OUTLIER DETECTION OF RFID DATASETS IN SUPPLY-CHAIN PROCESS

THESIS

Submitted in fulfilment of the requirement of the degree of

DOCTOR OF PHILOSOPHY

To

J C BOSE UNIVERSITY OF SCIENCE & TECHNOLOGY YMCA

by

MEGHNA SHARMA

Registration No: YMCAUST/Ph08/2010

Under the Supervision of

DR. MANJEET SINGH

PROFESSOR

Department of Computer Engineering

Faculty of Engineering and Technology

J C Bose University of Science & Technology YMCA

Sector-6, Mathura Road, Faridabad, Haryana, INDIA

July, 2019

DECLARATION

I hereby declare that this thesis entitled "**OUTLIER DETECTION OF RFID DATASETS IN SUPPLY-CHAIN PROCESS"** by **MEGHNA SHARMA,** being submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy in Department of Computer Engineering under Faculty of Engineering and Technology, J C Bose University of Science & Technology YMCA, Faridabad, during the academic year 2018-2019, is a bona fide record of my original work carried out under the guidance and supervision of **Dr. MANJEET SINGH, PROFESSOR, DEPARTMENT OF COMPUTER ENGINEERING, J C BOSE UNIVERSITY OF SCIENCE & TECHNOLOGY YMCA, FARIDABAD** and has not been presented elsewhere.

I further declare that the thesis does not contain any part of any work which has been submitted for the award of any degree either in this university or in any other university.

(MEGHNA SHARMA)

Registration No: YMCAUST/Ph08/2010

CERTIFICATE

This is to certify that this thesis entitled **“OUTLIER DETECTION OF RFID DATASETS IN SUPPLY-CHAIN PROCESS”** by **MEGHNA SHARMA** submitted in fulfillment of the requirements for the award of Degree of Doctor of Philosophy in Department of Computer Engineering, under Faculty of Engineering and Technology, J C Bose University of Science & Technology YMCA, Faridabad, during the academic year 2018-19, is a bona fide record of work carried out under my guidance and supervision.

I further declare that to the best of our knowledge; the thesis does not contain any part of any work which has been submitted for the award of any degree either in this university or in any other university.

Dr. MANJEET SINGH
Professor
Department of Computer Engineering
Faculty of Engineering and Technology
J C Bose University of Science & Technology YMCA, Faridabad

Dated: ______________________________________

The Ph.D viva-voce examination of Research Scolar Meghna Sharma (YMCAUST/Ph08/2010) has been held on 01/07/2019.

(Signature of Supervisor) (Signature of Chairman)

ACKNOWLEDGEMENT

I would like to express my sincere and deep gratitude to my guide **Dr. MANJEET SINGH**, Professor, Department of Computer Engineering, J C Bose University of Science & Technology YMCA, Faridabad for giving me the opportunity to work in this area. It would never be possible for me to take this thesis to this level without his innovative ideas, invaluable guidance, continuous support and encouragement. His knowledge of different perspectives of research provided me with the opportunity to broaden my knowledge and to make significant progress. I would like to express my gratitude towards **Dr. KOMAL BHATIA**, Chairman, Department of Computer Science Engineering, and all the faculties of Department of Computer Engineering, J C Bose University of Science & Technology YMCA, Faridabad, for their full support and encouragement throughout my research work.

I would like to specially thank my brother **Dr. ATHARVA SHARMA** for his constant guidance and support in my research work. My heartfelt thanks to my better half **Mr. VIKRANT SHARMA** for his technological assistance and unparalleled availability at all times during the course of my work and constant support and encouragement. My gratitude goes to my father **Dr. OM DATT SHARMA** and my mother **Mrs SAROJ SHARMA** for being my inspiration and motivation and my special thanks to my mother-in-law **Mrs SUSHMA SHARMA** for her constant support and encouragement.

My heartfelt thanks to my lovely sons, **VIHAAN SHARMA** and **AARAV SHARMA** for understanding me and giving me time for doing my research work. I would also like to thank my great father-in-law **Late Col. SHAM VED SHARMA** whose blessings have always been there for me**.** Special thanks to my friends **Dr. JAGDEEP KAUR** and **Ms. AMITA ARORA** for all their invaluable help directly or indirectly. I would also like to express my thanks to all my friends and my family for being always there in my tough times. I gratefully acknowledge my university colleagues and my fellow research scholars for their encouragement, support and invaluable suggestions in completing this research.

I express my gratitude to almighty God for giving me strength and courage to complete this thesis.

ABSTRACT

In this world of wireless communication, technologies like Wi-Fi, Global Positioning Systems (GPS), Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSN) play a very important role in lot of applications for the benefit and convenience of our society. The biggest advantage of using wireless communication is that it doesn't need any wired connection resulting in flexibility and mobility of the user.

Radio Frequency Identification (RFID) technology is a type of wireless communication-based technology in which automated receiving and sending of the information between tags (also called as transponders) and RFID readers is done. RFID chips/RFID tags are the small pieces of hardware consisting of an antenna for the transmission and reception of the radio signals. RFID are of use only when they are read by an interrogator known as RFID reader. It is the communication between reader and tag via radio waves which helps in the transferring of tag information stored in its memory to the reader as an when it is within the range of the reader. The tag should be in the range of reader to exchange the information without any manual intervention. There are attached antennas with tag and reader to transfer and receive signals or information. Whenever a reader is within required range and it sends appropriate signals to an object, the associated RFID chip in the tag, responds by sending whatever data it contains. The reader is connected to a centralised computer system which receives and stores the response data from the tagged object and use it for further processing.

Object or people-based tracking systems that use Radio Frequency Identification (RFID) have seen increasing usage over the past decade. These systems provide an effective tracking solution by leveraging the non–line-of–sight precise identification capability of RFID technology, however they still have to overcome a number of challenges posed by the nature of technology to improve their reliability and accuracy such as uncertain data that leads to location uncertainty.

Supply-chain process is the popular application in RFID-systems networks. RFID-enabled supply-chain process consists of all processes included in the flow of tagged objects from suppliers to consumer; within the chain of suppliers, manufacturers, distributors, retailers and consumers. It is quite a knowledge intensive process with big

complexity. Management and coordination of supply chain processes with RFID technology can significantly reduce this complexity.

An outlier is an anomaly or deviation from anything which is normal. Outlier detection in RFID-enabled supply chain is process of finding the abnormal or anomalous node in the supply chain network. An outlier may be generated because of unusual issues such as transport delays, thefts, etc. in the RFID supply chain.

The thesis work aims to create a full-fledged system for finding and predicting outlier points in the RFID enabled supply chain network. The work is divided into three sub-processes or modules consisting of RFID data collection and cleaning, clustering of supply chain trajectories for tracking outlier trajectories and particular outlier nodes and predicting the outlier points for future in the trajectory or path in supply chain if outlier trajectories or points are given.

There are certain limitations in the existing work or research done related to different sub-processes or modules in the thesis work. Most of the existing work in data cleaning uses fixed window sliding protocol due to which there is a trivial problem of fixing up the window size for cleaning the false positives, false negatives and duplication in reading the tag data by RFID readers. Setting very small window size can result in generation of false negatives and setting up a very large window size can result in generation of false positives. Some techniques like Kalman filter and bloom filter need too much of memory and speed of processing is low. Adaptive window sliding detection is the best as it is dynamic window adjustment according to the stream of RFID tag data. Window sub-range transition detection method is thoroughly studied for all its advantages over other RFID data-cleaning methods and it is also implemented and tested for performance analysis with supply chain RFID datasets. It has been used for the first time in supply chain domain as per our knowledge.

Most of works done for anomaly /outlier detection is based on clustering techniques. The main reason for it is non-availability of training data. Among the normal data, the anomalous node point data is very less. The distance measures generally used are Euclidian, Fréchet, edit distance, longest common sub-sequences, especially for trajectories but they have the drawbacks of either not taking into account the temporal factor into account with lag or lead of the objects due to speed variations or doesn’t work effectively for longer trajectories. Dynamic time warping is another similarity/distance measure which works efficiently in case of trajectories of different lengths as well as time lags due to speed variations in the objects following the trajectories. To cluster the trajectories along with planned trajectories many clustering approaches like k means, mean shift, hierarchical clustering, or cell-based partitioning is used but they either are not memory efficient, do not work well with longer trajectories or cannot cope up with the dynamic and uncertain RFID data stream. Furthermore, for the sub-process of predicting outliers using the previous history of trajectory data, many techniques like rules-based, matrix-based, hidden Markov model-based, decision tree-based, neural networks-based are used but they don't work efficiently in case of long length trajectories.

The literature study gives us the insight for the development of a complete framework to find out the outliers in the RFID-enabled supply-chain path. As per our understanding and study, till now there is no availability of such system in the mentioned domain. A system which can find out outliers with good accuracy with the consideration of long path sequences of the trajectories followed by the objects with varying speed and acceleration is required. The research work done aims for the same and overcome the problems in the existing systems. Scope of research is quite vast as the framework designed can be used to find out outliers or anomalies in various applications which are RFID enabled.

Amidst the uncertainty of RFID dataset in supply-chain process, it is very important to take care of the abnormal readings and report it to the stakeholders. In our work, we aim for creating a full-fledged integrated framework for the complete processing of RFID supply-chain data, starting from data cleaning till the prediction of outlier points in the path trajectory of RFID-enabled supply chain. Outlier detection in supply-chain process can be defined in terms of localization for determining deviation from the location, tracking for determining deviation from the complete path or trajectory to be followed by the object, logistics for any deviation from the normal flow of products and transportation for any failure in the transportation. If any inappropriate quantity of product as expected is found during tracking as well as tracing the shipments in supply-chain process, then these events can also generate alarm for being outliers or abnormalities. Tracing shipments could find inappropriate quantity and quality of the product and notify all trading partners in time. The thesis work is related to the localization process.

Our research work includes the cleaning of continuous stream of data being received from middleware attached to the RFID readers, then finding abnormal or outlier points in the supply-chain path and further using that information of outlier points in the supply-chain paths / trajectories as training data for the accurate prediction of outlier points using recurrent-based neural networks. Also, the comparative analysis of these proposed algorithms with the existing approaches is done. The complete sequence of processes can help in the domain of RFID-enabled supply-chain process with correct analysis of possible outliers in the complete chain of processes, making it more cost effective with better efficiency and manageability. Our research work may give better results in case of the scenarios where time lag factor is considered while checking the path similarities Also, in case of longer path sequences, the technique known as Long Short-Term Memory is going to give more accurate results unlike the other traditional techniques like sequential pattern mining or Hidden Markov Models generally used for path prediction.

The complete model follows a layered approach with pre-processing layer implemented using existing adaptive window-based cleaning technique for data cleaning which reduces number of false positives and false negatives followed by its output taken as input to the next layer for finding outlier path by using the clustering techniques on the supply chain path or trajectory with the scheduled or planned trajectories. The output of the outlier detection layer is the set of outlier nodes found from this layer and this is taken as input to the next layer of outlier node prediction using Recurrent Neural Network-based approach.

Our approach uses a density-based clustering technique specially designed for trajectory or path data generated by the tagged objects in a supply chain. With the modified approach the results showing the outlier paths among the normal ones are tremendously better. Further, the output of the novel clustering approach based on DBSCAN with Dynamic Time Warping can further help us predict the path followed by the tagged objects using Long Short-Term Memory (LSTM) approach for being an outlier node or not in the supply-chain path with much better accuracy even for longer sequences.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GPS:	Global Positioning Systems
RFID:	Radio Frequency Identification
WSN:	Wireless Sensor Networks
EPC:	Electronic Product Code
STSG:	Spatio-Temporal Sensor Graphs
WSTD:	Window Sub-range Transition Detection
DTW:	Dynamic Time Warping
LCSS:	Long Common Sub Sequence
EDR:	Edit Distance on Real Data
SMART:	Simple Monitoring enterprise Activities by RFID Tags
DBSCAN:	Density Based Spatial Clustering of Applications with noise
HMM:	Hidden Markov Model
RNN:	Recurrent Neural Networks
LSTM:	Long Short-Term Memory
IoT:	Internet of Things
SMURF:	Statistical sMoothing for Unreliable RFid data.
TRAJODBSCAN:	Trajectory Outliers with DBSCAN

CHAPTER 1: INTRODUCTION

In this world of wireless communication, technologies like Wi-Fi, Global Positioning Systems (GPS), Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSN) play a very important role in lot of applications for the benefit and convenience of our society. The most commonly used technology in wireless world is radio waves where distance range can vary from few meters as in case of Bluetooth to few kilometres as in deep sea communications. The thesis work is mainly focused on RFID networks which is based on automatic data capture technology and its one of very important application is supply chain management. This technology is actually a fourth-generation industrial revolution [1]. In RFID enabled supply chain process there are lot of possibilities of outliers or anomaly generation due to technical or environmental factors and the research work is done to design and implement the complete framework for the outlier detection in RFID datasets in supply chain process.

1.1 INTRODUCTION TO WIRELESS COMMUNICATION

Wireless communication is a communication through electromagnetic signals which are broadcast with the help of any wireless-enabled device within the air, physical environment or atmosphere. There is one sender and one receiver to propagate and receive wireless signals respectively. A wireless communication bridge is created between the sender and the receiver device. So, the biggest advantage of using wireless communication is that it doesn’t need any wired connection resulting in flexibility and mobility of the user.

1.2 TYPES OF WIRELESS COMMUNICATION TECHNOLOGIES

Wireless communication technology can be categorized into various types based on the range of communication; kind of devices used as well as the data range. The following are the different types of wireless communication technologies.

- **Radio and Television Broadcasting:** In this type of wireless data transmission, data is transferred with low frequency electromagnetic waves as medium of transfer via electric conductor and antenna.

- **Radar Communication**: Radar is used to detect objects by using radio waves for measuring the range, angle, or velocity of objects.
- **Satellite communication**: The communication is done via radio signals with the satellites orbiting around the Earth.
- **Cellular Communication**: A type of communication network via cellular network or mobile network where the last link is wireless.
- **Global Positioning System**: It is a system where dedicated receivers and satellites are used to navigate position and read location and speed of the devices on the surface of Earth.
- **Wi-Fi**: It is a type of wireless communication with access points to manage communication. It uses very low power with its usage very commonly seen in lots of modern electronic devices like laptops, systems, smart phones, etc.
- **Bluetooth**: Bluetooth technology helps transferring and sharing data by connecting to a variety of different electronic devices of low power.
- **Radio Frequency Identification**: Radio-frequency identification (RFID) uses electromagnetic fields for automatic identification of tagged objects by readers. The range of detection depends upon the type of tag and antenna used.

All of these communication technologies have different types of applications and architecture but they all have commonality of not having wired connection between their respective devices to initiate and execute communication. Our research work is in the domain of Radio Frequency Identification (RFID) networks. Though RFID usage started in 1940s and it gained momentum in 1970s, but the cost of tags and readers was very high at that time which restricted its usage commercially. Now with the decrease in the hardware cost, its usage is also increased in various applications. Its use is similar to bar code system but with several advantages such as: long service life, big reading distance, encrypted tag data, large storage capacity and information easy to be changed etc. and above all automated reading without any manual intervention.

RFID is most commonly used for tracking and monitoring in an automated manner. The most popular applications are pet and livestock tracking [2], management of inventory [3] and tracking of assets and people [4], managing the logistics in supply chain process [5], and tracking of vehicles [6] and even in evaluation of footprints of cattle [7]. There are so many varieties of industries that are turning or shifting to RFID enabled systems e.g. healthcare, manufacturing, retail type of industries and also can be used in business and home [8]. Lot of other areas are being explored [101].

There is more and more demand on RFID data reliability. According to the survey of the "information overload" by internationally-famous consulting company [9], results show that more than 90% of the enterprises believe that their information capacity is much linked to their competitiveness. So, they must analyse the data to obtain the useful information. Even the “exception” data is useful in many regards and because of this abnormal or anomalous data should be detected to be further used for analysis. It can also give very meaningful abstractions. One of a very important application is the supply-chain management process based on the RFID system technology [10]. The analysis of the abnormal data is important for the safety and effectiveness of supply chain [11].

Our research work mainly studies the outlier/anomaly-detection scheme of RFID-enabled supply-chain process based on Electronic Product Code (EPC) network in order to provide supply-chain data analysis and assistance for enterprise management decision.

1.3 RFID SYSTEMS

Radio frequency identification [12] technology is a type of wireless communication-based technology in which automated receiving and sending of the information between tags (also called as transponders) and RFID readers is done. RFID chips/RFID tags are the small pieces of hardware consisting of an antenna for the transmission and reception of radio signals.

Each manufactured tag is recognised by a unique serial number known as electronic product code. Whenever a tag is read by a reader it is recognised by this unique serial number only. It is generally written on the tag. Its usual size is 96 bits, although other sizes are also available depending upon the application. Out of these 96 bits, first 8 bits represent the standard protocol for communication, next 28 bits tell about the name of the company organisation which is involved in the management of data generated by tag. After first 36 bits, next 24 bits give information about the kind of product, the tag is attached with. Last 36 bits represents a serial number which actually give the unique identification to the tag a serial number. Tags come in variety of sizes and due to their variety of sizes, they can be easily inserted or attached to the different types of target objects depending on the application area. An RFID tag can be either passive or active type [13]. Most commonly used tags work in the passive mode. In a passive mode, tags don't have any battery power unlike as in active tags. Instead they do absorb the signals transmitted from the reader to set up a connection between the two and transforming of information. These radio signals are read and turned into energy sufficient for sending back the response to the reader. Passive tags have lower range but longer life and less expensive and active tags have higher range but smaller life and more expensive.

RFID are of use only when they are read by an interrogator known as RFID reader. It is the communication between reader and tag via radio waves which helps in the transferring of tag information stored in its memory to the reader as an when required by the reader. The tag should be in the range of reader to exchange the information without any manual intervention. There are attached antennas with tag and reader to transfer and receive signals or information. Whenever a reader is within required range and it sends appropriate signals to an object, the associated RFID chip in the tag, responds by sending whatever data it contains. The reader is connected to a centralised computer system which receives and stores the response data from the tagged object and use it for further processing. RFID systems operate in any of four radio frequency ranges like 125 to 134.2 kHz, 13.56 MHz, 856 MHz to 960 MHz and 2.45 GHz depending upon the application area or domain.

In the applications like supply chain, passive tags with ultra-high frequency range (856-960 MHz) are used. Figure 1.1 shows the interaction between reader, tag and middleware system. First RFID triggers a signal then antenna sends a signal to which tag responds. Reader recognizes the tag and notifies the middleware PC. After that RFID middleware applies local business logic and passes the filtered clean information to the asset-management application.

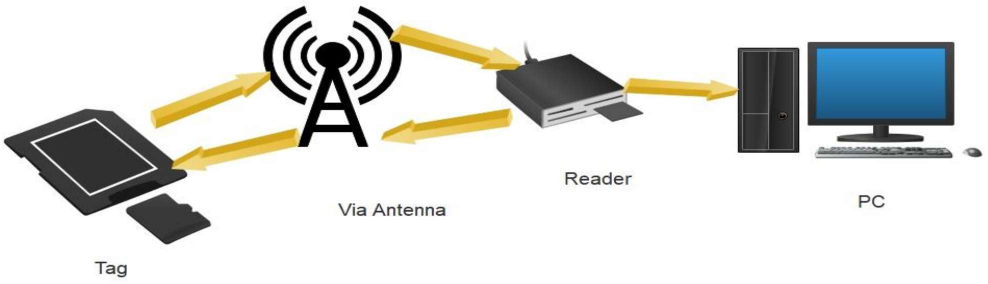


Figure 1.1 A Basic RFID System

1.4 RFID IN SUPPLY-CHAIN PROCESS

Supply-chain process is the popular application in RFID-systems networks. RFID-enabled supply-chain process [5] consists of all processes included in the flow of tagged objects from suppliers to consumer; within the chain of suppliers, manufacturers, distributors, retailers and consumers. It is quite a knowledge intensive process with big complexity. Management and coordination of supply chain processes with RFID technology can significantly reduce this complexity. RFID makes the supply-chain process very efficient, cost effective, and reliable and real time updated for better planning and monitoring by the administration. Figure 1.2 shows the visualisation of supply-chain path.

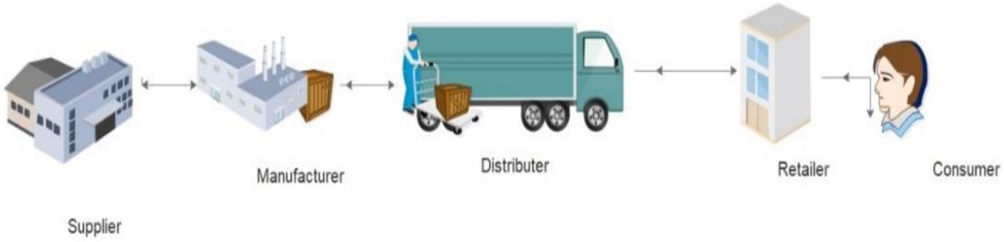


Figure 1.2 Supply-Chain Path

1.5 OUTLIERS IN RFID-ENABLED SUPPLY-CHAIN SYSTEM

An outlier is an anomaly or deviation [14] from anything which is normal. Outlier detection in RFID-enabled supply chain [11] is process of finding the abnormal or anomalous node in the supply chain network. An outlier may be generated because of unusual issues such as transport delays, thefts, etc. in the RFID supply chain. Mainly abnormal or outliers in the RFID supply chain process are explained with three ways, as described below:

- **Delay of transport -** In the supply chain, time of transit and arrival of the goods/objects is scheduled and for the products like fresh food materials without any preservatives is very important. So, delay in transport can have a derogatory effect which should be continuously monitored by the stakeholders to avoid significant losses and also know the reasons for delay to improve the system.
- **Theft -** Theft of the commodity is another common reason for the abnormal path followed by any object which can lead to the outlier or abnormal activity and these mostly occur in the time period of goods-packing by suppliers, goods-unloading by retailers or drivers' rest. Tracking of goods in the supply-chain system in real time, can largely reduce the loss rate.
- **Fake -**Inclusion of fake products by counterfeiting in a RFID enabled supply-chain path is a big problem. Fake products may not follow the pre-scheduled path which may lead to anomalous/outlier condition and thus constant monitoring of the products throughout the processes of production, circulation, sales and so on can help in avoiding the fake.

The framework for RFID supply-chain outlier detection system, can be explained in layered form with three levels:

- The data collection and pre-processing layer
- Outlier detection layer
- Predictive analysis layer.

Data collection is done at physical level by the reader from the tagged object and transferred and saved to the middleware attached to it for further analysis. Depending upon range of detection the reader keeps on collecting the data automatically from the tag till the time it is in its vicinity. This collected data is in raw form with too much of redundancy due to multiple reads and therefore needs to be pre-processed for removal of redundancy, false positives and false negatives. The data pre-processing layer is divided into time pre-processing and path pre-processing management. It cleans the RFID event data obtained from the EPC network and provides the upper layer with clean and useful data by removing the possible false positives, false negatives and duplicate values. The outlier detection and analysis layer is the core of the three-layer system and uses clustering-based approach to find out the outlier nodes and classification approach to predict outlier points.

1.6 RFID DATASETS AND CHALLENGES

Taking into the consideration, the possibilities of the anomalies, we need to work on the time and location error in readings by the RFID readers in the supply-chain path. The scheduled time and location can be pre-fed in the system and then compared with the followed path by considering temporal factor also in the path along with the location factor. Here temporal factor refers to difference in time of reaching a particular node in the supply-chain network due to difference in speed and acceleration. The path followed by the tagged objects must be accurate and any path deviation from the normal must be reported.

An RFID object tracking system consists of a collection of RFID readers at various points scanning for tags at periodic intervals. Each reader is referred by some location and after reading data from the tagged object, it generates a stream of time ordered records in the form (*EPC; location; time*) where, *EPC* is a unique “Electronic Product Code” associated with tagged object, *location* is the location of object detection, and *time* is the time of read. As the data read is generated redundantly due to multiple reads of the objects by the reader till the time they are in its range, significant data compression can be done by merging all the readings for an item that stays at a location for a period of time, into a tuple of the form (*EPC; location*, *in_time*, *out_time*) where, *in_time* is the time when the item is identified by *EPC* entered *location*, and *out_time* is the time when it leaves from the range of reader. Tag readings are sorted with respect to each EPC and a path database is generated, where sequence of locations called trajectories traversed by each object is stored. Entries in the path database are of the format (*EPC;* (*l*$_1$*; in_time*$_1$*; out_time*$_1$) (*l*$_2$*; in_time*$_2$*; out_time*$_2$)….(*l*$_k$*; in_time*$_k$*; out_time* $_k$) and a path is given an id.

Some specific properties of RFID datasets which make them to be handled differently from non-RFID datasets are as follows: -

I. **Spatial and Temporal**: RFID readers continuously read data from the tags in its range and along with the location the time of read is also recorded which makes the data temporal.
II. **Inaccuracy of Data**: Inaccurate measurements are a major issue in RFID system due to false positives which is reading the non-existing tags and false negatives which is missing tags in the vicinity of the readers**.** Such an erroneous data should be filtered.
III. **Continuous Streaming**: Readers continuously keep on reading the tagged object till the time they are in their range in periodic intervals. A tuple/record is continuously saved into system connected to reader, whenever it reads a tag in its vicinity. The continuous data generation in real time leads to the generation and storage of redundant data also which should be filtered and compressed. A lossless-compression technique is needed to compress such redundant data.
IV. **Granularity**: Depending on the applications the level of granularity (size of aggregation of items under consideration) for data collection can vary. The granularity of data collection can be a single object or object container consisting of many objects taken as a single unit. In our research we are taking the granularity of single object at a time.

