Classification Learner App of MATLABR2018b,and the most appropriate SVM model was selected EUP(Energy usage pattern) types were derived through K-modes clustering, and the household characteristics and energy consumption of each type were analyzed. As a result of K-modes clustering, seven EUP types and then an EUP-type prediction model with a prediction rate of 95.0% was implemented by training an SVM, and an energy consumption information model provided through GPR." | Behavioral data for actual occupants of single person households and energy usage pattern | Korean time use survey |
| Real Carbonneau.et.al | Forecasting distorted demand at the end of a supply chain (bullwhip effect). | "The objectives of this research are to study the feasibility and perform a comparative analysis of forecasting the distorted demand signals in the extended supply chain using non-linear machine learning techniques. More specifically, the work focuses on forecasting the demand at the upstream end of the supply chain. The forecasting techniques used in our analysis include: Naïve Forecast • Average • Moving Average • Trend • Multiple Linear Regression and Neural Networks • Recurrent Neural Networks • Support Vector Machines" | Two data sets for experiments: one obtained from the simulated supply chain, and another one from actual Canadian Foundries orders. | From actual Canadian Foundries orders. |