1.7 PROBLEM DEFINITION

Amidst the uncertainty of RFID dataset in supply-chain process, it is very important to take care of the abnormal readings and report it to the stakeholders. In the domain of supply chain management, most of the research work is done in data warehousing, path encoding, simulated data cleaning and very few researchers have worked in path clustering but not from the point of view of finding outliers. Moreover, the work done in the previous research work has not taken into account, the long sequence of paths particularly in the supply chain process. In our work we aim for creating a full-fledged integrated framework for the complete processing of RFID supply chain data, starting from data cleaning till the prediction of outlier points in the path trajectory of RFID enabled supply chain. As per our literature survey and knowledge gained through the study, no attention is paid on outlier detection in supply chain process in particular and no complete integrated framework exists which gives a complete solution starting from pre-processing till predictive analysis of outlier points in the RFID enabled supply chain. Outlier detection in supply-chain process can be defined in terms of localization for determining deviation from the particular scheduled location, tracking for determining deviation from the complete path or trajectory by an object, checking logistics for any deviation from the normal flow of products and analysing transportation for any failure in the transportation. If any inappropriate quantity of product as expected is found during tracking as well as tracing the shipments in supply chain process, then these events should alert the stakeholders for the outliers or abnormalities for necessary action. Tracing shipments could find inappropriate quantity and quality of the product and notify all trading partners in time. Our work is related to the localization process.

Our research work includes the cleaning of RFID data read by readers from the tags and the cleaned data is sent to the middleware systems attached to the RFID readers for finding abnormal or outlier points in the supply-chain path and further using that information of outlier points in the supply-chain paths / trajectories as training data for the accurate prediction of outlier points using recurrent based neural networks. Also, the comparative analysis of the proposed approaches with the existing approaches is done. The complete sequence of processes can help in the domain of RFID enabled supply-chain process with correct analysis of possible outliers in the complete chain of processes, making it more cost effective with better efficiency and manageability. Our research work may give better results in case of the scenarios where time lag factor is considered while checking the path similarities Also in case of longer path sequences the technique known as Long Short Term Memory is going to give more accurate results unlike the other traditional techniques like sequential pattern mining or Hidden Markov Models [15] which are most commonly used for path prediction.

1.8 OBJECTIVES

The problem of developing outlier detection system in RFID enabled supply chain is broken into following objectives:

- To design a data model for RFID data in supply chain and to perform pre-processing of data.
- To develop scheme for outlier detection in RFID-enabled supply-chain path.
- To perform the predictive analysis of outlier locations in supply-chain path using automatic machine learning technique.

1.9 ORGANIZATION OF THESIS

The thesis has been organized in six chapters; the descriptions of each of these chapters are as follows:

Chapter 1: It gives the introduction to the wireless RFID networks, RFID in supply-chain process and its importance in enterprise management and administration. The problem definition and objectives of the proposed work are also given in this chapter.

Chapter 2: It provides the detailed literature survey about techniques related to RFID data cleaning, outlier detection/path deviation techniques used in the domain of RFID-enabled supply chain and predictive analysis of outlier paths/trajectories. It explains about the existing problems in the domain related work and the scope of research in the discussed domain.

Chapter 3: This chapter presents the adaptive data cleaning of RFID data sets for reduction of false negatives and false positives and elimination of duplicate values to finally get cleaned and less redundant data**.** In this chapter the complete cleaning process is explained and the performance analysis is shown with cleaning and without cleaning approach.

Chapter 4: This chapter explains the complete schema of the tables used for managing RFID enabled supply chain processes. It shows the complete framework of outlier detection with unsupervised approach. A comparative study of different distance measures is explained on the basis of different parameters. The design of a novel density-based RFID supply chain trajectory outlier detection: TRAJODBSCAN and its implementation is explained along with the comparative analysis with traditional approaches of clustering.

Chapter 5: This chapter explains the various supervised approaches used for the predictive analysis of Outlier path in the supply chain process. Most commonly used methods predictive analysis of time series data is compared with Long Short-Term Memory (LSTM). The discretization of data to is done prior to input formatted for the methods. An exhaustive performance analysis among different techniques used along with performance tuning of hyper parameters is done with explanation for each.

Chapter 6: This chapter gives a conclusion and provides guidelines for future work in this area.

1.10 PROPOSED WORK FLOW OF PROCESSES

The proposed work flow is logically divided into following steps:

I. Construction of data model for RFID supply-chain data and pre-processing.

II. Supply-chain trajectories' outlier detection using a novel density-based clustering method TRAJODBSCAN.

III. Construction of the framework for predictive analysis of outlier path in supply-chain process. The sequence of processes is as depicted in as shown in Figure 1.3.

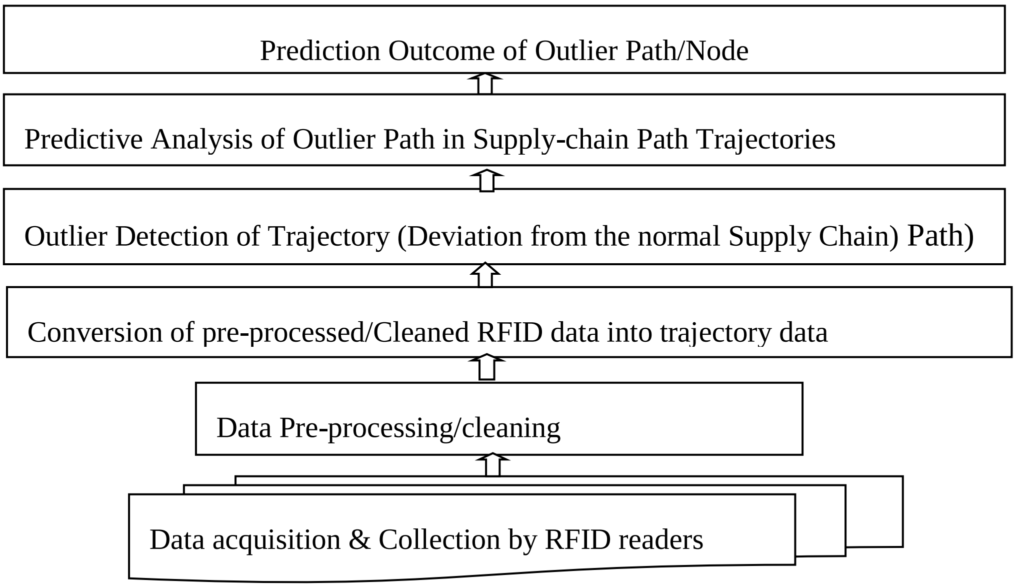

Prediction Outcome of Outlier Path/Node
Predictive Analysis of Outlier Path in Supply-chain Path Trajectories
Outlier Detection of Trajectory (Deviation from the normal Supply Chain) Path)
Conversion of pre-processed/Cleaned RFID data into trajectory data
Data Pre-processing/cleaning
Data acquisition & Collection by RFID readers

Figure 1.3 Sequence of Process Flow

CHAPTER 2: LITERATURE SURVEY

RFID technology has been mostly used in the domain of manufacturing, which includes designing of product and process, assembly, planning of materials, controlling quality, scheduling the processes, after maintenance, etc. Most of literature studies are related to objects tracking and product management in the domain of supply chain but very few researchers have worked on the abnormal condition or outlier detection while monitoring of RFID tagged objects. Here by abnormal condition or outliers we meant any kind of deviation of a node from its normal processing or behaviour.

Our research is specific to the supply-chain process using RFID system and in that too specifically for the abnormality detection in the localisation process in supply-chain process. Inaccurate localisation of objects can be due to several reasons like theft, counterfeiting, traffic problem, environmental factors like bad weather, road repairing etc., vehicle carrying tagged objects is not working and so on.

2.1. INTRODUCTION

Supply-chain process enabled with RFID is still in the nascent stage due to challenges in shifting from the traditional approach which involves high cost of implementation and skills and time required in restructuring. The main reasons for the issues are lack of literature availability for the complete makeover from the non-RFID system to RFID implementation and its deployment in the current business processes. Improvements in supply chain were hindered due to lack of academic research and understanding of the technology though it’s now-a-days taking up leap. We hope that this study can further develop insight into the challenges and opportunities of RFID and can direct academicians for further research on the areas of RFID that are most pertinent to practitioners.

This chapter provides the literature survey about techniques related to RFID data cleaning, outlier detection/path deviation techniques used in the domain of RFID-enabled supply chain and predictive analysis of outlier paths/trajectories.

As already discussed in previous chapter, RFID is used to keep the track of processes and trace the supplies in construction and assembly industries. Ngai et al. [12] studied the literature of RFID technology by organising their studies into technological based, application based, policy based and security-based categories. Not much of the literature was available to study the technology. Overall only 85 research papers were available. Their analysis gave useful insights on the anatomical details of the RFID literature. As RFID technology matured, so many applications [16] were unleashed to exploit inexpensive and highly available automatic identification. A complete framework for monitoring the progress using smart objects like RFID along with web service technologies in ubiquitous manufacturing had been proposed by Qu et al. [17].

RFID technology has shown the remarkable improvement in the domain of production planning and scheduling [18]. RFID systems do real time coordination and interaction among various levels like production level, planning level and scheduling level for achieving the lean control of processes in manufacturing [17].Smart manufacturing shop floors are created with RFID technology [19]. Supply chain with RFID technology has lot of advantages [20].

Some works focus on managing and mining RFID stream data. Hector Gonzalez et al. have done extensive work in this domain in various aspects. Traditional data warehousing multidimensional models won't fit into RFID datasets due to various properties of RFID data as discussed in the previous chapter. A new model for warehousing RFID data has been proposed [21] in which object transitions' preservation is considered owing to the temporal feature RFID data. In order to represent the transportation of objects FlowGraph method has been proposed [22]. This representation can be very effective in the multi-dimensional analysis of flow of objects. Easy capturing of the movement of objects and monitoring exceptions in RFID flows has been proposed by using compressed probabilistic workflow method [23].

Elio Masciari [24] has researched on the outlier mining in RFID data stream to find out the outliers/anomalous nodes in the supply chain network. This research work used discrete Fourier-transform method to check for the similarities among various paths in the supply chain. Some research is done in area of mining RFID data. A framework called Rule-and-Motif-based anomaly detection has been designed for anomaly detection in moving object [25] but it does not fit well for the data with too much of rules as it adds to the complexity.

Taking the combined data in a supply-chain path which can be taken as a trajectory data, a novel partition-and-detect framework [26] for outlier detection of trajectory or path followed by the moving object had been proposed in which each trajectory or path followed by the object is first split into parts and then checked for similarity using all the angles between the distances in trajectory paths. This approach used clustering based approach but still didn’t consider the speed variations of the objects in trajectories. Our research work has solved this issue though the base is similar to this research work. A new trajectory classification method for classifying various trajectories in supply chain called TraClass which uses hierarchical region-based approach is proposed by the same authors who used partition- and-group -detect framework [27] but it is not scalable.

Sensor network has some typical traits like limitation of resources, easy deployment of sensors, multiple hops in the network, massive data generation, and low maintenance requirement. Data mining with such traits is also different that way. Under the constraints of computational/memory/power limitations, A general probabilistic framework which uses supervised learning has been proposed by Ghosh et al. [28] considering computational, memory related or power related constraints. The main issue with supervised approach is the availability of training data which may not be the case in all type of applications. Moreover, we need to see over fitting problem too. One more approach which uses Spatio-Temporal Sensor Graphs (STSG) to model and mine sensor data for finding anomaly patterns and centralised locations at each time interval, has also been proposed [29] but for very long sequences it may not be very effective. An adaptive mining framework which can adapt according to changes in data has also been proposed by Cook et al. [30].

There are several contributions towards data mining from Internet of Things (IoT), our main focus is on the one of the very important rudiment of IoT i.e. RFID. As a completely new paradigm in the research area, RFID-related applications still lack sufficient models and theories for the application of machine learning or data mining techniques. In the next section, a detailed study of the existing works done in the respective areas of sub-processes of the research work is explained along with the summarisation for the research gaps.

2.2. EXISTING WORK IN SUB-PROCESSES OF RESEARCH WORK

Our research work is defined by three sub processes and according to this, we have shown the related work/existing work in these areas. As discussed in the previous chapter, our Outlier Detection Framework consists of the three layers:

I. The Data Pre-Processing Layer:

It is divided into time pre-processing and path pre-processing management. It cleans the RFID event data obtained from the EPC network and providing the upper layer with clean and useful data by removing the possible false positives, false negatives and duplicate values.

II. The Outlier Detection and Analysis Layer:

It is the core of the three-layered system and it uses clustering-based outlier detection test in this work. Density based clustering technique is used for finding outlier clusters.

III. Predictive Analysis of Outlier Nodes:

Predictive analysis of outlier nodes is done by recurrent-based neural networks, specifically Long Short-Term Memory (LSTM) [31].

2.2.1 Existing Work Related to Data Cleaning

RFID data generated by reader as read from tag is quite unreliable and redundant due to many external technical as well as environmental factors. Many Methodologies are proposed in literature to improve the reliability of RFID data. Work is done on both the hardware and the software aspects in a RFID system with respect to cleaning of RFID data [12]. Middleware solutions which are software based, refers to the implementation of algorithms for the correction of data streams coming through readers before being passed to the final database saved in the system for further analysis. There are many RFID-middleware solutions [32] [33] which are based on simple filtering techniques using fixed temporal sliding window filter to remove false negatives and false positives from RFID data and in these applications the very important thing is setting up of the window size for sampling the data and it's a drawback too. In the dynamic environment, it's not trivial to set up the appropriate smoothing-window size. Here the data generated in the environment is the continuous stream of datasets. A balance needs to be maintained between the tasks of ensuring completeness for the readings because of system unreliability and also ensure full capturing of the dynamics of tag as it moves in and out of the detection range of RFID reader. If a window size is selected as large then although it ensures the completeness but the system is not able to efficiently detect transitions of tags from inside to outside the window or outside to inside the window. On the other hand, if a small window size is selected then the system is able to detect transitions but it cannot ensure completeness due to missed readings. So if the size of window is set as small then it is possibility that some tags may miss being read leading to generation of *false negative* errors in which the tag is mistakenly assumed to be absent while it is actually present and if the size of window is set to very large then it can lead to generation of false positive errors as due to interpolation of the readings of the tag read by reader some tags which have already exited the detection region are also considered to be present and their information in stored by the system. So, in the real-world scenario experimentally no particular single sized window can consistently perform correctly for finding correct tag reads. We have studied and used an adaptive window-based approach called as window sub range transition detection algorithm (WSTD) [34] which can perform very well even in harsh environment with many dynamic changes. This approach also overcomes all the disadvantages of the fixed window protocol for filtering RFID data. The decision to use WSTD is based on the following literature study of the various techniques being used for the existing scenario.

Bai, Y, F Wang and P Liu [35] suggested the cleaning of RFID data in raw form into semantic application data. The false positive and negative readings as well duplicated readings should be filtered before being converted into the semantic form so that they can be used in different applications. The authors have proposed several effective methods to filter RFID data, for removing noise and eliminating duplicates. They have used sliding window protocol with fixed size and threshold limit for removal of noise to be passed as input. In a dynamic environment of RFID data streaming continuously, setting up the parameters is quite a difficult task.

Gonzalez, H., J. Han and X Shen [36] proposed method based on Bayesian Networks which has the advantage of adjusting dynamically based on the probability of the tag existence. Due to its dynamism it is called as dynamic Bayesian network this method is computationally expensive due to sigma functions and cross-corpora calculations and performs very poorly in case the data set used is small.

Shen, H. and Y. Zhang [37] have used counting bloom filter-based technique known as decaying bloom filter. As new tags continuously enter in the sliding windows, the old and unused tags are removed. The method used can efficiently detect duplicate readings also. The problem with this filter is detection of false positive errors only but it cannot handle false negative errors.

Fan, H., Q.Y. Wu and Y.S. Lin. [38] proposed stream data processing by converting the semantic data in different logic rules to monitor the abnormal conditions of the work pieces in the manufacturing workshops. Work piece can be analysed based on real time processing as well as history-oriented tracking. It works fine for small amount of data but doesn't fit well on large number of datasets.

Jeffery, Shawn R., Minos Greatlakes, and Michael J. Franklin [39] proposed an adaptive smoothing filter which can help in fixing the problems related to fixed window sliding protocol to reduce false positive and false negatives. An adaptive smoothing filter aggregates the RFID data and interpolates for lost readings. The algorithm known as Statistical sMoothing for Unreliable RFid data (SMURF) uses sampling theory and cleans the data by taking statistical sample of tag ids of the physical world and thus helps in modelling the unreliability of the RFID data readings by the reader. It uses binomial sampling and π-estimators; SMURF does the setting of correct window size automatically with continuous adaption over the historical and currently observed data readings. The effect of this cleaning algorithm is optimal only when the RFID tags move at a uniform speed. But in case of high-speed movement of tags in and out of readers' detection range the performance of this algorithm starts decreasing. Many variations like VSMURF [40] have been proposed based on SMURF concept.

Y. Wang, B.-Y. Song, H. Fu, and X.-G. Li [41] proposed a cleaning method KAL-RFID which is based on Kalman Filter. It is used to find false negative and false positive readings as well as solved the problem of delay occurrence in the transition time of tag stream. Kalman Filter update process consists of updating of time and measurement. This process needs lot of memory for storing the tags.

Wang, Y. L., C. Wang and X. H. Jiang. [42] proposed a cleaning method based on bloom filter for handling the redundant data generated in the distributed data flow environment.

Massawe, L.V., J.D.M. Kinyua and H. Vermaak [43] proposed an improvement over SMURF which is also an adaptive sliding-window based approach known as Window Sub Range Transition Detection (WSTD). It can handle environmental variation and tag dynamics very efficiently. It can very well adjust the smoothing window size and thus can cope up with the changes in the environment which could lead to variations in the tag-reader performance. This thesis work cleaning method is based on WSTD for supply chain datasets. The existing work and the research gaps are summarised as shown in the Table 2.1 below:

Table 2.1 Summary of Existing Approaches in RFID Data Cleaning

Sr. No.	Approach	Methodology	Research Gaps
1.	Bai, Y, F Wang and P Liu [35]	Fixed Window Protocol	Problem in setting window size
2.	Gonzalez, H., J. Han and X Shen [36]	Bayesian Networks	Computationally extensive, perform poorly for small datasets
3.	Shen, H. and Y. Zhang [37]	Decaying Bloom Filter	Can’t detect false negative errors
4.	Fan, H., Q.Y. Wu and Y.S. Lin. [38]	Conversion of semantic data into logical rules	Not efficient for real time data processing

5.	Jeffery, Shawn R., Minos Garofalakis, and Michael J. Franklin [39]	Adaptive window sliding protocol	In case of high-speed movement of tags in and out of readers' detection range the performance of this algorithm starts decreasing.
6.	Y. Wang, B., Y. Song, H. Fu, and X., G. Li [41]	Kalman Filter	Process needs lot of memory for storing the tags.
7.	Wang, Y. L., C. Wang and X. H. Jiang. [42]	Based on Bloom Filter	Preprocess redundant data only, not false positives and false negatives

2.2.2 Existing Work in Outlier Detection of Trajectories in Supply-Chain Process

Outlier detection in RFID enabled supply-chain process requires the study on the trajectories identification which are not normal or deviating from the normal path called as outliers and to check for the similarity between various trajectories, different similarity measures are studied. The dataset formed in the supply-chain process is read in terms of trajectory as it contains the component of time and location in a particular order as read by the readers at various locations of the supply-chain path.

In literature, several authors have used many types of techniques [44] like clustering [45], classification, sequence pattern matching, probabilistic statistical techniques to find out outlier points or set of points from the given set of points. For the clustering approaches the training/labelled dataset is not required [46] but for the classification [44] there should be the availability of training points which is not available in all kind of applications. Any clustering algorithm/technique is based on the concept of finding dissimilarity/similarity between the objects to be clustered. The type of objects depends on the application being studied. Our research work deals with the trajectories followed by objects in the supply-chain path so here the similarities or differences between trajectories are considered and related work is studied. There are many similarity measures [47] studied and implemented with each having their own pros and cons.

Euclidean distance [48] is the most commonly used distance measure with the condition of equal length in case of trajectory data [47] but it's not suitable for the cases in which the length of the trajectories are unequal. Also, it does not consider the time lagging factor within the path from source to destination. Other similarity measures like Hausdorff measure [49], Edit Distance [50], Fréchet Distance [51], Longest Common Subsequence [52], Dynamic Time Warping [53] etc. are also proposed for different applications. The following section covers the detail of related works in the area of trajectory mining as this is the base taken in our research for finding outlier points/nodes in the RFID enabled supply-chain network.

Wang, Haozhou, et al. [47] have done a comparison of various trajectory similarity measures. Methods from time series analysis can be applied for the computation of trajectory similarity as the structure is same. Methods like Dynamic time warping (DTW), Longest Common Sub sequences (LCSS) and Edit Distance are quite commonly used methods. DTW, Edit Distance, and LCSS allow flexibility in finding match without any requirement of matching points at corresponding times. DTW and Fréchet distance measures don't require exact time correspondence [54]. Time correspondence can't be ignored so simple Euclidian distance measure can't solve the purpose.

Berndt, D.J. and J. Clifford [53] proposed Dynamic Time Warping (DTW) distance measure as the technique to find the similarity between various speech patterns. Later this measure is used in the variety of problems in various other domains too. It is based on Euclidian distance but with the consideration of lagging or leading time factor over the complete path and thus is a solution to the weaknesses of Euclidian distance metrics. Due to this approach the time series or sequences which are not in phase locally due to temporal factor but are still similar are also taken into consideration. Although the time complexity of this approach is quadratic it is still considered an efficient way in terms of accuracy of finding similar time series/sequence data and is popularly used in various application areas like bioinformatics, medicine, engineering, entertainment etc.

Jeung, H, et al [55] proposed a hybrid prediction model to study the trajectory pattern being followed so that it can help in estimating status of any node in the trajectory network. Oobject’s movements are based on many environmental factors like road networks and traffic jams for vehicles, turbulence places for aircraft, and so on which makes use of mathematical formulas to represent the patterns followed in a path/trajectory, an inefficient way. So, the authors have proposed a novel approach which is a combination of Apriori [56] [57] for detecting frequent trajectory patterns and DBSCAN [58] for further clustering the sub trajectories. Use of Apriori for trajectory patterns however is not very memory efficient due to lot of candidates’ generation in the intermediate steps.

Chun-Hee, L. and C Chin-Wan [59] proposed a path encoding schema using the concept of Chinese remainder theorem for the processing of large amount of RFID data for supply-chain management. However, with the increasing of the tag numbers in system, the storage cost of data and the time cost are not utilized efficiently.

Masciari, E. [60] proposed asystem called as SMART (Simple Monitoring enterprise Activities by RFID Tags) which is based on defining a template for detection of outliers. The templates cover all the outlier scenarios and based on them the matching is done to find the outliers in RFID data streams. The templates consider sample taken(P), type of monitored objects (O) and the attributes (A), the outlier definition by means of a suitable function F (P, A, O) → $\{0, 1\}$. It is defined like a rule but for too many samples and objects this is not going to work efficiently. Scalability is an issue here.

Fan, H., et al. [38] proposed a model using the concept of a tree based structure path spliiting so that it records the movement of the products/tagged objects in trjactory path. This tree model finds out any deviation from the normal path and thus can be used to find out outliers, but for longer paths, it would increase the time and space complexity. Also redesigning of relational schema which can also store path and time information is required.

Hanning C..et al. [61] proposed a novel method based on K means clustering algorithm [62]. Sequence of locations and time i.e. spatio-temporal elements are used to construct path network. Both K Means and Mean Shift [63] algorithms are compared for clustering similar paths and comparatively Mean Shift algorithm performed better but either these algorithms need the number of clusters to be formed in advance and also there is the restriction of the spherical shape of the clusters or they are very slow, it is also embarrassingly parallelizable, as each point could be shifted in parallel with every other point.

Liu, X., et al. [64] worked on finding outliers in RFID trajectories by doing spatial analysis of the association among various discrete points in the path. Kriging method [65] is used for the interpolation of the number of points. Spatial and temporal variation in the accuracy of RFID readings is assessed quantitatively for finding or predicting the missed information values. It is not effective with large number of discrete points.

Kwon K., Kang D., Yoon Y., Sohn J.S., Chung I.J. [66] proposed a method called Procedure Tree for the mining of the massive data flow generated by tagged objects read by readers. The proposed system can perform better as compared to traditional systems for the tracking of objects but for longer supply-chain paths its efficiency decreases in terms of time and space utilization.

Huang S.P., Wang D. [11]. proposed distance based and rule-based approaches to detect the anomalies like delay in transiting and steal of the packages in the supply-chain path. The system can provide some help for the enterprise management so as to help enterprises to effectively control the information of supply chain.

The existing work and the research gaps are summarised as shown in the Table 2.2 below:

Table 2.2. Summary of Existing Approaches in RFID Outlier Detection using Clustering

Sr. No.	**Approach**	**Methodology**	**Research Gaps**
1.	Wang, Haozhou, *et al.* [47]	Study of distance measures like Euclidian, Dynamic Time Warping (DTW), Longest Common Sub Sequences	Euclidian Distance not appropriate approach with temporal factor

Sr. No.	**Approach**	**Methodology**	**Research Gaps**
		(LCSS), Edit Distance	
2.	Jeung, H, *et al* [55]	Hybrid model (Apriori + DBSCAN)	Not memory efficient
3.	Chun-Hee, L. and C Chin-Wan [59]	Path Encoding Scheme	Memory and Time inefficient with increasing tag numbers
4.	Masciari, E. [60]	Rules/template creation	It doesn't work with high scalability
5.	Fan, H., *et al.* [38]	Tree based model	It doesn't work efficiently with long sequences of trajectory path.
6.	Huang, S.P. and D. Wang [11]	K Means and Mean Shift based Algorithm	Forms only spherical clusters and not efficient for finding outliers or it is too slow and also embarrassingly parallelizable, as each point could be shifted in parallel with every other point.
7.	Liu, X., *et al.* [64]	Kriging Method of Interpolation	It doesn't work efficiently for dynamic system
8.	Kwon K., Kang D., Yoon Y., Sohn J.S., Chung I.J. [66]	Procedure Tree Method	It doesn't perform efficiently for longer sequences

Sr. No.	Approach	Methodology	Research Gaps
9.	Huang S.P., Wang D. [11]	Rule and Distance based method	Too many path sequences lead to too many rules generation. It is not memory efficient

2.2.3 Existing Work in Predictive Analysis of Outlier Points in the Trajectories

The concept used in this research is based on the previous information about the location e.g. status of any particular node or set of nodes in RFID network paths. It can be an outlier point or non-outlier point. location-based prediction as done in the research work where every location is confirmed for non-outlier or outlier status. Training data is taken as the data generated after applying our proposed TRAJODBSCAN. Finally, outlier status is predicted based on the previous points in time series RFID trajectory path points. Many location-prediction algorithms are available which can predict the next location given the current location. The simplest way to handle this is by using speed and direction of movement but it’s not easy in the real-world scenario in which the problems like traffic jam, theft, poor weather conditions do exist. As reviewed by Giannotti et al. [67] there are many methods for various applications specifically for data mining of trajectory. If the previous history data is available about the paths or trajectories being followed along with information about the deviation or outlier points in the path, prediction of any deviation or outlier path can be predicted. Main focus is improving the accuracy of prediction and according to literature available many techniques like pattern mining of moving objects [68] and model-based mining [69] are most commonly used. Li et al. has worked on the path prediction of [70] based on their moving pattern and behaviour. The trajectory data is transformed in form of cell points for all the points and mining is done on that format. Yavas et al. [71] also worked on frequent pattern mining for path detection or deviation with Apriori algorithm as a base algorithm. Locations, which are co-occurring are extracted from the frequent patterns generated and analysed. Further Morzy et al. also considered temporal and spatial features and developed a modified Prefix-Span algorithm [72].A multi-centre Gaussian model is proposed by Cheng et al. [73] for predicting the distance between the various patterns generated but in this method the sequencing or ordering is not considered. A hybrid technique based on Hidden Markov Model has been proposed by Mathew et al. [74]. Jeung [55] used cell partition-based algorithm to map the trajectory points into frequent regions but again the prediction accuracy here is constrained by the granularity of the cells.

Sequence mining gained popularity and use of neural networks became the favourites for many researchers. Recurrent Neural Networks (RNN) [75] are very popularly used for time series data in wide areas of applications in the sequence mining [76]. In recurrent neural networks each layer actually represents each time step in the series of timestamp values. They are first kind of networks with internal memory and due to this reason, they are the most suitable ones for sequential data where previous steps need to be memorised. Liu et al. [64] extended traditional RNN spatial and temporal contexts to predict the spatio-temporal data. For small sequences or trajectories, RNN gives good results but when the length of the trajectory is very long, even more than twenty steps, the problem of vanishing and exploding gradients comes in and can’t be handled by RNN.

We hypothesize that Long Short-Term Memory (LSTM) [31] may be the solution to this problem. LSTM uses the memory blocks in any of the layers and thus remembering the context and value for much older than recent previous history. Specifically, LSTM-based architecture is used for our outlier point’s prediction in the supply-chain path. This concept has been used in this domain for the first time as per our knowledge. The related work as studied by us in the existing literature is discussed as follows:

Yavas G. Katsaros, D. Ulusoy O. Manolopoulos, Y. [71]. proposed a three-layered approach for the path prediction of users on personal communication network which is divided in the form of cells. In the first layer, the mobility patterns of the user are processed on the basis of the historical data of user trajectories. In the second layer, patterns are extracted in form of rules based on the mobility of users and in the third and final layer; the rules generated in the previous layers are used to predict the path in the communication network for the users. The Mobility rule-based prediction method is also compared with Mobility Prediction based on Transition Matrix (TM) which takes on the historical data and Ignorant Prediction method which does not take historical data. The accuracy of the proposed method is better but it takes toll on the memory requirements of the process.

Morzy M. [72] proposed movement rules method for finding frequent patterns in trajectories. Any trajectory followed by a moving object is compared against the saved movement rules and a probabilistic model to locate the objects in a path/trajectory is used. Proposed algorithm gives reasonably good prediction accuracy (80%). With the increasing network of trajectories, the proposed system can become very complex and with too many rules generation, the memory requirements would also be very high. Also matching with the rules will require higher processing time.

Cheng C., Yang H. King, I., Lyu M.R. [73] proposed a model based on matrix factorization method to find out the probability of check in by the user on any location in the path of the network. A framework with matrix factorization as well as social information of the user is used to demonstrate that the fused matrix factorization framework with multi Gaussian method uses the distance information and helps in the predicting the patterns of user check- ins in a particular network environment. Generation of matrix, though, is a high memory requirement process.

Jeung, H. Shen, H.T. Zhou, X. [55] proposed a novel approach which overcomes the problems associated with issues related to cell partitioning based processing. They explain the association between the frequent regions and the partitioned cells by using trajectory pattern models based on hidden Markov process. With the proposed approach, the movement of any object is defined by the partitioned cells structure but the trajectory patterns used by the objects are defined by the frequent regions being followed by reading those cells. Deciding the granularity level of the cells is a problem. Moreover, this approach doesn’t work well with longer trajectory paths.

Mathew W., Raposo R., Martins B. [74] proposed a representation learning method based on Hidden Markov Model (HMM) approach for Location-Based Social Networks, to be used for location recommendation and link prediction. The method helped in removing the drawbacks of the existing methods which focus only on topology patterns but not the sequences of check-ins. So, the approach works well in dynamic environment with hierarchical network. This approach however doesn’t work well for longer sequences in a trajectory or path.

Kim, Moon-Chan, et al [77]. proposed a fuzzy cognitive map model. The weight matrix uses genetic algorithm with previous states data based on which the analysis of next state is done. It also took care of sudden change in the state and the cause for it base on the previous state data. The problem with this approach is that it does perform efficiently for very long paths or trajectory.

Graves, A. Mohamed, G. Hinton [78] studied about Recurrent Neural Networks (RNNs) for sequential data. The authors investigate deep recurrent neural networks and concluded RNN performance degrades with the increase in sequence path length.

The existing work and the research gaps are summarised as shown in the Table 2.3. below:

Table 2.3. Summary of Existing Approaches in Predictive Analysis using Classification

Sr. No.	Approach	Methodology	Research Gaps
1.	Yavas G. Katsaros, D. Ulusoy O. Manolopoulos, Y. [71]	Rules based	Memory inefficient and high time processing requirements
2.	Morzy M. [72]	Rules based	Memory inefficient and high time processing requirements
3.	Cheng C., Yang H. King, I., Lyu M.R. [73]	Factorization matrix	Memory inefficient
4.	Jeung, H. Shen, H.T. Zhou, X. [55]	Hidden Markov Model based	It doesn’t work efficiently with very long trajectory paths
5.	Mathew W., Raposo R., Martins B. [74]	Hidden Markov Model based	It doesn’t work efficiently with very long trajectory paths
6.	Kim, Moon-Chan, *et al* [77]	Fuzzy Cognitive Networks with Genetic Algorithm	It doesn’t work efficiently with very long trajectory paths

2.2 LIMITATIONS OF EXISTING RESEARCH

There are certain limitations in the existing work or research done in the different sub-processes’ domain. Specific to the application RFID enabled supply chain and processing and analysis of the data generated, not much of the literature is available. The related works are studied for similar kind of applications and data generated in RFID stream like traffic trajectories, healthcare applications, human trajectories etc. Here trajectory refers to the node's points combined to form a supply-chain path or any type of path depending upon the application.

The following conclusions are drawn after going through the complete study of existing literature on sub processes used in our research work i.e. data cleaning, outlier detection using clustering approach and predictive analysis of path points in a trajectory.

Most of the existing work in data cleaning uses fixed window sliding protocol due to which there is a trivial problem of fixing up the window size for cleaning the false positives, false negatives and duplication in reading the tag data by RFID readers. Setting very small window size can result in generation of false negatives and setting up a very large window size can result in generation of false positives. Some techniques like Kalman filter and bloom filter need too much of memory and speed of processing is low. Adaptive window sliding detection is the best as it is dynamic window adjustment according to the stream of RFID tag data.

Most of works done for anomaly /outlier detection is based on clustering techniques. The main reason for it is non-availability of training data. Among the normal data the anomalous node point data is very less. The distance measures generally used are Euclidian, Fréchet, edit distance, longest common sub sequences, especially for trajectories but they have the drawbacks of either not taking into account the temporal factor into account with lag or lead of the objects due to speed variations or doesn't work effectively for longer trajectories. Dynamic Time Warping is another similarity /distance measure which has a drawback of high complexity but works efficiently in case of trajectories of different lengths as well as time lags due to speed variations in the objects following the trajectories. To cluster the trajectories along with planned trajectories many clustering approaches like k means, mean shift, hierarchical clustering, cell-based partitioning is used but they either are not memory efficient, doesn't work well with longer trajectories or can't cope up with the dynamic and uncertain RFID data stream. Further for the sub process of predicting outliers using the previous history of trajectory data, many techniques like rules based, matrix based, hidden Markov model based, decision tree based, neural networks based are used but they don’t work efficiently in case of long length trajectories.

2.3 SCOPE OF RESEARCH

The literature study gives us the insight for the development of a complete framework to find out the outliers in the RFID enabled supply-chain path. As per our understanding and study till now there is no availability of such system in the mention domain. A system which can find out outliers with good accuracy with the consideration of long path sequences of the trajectories followed by the objects with varying speed and acceleration is required. The research work done aims for the same and overcome the problems in the existing systems. Scope of research is quite vast as the framework designed can be used to find out outliers or anomalies in various applications which are RFID enabled. Apart from outlier detection of RFID supply chain datasets, they can also be used in RFID path deviation detection, RFID enabled healthcare process, RFID enabled toll process, RFID enabled car parking, RFID enabled tracking of things and so on.

CHAPTER 3: ADAPTIVE DATA CLEANING FOR RFID DATASETS

RFID data in raw form contains many errors and is not cleaned enough to be used in analysis. The uncleaned data, if used for further analysis can give wrong interpretation and analysis, so it must be taken care of in the beginning itself. This is called pre-processing step before the data is actually used for processing based on the analytical requirements and model generation and its evaluation in the process of its data mining.

3.1 INTRODUCTION

Before any kind of processing of data is carried out, cleaning of same raw RFID data is an absolute must. The data generated by RFID tags is further read by RFID readers consists of many false negatives, false positives and it is redundant too. Here false negatives mean the missed information about the tags which were actually in the vicinity of the RFID readers but could not be recorded. False positives which are actually side effects of false negatives due to interpolation, are the readings recorded by the readers which are actually not present in the range of readers. Redundant data is duplicated readings due to multiple readings of the same tag ids by readers till the time they are in their vicinity. This chapter focuses on the pre-processing stage of the datasets generated in the RFID-enabled supply chain process and it mainly explains the cleaning process of RFID data. It starts with the basic introduction of the various ways discussed in the literature and then the technique used by us in our research work. Performance analysis is also done with the use of adaptive window-based technique for reducing false positives, false negatives and redundant data generated.

The literature survey done for the data cleaning of RFID data shows that most of the data cleaning techniques are based on fixed-size window sliding algorithm. Now here window size means the width of time duration set for reading of the tags by reader. Here the size of time window for the collection of data is fixed. It is a parametric approach and fixing of window is not easy. It depends on the analysis of the historical data. The biggest disadvantage of fixing the window size is either missing of false negatives, if the window size is set very less or missing of false positives, if the window size is fixed to be very large [35] [33] [66]. So, setting up optimal size is a very critical task. False positives are basically the byproducts of cleaning of false negatives due to interpolation of missed readings in big window size. Considering the above problems in setting the fixed temporal window size, the adaptive window-based technique is followed in this thesis work.

The technique is used along with some of the concepts of Statistical Smoothing for Unreliable RFID data (SMURF) [39] but with a greatly enhanced transition-detection mechanism as SMURF is not effective when data stream is unstable. The result has been evaluated on RFID data stream which takes at least 75 samples on an average, automatically as it is adaptive window-based technique where the size of the window of reading is not fixed but set automatically according to the continuity of the data stream using probability distribution concepts. The results of implementation of WSTD technique in MATLAB are showing WSTD approach deals with RFID with good accuracy avoiding the drawbacks of fixed sliding window protocol.

In RFID Middleware, the most popular techniques for RFID data cleaning for reducing number of false positives and false negatives are based on sliding window filters. Sliding window means the time frame of the readings that moves with time. The readings which are missed by reader but are expected to be read are recovered by the process of interpolation. Using the aggregate function such as count can help in removal of anomalous readings. Window is described at a particular time instant by some time interval e.g. at one point of time the time coordinate of window size is from t1 to t1+window_size and after d interval of time, the boundaries of window becomes t1+d and t1+window_size+d. and it goes on like this. It has starting time and end time with some gap of time which is called width of the window. In fixed size this width is constant but it keeps on varying in adaptive window-based method so that there is the minimum occurrence of false positives as well as false negatives. The readings as read by readers from the tag are datasets of the format: reader_id, tag_id, timestamp). Here the timestamp is epoch time in absolutely raw form which is the value of number of seconds passed since 1st January 1970 and it can be easily mapped to date, hours, minutes and seconds combination and as the time frame moves, the value of the readings of the previous time frame gets expired. Same reader is analysed for the sliding process and false negatives as well as false positives are cleaned with the setting of appropriate window size. In our research work, we have used adaptive setting up of window size and this adaptive data cleaning used by is known as Window Sub-Range Transition Detection (WSTD) [43]. WSTD has the ability to automatically set the window size according to the data samples varying due to the fluctuations in tag-reader performance factoring the environment conditions. The transition points from inside the window to outside it for a tag are detected accurately as compared to other filtering algorithms.

3.2 RFID DATA CLEANING SYSTEM

RFID data cleaning system is typically categorised into three types:

I. Physical Solutions
II. Middleware Solutions
III. Deferred Solutions

In physical solutions [79], the hardware components associated with the RFID systems are checked upon and deployed till the major improvement is seen in reducing errors e.g. number of readers, changing the orientation of the antenna, range of antenna and so on.

In the middleware solutions, the RFID stream read by the reader is first corrected for errors before passing it further by the systems attached to the readers for further analysis. In this thesis work middleware solution specifically, time window based smoothing method is used.

In case of deferred solutions [80], correction of errors is done after the reader passes RFID data with errors to the applications via middleware systems attached to the readers. Within the application there are intelligent techniques to correct the data for errors.

The most commonly used middleware solution uses concept of temporal-based window sliding protocol where the time window is set for reading the RFID data. Now the setting of window size can be static or adaptive. The aim is to reduce the chance of missed/dropped readings by giving the tag a chance to be read by the reader within its respective time window. Sliding window essentially interpolates lost readings from each tag within the respective time window.

Smoothing window size needs to ensure that all the tags present in the reader's detection range are read and should be able to detect the dynamics of tag i.e. accurately identifying the in and out of tag from the reader's range. So, setting a fixed window is not easy and reliable.

Adaptive window sliding algorithm which sets up the window size automatically based on the observed readings over a lifetime of the system is better way. The main challenge for this scheme is to differentiate between the states when there are missed readings and when tag can't be read due to its not being in the vicinity/range of the reader. Existing adaptive window-based approaches like SMURF [39] and its enhancement WSTD [43] used approaches based on statistical sampling of RFID data streams and according analysing the setup of window size. There are other variations of SMURF like VSMURF [81] but the basic concept followed by all the variations is same. WSTD approach is better if the tags' stream is too dynamic [43].

3.3 RFID DETECTION MODEL

Readers have some range of detection. Up to some distance the signal is strong enough to be called as strong detection region and after that distance tag is read but signals starts attenuating and that region is called as weak detection region. The tag can't be read at all beyond weak detection region. From strong to weak region the transition state of the tag starts in and out of time window as shown in Figure 3.1.

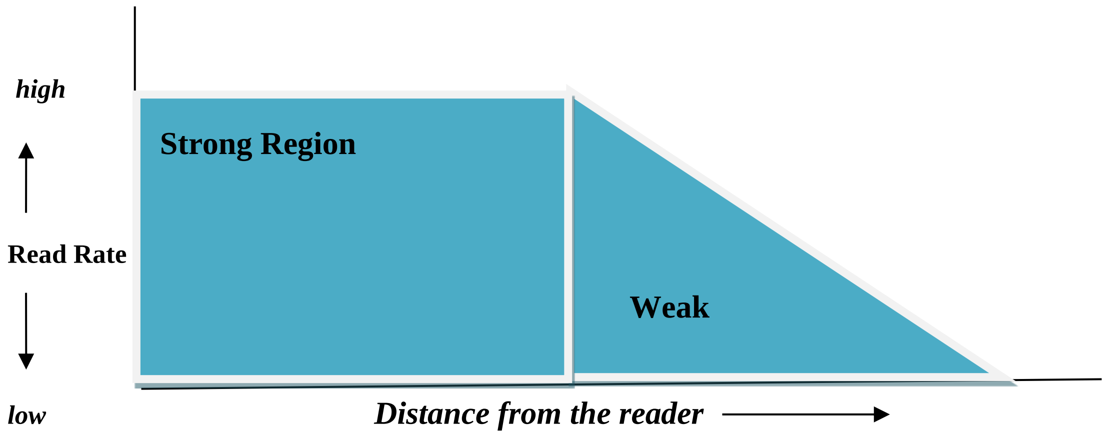


Figure 3.1 Reader-Detection Model

The observed inflow and outflow of the RFID tags in the range of the reader at the particular time frame is used to detect any significant statistical changes for fixing up the size of smoothing window automatically. Smoothing window is based on the completeness requirements and signals transition. In the strong detection range, up to 95 percentage of detection rate is there. Beyond weak detection range there is no detection of any tag by the readers.

3.4 WINDOW SUB-RANGE TRANSITION DETECTION ALGORITHM (WSTD)

Window Sub-Range Transition Detection Algorithm is based on probabilistic sample observation of the tags read by the reader within a particular time frame and then doing the statistical analysis of the outcome probabilities of read, to set up the size of the temporal window such that generation of false positives and false negative readings is reduced as much as possible. The RFID readings actually observed are treated as samples of all the tags in the physical world, generated randomly. There are many missed readings also due to various environmental or external factors due to which only a portion of the complete population of tags are observed and stored. Output is finding the adaptive window size after repeated random sampling trials for multiple epochs. An epoch refers to the unit of time taken by one read cycle by RFID reader. Following section explains the whole concept and steps used in adaptive window sliding technique: Window Sub-Range-Detection Technique.

Let the sample space consisting of complete population of tags is denoted as M_t at a particular epoch say t. The size of the population is unknown. A sample O_t is the subset of the population with values ranging from 1 to M_t during that epoch. Now the probability of picking of random sample is not equal. The probability of any i^{th} tag being selected at particular epoch t is calculated as the number of read requests per number of responses by the readers. It is denoted by $P_{i,,t}$. Equation 3.1 explains the calculation below:

$$\boldsymbol{P_{i,t}} = \frac{\boldsymbol{Total\ number\ of\ responses}}{\boldsymbol{Total\ number\ of\ requests}} \qquad \textbf{(3.1)}$$

Here each epoch is explained as an independent Bernoulli trial with successful observations of tag **i** in the time window. $\mathbf{W_i}$ is the time window with $\mathbf{n_i}$ epochs i.e.

W_i = (t – n_i, t). The binomial distribution with the probability of the success **$P_{i,t}$** of the tag being read by the reader is expressed as **Binom (n_i, $P_{i,t}$).**If the tag probabilities as expressed in equation 3.2 are assumed to be relatively homogeneous then if the average of the probabilities is taken, it will give an estimate of the actual probability of tag **i** within the window **W_i** . So, the following equation 3.2 defines the average read rate of the tags over the observations read cycles or epochs

$$\mathbf{P}_{\mathbf{i}^{\mathbf{avg}}} = \frac{\mathbf{1}}{|\mathbf{O_i}|} * \sum_{t \in O_i} P_{i,t} \qquad \textbf{(3.2)}$$

Also, **O_i** can be seen as binomial sample of epochs in **W_i** i.e. a Bernoulli Trial Probability **P_i^{avg}** for success and **$|O_i|$** as a binomial random variable with binomial distribution Binom (**n_i, $P_{i^{avg}}$**). Hence, from standard-probability theory, the expected value and variance of O_i is given as shown in equation (3.3) and equation (3.4)

$$\boldsymbol{Expected}[|\boldsymbol{O_i}|] = \boldsymbol{n_i} * \boldsymbol{P}_{i^{avg}} \qquad \textbf{(3.3)}$$

$$\textbf{Variance}[|\boldsymbol{O_i}|] = \boldsymbol{n_i} * \boldsymbol{P}_{i^{avg}} * (1 - \boldsymbol{P}_{i^{avg}}) \qquad \textbf{(3.4)}$$

The sampling model explained above ensures that the size of window would be set in a way as to ensure sufficient read cycles in the window frame say W_i for any tag say i can be read based on the reads which are confirmed by the reader. Equation 3.5 explains that the number of read cycles fixed to ensure that tag i is read within the window W_i will have probability greater than (1- $\boldsymbol{\delta}$) **.** Here, $\boldsymbol{\delta}$ refers to required completeness confidence. Now here completeness means stay of tag during the reading time in the time window while being read. We have set the completeness confidence to **0.1**. Each epoch (reading cycle) is about **0.2-.25 secs** [82].

$$\boldsymbol{n_i} \geq [\frac{\mathbf{1}}{\boldsymbol{P}_{i^{avg}}} * \left(\mathbf{ln}\frac{\mathbf{1}}{\boldsymbol{\delta}}\right)] \qquad \textbf{(3.5)}$$

In order to maintain a balance between capturing tag dynamics and also guaranteeing completeness, the WSTD algorithm uses some very simple rules along with statistical analysis of the underlying data stream to adaptively modify the cleaning-window size. If the number of readings which are actually observed are less than the readings which are expected to be read within that time frame then it means there is in and out of range transition within the window and there is big variation in the observations of tags. It can be very well explained using Central Limit Theorem. Equation 3.6 explains the concept as shown below:

$$\mathbf{|\,|O_i| - n_i\, P_i^{avg}\,| > 2{*}\sqrt{\ n_i\, P_i^{avg}\ (1- P_i^{avg})}} \qquad \mathbf{(3.6)}$$

Window size is additively reduced incrementally by two epochs in every loop such than number of tags which are recorded read but are not actually present due to wrong transition detection within the window time frame, can be reduced. The window is divided into two halves. In the first half the range of detection is maximum and during second half the tag will be moving out of the area of detection. Second half of the window starting from mid half is checked for the probability of tag detection and if a tag detection is showing consistent fall in the probability then it can be concluded that the tag is moving out of the window and detection range.

In case the number of samples as expected to be detected, are less than the actual number of samples detected then there are chances of false negatives and the size of the sliding time window should be increased. It actually means that the detection probability which is defined as the average of the probability $\mathbf{P_i^{avg}}$ is less than expected, increase in window size will help in detection of missed readings and reduction of false negatives. This is how window size is adjusted adaptively. When the observed readings are as expected are greater than or equal to the sample size actually detected then size of window remains as it is. There is no need to change the window size.

Steps of WSTD

The detailed steps in form of pseudo code of WSTD per tag-cleaning algorithm are given in the following Figure 3.2.

```
Input:     O=A complete set of all observed tag ID
δ=required completeness confidence
Output:     T=set of all present tag IDs
 /* window size is set to one epoch initially*/
for all i belongs to T, $n_i$ ⟵ 1
While (getNextEpoch) do
For (i in T)
/*Process window estimate the value of parameters $P_i^{avg}$ and $O_i$*/
```

```
Process window ($W_i$) ⟶ $P_{i,t}$, $P_i^{avg}$, $|O_i|$
if (tagidExist ($|O_i|$)
    Output i
end if
/* set up the required window size based on the required completeness confidence and average probability. */
$n_i$* ⟵ required_WindowSize ($P_i^{avg}$,$\delta$)
/*Window is split into two halves and then check for the presence of tag in first in first half and then in second half and accordingly increase or decrease the size of window to ensure completeness as well as detection of in and out movement of the tag from the window. */
existingtag ← movementdetection ($P_{i,t}$)
/*checking if the tag exists in the second half of the window i.e. When the tag is exiting the detection range. If it's not detected then the size of window is reduced to half to reduce false positive readings. */
if (existingtag and $|O_{2i}|$ =0)
    ni ← max(min($n_i$/2,$n_i$*),3) // 3 is the set as the min. window size
else if (detect_transition ($|O_i|$, $n_i$, $P_i^{avg}$))
    /*In case tag transition detection is seen, window size is reduced to 2 iteratively as it is sufficient to detect tag readings or else it can result in false positives. */
    $n_i$ ← max(($n_i$-2),3)
    /*Reduction in false negatives is ensured by incrementally increasing the size of the window by two, if the computed size of the window size is more than the current size of the window and expected number of observation samples are less than the actual number of observed samples */
else if(($n_i$*>$n_i$) and $|O_i|$<$n_i$* $P_i^{avg}$)
    $n_i$ ← min ($n_i$+2, $n_i$*)
end if
end for
end while
```

Figure 3.2 Detailed Steps of WSTD

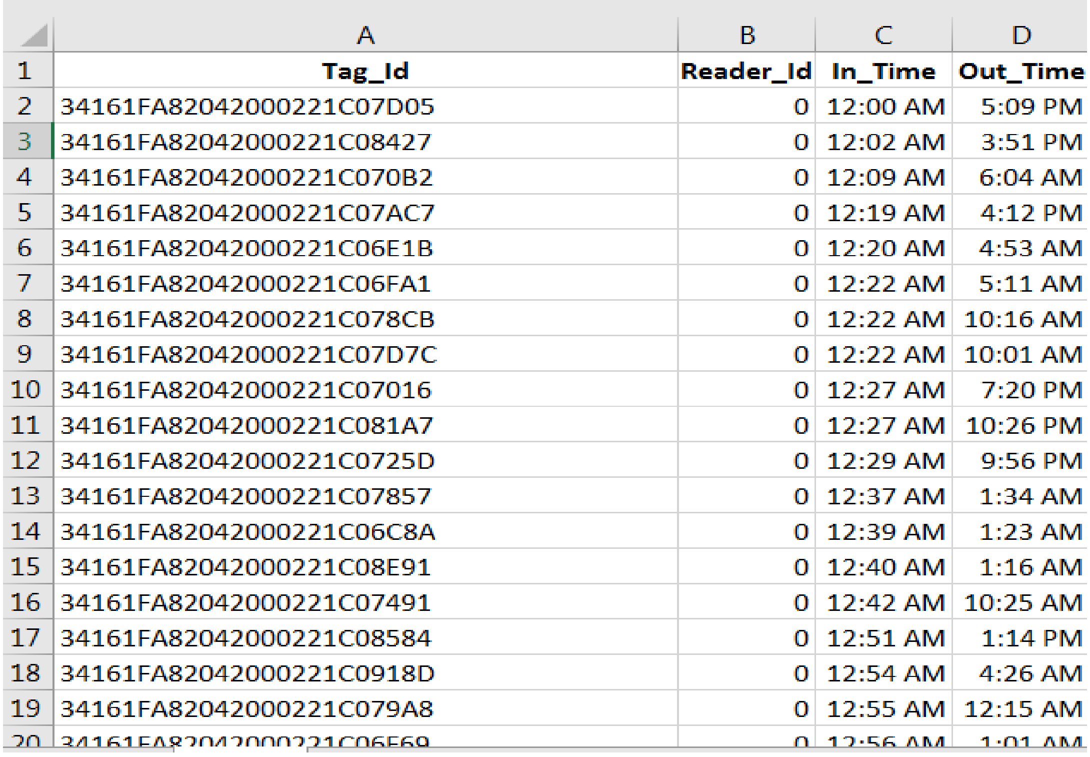
Figure 3.2 is a depiction of the WSTD per-tag-cleaning algorithm for removing false positives and false negatives while reading of tags by the readers in RFID system. The duplicate readings are handled by using hash table and saving only unique tag ids from the particular epoch. Use of hash tables gives faster indexing and searching of the existing stored values.

The above algorithm helps in setting the time window and also average number of samples of tags for various epochs and window frames can be observed.

3.5 EVALUATION RESULTS USING WINDOW SUB-RANGE DETECTION

In our experiments, we used a maximum read rate of 95% which is generally the read rate within the strong in-field region. The detection range for maximum values is 4.5 m and the percentage of strong in-field and the distance between reader and the tag is varied. Samples of 200 tags are taken as input and total number of read cycles or epochs are taken as 1000 read cycles. Default minimum size is taken as three as suggested by Massawe et al. [43] as it's an optimized size for maintaining the balance between reducing the false positive errors and the smoothing effect of the algorithm.

The data in the raw format after epoch time converted to unix time format looks as shown in Figure 3.3.



	A	B	C	D
1	Tag_Id	Reader_Id	In_Time	Out_Time
2	34161FA82042000221C07D05	0	12:00 AM	5:09 PM
3	34161FA82042000221C08427	0	12:02 AM	3:51 PM
4	34161FA82042000221C070B2	0	12:09 AM	6:04 AM
5	34161FA82042000221C07AC7	0	12:19 AM	4:12 PM
6	34161FA82042000221C06E1B	0	12:20 AM	4:53 AM
7	34161FA82042000221C06FA1	0	12:22 AM	5:11 AM
8	34161FA82042000221C078CB	0	12:22 AM	10:16 AM
9	34161FA82042000221C07D7C	0	12:22 AM	10:01 AM
10	34161FA82042000221C07016	0	12:27 AM	7:20 PM
11	34161FA82042000221C081A7	0	12:27 AM	10:26 PM
12	34161FA82042000221C0725D	0	12:29 AM	9:56 PM
13	34161FA82042000221C07857	0	12:37 AM	1:34 AM
14	34161FA82042000221C06C8A	0	12:39 AM	1:23 AM
15	34161FA82042000221C08E91	0	12:40 AM	1:16 AM
16	34161FA82042000221C07491	0	12:42 AM	10:25 AM
17	34161FA82042000221C08584	0	12:51 AM	1:14 PM
18	34161FA82042000221C0918D	0	12:54 AM	4:26 AM
19	34161FA82042000221C079A8	0	12:55 AM	12:15 AM
20	34161FA82042000221C06F69	0	12:56 AM	1:01 AM

Figure 3.3 RFID Data Schema

The algorithm is applied to remove false negative errors. As false negative reading errors occurs when tag is not detected by the reader but present in the database. By reading RFID data set, out of 75 sample readings, 14 errors were found as shown in Figure 3.4 RFID Data Containing False Negative Errors are as shown in Figure 3.4. Here X-axis denotes the tag-ids and Y-axis denotes the epoch time in form of timestamp.

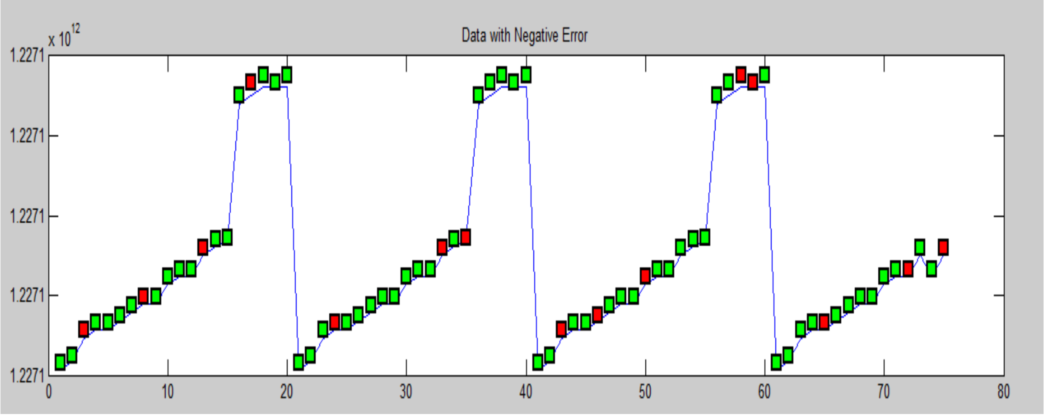


Figure 3.4 RFID Data Containing False Negative Errors

After applying the WSTD algorithm there is a vast reduction in errors by an approximation of 70% as compared to existing approach, as shown in Figure 3.5 below.

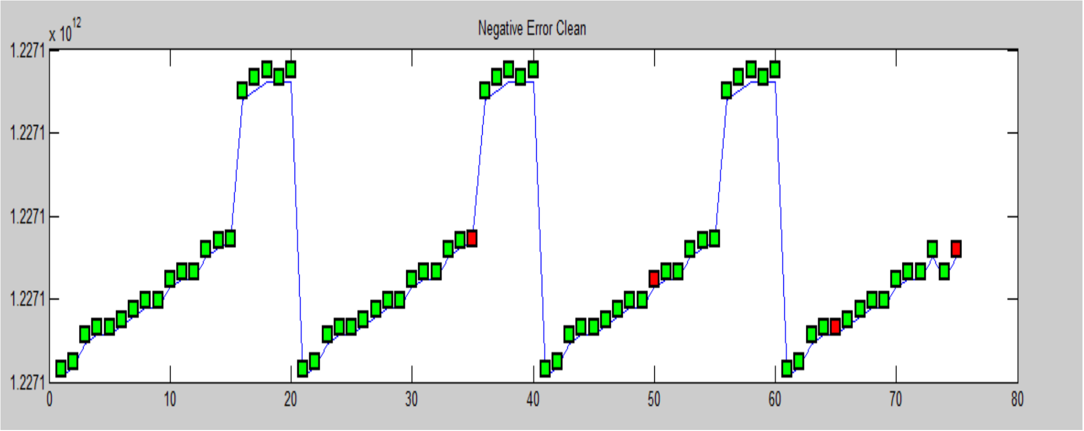


Figure 3.5 RFID Data After Removing False Negative Errors

False positive readings are also removed. The proposed algorithm is also applied to remove false positive errors. As false positive reading errors occur when tag is detected by the reader but not present in the database. By reading RFID data set, out of 95 sample readings 21 errors were found as shown in Figure 3.6 below.

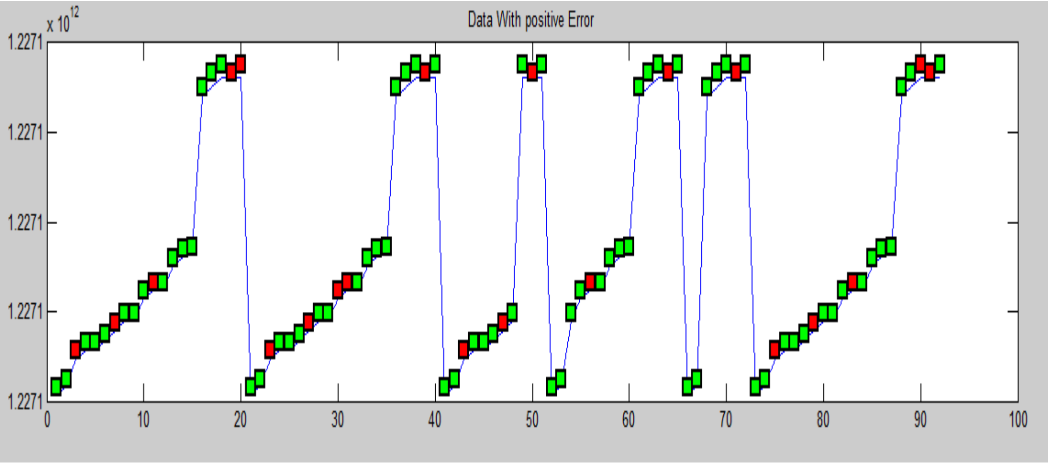


Figure 3.6 RFID Data Containing False Positive Error

After applying the proposed algorithm there is also a vast reduction in errors in false negative readings by an approximation of 70% compared to existing approach as shown in Figure 3.7

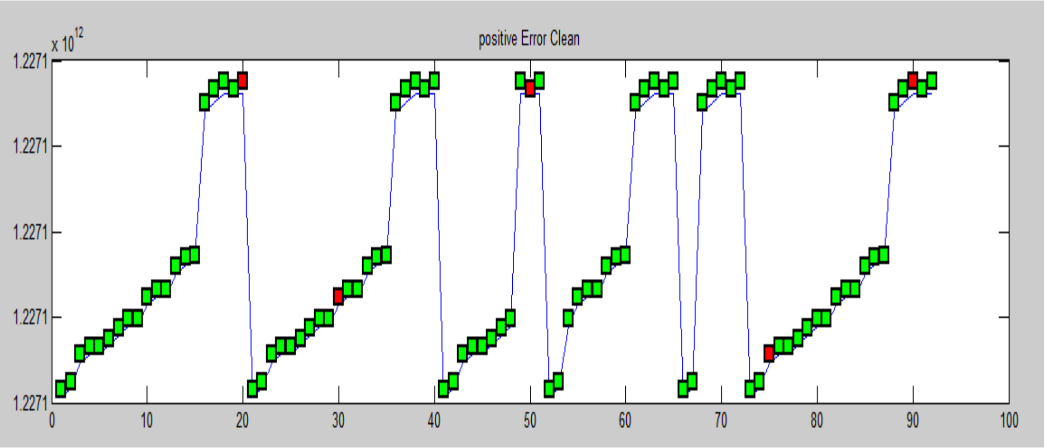


Figure 3.7 RFID Data After Removing False Positive Errors

Average number of errors are calculated on the basis of total number of false positives and false negatives. So, total number of true positives and true negatives give correct readings. Size of the window changes to different sizes according to tag and reader behavior. It ensures better capturing of false negatives and false positives and thus better cleaning of RFID data. Figure 3.8 shows the change in window size with number of epochs. We may conclude that after half the number of read cycles, window size grows then contracts for adjusting the number of false positives as well as negatives.

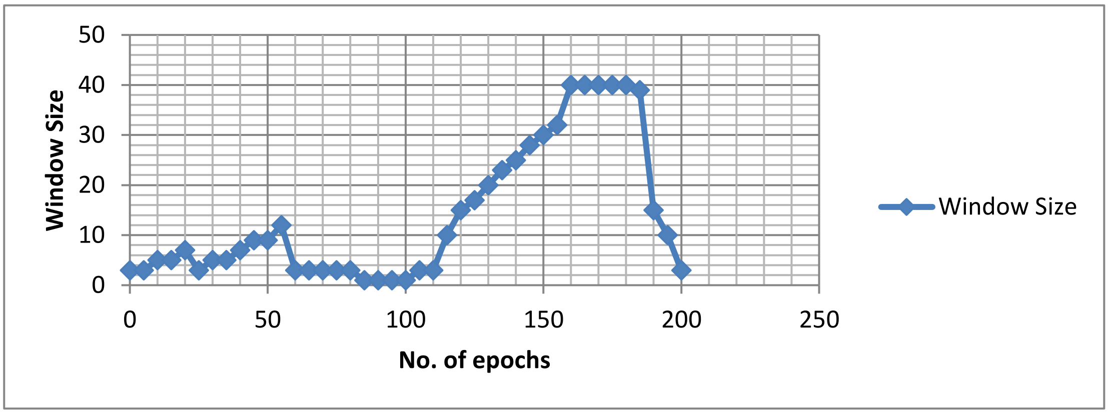


Figure 3.8 Window Size Vs Number of Epochs

3.6 CONCLUSION

Unreliability of RFID data streams can be managed by effectively handling data cleaning to reduce errors like false positives or false negatives. Adaptive sliding window-based approach called WSTD can efficiently cope up with environment variation and dynamics of the tag by reducing the percentage of errors to up to 70% in our RFID data which is reasonably good for the uncertain data like RFID data. Reader’s reading range, reading frequency, speed of tag movement are the factors which can affect reader’s reading accurately. Due to adaptive window size, the optimisation of readings is done and ensures least number of false positives or false negatives generated while reading tags by reader.

CHAPTER 4: OUTLIER DETECTION OF RFID DATASETS USING TRAJODBSCAN

RFID enabled supply-chain process is the research domain to study and implement the outlier detection points in the process. As discussed in detail, in the previous chapters, supply-chain process is one of the areas of research especially when the whole process is automated through Radio Frequency Identification (RFID) method with readers to read the tags on objects automatically and thus generating too much of raw data in the form of Tag_ID (EPC of tag), Reader_ID (here Reader Id is mapped to location coordinates) and timestamp(It is epoch time which is converted to datetime format) of read. Mining such data is a challenging problem. For each product or group of products, the schedule is planned by the stakeholders that include suppliers, manufacturers, distributors and retailers. From suppliers to retailers many routes in the supply chain can be generated. These routes are known as trajectories due to element of time and location read by readers.

4.1 INTRODUCTION

The main aim of research is to find outlier trajectories after the pre-processing of the data for reduction of false positives, false negatives and redundant raw RFID data, by clustering the scheduled or planned trajectories of the objects with the current trajectory. Any deviation or abnormal path/trajectory will be the outlier one.

A novel density-based approach with the combination of dynamic time warping distance measure along with DBSCAN [46] [83] based approach is proposed. A comparative study of different similarity measures for the trajectory/path data [48] [67] points is done followed by comparison of our technique with other clustering algorithms modified for trajectory data. The accuracy achieved by the novel outlier detection technique known as TRAJODBSCAN is found far better than other clustering methods.

4.2 RFID DATA MODEL

The raw data generated by the RFID tags and readers is cleaned and then converted into meaningful information. In raw form, the data are stored as (EPC, Location, Timestamp). Here EPC is the unique electronic product code of the tag, Location refers to location of the reader and Timestamp refers to the time of read of tagged object by the reader. In RFID-enabled system, the object is read multiple times when it is in the vicinity of readers, so data compression is must which can be done by simply converting the raw data into the form (T*ag_Id*, *Reader_Id*, *In_Time*, *Out_Time*). After the pre-processing of data transferred in the systems attached with the readers, the data values at different nodes of a trajectory or path is transformed to the trajectory data which is sequence in path in form of: [(location$_1$, time$_1$), (location$_2$, time$_2$… (location$_n$, time$_n$)], where n is length of the path. Each node in the path has spatio-temporal feature. Each location of reading is defined in terms of latitude and longitude converted to Cartesian coordinates using haversine formula for further analysis.

The data to be processed for further analysis Figure 4.1 is mapped according to the schema diagram as shown in. This schema diagram is designed according to the characteristic of RFID data and its mapping with the required and available fields in the supply-chain domain specifically. Traditional ER diagram won’t fit into it due the spatio-temporal factor associated with the data.

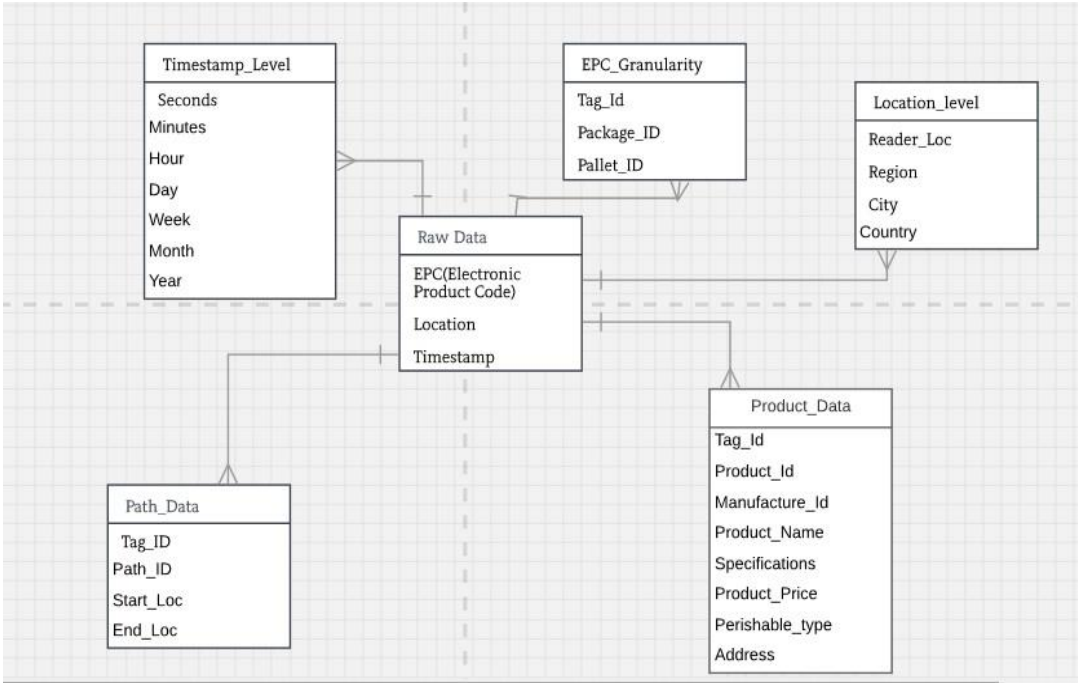
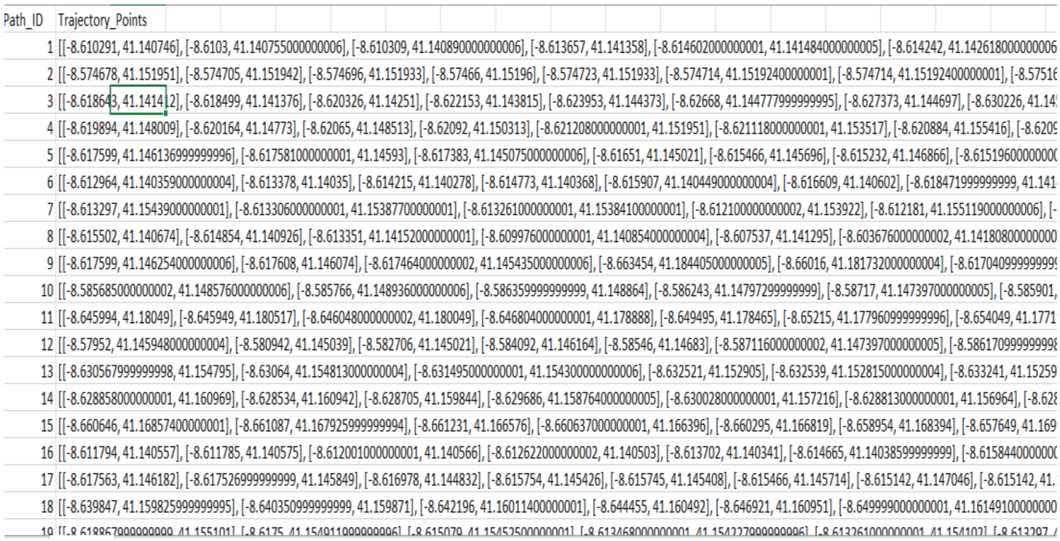


Figure 4.1 Supply-Chain RFID Data Schema

The above schema diagram can explain the information that can be mapped with the basic information of raw data which just contains the tag information and its time and location of read. This information can be linked with multiple tables with product information details, location hierarchy levels with the level of details from exact location point in form of latitude and longitude to the aggregated form like, region, city, state or country depending upon the level of information we need about the location. Similarly, granularity level of product is also stored in EPC hierarchy table i.e. whether the product read is individual or in some pallet (containing multiple numbers of product items). Path database gives the information about nodes of particular path ID assigned to the trajectories for tagged objects.

The schemas for RFID datasets in supply chain include planned path/trajectory as well as generated path/trajectory. Grouping it by each individual tag id and sorting in increasing order of time a sequence of paths called trajectories is formed as shown in Figure 4.2. Each trajectory is a sequence of Cartesian points in order of the time of the read. For each Tag, planned trajectories/paths are already stored for comparing the valid or outlier path.



Path_ID	Trajectory_Points
1	[[-8.610291, 41.140746], [-8.6103, 41.140755000000006], [-8.610309, 41.140890000000006], [-8.613657, 41.141358], [-8.614602000000001, 41.141484000000005], [-8.614242, 41.142618000000006
2	[[-8.574678, 41.151951], [-8.574705, 41.151942], [-8.574696, 41.151933], [-8.57466, 41.15196], [-8.574723, 41.151933], [-8.574714, 41.15192400000001], [-8.574714, 41.15192400000001], [-8.5751
3	[[-8.618643, 41.141412], [-8.618499, 41.141376], [-8.620326, 41.14251], [-8.622153, 41.143815], [-8.623953, 41.144373], [-8.62668, 41.144777999999995], [-8.627373, 41.144697], [-8.630226, 41.14
4	[[-8.619894, 41.148009], [-8.620164, 41.14773], [-8.62065, 41.148513], [-8.62092, 41.150313], [-8.621208000000001, 41.151951], [-8.621118000000001, 41.153517], [-8.620884, 41.155416], [-8.620
5	[[-8.617599, 41.146136999999996], [-8.617581000000001, 41.14593], [-8.617383, 41.145075000000006], [-8.61651, 41.145021], [-8.615466, 41.145696], [-8.615232, 41.146866], [-8.6151960000000
6	[[-8.612964, 41.140359000000004], [-8.613378, 41.14035], [-8.614215, 41.140278], [-8.614773, 41.140368], [-8.615907, 41.140449000000004], [-8.616609, 41.140602], [-8.618471999999999, 41.141
7	[[-8.613297, 41.15439000000001], [-8.613306000000001, 41.15387700000001], [-8.613261000000001, 41.15384100000001], [-8.612100000000002, 41.153922], [-8.612181, 41.155119000000006], [-
8	[[-8.615502, 41.140674], [-8.614854, 41.140926], [-8.613351, 41.14152000000001], [-8.609976000000001, 41.140854000000004], [-8.607537, 41.141295], [-8.603676000000002, 41.14180800000000
9	[[-8.617599, 41.146254000000006], [-8.617608, 41.146074], [-8.617464000000002, 41.145435000000006], [-8.663454, 41.184405000000005], [-8.66016, 41.181732000000004], [-8.617040999999999
10	[[-8.585685000000002, 41.148576000000006], [-8.585766, 41.148936000000006], [-8.586359999999999, 41.148864], [-8.586243, 41.14797299999999], [-8.58717, 41.147397000000005], [-8.585901,
11	[[-8.645994, 41.18049], [-8.645949, 41.180517], [-8.646048000000002, 41.180049], [-8.646804000000001, 41.178888], [-8.649495, 41.178465], [-8.65215, 41.177960999999996], [-8.654049, 41.1771
12	[[-8.57952, 41.145948000000004], [-8.580942, 41.145039], [-8.582706, 41.145021], [-8.584092, 41.146164], [-8.58546, 41.14683], [-8.587116000000002, 41.147397000000005], [-8.586170999999998
13	[[-8.630567999999998, 41.154795], [-8.63064, 41.154813000000004], [-8.631495000000001, 41.154300000000006], [-8.632521, 41.152905], [-8.632539, 41.152815000000004], [-8.633241, 41.15259
14	[[-8.628858000000001, 41.160969], [-8.628534, 41.160942], [-8.628705, 41.159844], [-8.629686, 41.158764000000005], [-8.630028000000001, 41.157216], [-8.628813000000001, 41.156964], [-8.628
15	[[-8.660646, 41.16857400000001], [-8.661087, 41.167925999999994], [-8.661231, 41.166576], [-8.660637000000001, 41.166396], [-8.660295, 41.166819], [-8.658954, 41.168394], [-8.657649, 41.169
16	[[-8.611794, 41.140557], [-8.611785, 41.140575], [-8.612001000000001, 41.140566], [-8.612622000000002, 41.140503], [-8.613702, 41.140341], [-8.614665, 41.140385999999999], [-8.6158440000000
17	[[-8.617563, 41.146182], [-8.617526999999999, 41.145849], [-8.616978, 41.144832], [-8.615754, 41.145426], [-8.615745, 41.145408], [-8.615466, 41.145714], [-8.615142, 41.147046], [-8.615142, 41.
18	[[-8.639847, 41.159825999999995], [-8.640350999999999, 41.159871], [-8.642196, 41.16011400000001], [-8.644455, 41.160492], [-8.646921, 41.160951], [-8.649999000000001, 41.16149100000000
19	[[-8.618867999999999, 41.155101], [-8.6175, 41.154911999999996], [-8.615079, 41.15452500000001], [-8.613468000000001, 41.154227999999996], [-8.613261000000001, 41.154102], [-8.613297, 4

Figure 4.2 Path Database

Due to the non-availability of historical data for outlier points in the supply-chain path, clustering technique can be used instead of classification to segment the points as outlier point or non-outlier points in a supply-chain path. Here, cluster of planned trajectories is taken and density-based approach is used as base technique to cluster normal trajectories and detecting outlier trajectories. Figure 4.3. illustrates two trajectories. Here x-axis represents discrete points in the trajectories and y-axis represents location points at those points or nodes.

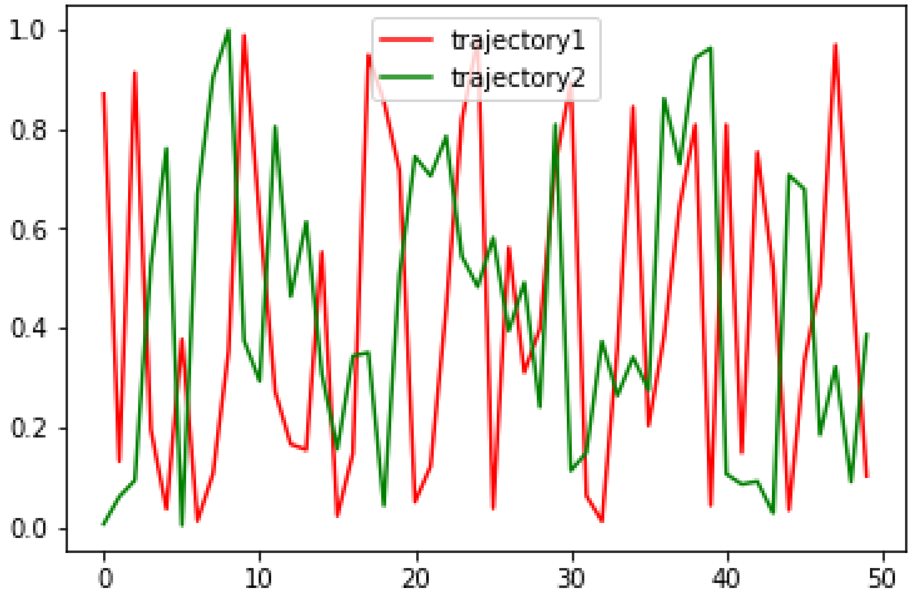


Figure 4.3 Trajectories with 50 points

Trajectories not belonging or similar to any clusters are outlier trajectories or invalid paths.

4.3 OUTLIER DETECTION FRAMEWORK

Outlier detection of the trajectory needs to be treated differently unlike traditional approach where point to point similarity or dissimilarity is measured. Not much of the work is done for the outlier detection specific to supply-chain RFID data, though increased usage of RFID in supply-chain application is quite prevalent. Interesting research has been conducted on warehousing RFID data sets [21]. Moving objects can be handled with compression in data and preservation of path structure with a novel aggregation method. It is quite a challenging task because the data generated with reading of moving object is quite massive and it has spatio-temporal properties too. FlowGraph [23] is the method proposed by Hector et al. It is a probabilistic model which captures the main trends and exceptions in moving object data and FlowCube [22] is a multidimensional extension of FlowGraph and its biggest advantage is adaptive fastest path technique which does real time computation of routes based on patterns of driving. If moving objects are studied for their movements with automatic identification then one of the applications could be tracking of any suspicious movement. J G Lee *et al*. proposed partition and detect framework for finding outliers in trajectories/paths [26]. They used mathematical calculations based on minimum description length (MDL) to find out the dissimilarities between various trajectories by first splitting the trajectories into parts or line segments and then detecting the abnormal or outlier line segments. In this approach however point to point similarity or dissimilarity goes unnoticed and may give inaccurate results.

Clustering approach is based on the similarity or differences between objects as there is no reference data to be compared with. The data used in our research is trajectory data. Distance/similarity measures used for comparing trajectories are normally different from point to point distance/similarity measure. Euclidian distance [84] is most commonly used distance measure. Bayesian approach [85] is also used to find outliers in data collected by a wireless sensor network. Many measures of similarity have been discussed by many people. Each of the similarity measure has its own pros and cons. Euclidian distance [47] , dynamic time warping (DTW) [53], longest common sub sequence (LCSS) [52], edit distance [50], Fréchet distance [51], Hausdorff distance [49], are the most popular ones [48] [86]. Edit distance and LCSS are the non-metric ones and stress more upon the shape matching but do not give time to time details. Edit distance on real data (EDR) is also a good measure against real sequences with noisy data. It is based on the number of edits required to match one trajectory to another to calculate the similarity or distance between trajectories. This measure is also more efficient when approximate similarity is to be calculated. Hausdorff and Fréchet distance measures are used to calculate the geometrical similarity of the trajectory. With time shift, this distance measure behaves well but it is better used in the cases where only shape similarity should be checked. It cannot give exact time correspondence. Trajectory clustering based on partitioning of trajectories into line segments and detecting the similarity among them is an efficient approach. Here, the similarity measure used for the line segments focuses only on geometrical similarity, and does not take the temporal component into account. This similarity measure in a clustering method can be more appropriate for shapes that are polygonal. For supply-chain process trajectory clustering, it may not be justified. Dynamic Time Warping (DTW) calculates point to point matching with flexibility of time shift due to the difference in acceleration of objects read in the trajectory path. It is used in many suitable domains [87].This distance measure appears most suitable for the domain area being studied.

The primary aim of the research is to propose a solution for clustering of trajectories. Most of the work done in clustering of the trajectories covers either the temporal aspect or structural aspect for similarity. Trajectory clustering and similarity measures have been studied in different areas of applications like web clustering, outdoor surveillance but none of the work is related to RFID-enabled supply-chain process as per the knowledge of the authors. Depending on the applications, different similarity measures are used. For clustering, different categories of algorithms can be used like partitioning-based, hierarchical-based, density-based, grid-based and model-based algorithms [44] Outlier detection can be considered as an outcome of clustering. It can be the data unit which does not belong to any of the clusters formed. This data unit can be a point or sequence of points giving any geometrical shape like a trajectory, which diverts the topic to trajectory clustering finally leading to the finding of trajectory outliers.

Trajectory outliers or anomalies [11] can be the trajectories or part of trajectories, which are significantly different or abnormal when compared for similarity with other trajectories or part of trajectories. Events that do not follow the pattern (e.g. unexpected path taken by a tagged object or abnormal stay of any object or set of objects at one place) lead to the formation of abnormal or outlier trajectories. A complete survey on outliers or anomaly detection methods [14] used for different types of domain or data types can be found in literature.

The complete framework of Outlier Detection system is designed with three layers as discussed in the previous chapters. It gives the complete solution including pre-processing, clustering of trajectory outliers and classification of outlier points and thus predicting the outlier nodes in the supply chain Framework of Outlier Detection System is as explained in Figure 4.4. The input data is taken as pre-processed data which is in the form of Tag_Id, time when the tag comes in the range of reader, time when the tag moves out of the range of reader and the location of the reader. This data is not in the form of trajectory, so it is mapped in trajectory or path form for each object. The trajectories are input to the proposed algorithm TRAJODBSCAN for finding outlier points in the trajectories. Information about outlier trajectory or point in a trajectory is given to the stakeholders in supply chain for further action. Here trajectories are partitioned into batches and then checked for outlier trajectories and hence the outlier nodes in the trajectories.

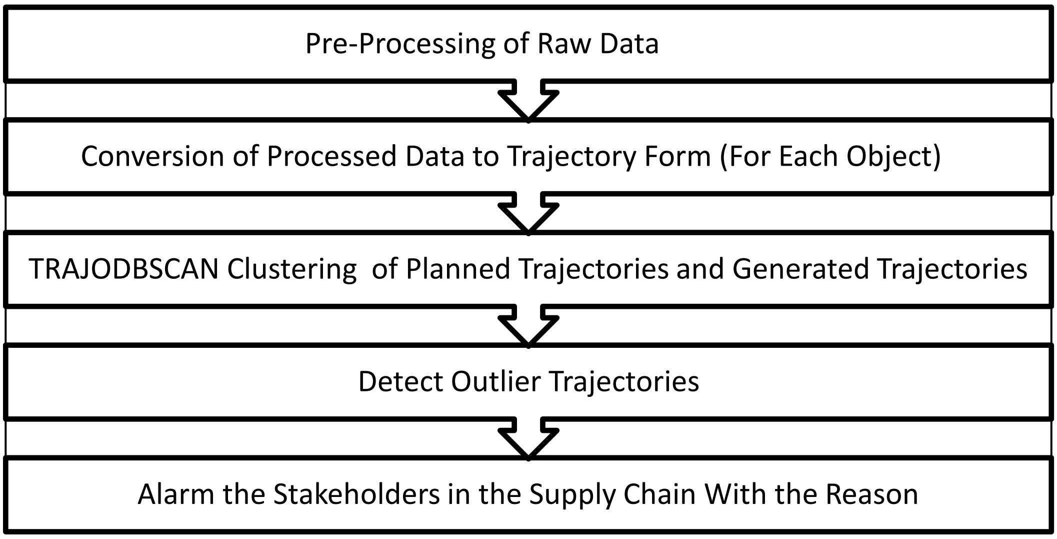

Pre-Processing of Raw Data
Conversion of Processed Data to Trajectory Form (For Each Object)
TRAJODBSCAN Clustering of Planned Trajectories and Generated Trajectories
Detect Outlier Trajectories
Alarm the Stakeholders in the Supply Chain With the Reason

Figure 4.4 Framework of Outlier Detection System

4.4 OUTLIERS IN SUPPLY-CHAIN PROCESS

Outliers in supply chain refer to any possible deviation from the normal supply-chain process especially in the automated RFID-based. Possible outliers in supply-chain process can be majorly due to delay in transport (due to path deviation or problem in the transport), theft of goods on the way to particular path and fake goods.

Detection of the outliers can be done by various ways and each technique has its own constraints. Flow analysis of events to detect the events that are not in normal flow in the chain. The classification technique is used in case the class labels with outlier/no outlier are known for training dataset. Outliers are rare events, so classification approach is not very efficient if sufficient data for training the classification model.

4.5 COMPARISON OF DIFFERENT DISTANCE MEASURES FOR TRAJECTORY DATA

In this section different distance measure methods are studied to find out the differences and the similarities among them so that the most effective method could be selected. It may be noted that the trajectory data or individual trajectories are referred as path data in the supply chain. Particular locations in path sequences are periodically sampled as a finite sequence of time-stamped locations. Before proceeding for the comparative analysis, one should know about the most commonly used distance/similarity measures. Generally, for RFID type of spatio-temporal data distance measures like dynamic time warping, longest common sub sequence and edit distance-based measures are most popular ones [48].

For the data where spatial factor is dominating with discrete points of specification then Lp-norm based point to point distance measure is the most popular one. If the data is spatio-temporal then for discrete points paths, dynamic time warping, longest common sub sequence and edit distance with real time, Hausdorff and Fréchet are used. Each of them has their own pros and cons depending on their applications. In the coming sub section, we will discuss each of them in detail and will then show the comparative study of all the measures with our trajectory data.

4.5.1 Euclidian Distance

Euclidean distance or L2 norm [84] [88] can be used as similarity or distance measure for many types of applications. This distance measure is free from defining any parameter and is quite easy to implement. It can manage the large size trajectory data due to its linear complexity. The values in the objects with the same time instance are compared. If there is any time offset between the objects to be compared then this approach is not efficient. While calculating the distance between trajectories, one to one mapping between various points or nodes in the trajectories is done. It's useful only when trajectories are of equal length. If the trajectories are similar but the sampling rates are different, Euclidian distances will not calculate them as similar. This distance measure is very sensitive to noise. Euclidian distances can perform well only in case of straight-line distances. Euclidian distance between each mapping points in the compared trajectories is defined as shown in equation 4.1. Let us say p (x_1, y_1) and q (x_2, y_2) are two points with location coordinates representation then the distance d (p, q) is as follows:

$$\boldsymbol{d(p,q) = \sqrt{(x2 - x1)^2 - (y2 - y1)^2}} \tag{4.1}$$

4.5.2 Longest Common Sub-Sequence

Longest common subsequence (LCSS) [89] is one of most popular measure for checking similarity between sequences. Its first application was to find out the similarity in strings by comparing each and every character of strings. Li et. al [90] applied LCSS measure for similarity measure in trajectory data. A threshold parameter is used to define the number of matching characters in strings. The basic idea of LCSS is that it allows some sample points unmatched to match some sequences in trajectories. LCSS is good for processing with low-quality trajectory data (i.e. noisy trajectory data). It is useful for the detection of similar trajectories with different sampling rates. It uses dynamic programming approach. Here matching can be defined by a distance threshold. So, selecting threshold is again an issue. Due to mentioned reasons it’s good with noise and outliers but if one need to find out outliers then it is an inefficient method as it ignores the parts that don’t match.

The optimal substructure of the LCSS problem gives the recursive formula as equation 4.2. If we have two sequences a and b and LCSS (i, j) is the distance of the LCSS of a (1...i) with b (1...j). The distance measure can be as explained below in the equation 4.2:

$$\boldsymbol{LCSS(i,j)} = \begin{cases} \mathbf{0} \qquad \boldsymbol{if\ i = 0\ or\ j = 0} \\ \mathbf{1} + \boldsymbol{LCSS[i-1, j-1]\ if\ a_i = b_j} \\ \mathbf{max}(\boldsymbol{LCSS[i-1, j], LCSS[i, j-1]} \end{cases} \tag{4.2}$$

The equation 4.2 above describes the distance between two sequences of lengths i and j respectively and if either of the lengths is zero then the distance is calculated as zero. If the last points of the sequences match i.e. they are equal then distance is defined by the recursive definition of LCSS as 1+recursive LCSS distance function for length of the sequences up to i-1 and j-1.If last points of the two sequences do not match then LCSS distance is defined as maximum of distance between (i-1) points of one sequence to j points in another sequence or i points of one sequence with j-1 points of another sequence.

4.5.3 Edit Distance

Edit Distance on Real sequence is adapted from Edit Distance on strings. The distance is calculated based on the number of insertions, deletions and replacements needed to convert one object points to another so that they become similar. It is also threshold-based equality relationship. Two locations are regarded as equal if they are close to each other with respect to a threshold. Resulting value of the distance defines the number of operations required for converting t. Here distance value means the number of operations, not distance between locations. It is not sensitive to noise too. If LCSS and Edit Distance on real sequence are compared then they both are based on counting the similarities or dissimilarities. LCSS counts the number of pairs of points that match and EDR counts the cost of operations required to fix the pairs which are not matching. They are inversely proportional to each other. Equation 4.3 gives the summary.

$$\boldsymbol{d(i,j)} = \mathbf{min}\begin{cases} \boldsymbol{d(i-1,j)+1} \\ \boldsymbol{d(i,j-1)+1} \\ \boldsymbol{d(i-1,j-1)+\delta(x(i-1),y(j-1))} \end{cases} \qquad \mathbf{(4.3)}$$

Here d (i, j) represents the distance between two points on two sequences respectively. If the lengths of the two sequences are m and n, then a two-dimensional matrix d [0…m,0…n] is used to keep edit distance values. Here δ(x(i-1), y(j-1)) is either 0 or 1 depending upon the condition whether the current points i and j are same position or not. The value of d (,) is calculated row by row. The value of current row i.e., d (i,) is dependent upon the value of previous row or we can say previous point i.e. d(i-1,). The time complexity of the algorithm for this measure is O(m*n) where m and n are the lengths of the sequences respectively. The drawback of Edit Distance is that it does not take into account direction of movement which makes it not useful in many applications where direction of movement in trajectories are very important and it can be handled by Fréchet distance measure discussed in the next sub-section.

4.5.4 Fréchet Distance

Fréchet distance is also a distance measure used for measuring the distance between two curves. It takes into consideration the location as well as order of sequence points in trajectories. It is a shape-based similarity. Here each pair of different points are separated from matching points by a disjoint open set. In this approach the distance is not measured point to point as in case of previously discussed distance measures. It rather follows many to many correspondences. It is also known as “Dog Leash” distance due to the analogy with minimum length of a rope required to connect owner with his dog. Their movements are constrained to two separate paths with the possibility of movements with different velocities but without backtracking. So, the distance is minimum length of the rope or leash which is sufficient enough to join a point moving in forward direction along one path say p and one moving in forward direction along q. The rate of movement may not be uniform for either of the curve points. It can be solved by recursion concept using dynamic programming.

$$\boldsymbol{d_{frechet}(i,j) = \max\begin{cases} \min(d(i-1,j), d(i-1,j-1), d(1,j-1)) \\ d(i,j) \end{cases}} \tag{4.4}$$

Here d (i, j) is the Euclidian distance between any two points in the curves or trajectories taken parallelly to record the distance between curves on the ith and jth points respectively.

4.5.5 Hausdorff Distance

The Hausdorff distance [91] is a based-on metric space. It measures the distance between two sets of metric spaces. Here each pair of different points are separated from matching points by a disjoint open set. The distance measure is not done by point to point or one to one mapping, here both are trajectories to be compared are taken two different set of points for each path. Now the minimum distance between two sets for every point in one set to any other point in second set is calculated and maximum of all these calculated distances is known as Hausdorff distance. Using Hausdorff distance we can calculate how different are points between two paths and therefore how different are two trajectories or path from each other. The calculations done are quite resource intensive.

This distance is different from some of the previously discussed measures where, rather than one to one mapping between points in two trajectories, one to many or many to many mappings is done to measure the distances. To compute HAUSDIST (A, B), both of these methods consider the trajectory B as a single object and traverses the index of A to identify the point a in A that yields as shown in equation 4.5.

$$\mathbf{HAUSDIST(A,B) = MAX\{MINDIST(a,B): a \in A\}} \qquad \textbf{(4.5)}$$

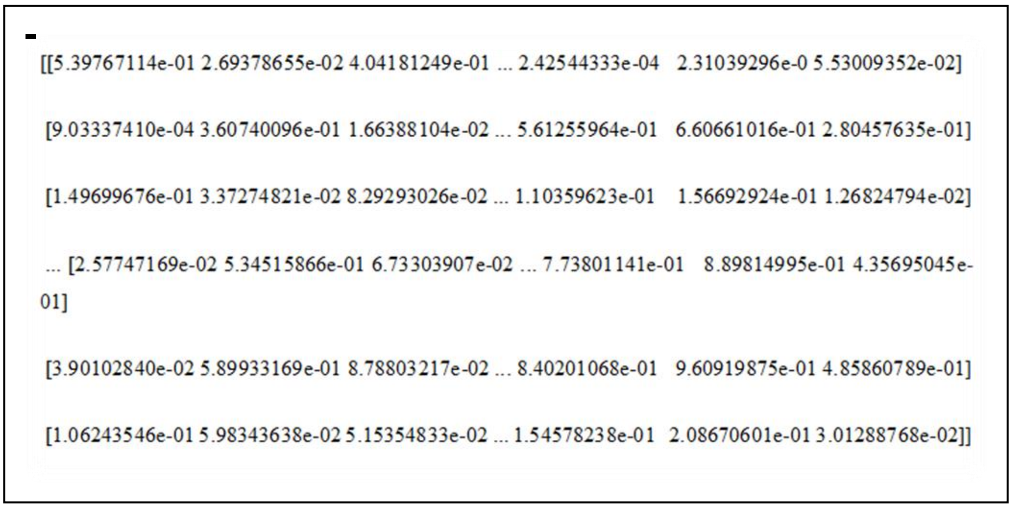
In the equation 4.5 above the two sets of points for path A and B are compared. The disadvantage of Hausdorff distance is that even if the shapes of two trajectories or paths does not look very similar, still they may give small Hausdorff distance.

4.5.6 Dynamic Time Warping

Dynamic Time Warping (DTW) is a well-known algorithm for finding similar trajectory patterns between two trajectories. The distance calculated is based on Euclidian distance only but the process in optimised for paths or trajectories followed by objects with varying speed. Pairwise comparison of two sequences is done unlike the way it is done using Hausdorff method. The sequences of observations are represented in form of grid where rows and columns represent points of each path/trajectory and each cell of the grid has the distance value between the combination of i^{th} point of one trajectory with j^{th} point of another trajectory. This way all the possible point combinations between two trajectories are available and the best match between two paths is the one that can minimize the total distance between them. The biggest advantage of dynamic time warping is finding the distance between two different trajectories which are of different lengths and it is suitable for both spatial as well as spatio temporal paths. Other than this, no extra parameter is required like it is in case of EDR and LCSS where threshold distance to conclude for similarity or dissimilarity is required. If one slow-moving object and another fast-moving object follow same path, the similarity of the paths followed can be very well observed by DTW measure as it considers the time lag correctly. Myers *et al*. introduced this measure first time for computing distance between time series data (Myers *et al.* 1980). Later Kruskal *et al*. used this measure for finding distance between straight paths. (Kruskal 1983, Soong & Rosenberg 1988, Picton *et al.* 1988, Ostendorf & Roukos 1989). The base DTW is complex and thus takes more time to process but this can be very well managed by using indexing scheme in which similar sequences' retrieval can be done by taking only subset of sequences as many points in the sequence are of no use in computation and thus, they are pruned. It gives accurate measure compared to others especially when there is a need to detect similar trajectories with different sampling rates. The coming sub section will discuss the steps of DTW in detail.

Before calculating the distance between various trajectories, the data is input in the form of distance matrix with each x and y axis showing trajectory 1 and trajectory 2 coordinates. They can be of same lengths or different lengths and depending on that the appropriate distance measure can be taken. The distance measure methods are implemented in python language.

The snapshot in Figure 4.5 shows the format of the input data matrix with each value in the matrix representing the distance of any one point in trajectory 1 to distance of some point in trajectory 2.



```
[[5.39767114e-01 2.69378655e-02 4.04181249e-01 ... 2.42544333e-04   2.31039296e-0 5.53009352e-02]

 [9.03337410e-04 3.60740096e-01 1.66388104e-02 ... 5.61255964e-01   6.60661016e-01 2.80457635e-01]

 [1.49699676e-01 3.37274821e-02 8.29293026e-02 ... 1.10359623e-01   1.56692924e-01 1.26824794e-02]

 ... [2.57747169e-02 5.34515866e-01 6.73303907e-02 ... 7.73801141e-01   8.89814995e-01 4.35695045e-
01]

 [3.90102840e-02 5.89933169e-01 8.78803217e-02 ... 8.40201068e-01   9.60919875e-01 4.85860789e-01]

 [1.06243546e-01 5.98343638e-02 5.15354833e-02 ... 1.54578238e-01  2.08670601e-01 3.01288768e-02]]
```

Figure 4.5 Distance Between Points in Trajectories in Matrix Form

As already discussed, the various distance measures and their pros and cons and the kind of data they are more suitably used, Table 4.1 gives the comparison study results between various distance measures methods used for finding distances between two trajectories based on many parameters. In the Table 4.1, different trajectory similarity measures are reviewed to find the most similar trajectories to the query. The implementation was done using Python language environment. Here the comparison is done to find the similarity between trajectories with on an average 100 points of location in the path. Depending on the type of distance measure used, each trajectory can give different similarity or distance calculation. Each type has its own complexity and parameter requirements. Some measures are good in terms of accuracy and some are better in terms of the time and memory taken. It all depends on the application area for which we need to calculate the distance. Some commonly used parameters like the behaviour of the measure when some noise is added in the input, lengths of the paths (whether they are similar or not), support of time shift i.e. the effect on the accuracy of measure when two trajectories or paths are similar in shape but may lag in time. All these observations are summarised for the different similarity measures in Table 4.1 below:

Table 4.1 Comparison of Distance Measures

Parameters	**Euclidian**	**Fréchet**	**DTW**	**Edit time**	**LCSS**	**Hausdorff**
Add noise	Sensitive	Sensitive	Robust	Sensitive	Robust	Sensitive
Length of two trajectories	Equal lengths	Different lengths	Different lengths	Different lengths	Different lengths	Different lengths
Support time shifting	No	No	Yes	Yes	Yes	Yes
Computation cost. In terms of time	O(n) Where n is the no. of points in the trajectories	O(pqlog(pq)) Where p & q are no. of points in the trajectories	O(pq) Where p & q are no. of points for different trajectories	O(pq) Where p & q are no. of points for different trajectories	O(pq) Where p & q are no. of points for different trajectories	O(pq) Where p & q are no. of points for different trajectories
Time taken (ms)	**139**	**7430**	**865**	***687***	**676**	**778**

LCSS does not give very accurate results as it tends to ignore many important points in between the trajectory though it works well with noisy data. Euclidian distance reflects similarity in time and DTW reflects similarity in shape. LCSS does not take into account the unmatched points and matches only similar parts. So, it is not a good approach when we need to find the abnormal paths specifically.

DTW and Euclidian try to match each and every element so can be good for matching two or more supply-chain path as trajectories more efficiently. Hausdorff and Fréchet are also the metric measures like Euclidian but they also consider the shape similarity, not the exact temporal arrangement. The DTW method can measure distance between time series even if they vary in time or speed. It is widely used in the comparison of time series, and due to the same reason, it may be appropriate for supply chain path trajectories which are series of reader points reading tags at a particular time in a particular order.

Now considering the above statistics for distance measure, we intend to use DTW for our proposed approach of clustering the trajectories of supply chain. Here, each point of the trajectory can refer to supplier, manufacturer, distributer, retailer or consumer. There can be many distributers in between the chain and so is the case of other representations.

4.6 TRAJECTORY CLUSTERING

In trajectory clustering, not only shape but speed and direction are also important parameters as they all combine to form a trajectory. Among many clustering approaches, density-based may be the most appropriate to find outliers due to many advantages over the other clustering approaches.

Density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering approach with several advantages over other clustering approaches, the major one being the ability to find clusters of arbitrary shape. It also does not require the user to know the number of clusters to be formed beforehand. DBSCAN is very robust to outliers and requires just two parameters, minimum radius and minimum no of points within the minimum radius given.

In the proposed algorithm, DTW is used for distance measure rather than Euclidian distance as Euclidian distance cannot take into the consideration the speed of travel. Moreover, the Euclidian distance requires trajectories to be of equal lengths.

For comparing the trajectories with different lengths, and in terms of accuracy DTW gives the best results although the time taken is little more, which can be managed with better configuration systems as well as using various variations of DTW like LB Keogh [92]. It is a very appropriate method for measuring similarity between two temporal sequences which may vary in time or speed. In supply-chain process, this would be a common scenario where path followed is same in routine but speed and time may vary.

4.6.1 Basic DBSCAN Algorithm

Before giving the proposed algorithm, we are going to discuss the DBSCAN [83] algorithm. Density-based spatial clustering of applications with noise (DBSCAN) is a well-known data clustering algorithm that is commonly used in data mining and machine learning. There are many features of DBSCAN which makes it suitable for finding outliers in trajectory followed by tagged objects in supply chain. Unlike K-means, there is no need to specify the number of clusters in advance which is not a trivial task. Clusters formed are of arbitrary shapes, not fixed spheres.

DBSCAN is quite robust method when it comes to finding outliers in the clusters. It requires two parameters as input to do clustering. It works in a greedy manner by grouping points that are similar to each other in clusters and clusters with very few points are outliers. With the use of indexing schemes like R* tree [93], the performance of DBSCAN can be further enhanced. The next sub section explains the selection of parameters for the algorithm.

4.6.2 Parameter Estimation

The parameter estimation is a non-trivial task in implementation of any data mining technique. The performance of the algorithm depends on the parameters chosen and it depends a lot on previous history of their usage for any particular application.

In DBSCAN, two parameters are required as defined below:

I. **ε:** It refers to the radius around any point which is checked for the density of points around it. The minimum number of points is another parameter required as explained in the next point. In our research work, the unit is trajectory rather point, so we need to focus on the radius around which the number of trajectories need to be checked rather than points**.** The value for ε in chosen by our algorithm is using a k-distance graph by plotting the distance to the *k* = *minPts*-1 nearest neighbours starting from the maximum to the minimum value. Here k i.e. min number of trajectories is chosen according to the average of the scheduled trajectories taken by the objects in the supply chain. Too small a value for radius will affect clustering in a way that large part of data won't be clustered. Some points won't be clustered at all because they won't satisfy the density condition i.e. the number of points around it within small radius. On the other hand, if the radius value for density is taken as very high then most of the points will fall around the big radius and rather than making separate clusters, they will merge to become single cluster. The eps should be chosen based on the average distances of the dataset (as we have used k-distance graph).

II. ***MinPts*****:** MinPts are the minimum number of points around the radius which can make it a core point having enough dense values around it to be connected within a cluster. According to Sander *et al* [94]. the thumb rule says that the number of minimum points is affected by number of dimensions in the data set. It says MinPts should be at least 2 times the number of dimensions. Minimum points should be greater than number of dimensions plus one. Still it should be chosen with precautions, obviously very less value like one carries no meaning because that will create cluster for each individual point on its own. If minimum points would be taken as any value less than or equal to two then again, the result would be similar to hierarchical clustering. For the value with too much noise as well as for the data containing too many duplicate values, larger values (at least large than 3) give better results and result in more significant clusters [95].

Now after finalizing the two important parameters, distance function is the most important factor to create an impact on the accuracy of the results. The research work focused on the selection of best distance measure and considering the trajectory type of data and doing the comparative study of various distance measure relevant to the dataset used, we hypothesize DTW as the best choice. Now our proposed work is the combination of Dynamic Time Warping with DBSCAN as a novel technique and specially used first time with supply chain datasets as per our knowledge.

4.6.2 INTRODUCTION TO DYNAMIC TIME WARPING

The brief introduction to dynamic time warping (DTW) algorithm is given in previous section of distance measures. After the comparative study of all the distance measures and hypothesizing DTW as a suitable approach for the RFID enabled supply chain data, we would be discussing DTW method in detail.

The Dynamic Time Warping (DTW) distance measure is a technique that has long been first used in the domain of speech recognition. It can be used for major data mining techniques which include clustering, classification and detection of outliers as it is helpful in finding the distance between any two objects by allowing a mapping of one signal to another signal even if it is non-linear and the target is minimising the distance between the two signals. So, number of steps required to minimize the distance can help in determining the similarity of two signals.

This particular technique has shown outstanding results in terms of accuracy and now it is used in variety of problems in various domains. One of the important features is its usage in time series data where the sequence pattern may be same but the temporal factor may vary. In the applications like trajectory mining where the study of trajectories followed by moving objects or people, needs to be analysed, DTW comes up handy.

In the real-world scenario, the moving objects or people may not have consistent speed and this should be taken into consideration. Dynamic Time Warping (DTW) is an appropriate algorithm when the temporal sequences may vary in speed or have different acceleration. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension.

Figure 4.6 shows the sequence of steps for distance matrix generation with DTW.

```
/*For the trajectory1 and trajectory2 of length L1 and L2 respectively*/

L1← length of trajectory 1
L2←length of trajectory 2
/* Initial the time warping matrix: Time_warp with 0 value for 0th row and 0th column.  Here
    c matrix represents the simple pairwise distance between ith and jth element. */
Time_warp [] ← matrix of size L1 X L2
Time_warp [0,0] ←0
/*Fill the value of distance matrix after warping for all the rows of first columns as 0 +
    simple distance between ith row and 1st column*/
for i-1; i<= L1; i++ do
Time_warp[i,1]←Time_warp[i-1,1] + c[i,1]
end for
/*Fill the value of distance matric after warping for all the rows of first columns as 0 +
    simple distance between 1st row and jth column*/
for j=1; j<=L2; j++ do
Time_warp [1, j]←Time_warp [1, j-1] +c[1,j]
end for
/*Fill the value of distance matric after warping for all the rows of first columns as minimum
    of the distance between (i-1,j or I,j-1 or i-1,j-1) + simple distance between ith  row and
    jth column*/
 for i=1; i<=L1; i++ do
     for j=1; j<=L2; j++ do
     Time_warp [i, j] ← c [i, j] + min {Time_warp [i-1, j], Time_warp [i, j-1],
    Time_warp [i-1, j-1]}
     end for
 end for
return Time_warp
```

Figure 4.6 Steps to Find Distance Matrix

After the Time_warp matrix is created with all the distances, warping path can be calculated by backtracking. The detailed steps are as given in the Figure 4.7.

```
/*Backtracking from top right corner of the Time_warp matrix generated in the above section*/
Warped_path []
i = number_of_rows (Time_warp)
j = number_of_columns (Time_warp)
while (i > 1) & (j > 1) do
    if i == 1 then j = j -1
    else if j == 1 then i = i -1
    else
        /*Finding the minimum value either horizontally, vertically or diagonally*/
        if Time_warp [i-1, j] == min (Time_warp [i-1, j], Time_warp [I, j-1], Time_warp [i-1, j-1] then
            i = i - 1
        else if Time_warp (I, j-1) == min (Time_warp [i -1, j], Time-warp [i, j -], Time_warp [i -1, j-1]) then  j = j -1
        else
            i = i -1;
            j = j-1
        end if
        /*All the values of path traversed starting from Time_warp [i, j] until Time_warp [0,0] while picking up minimum values along the path give the final warped path. */
        Warped_path.add ((i, j))
    end if
end while
return path
```

Figure 4.7 Steps to Find Warping Path in DTW

4.7 PROPOSED APPROACH: TRAJODBSCAN (TRAJECTORY OUTLIERS WITH DBSCAN) ALGORITHM

TRAJODBSCAN is a novel approach to cluster the given trajectory along with the scheduled set of trajectories and trajectories which do not belong to any cluster are outlier trajectories. They are put into separate set. Rest are assigned some clusterId with each pair of trajectories compared for the similarity among them the dynamic time warping path is calculated and then with the given minimum radius and given minimum number of trajectories as parameters the number of core trajectories are observed and based on the trajectories which are density reachable and density connected the clusters of trajectories are calculated. Figure 4.8 shows the steps of TRAJODBSCAN

Algorithm

Let us, first of all, give notations being used in the algorithm:

- The $N\mathcal{E}(T_i)$ of trajectory $T_i \in T$ (set of trajectories) is defined as Trajectory space.
- $N\mathcal{E}(T_i)= \{T_j \in T|DTW (T_{i,} T_j) \leq \mathcal{E}$ (minimum radius given)$\}$
- T_i is core Trajectory w.r.t. $\mathcal{E}$ and min_trajs (minimum no. of trajectories), If $N\mathcal{E} (T_i) \geq$ min_trajs.
- Trajectory $T_i \in T$ is directly density reachable from trajectory $T_j \in T$ w.r.t $\mathcal{E}$ and min_trajs. If Ti $\in N\mathcal{E} (T_j)$.
- A $T_i \in T$ is density reachable from $T_j \in T$ w.r.t to If there is a chain of trajectories $T_{j,}$ $Tj_{—1}$, Ti such that the adjacent trajectories in between the chain are direct density reachable from each other.
- $T_i \in T$ is density connected to $T_j \in T$ w.r.t $\mathcal{E}$, if some $T_c \in T$ is there and T_i and T_j are density reachable from T_c w.r.t $\mathcal{E}$.

- **Input**

  A set of Trajectories T= $\{T_1, T_2, \ldots, Tn\}$

  I. Minimum radius parameter as $\mathcal{E}$ and minimum number of trajectories as min_trajs. $\mathcal{E}$ is chosen based on the k–distances on history data. It is based on the average slope of sorted K distances of each trajectory path of the neighbour paths of the trajectory.

  II. Min_trajs is also the average number of neighbouring trajectories based on the history of supply-chain paths followed by the objects.

  III. Initial/starting time of the read in supply-chain path and destination time of the object as t_0 and t_f, respectively

- **Output.:** A set of clusters C= {C_1, C_2, C_3…., Cn}and set of outliers cluster O.

```
Mark all the trajectories in T as unclustered and initialise clusterid=0.
 for each (Ti ∈ T) do
   if (Ti is unclustered) then
   compute   NƐ (Ti) using Dynamic time warping distance measure.
       if (|NƐ (Ti)|) >=min_trajs) then
           Assign clusterid to all Trajectories ∈ NƐ (Ti);
                  Insert NƐ (Ti) –{Ti} into L //L is list of trajectories with clusterids
                  IncreaseCluster (L, clusterid, Ɛ, min_trajs)
                  clusterid=clusterid+1;
          else
                  Mark Ti as outlier and assign to set O //O is outlier trajectories list
                 Assign for all Ti∈ T a clustered
   end for
 /* All density reachable and density connected trajectories (based on DBSCAN)
are merged in the same cluster with same clusterid. */
     IncreaseCluster (L, clusterid, Ɛ, min_trajs)
      While (L! =Ø) do
           {
             for Ti in L /*First trajectory in the list*/
                      Compute NƐ (Ti ) ;
                      If (|NƐ (Ti )| >= min_trajs) then
                     for each (X ∈ NƐ (Ti )) do
                          Assign clusterid to X;
                    If (X is unclustered) then
                            Insert X to L;
                           Remove Ti from L;
            }
```

Figure 4.8 TRAJODBSCAN Algorithm

DBSCAN algorithm has many advantages over other algorithms as discussed in the previous section. For finding outliers, density-based algorithm is the most appropriate type. This approach would be the base to Trajectory DBSCAN for finding outliers.

TRAJODBSCAN uses DTW for finding the similarity or differences between paths taken in supply-chain process. Each path can be called as trajectory as each stop in the path where object is read can be defined in terms of time and location coordinates.

Use of DTW as measure justifies the real scenario of supply-chain process where small time shifts due to difference in acceleration is not given much of attention provided the hops and final destination is matching.

4.8 EVALUATION ENVIRONMENT

Python 3.4.5 using Spyder environment is used for implementation. The experiments were conducted on a Lenovo laptop V 460 with Intel(R) 2.27 GHz processor and 4 GB RAM. Minimum number of trajectories is chosen as 25 based on the scheduled paths for particular tagged objects and minimum radius is taken as 10 m based on the minimum average of the distance among the trajectory paths given in the data.

4.8.1 Performance Evaluation

The comparison between different clustering methods is done and TRAJODBSCAN performed best in terms of accuracy. Accuracy is defined in terms of true positives and true negatives detected for being or not being outliers. K-Means is not showing any outlier cluster with our dataset but the cluster with the number of trajectories below five (threshold value based on average number of nearest trajectories) can be taken as outlier trajectories. 91.2% accuracy in case of clustering can be considered as reasonably good as here unlike classification technique there is no reference data for outliers. The similarity measures and algorithms' code are implemented in python language. There are total 10575 of rows in the dataset. After conversion of rows into the path database overall 210 paths are extracted with on an average 50 points in each path. Total trajectories taken are 100 with 50 points of location. Number of clusters for K-Means algorithm are taken based on elbow method. The elbow method can help in determining the number of clusters good enough to give have more cohesion within the clusters as compared to separation among the clusters The value is checked for different values of k in increasing order and at one it stops decreasing further with further increase in the number of clusters and that point is called elbow point and can decide on finding the value of k in k means clustering algorithm. In thesis work the objects checked for within the sum of squares are trajectories.

The following Figure 4.9 illustrates the decision of total number of clusters taken. with score representing the distance of the points in the cluster with the mean values. After five clusters, this score becomes constant and that’s why we have chosen the total number of clusters as five.

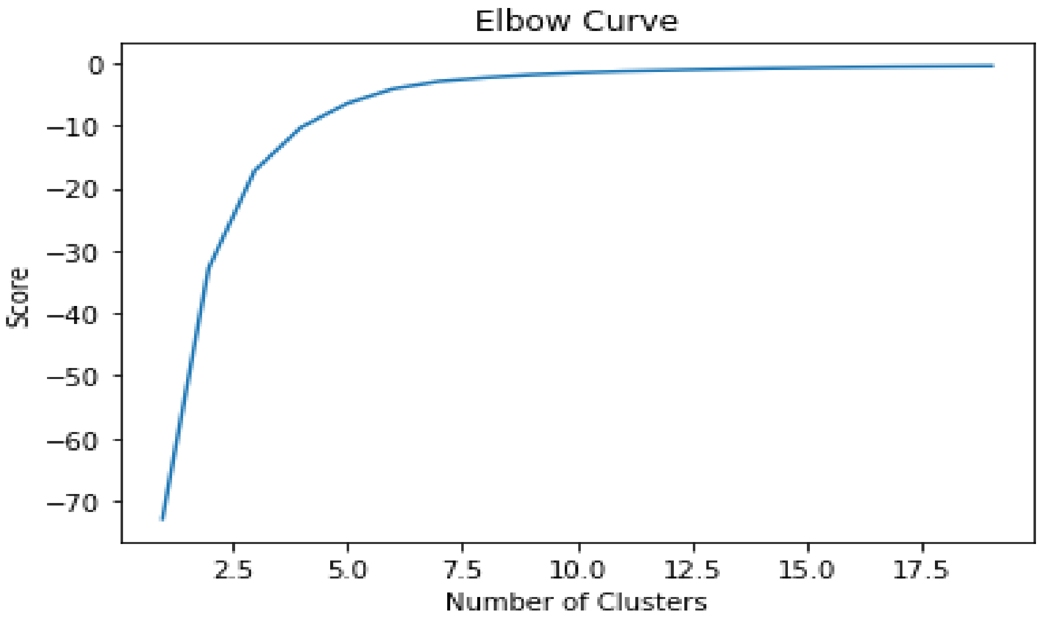


Figure 4.9 Elbow Method

The elbow method is quite helpful in selection of number of clusters in k means algorithm. Similarly, in hierarchical clustering also, agglomerative approach is followed where the clustering starts from individual clustering and then based on single linkage distance i.e. Minimum distance between trajectories among all the distances between each trajectory is calculated and then number of trajectories following the similarity measure are combined to form single cluster. This is done in iteration till all the trajectories are clustered into one cluster. Then backtracking the path of the tree generated the number of optimum clusters are chosen based on intra-cluster similarity and inter-cluster similarity.

Table 4.2 gives a comparative analysis between various clustering algorithms with proposed clustering approach to find outliers i.e. TRAJODBSCAN.

Table 4.2. Performance Analysis

Clustering method	**Mean time (s)**	**Outliers accuracy**	**Complexity**
DBSCAN	5.23	70.2% (58 out of 82 detected)	O(kN^2): k = no. of dimensions
TRAJODBSCAN	90.06	91.2% (75 out of 82 detected)	O(kN^2(logN)): k = no. of dimensions
K-Means (with K = 5) (here k = 5 is selected based on elbow method)	4.03	No outliers detected	If *k* and *d* (the dimension) are fixed, the problem can be exactly solved in time O(N^{dk+1}), where N is the number of entities to be clustered
Hierarchical (Agglomerative) **(checked for 5 clusters)**	12.23	56.4% (47 out of 82 detected)	O(N^{2}logN)

Table 4.2 is giving the clear picture of TRAJODBSCAN outperforming other clustering algorithms. Each of the traditional algorithms is modified according to trajectory inputs as our research focuses on trajectory clustering rather than points clustering. Here the basic unit is a trajectory. The distance measures used in traditional approaches is Euclidian measure. Although the tie taken by TRAJODBSCAN is more because of the time complexity of dynamic time warping (DTW) distance approach still the accuracy factor dominates as compared to others. Time complexity can be further managed with optimizations of DTW methods like use of lower bound methods along with indexing schemes like R*.

4.9 CONCLUSION

The proposed approach for finding the outliers or anomalies in the trajectories formed by the RFID-enabled supply-chain process may help us to alert the various stakeholders in the chain for prompt action in case any abnormality or outlier is detected. Weights can be assigned the values based on the requirements of similarity measures. This approach is likely helping us to find the correct number of outliers in an efficient manner. Density based clustering approach is a good choice for finding the outliers as outliers are not frequent events and not all the clustering algorithms can handle them effectively. Along with the dynamic time warping measure there is a tremendous increase in the accuracy of the clusters generated for the cases where straight line path is not followed point to point. For the time series data where time is a prevalent factor it is very important to consider the time lag as it’s a norm in supply chain path process for a tag moving from supplier to customer with many hops in between. Our approaches have proven quite effective in terms of accuracy for checking the deviation of the path.

CHAPTER 5: PREDICTIVE ANALYSIS OF RFID SUPPLY-CHAIN PATH USING LONG SHORT-TERM MEMORY (LSTM)

Prediction of location has gained lot of attention in different applications areas like predicting the path or any deviation like taxi-route, bus route, human trajectory, robot navigation. Prediction of the next location or any path deviation in RFID enabled supply-chain path followed in the process is quite a novel area for the related techniques.

5.1 INTRODUCTION

In this chapter, predictive analysis of outliers in RFID supply-chain process is studied and implemented using Hidden Markov Model, XGBoost, Recurrent Neural Networks and Long Short-Term Memory techniques.

Different classification models are compared for the accurate prediction of the outlierness of the path followed by the tagged objects read by RFID readers during the supply-chain process. Comparison of Hidden Markov Model (HMM), XGBoost (decision-tree-based boosting), and Recurrent Neural Network (RNN) and state of the art technique in RNN known as Long Short-Term Memory (LSTM) is done. To our knowledge, the above-mentioned classification techniques have never been used for this application area for outlier point prediction. It may be concluded that for the longer path sequences, LSTM has outperformed over other techniques. The training datasets used here are in the form of the record of the outlier positions in particular path and at particular time and location.

The main objective of the research is to predict the outlierness of the trajectory path followed by the objects in supply chain starting from supplier to consumer. Supply chain is quite a long sequence of points where tagged objects are automatically read by readers with both space and time component i.e. location where the data is read and at what time it is read. This data can be considered as time series data and thus, we hypothesize that Recurrent Neural Network and Long Short-Term Memory (LSTM) techniques may be best suited for such types of data.

5.2 SELECTION OF PREDICTION METHOD

Traditional models like Hidden Markov Model [15] and Recurrent Neural Networks [71] [78]are being used to handle the time series (spatio-temporal) data. But they are efficient only for sequences or trajectories with short lengths; that is for longer sequence of points in the chain, performance of traditional models deteriorates as is verified for many other systems [31]. LSTM, a Recurrent Neural Networks -based technique, can save information about the previous sequence points, thus is a preferable method for longer sequence of spatio-temporal points in a trajectory for further prediction and classification. Following sub-sections will give the overview of each technique used for prediction and classification.

5.2.1 Hidden Markov Method

The Hidden Markov Model (HMM) is a method to calculate probability distribution of the observations or outputs of the sequence of states. It is a very popular model used for sequential type of data where based on the sequence of observed states prediction of future states can be done. Although it is based on Markov process where the states are not hidden and the parameters available are state transition probabilities, HMM model has hidden states. Here only the outputs of the states are observed without any information about states. So, the processes involved in generating the data are hidden from the user. Sequences of outputs known as tokens are used for the prediction of next state outputs in the sequence.

In Markov process the current state is used to predict or find the probability of next state and this makes the system ‘memory less’. This ’memory less’ property of a process is called the Markov property, whereas an HMM is one in which the states, though known, are not directly visible to the observer. Therefore, the sequence of observed output values provides information about the sequence of states. We have used the concept of HMM to design and implement the prediction of outlier point in a RFID supply-chain path. Here the output with probability of being an outlier point or not will help in the prediction of next point in the path for outlier point.

Parameters

In general, consider the parameters required for prediction of the next point in the sequence is number of hidden states and number of observation states. Here the hidden states in our implementation refer to the location type of a particular path_id and number of observation states are two i.e. outlier and non-outlier. Other parameters are the length of sequence or path taken. Here it’s taken as 50. Discrete set of observation symbols consist of o1 and o2 for outlier and non-outlier. Probability of being in a particular state is adjusted according to the set of sequences taken in the training sets. Probability of being in a state i at the beginning of experiment as state initialization probability. State transition probabilities and emission probabilities are required to finally output the observation symbol of the next time step in the sequence. The observation sequence for the hidden state can help in finding the next observation.

State transition probability tells us that the probability of moving from one state to the other state. Here, observation sequence represents outlier or non-outlier points in the supply-chain path.

5.2.2 XGBoost Prediction Method (Tree-Ensemble Based Method)

XGBoost [96] is an ensemble method which can be used to do classification and prediction. It is and advanced implementation of gradient boosting algorithm with faster computation and better management of resources parallelly basic concept is boosting where many weak learners combine to give a strong learner with high accuracy. The classification is done either linearly or in the form of decision tree. It’s just that here instead of single decision tree, multiple decision tree models are used. The outcomes of the model at a particular time instant t are weighed according to the previous step t-1 output. Those outcomes which are predicted correctly are given less weightage as compared to the ones which are misclassified so that at each step the misclassification can be corrected. So, in boosting weak learners (which are just better than random classifiers) are trained sequentially by correcting their predecessor weak model. If decision trees [96] are used for modelling then they can be faster than other models. The tree ensemble model is a group decision trees created iteratively with the attempts to reduce the misclassification rate at each step. There are many boosting algorithms like AdaBoost (Adaptive Boosting) [97], Gradient Tree Boosting and XGBoost but

XGBoost (eXtreme Gradient Boosting) is the most optimized distributed gradient-boosting library.

Parameter Tuning

In XGboost few parameters need to be tuned during training of the models for optimized output. The parameters include booster for inputting the type of model used, learning rate represented as eta with default value as 0.3, minimum sum of weights of the outcomes required in child of the tree and normally smaller value is chosen. Minimum child weight parameter is used to prevent overfitting. It's better to use low value or else if higher value is chosen, it can lead to underfitting also. Maximum depth of the tree should also be set It is tuned with inbuilt cross validation and the range of this parameter lies between 3-10. Gamma is another parameter. Node of the tree model can split further into separate paths when the split results in the positive reduction in loss function and Gamma helps in specifying the minimum reduction in loss needed for further splitting. Its default value is set a zero. Subsample parameter can also be set for controlling the fraction of observations to be selected for each tree. Very high value prevents overfitting and very small value can lead to under fitting. Typically, its value lies between 50 percent to 100 percent. Similarly fraction of columns to be sampled can also set by using colsample_bytree parameter. scale_pos_weight |parameter is used to control class imbalance and helps in faster convergence of the tree. Because of high-class imbalance. There are some regularization parameters also known as lambda and alpha to reduce the complexity of the model but they are generally required in case of very high number of dimensions and scalability otherwise they are avoided Generally the process of parameter tuning starts with high learning rate and then it is lowered iteratively and checked for the performance. Based on the dataset available XGBoost finally gives the optimal parameters.

5.2.3 Recurrent Neural Networks (RNNs)

A recurrent neural network (RNN) is a kind of artificial neural network where connections between units form a directed cycle. RNNs have an internal memory for the processing of sequence data or trajectory data. They are the type of feed-forward neural networks but they transfer information from one stack to another in sequence of time. Input is given in form of feature and the type of information at each time step is same. Number of hidden layers is same at each time step. RNN is, therefore, considered a suitable approach for time series or sequence data. It can also perform well with time series data of varying lengths. In supply-chain process also, chain of different reading points can be of different lengths, so RNN is a good choice. The output of RNN is dependent on input from the previous time steps as well as current state input. Unlike Hidden Markov Models, RNN can capture long range of dependencies with time.

Hidden number of layers is the representation of the number of states or time steps. As the number of nodes in the layer increases, the number of states also increases exponentially. So, there is always a trade-off between accuracy and time.

Recurrent Neural Networks have the drawback of exploding or vanishing gradient problem which occurs when the gradients become too large or too small and make it difficult to model long-range dependencies. Generally, after ten or more-time steps, its performance starts degrading. LSTM appears to be the solution to this drawback.

5.2.4 Long Short-Term Memory Network (LSTM)

Long Short-Term Memory Network (LSTM) has the ability to remember important long-term and important short-term information. LSTM can decide, which pieces of information are important to remember for the short term and which are important for the long-term. Hidden units in LSTMs are referred to as memory cells, and are modified to have an input node: g(t), as an input gate: i(t), as a forget gate: f(t), output gate: o(t), and internal state: c(t). Fundamentally, the architecture of LSTM and RNN is same but the functions used to compute the hidden states are different These differences make LSTM more efficient in capturing long-term sequences. Current LSTMs have the corresponding update equations.

$$\boldsymbol{g(t) = tanh\left(W^{gh}x^{(t)} + W^{gh}h(t-1) + b_g\right)} \tag{5.1}$$

$$\boldsymbol{i(t) = sigmoid\left(W^{ix}x(t) + W^{ih}h(t-1) + b_i\right)} \tag{5.2}$$

$$\boldsymbol{f(t) = sigmoid\left(W^{fx}x(t) + W^{fh}h(t-1) + b_f\right)} \tag{5.3}$$

$$\boldsymbol{o(t) = sigmoid\left(W^{ox}x(t) + W^{oh}h(t-1) + b_0\right)} \tag{5.4}$$

$$\boldsymbol{c(t) = g(t)\Theta\, i(t) + \mathrm{c(t\text{-}1)}\Theta f(t)} \tag{5.5}$$

$$\mathbf{h(t) = tanh(c(t))\, \Theta\, o(t)} \tag{5.6}$$

The above-mentioned equations from 5.1 to 5.6 describe the input and output to LSTM. Here Θ represents the point wise multiplication and W^{gh}, W^{ix}, W^{fx} , W^{oh} and W^{ox} are the weight matrices from one input to another output.

The input node g(t) takes input and the previous hidden layer in the standard way. tanh is used here and the output of tanh lies between -1 and 1. The internal state c(t) consists of a self-connected recurrent edge with fixed unit weight. This allows error to flow in back propagation through time steps easily and solves the problem of vanishing or exploding gradients. The input gate, i(t) helps to modulate how much of the input that we should utilize since it is point wise multiplied by g(t) when calculating c(t). If i(t) consist of 0s, then we completely disregard the current input. If it consists of 1s, we utilize the whole current input. Similarly, forget gate f(t), allow us to forget unneeded past internal state. Lastly, the output gate, o(t) allows us to remember important information when calculating the next hidden state, h(t). The standard SoftMax can then be applied to obtain predictions.

LSTMs are currently widely claimed to be state of the art in RNN literature. Many have noted that recent breakthroughs in sequence prediction for various problems have been due to LSTMs, not RNNs. Recently, LSTMs have started to become start of the art in speech translation, machine translation, image captioning, and question answering systems. All of these problems boil down to sequence problems, and LSTMs have been able to perform very well on each of these. Therefore, it will be interesting to see if LSTM outperforms RNN and other methods like boosting decision tree based, Hidden Markov Model which is popularly used for the user trajectory problem.

Vanishing gradient problem is a problem in RNN where weights are updated at each iteration of learning with respect to the partial derivative of the error function with respect to current weight in each iteration of training. Now with multiplication of the calculations at each step the gradient would become so small that it starts vanishing and prevent the weight from changing the value according to error and due to this further learning process is stopped. Figure 5.1 defines a single LSTM cell with input gate(i(t)), Forget Gate(f(t)) and Output Gate(O(t))

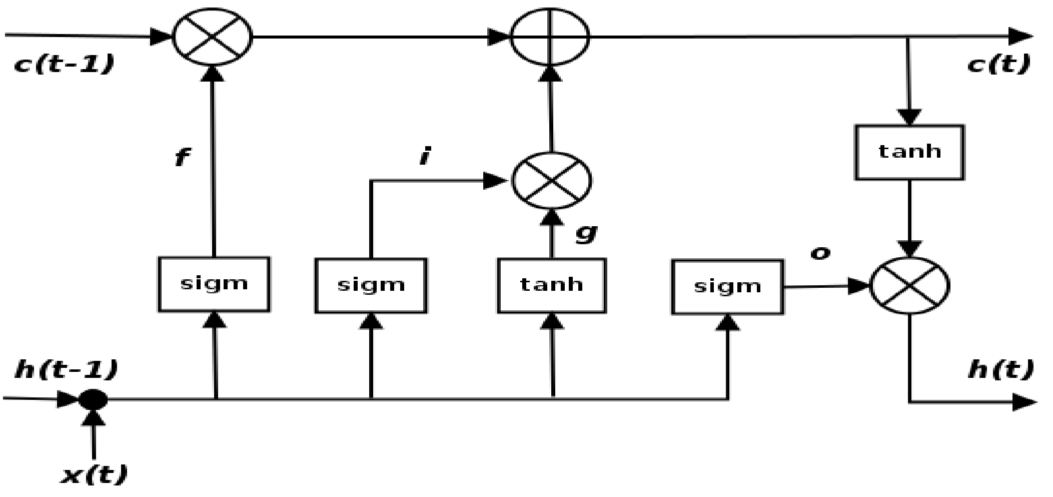


Figure 5.1 Detailed LSTM Cell

So, after the discussion of LSTM and RNN, we may conclude Standard RNNs (Recurrent Neural Networks) suffer from vanishing and exploding gradient problems. LSTMs (Long Short-Term Memory) deal with these problems by introducing new gates, such as input and forget gates, which allow for a better control over the gradient flow and enable better preservation of "long-range dependencies."

Figure 5.2 and Figure 5.3 give the pictorial representation of the differences. Here sigma function is expressed as "σ".

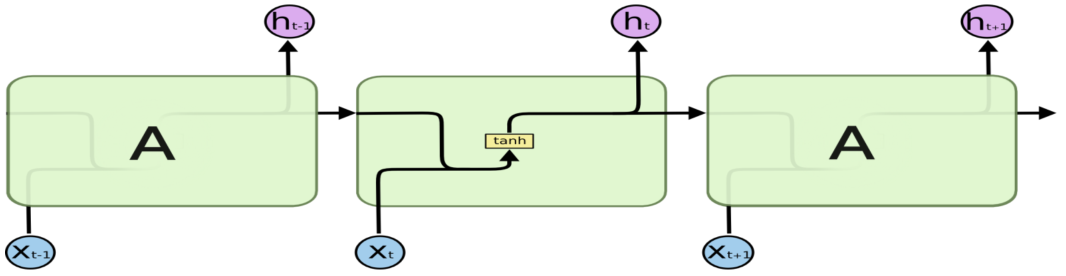


Figure 5.2 Recurrent Neural Network Cell

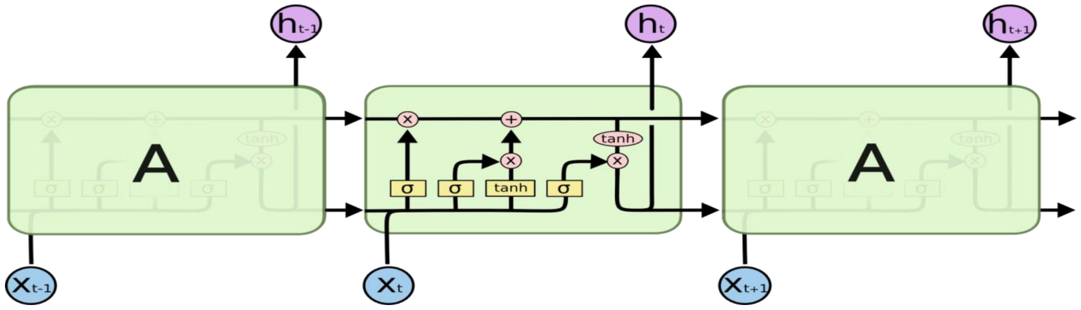


Figure 5.3 Long Short-Term Memory Cell

5.3 RELATED WORKS

The concept used in the research uses location-based prediction as here with every location outlier status is also given in the data and thus predicting the location can also predict the outlier status. Many location-prediction algorithms have been implemented which can predict the next location given the current location. Simplest method is use of direction and speed but it can vary according to various circumstances like traffic jam, weather, theft etc. and may not give correct results. In recent time, many methods are popular for the mining of trajectory data based on history. Based on the historical path followed and the status of the path and recent few locations and their status (here the status of path deviation or any outlier) can be predicted with accuracy. Methods like movement pattern mining and model-based methods are most commonly used.

Movement pattern mining uses the concept of frequent pattern followed. Next location is predicted based on the previous one in the pattern. Jiang *et al*. [98] studied taxi trajectories and found their moving pattern and behaviour. The trajectories here need to be converted into cells first. Jeung [55] used Apriori algorithm to extract movement patterns. An improvement of Apriori algorithm is used by Yavas *et al*. [71]. Co-occurrences of the locations can be extracted from the frequent patterns generated. A modified Prefix-Span algorithm was developed by Morzy [72] as both temporal and spatial features were taken into account. So sequential pattern method was used and gave better results. I used pattern-based prediction method by clustering the frequent places and using Fourier transform to detect specific period. Clustering-based algorithm had the problem of losing a lot of information for the points not lying in the clusters thus gave low accuracy in prediction. Cheng proposed multi-center Gaussian model for finding the distance between the patterns generated but in this method the sequencing or ordering is not considered. Mathew proposed a hybrid hidden Markov model. Jeung converted the trajectories into frequent regions with cell partition algorithm but again the prediction accuracy here is constrained by the granularity of the cells.

Recently, Recurrent Neural Networks is a very popular method in the sequence mining. When the RNN feed-forward network is unfolded, the different layers representing different time steps can be expressed. Multiple hidden layers in RNN can adjust dynamically with the input of behavioural history; therefore, an RNN is suitable for modelling temporal sequence. Liu extended traditional RNN spatial and temporal contexts to predict the spatio-temporal data. For small sequences or trajectories, RNN gives results but when the length of the trajectory is very long., even more than 10 steps the problem of vanishing and exploding gradients when the error back propagates through many steps. Comes in. and can't be handled by RNN.

LSTM is the solution to this problem by using the memory blocks in any of the layers and thus remembering the context and value for much older than recent previous history. Most common applications where this technique is used till now are translation of speech, machine translation, automatic question answering systems. All of these are all sequence-related applications. So, it is expected that LSTM should perform better than other techniques in case of prediction of outlier position on the long length supply-chain trajectory. It's never been tried with this application as per authors' knowledge.

5.4 SYSTEM DESIGN

In this section, the design of the proposed system will be provided. First, the basic scheme of working of the system will be given and then architecture of the learning module will be given. Finally, the detailed design of the system will be provided by describing the format of the data, discretization of the data and setting of hyper parameters.

5.4.1 Basic Scheme

The main focus of the research is to develop a system which can predict an outlier point based on the training data with the class label for outlier points.

The overall system will first do the learning with different techniques like hidden Markov modelling, an ensemble method XGBoost, Recurrent Neural Networks and Long Short-Term Memory method. For each different learning technique, different parameters are tuned and set accordingly to generate the different learners.

These different learners are trained by providing input data (Trajectory data) and respective models are generated. Later, these reasoners are used to predict the outliers among the paths by taking test data as input. Finally, based on the predicted outputs, an alert may generate for the predicted output points. The same is illustrated in the Figure 5.4.

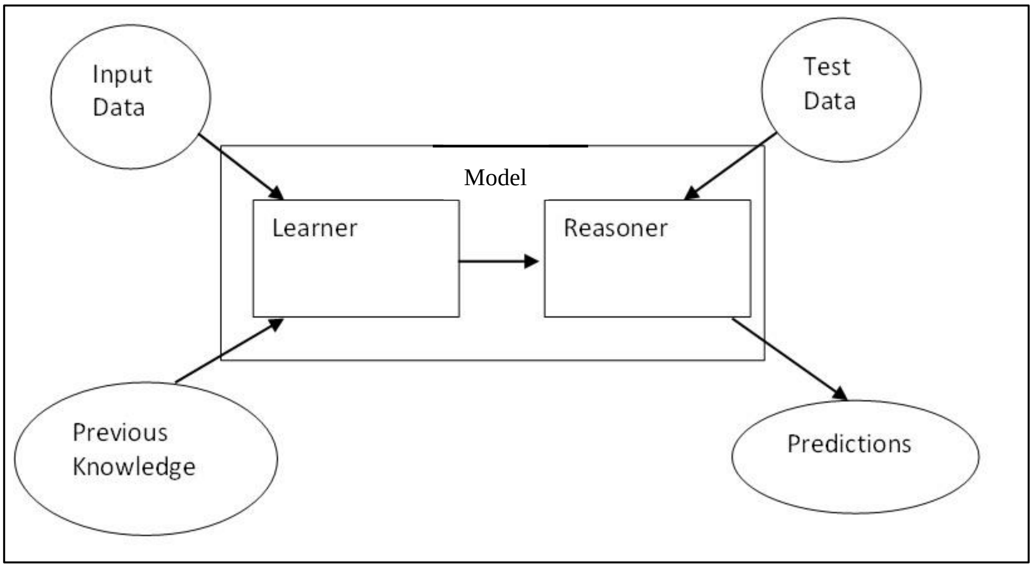


Figure 5.4 Basic Scheme of Learning Model

5.4.2 Data Preparation

The proposed architecture focussing on LSTM networks is based on the hypothesis that it is more efficient than other discussed techniques for longer sequences. Here the trajectory data taken as input vector consists of discretized longitude, latitude, time and track_id of supply-chain path. Using the class label outlier which defines whether the particular point of the particular path id is outlier or not in the historical data. Prediction for the particular point/node in the supply-chain path can be done for being an outlier or not and the alert can send to the stakeholders like supplier, manufacturer, distributer or retailer involved in the process. For each LSTM cell that is initialized, the number of hidden units in the LSTM cell has to be supplied. The most optimal value can be selected by repeated experiments. There is no fixed rule for this. It depends on the application data and consideration of the time taken to train the model and test it. Too high a value may lead to over fitting or a very low value may give very poor results. Selecting the right parameters is very important and it depends on the application data also. The dimension of the weights will be number of hidden layers multiplied by the number of classes as output. Thus, on multiplication with the output (Val), the resulting dimension will be batch size multiplied by number of class. The following Figure 5.5

Outlier Prediction Analysis Process of Trajectory Point explains the detailed view of the processes involved in the proposed system.

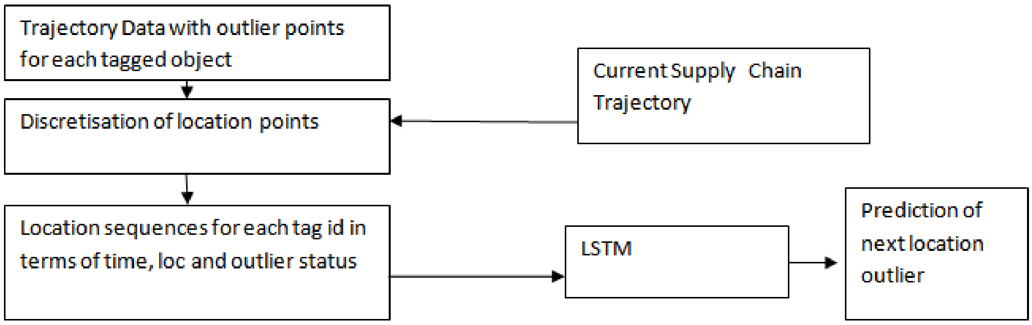
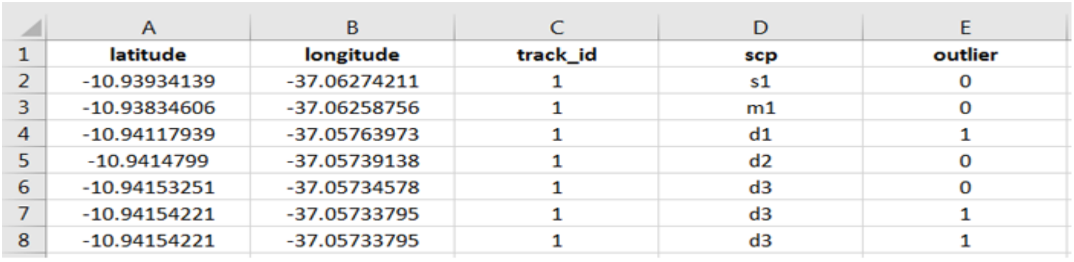

Trajectory Data with outlier points for each tagged object
Discretisation of location points
Current Supply Chain Trajectory
Location sequences for each tag id in terms of time, loc and outlier status
LSTM
Prediction of next location outlier

Figure 5.5 Outlier Prediction Analysis Process of Trajectory Point

In the next sub section, we are going to discuss the format of the data to be taken as input to the system. The datasets taken as input is a type of spatio-temporal data with time and location information and thus can be converted into trajectory data which is a sequence of locations in order of time. Datasets considered here is the historical data stored (here it is the output of TRAJODBSCAN as discussed in the previous chapter) with the information about outlier positions with the details of latitude, longitude, path/Track_id, timestamp and a class label showing whether it's an outlier position in the previous history or not in form of 0/1. The datasets as shown in the Figure 5.6 are taken as training datasets for developing predictive model using LSTM technique and also compared with other techniques like XGboost, Hidden Markov Model and Recurrent Neural Network as they are the most popular ones when classification or prediction in trajectory/sequence is done. The process follows many numbers of iterations to get most optimal and accurate analysis.



	A	B	C	D	E
1	latitude	longitude	track_id	scp	outlier
2	-10.93934139	-37.06274211	1	s1	0
3	-10.93834606	-37.06258756	1	m1	0
4	-10.94117939	-37.05763973	1	d1	1
5	-10.9414799	-37.05739138	1	d2	0
6	-10.94153251	-37.05734578	1	d3	0
7	-10.94154221	-37.05733795	1	d3	1
8	-10.94154221	-37.05733795	1	d3	1

Figure 5.6 Supply-Chain Training Dataset

The data (with latitude, longitude, track_id, time, outlier and stakeholder's information) as shown in Figure 5.6 is converted into a feature vector for each time step in the sequence in a trajectory. Each track_id represents a single trajectory. scp attributes having values like s1, d1, m1 etc. are representing supplier 1, distributor 1, manufacturer 1 and so on. Total number of tuples taken is 10575 in number. This data is mapped to location with latitude and longitude.

5.4.2.1 Discretization of Data

Trajectory's coordinate information is continuous in space. It is quite complex to model it. Value of trajectory location points can be mapped into projected points on road. Margin of displacement of points' threshold up to 10 m is fixed. It can be fixed based on precision required and length of the road if the projected point is not discrete one. Open Street Maps (OSM) [99] is referred for road information of roads.

Each supply-chain path trajectory is represented as line T_i= (n(s), n(e)) with a start node $n(s)_s$=(longitude(s), latitude(s)) and the end node n(e)=(longitude(e), latitude(e)).The point p(p)=(longitude(p), Latitude(p)) of projection of longitude and latitude on road is calculated by equation (1) for any point n(i) =(longitude(i), latitude(i)) on road with k as slope of the line of the road. This is a feature extraction step extracting discrete points on the road so that it's much easier to develop predictive model after that. Equations 5.7 and 5.8 define the conversions.

$$\mathbf{longitude(p)} = \frac{\mathbf{k \times longitude(s) + \frac{longitude(i)}{k} + latitude(i) - latitude(s)}}{\mathbf{\frac{1}{k} + k}} \qquad \textbf{(5.7)}$$

$$\mathbf{latitude(p)} = -\left(\frac{\mathbf{1}}{\mathbf{k}}\right) \times (\mathbf{longitude(p)} - \mathbf{longitude(i)}) + \mathbf{latitude(i)} \qquad \textbf{(5.8)}$$

The length of the road L can be calculated by equation (5.9) where $Ø_1$ is the radian of latitude(s), $Ø_2$ is the radian of latitude(e), ΔØ the radian of (latitude(s)-latitude(e)) and Similarly φ_1 is longitude_s and φ2 is longitude(e) and Δφ is the radian of (longitude(s)-longitude(e)). R is the radius of the earth taken equal to 6371 km.

The fixed points in the trajectory on road can be calculated by n(s), n (e), k and the projected points.

$$\mathbf{L} = \mathbf{2} \times \mathbf{R} \times \mathbf{arcsin}\left(\sqrt{\left(\mathbf{sin2}\left(\frac{\Delta \text{Ø}}{2}\right) + \mathbf{cos}(\text{Ø}\mathbf{1}) * \mathbf{cos}(\text{Ø}\mathbf{2}) * \mathbf{sin2}\left(\frac{\Delta \varphi}{2}\right)\right)}\right) \qquad \textbf{(5.9)}$$

Considering the above calculations, all the latitudes and longitudes are converted to projected points which can be taken as input to the training model and help to do the calculations in an efficient way.

Timestamp can also be discretized by assigning time ids by diving time into intervals. The size of time is taken as 10 minutes. It creates six-time ids in an hour and total 144-time ids from 1 to 144 in a day. So, combination of day of the month and time id of the day gives discrete value of timestamp.

Here the test data is in the same format as training data except the class label. The technique used for segregating test and training data is ten cross fold validation. The input and output scheme of the datasets used to model the classification process as shown in Figure 5.7 below

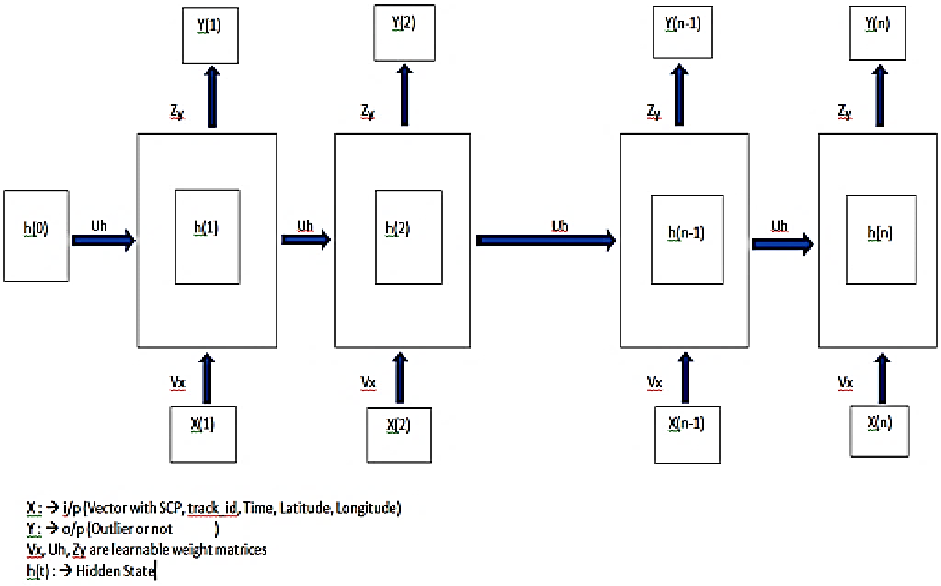

Y(1)
Y(2)
Y(n-1)
Y(n)
Zy
Zy
Zy
Zy
h(0)
Uh
h(1)
Uh
h(2)
Uh
h(n-1)
Uh
h(n)
Vx
Vx
Vx
Vx
X(1)
X(2)
X(n-1)
X(n)
X : → i/p (Vector with SCP, track_id, Time, Latitude, Longitude)
Y : → o/p (Outlier or not)
Vx, Uh, Zy are learnable weight matrices
h(t) : → Hidden State

Figure 5.7. Recurrent Neural-Network Model for RFID Supply-Chain Trajectory

In Figure 5.7. Recurrent Neural-Network Model for RFID Supply-Chain Trajectory X is input vector and X(i) stands for input at time i. It's a vector with latitude, longitude as projected positions on road, time id according to discretized timestamp and track_id and position's stakeholders' information as dimensions of the vector. V_x is matrix of weights from input to hidden layer; U_h is matrix of weights from previous hidden layer to current one and Z_y is the matrix of weights from hidden layer to the output.

5.5 HYPER PARAMETER SETTING

To train the network, training sets, validation sets or test sets are taken in a standard way. One set of trajectories are taken for the training and another set is taken for the validation part. One epoch is the complete pass through training data. In each pass a candidate model is used for the prediction of test set.

Implementation of RNN and LSTM models is done using tensor flow in python. XGBoost method is implemented by importing XGBoost package in python. Optimal parameters taken here are: number of rounds = 400, eta =0.05, maximum depth=5, scale position weight=5 and minimum child weight=1. HMM is also implemented in python with cost parameters set to 0.1 with forward-backward algorithm and rest of the parameters are default ones as they are the most tuned ones to give the optimal values, learning rate taken as 0.01 and maximum iterations =100.

The basic steps to design and run the models for prediction are as follows:

- Build the computation graph for defining calculations and functions executed in runtime.
- A Tensor Flow session is created and execution of the graph/network created in the previous step is done with the supplied data.
- Calculation of the probability scores of each output.
- Check the accuracy of the model by calculating the loss. Cost function is used to compare predicted value and actual value. The aim is to reduce or minimize the cost function.
- Last step is of optimization. TensorFlow has optimization functions like Adam Optimizer, RMSPropOptimizer. We have chosen Adam Optimizer. The use of optimization is to minimize the loss as much as possible.

Here the trajectory datasets of supply-chain paths followed is on an average 989 points of read in a day with on an average 200 trajectories in a day. Training and Test is split by 10 Cross Validation method. For LSTM, number of hidden units are experimented from 2 to 250 and after 120 the model is showing constantly better results. So, the optimal value is taken as 120.

The LSTM Recurrent Neural Network Approach has never been applied in the path prediction and anomaly detection for supply-chain path. Python packages are used for the implementation**.** Figure 5.8 and Figure 5.9 represents the evolution of classification accuracy and loss minimization on the training data. To make the plot readable, tensor flow plot starts from step 300 Number of features taken in the feature vector are five. They are SCP, track_id, Time, Latitude and Longitude. Here SCP defines the type of location: s for supplier, c for consumer, p for producer and so on. track_id is path_id representing the path or trajectory number, time is the time id generated after discretization, latitude and longitude are the discretized location coordinates. The batch size is taken as 100. Training or learning is run for 140, 000 iterations. AdamOptimizer function with learning rate 0.0001 is chosen. Drop Out of each layer in LSTM and RNN is set to 0.2 to avoid overfitting.

5.6 RESULTS ANALYSIS

The accuracy and loss are accumulated to monitor the progress of the training. 40,000-50000 iterations are generally enough to achieve an acceptable accuracy.

As already discussed in the previous section about the basic scheme, first the classification model needs to be trained based on the training input data. Now the data input here is the trajectory data with outlier points and based on the sequences of outlier points in a trajectory the system generated a classification model. The system is trained for many numbers of iterations till the loss is reduced to minimum and accuracy is reached to the maximum. The following Figure 5.8 and Figure 5.9 give the illustration of accuracy and loss functions respectively during training/learning process.

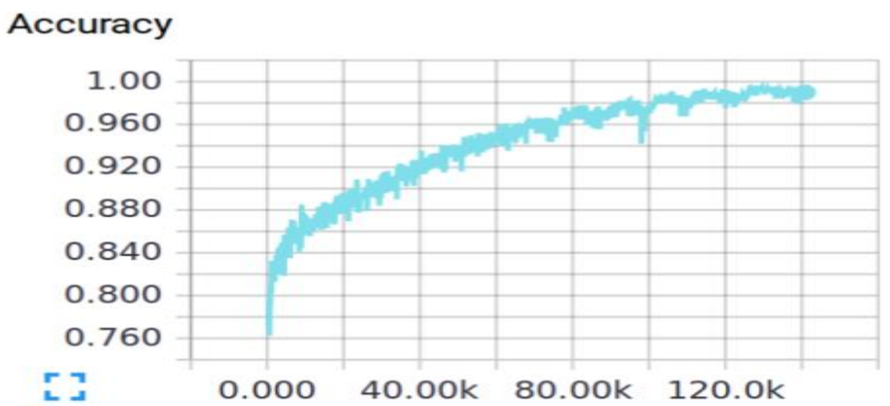


Figure 5.8. Accuracy Function

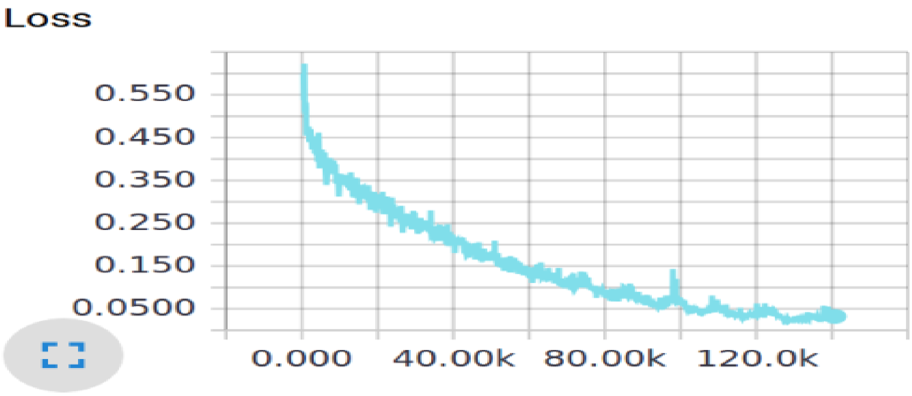


Figure 5.9 Loss Function

Table 5.1 gives the description of the values for average loss and average accuracy based on the number of iterations. The iterations are performed till a reasonably good accuracy level is reached and the loss is minimized.

Table 5.1 Average Accuracy and Loss with Iterations

Number of Iterations	Average Loss	Average Accuracy
20000	.301534	88.12%
40000	.213423	92.13%
80000	.098636	96.27%
120000	.049879	98.11%
140000	.001296	99.79%

RNN and LSTM models are compared with HMM and XGBoost for the accuracy against the test data and table 5.6. shows the results in form of precision, recall and f-measure. LSTM is giving better results in terms of finding true positives from set of true positives and false positives. The experiments are executed on Windows 7 platform with a 2.8 GHz i7 CPU and 4 GB RAM.

In this kind of data accuracy of the classification can't be calculated by finding the ratio of total number of true positives (TP) and true negatives (TN) among all the values.

The value of true negatives will outweigh others and will give similar accuracy for all the approaches. So, accuracy should be measured in terms of Precision Recall only. Precision(p) and Recall (r) are used to measure the accuracy of the classification approached are expressed as shown in the equations 5.10 and 5.11. Here is TP is true positive, FP is false positive and FN is false negative.

$$\mathbf{p} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \tag{5.10}$$

$$\mathbf{r} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \tag{5.11}$$

Precision *p* is the number of correctly classified positive examples divided by the total number of examples that are classified as positive. Recall *r* is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set. So, precision gives the value of correct classification among the predicted ones and recall tells us about the correct values from the existing ones.

F score combines precision and recall into one measure as shown in Equation 5.12.

It can be defined as the weighted harmonic mean of the precision and recall of the test. So, in a way it measures the balance between precision and recall.

$$\mathbf{F} = (\mathbf{2} * \mathbf{p} * \mathbf{r})/(\mathbf{pr} + \mathbf{r}) \qquad \textbf{(5.12)}$$

Here F stands for F score. Higher value of F- Score shows more accuracy. The confusion matrix table and table for all the supervised methods are as shown below. As we can see from the confusion matrices that number of true negatives are too large in number. These are the points which are detected not to be outlier and they are actually not outlier points in the trajectories of RFID supply chain paths. The main focus here is to classify the accurate number of outliers from the actual ones. The results are as shown in confusion matrices because outliers are rare events and so is the case with supply chain process.

Tables 5.2, 5.3, 5.4 and 5.5 give the confusion matrices for different supervised approaches used in the thesis work. They give the comparative analysis of XGBoost, Hidden Markov Model, Recurrent Neural Network and LSTM in terms of number of false positives, false negatives, true positives and true negatives.

Here the number of true negatives is quite large in number as compared to true positives, false positives and false negatives because here the classification is done on rare events i.e. outliers. Outliers are abnormal events which are not frequent in occurrence.

Table 5.2 Confusion Matrix for RNN

N=10575	**Predicted (Positive)**	**Predicted (Negative)**
Actual (Positive)	63	19
Actual (Negative)	13	10480

Table 5.3 Confusion Matrix for LSTM

N=10575	Predicted (Positive)	Predicted (Negative)
Actual (Positive)	72	10
Actual (Negative)	5	10488

Table 5.4 Confusion Martix for HMM

N=10575	Predicted (Positive)	Predicted (Negative)
Actual (Positive)	44	38
Actual (Negative)	28	10465

Table 5.5 Confusion Matrix for XGBoost

N=10575	Predicted (Positive)	Predicted (Negative)
Actual (Positive)	58	24
Actual (Negative)	23	10470

Table 5.6 Comparison between different Models

Model	Precision	Recall	F-Score	Training Time (Secs)	Prediction Time (secs)
LSTM	.94	.88	.90	256	2.34
RNN	.83	.77	.79	82	1.4
HMM	.67	.54	.59	2830	18.3
XGBoost	.72	.71	.71	340	1.1

If we analyse the table graphically as shown below then we can see that LSTM has better precision, recall and F score as compared to RNN, XGBoost and HMM. The results show the worst performance of HMM among all. Training time is also quite high for HMM and has a remarkable difference from XGBoost, RNN and LSTM. For rest of the three the training time is quite comparable. Same is for prediction time.Figure 5.10,5.11 and 5.12 shows the results graphically.

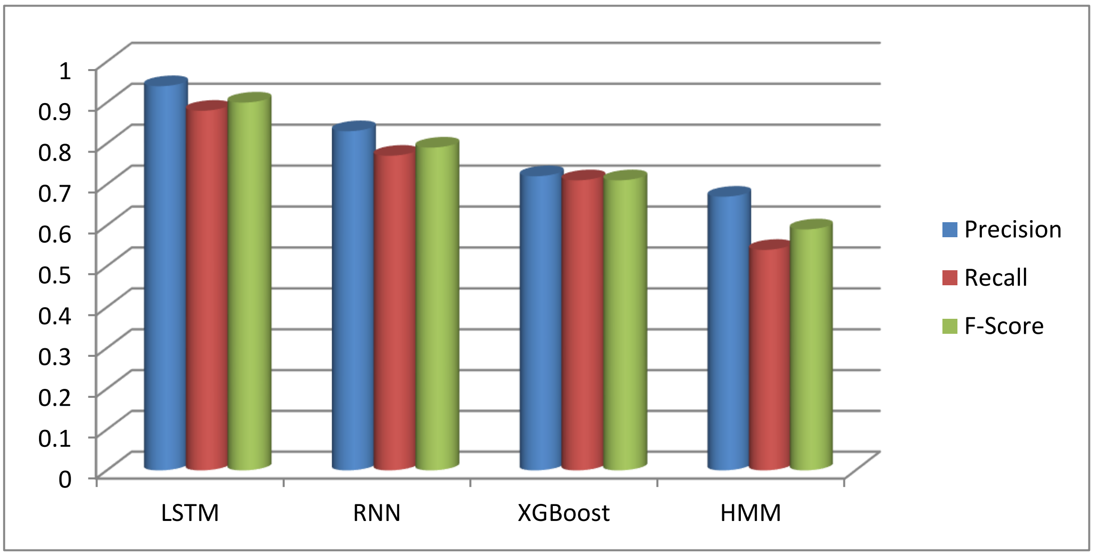


Figure 5.10. Comparative Analysis of F- Score

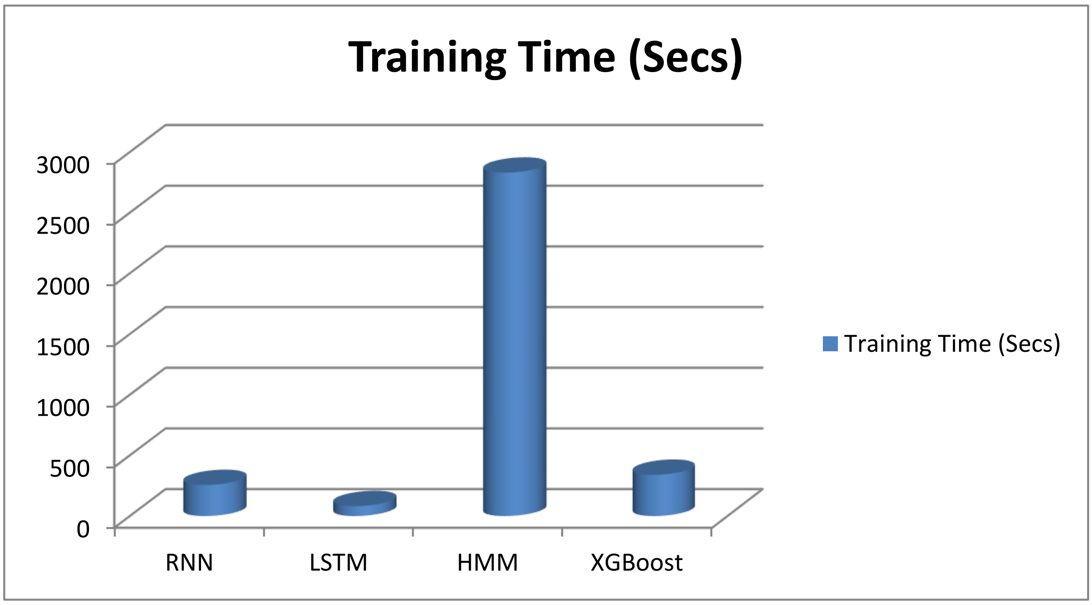


Figure 5.11. Comparison of Training Time

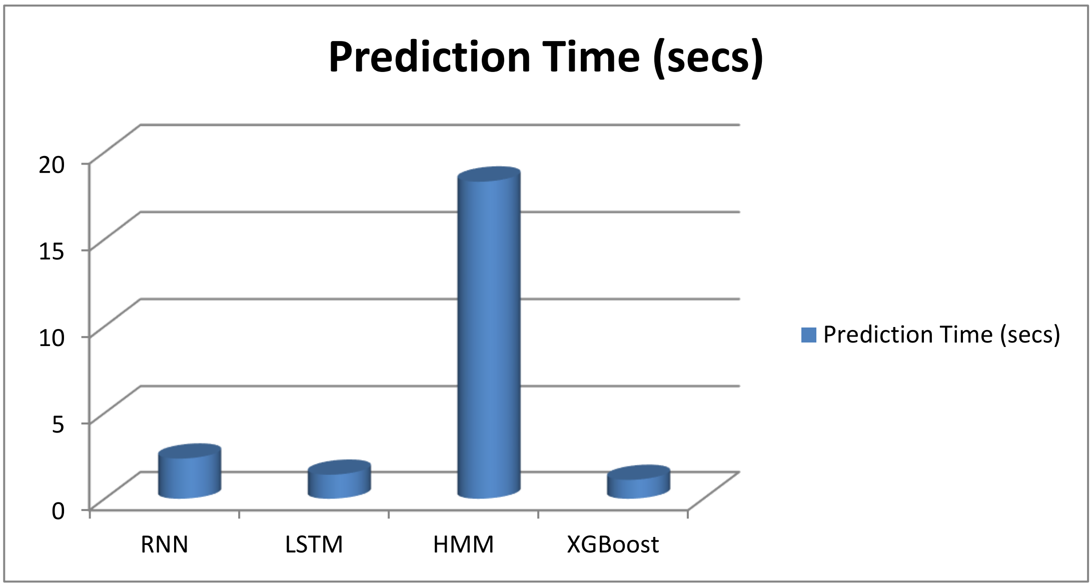


Figure 5.12. Comparison of Prediction Time

5.7 CONCLUSION

All the above figures 5.10, 5.11, 5.12 and table 5.6 can be interpreted for the performance of LSTM with other classification algorithms for prediction of outliers. The number of true positives found in LSTM is far better than number of True positives found using other techniques. Though the time taken in RNN is better than LSTM but the difference is very less as compared to the accuracy measure.

HMM shows the worst performance with maximum number of false positives and negatives. Also, the time taken for prediction is also much beyond the value of time taken by model designed by rest of the approaches. XGBoost also performs better in terms of time taken and performance wise shows similar to RNN. So, overall LSTM gets the lead both in terms of accuracy as well as time taken for the prediction.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

RFID technology has a great impact on the monitoring and controlling of processes in an automated manner. It is a great challenge to handle such a massive data in an effective manner. In this work we have proposed techniques for outlier detection and prediction successfully. The more detailed conclusion about proposed work and future insights are given in the coming sections.

6.1 CONCLUSION

The domain studied in the research work is supply chain management with RFID networks. Due to the continuous generation of data by readers and tags, there is lot of data stored by the systems attached to the readers. This data can carry lot of information about the processes involved in supply chain network and must be used effectively for analysis. Despite being the automated system, it is not full proof and lot of issues need to be checked like uncertainty in the data, generation of false positives, false negatives and redundant data, environmental and physical factors affecting the correct data information.

Data mining on such data is very important for the proper monitoring of the processes. Our research work is focussed on the path deviation related anomalies in the supply chain process which is RFID enabled. A comprehensive framework is developed by us that can handle various tasks starting from pre-processing of the data then finding outlier paths in the supply chain network and also predicting the outlier points in anticipation based on the historical data about the outliers. Here outliers can be due to many factors like change of normal scheduled path because of traffic or poor weather conditions, it can be due to any theft case also due to which the regular or normal path is deliberately avoided by the carrier of tagged objects, it may be due to malfunctioning of the carrier carrying the tagged objects due to which the objects are not read by the reader according to their scheduled time and location of read.

The proposed framework addresses the following key challenges present in tracking of RFID tagged objects:

i. Data cleaning
ii. Outlier Detection
iii. Predictive Analysis of outliers

Contribution i:

Data is generally not clean. Many tags are not detected at all or read incorrectly. Multiple reads of the same objects lead to the redundant data generation. This uncleaned data needs to be pre-processed for cleaning so that it can be further used for analysis. Many cleaning methods are discussed in detail with their pros and cons. It may be concluded from the literature survey that middleware system cleaning methods based on sliding window are the most commonly used methods and can effectively reduce number of false positives, false negatives and redundant data. The technique used in the thesis is based on the adaptive sliding window cleaning known as window sub range detection method. It is tested for single tag reading by the readers within a particular time frame in such a way that the readings shouldn't be missed and shouldn't be read when not in the range of reader. It's quite an effective method, good enough to reduce the number of errors up to 70%. Along with WSTD method which is based on statistical sampling through binomial distribution, Poisson distribution method is also used to check the probability of the number of tags that can be read within a particular time interval, given the average number of tags reads within that time interval by the readers. It can help in checking errors of tag reads for any time frame. Also, the change in sliding window size according to number of read cycles or epochs is also analysed. It can be helpful in checking the optimal number of epochs required for change in window size.

Contribution ii:

Outliers in any datasets are rare recordings as outliers or anomaly are rare events. Generally, such kind of information is not available historically and in such a scenario, a system should be available which can analyse the anomalous situation by checking the deviation from the normal state. Our second layer of the proposed framework does the same and we have used an unsupervised approach to find out outlier points in the supply chain network. The core of the dissertation uses the RFID data schema for mapping the raw data with other tables like product information, detailed location information, time hierarchy, manufacturer information and so on. The data is transformed to trajectory or path data and then this transformed data is used to check for abnormal or outlier paths. Since no training dataset is available in this layer which can have information about any previously tracked outlier points in the trajectory, a novel density-based technique is proposed to find out the outlier trajectories from the normal ones. The normal scheduled trajectories are already available and any new set of trajectories can be checked for any outlier points by doing clustering and finding out the clusters with scarce number of trajectories. They would be the outlier ones. The technique used in the thesis work is a blend of DBSCAN with dynamic time warping method. The combination of the two is giving reasonably good results in comparison with traditional clustering methods like k-means, hierarchical and basic DBSCAN. In fact, these basic clustering methods are also modified for trajectory clustering rather than point to point clustering. The unit of data to be clustered is trajectory here rather than a point object. They are all based on Euclidian distance measure unlike in our proposed algorithm where dynamic time warping measure is used and in terms of accuracy in finding the outlier trajectories, it has outperformed others. The reason for this performance is its consideration of temporal factor with variation of speed of different objects in a particular trajectory. The accuracy achieved is 91.2% which is considerably good in case of an unsupervised approach.

Contribution iii:

Once the data is available with outlier points in the trajectory, it can be exploited for further analysis and prediction of likely outlier points can be really helpful to the stakeholders in the supply chain path for avoidance of those paths which may be anomalous or consisting outlier points. Since RFID data can be expressed in terms of time series, recurrent neural networks can be the used for path prediction. Recurrent neural network is a self-learning technique which can save information about the previous time steps and predict the future time steps. Along with recurrent neural networks, methods like hidden Markov model and decision tree-based ensemble methods are also used for path prediction as studied in the literature. In the thesis work all the methods are compared for prediction of outliers in the trajectory of tagged objects in supply chain, with a variation of recurrent neural networks known as Long Short-

Term Memory (LSTM) method with tuned hyperparameters. The system was implemented using Python language environment. The other supervised methods such ad XGBoost are compared for their training time, prediction time, precision and recall. It has been observed that LSTM has outperformed others when the length of trajectory or sequence time series is too long.

6.1. FUTURE SCOPE

The combination of the above-mentioned sub-systems combine to give a full-fledged outlier detection in RFID enabled supply chain process. Detection of anomalies or outliers can also be used in to reduce congestion in the traffic, tracking of products and can be helpful to transportation engineers in forecasting traffic and designing of road networks. Still in terms of accuracy and speed of performance further improvement is needed due to size, complexity and dynamics of RFID data. In our research main emphasis is on developing algorithms for outlier detection and prediction of deviation supply chain data.

Therefore, in our view, the proposed techniques can be further enhanced in the following directions. One of possible directions of future work is to improve the efficiency of the presented algorithms by applying more sophisticated data structures like spatial indexes, R* tree in TRAJODBSCAN algorithm. We envisage optimisation of the algorithms in the continuation work. Optimisation of the proposed methods can be done by using meta-heuristic techniques. Another direction is performance enhancement with increased parallelism and distributed environment by using complex networking of supply chain nodes. The RFID data generated in the system is a massive data which is continuously generated and with the usage of big data map reduce based infrastructure, the performance can be further improved as the processing could be done in distributed environment with parallel computations.

The current work has focused upon the outliers related to deviation from the normal path but future work should not only focus on addressing the missed reads and path deviation but also differentiating between addition or reduction or stealing of products on the way to supply chain path.

The outlier framework designed can be extended to be used in the inventory management in RFID enabled supply chain process. Poisson distribution [100] can be also be used for RFID data cleaning in order to get the probability of the occurrence of the events in a particular time interval, given the average occurrence of the event within the given interval.

REFERENCES

[1] Shin, Y. J., Oh, J. S., Shin, S. H., & Jang, H. L. (2018).A Study on the countermeasures of Shipping and port logistics industry in responding to the progression of fourth industrial revolution. *Journal of Korean Navigation and Port Reserch*, 42(5), 347-355.

[2] Voulodimos, A. S., Patrikakis, C. Z., Sideridis, A. B., Ntafis, V. A., & Xylouri, E. M. (2010). A complete farm management system based on animal identification using RFID technology. Journal of *Computers and Electronics in Agriculture*, 70(2), 380-388.

[3] Ozguven, E. E., & Ozbay, K. (2012). An RFID-based inventory management framework for efficient emergency relief operations. In Proceedings of 15th International IEEE Conference on Intelligent Transportation Systems,1274-1279.

[4] Oztekin, A., Pajouh, F. M., Delen, D., & Swim, L. K. (2010). An RFID networks design methodology fir asset tracking in healthcare. Decision Support Systems, 49(1), 100-109.

[5] Lin, C. Y., & Ho, Y. H. (2009). RFID technology adoption and supply chain performance: an empirical study in China's logistics industry. International Journal Supply Chain Management, 14(5), 369-378.

[6] Bajaj, D., & Gupta, N. (2012). GPS based automatic vehicle tracking using RFID. International Journal of Engineering and Innovative Technology, 1(1), 31-35.

[7] Cole, N. A., Parker, D. B., Todd, R. W., Leytem, A. B., Dungan, R., & Ivey, S. L. (2017). 738 Use of new technologies to evaluate the environmental footprint of feedlot systems. *Journal of Animal Science*, *95*(4), 359-359.

[8] Zhu, X., Mukhopadhyay, S. K., & Kurata, H. (2012). A review of RFID technology and its managerial applications in different industries. *Journal of Engineering and Technology Management*, *29*(1), 152-167.

[9] Puffenbarger, E., Teer, F. P., & Kruck, S. E. (2007). RFID: New Technology on the Horizon for its Majors. *International Journal of Information and Communication Technology Education*, *3*(4), 50-63.

[10] Qianli, D., & Zhang, M. Y. (2016). Usage of RFID technology in supply chain: Benefits and challenges. International Journal of Applied Engineering Research, 11(5), 3720-3727.

[11] Huang, S. P., & Wang, D. (2014). Research on supply chain abnormal event detection based on the RFID technology. In Applied Mechanics and Materials, 513(2), 3309-3312.

[12] Ngai, E. W. T., Moon, K. K., Riggins, F. J., & Candace, Y. Y. (2008). RFID research: An academic literature review (1995–2005) and future research directions. International Journal of Production Economics, 112(2), 510-520.

[13] Domdouzis, K., Kumar, B., & Anumba, C. (2007). Radio-Frequency Identification (RFID) applications: A brief introduction. Advanced Engineering Informatics, 21(4), 350-355.

[14] Zhang, Y., Meratnia, N., & Havinga, P. J. (2010). Outlier detection techniques for wireless sensor networks: A survey. IEEE Communications Surveys and Tutorials, 12(2), 159-170.

[15] Rabiner, L. R., & Juang, B. H. (1986). An introduction to hidden Markov models. IEEE ASSP magazine, 3(1), 4-16.

[16] Nath, B., Reynolds, F., & Want, R. (2006). RFID technology and applications. IEEE Pervasive Computing, 5(1), 22-24.

[17] Qu, T., Yang, H. D., Huang, G. Q., Zhang, Y. F., Luo, H., & Qin, W. (2012). A case of implementing RFID-based real-time shop-floor material management for household electrical appliance manufacturers. Journal of Intelligent Manufacturing, 23(6), 2343-2356.

[18] Zhong, R. Y., Huang, G. Q., Lan, S., Dai, Q. Y., Chen, X., & Zhang, T. (2015). A big data approach for logistics trajectory discovery from RFID-enabled production data. International Journal of Production Economics, 165(3), 260-272.

[19] R. Y. Zhon, "Analysis of RFID datasets for smart manufacturing shop floors," in *In Networking,Sensing and Control(ICNSC), IEEE 15th International Conference*, Zhuhai, China, 2018.

[20] Sarac, A., Absi, N., & Dauzère-Pérès, S. (2010). A literature review on the impact of RFID technologies on supply chain management. International Journal of Production Economics, 128(1), 77-95.

[21] Han, J., Gonzalez, H., Li, X., & Klabjan, D. (2006, August). Warehousing and mining massive RFID data sets. In International Conference on Advanced Data Mining and Applications,1-18.

[22] Gonzalez, H., Han, J., & Li, X. (2006, September). Flowcube: constructing RFID flowcubes for multi-dimensional analysis of commodity flows. In Proceedings of the 32nd VLDB International conference on Very large data bases, 834-845.

[23] Gonzalez, H., Han, J., & Li, X. (2006, November). Mining compressed commodity workflows from massive RFID data sets. In Proceedings of the 15th

ACM International Conference on Information and Knowledge Management, 162-171.

[24] Masciari, E. (2007, September). A Framework for Outlier Mining in RFID data. In 11th International Database Engineering and Applications Symposium (IDEAS 2007), 263-267.

[25] Li, X., Han, J., Kim, S., & Gonzalez, H. (2007, April). Roam: Rule-and motif-based anomaly detection in massive moving object data sets. In Proceedings of the 2007 SIAM International Conference on Data Mining,273-284.

[26] Lee, J. G., Han, J., & Whang, K. Y. (2007, June). Trajectory clustering: a partition-and-group framework. In Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, 593-604.

[27] Lee, J. G., Han, J., Li, X., & Gonzalez, H. (2008). TraClass: trajectory classification using hierarchical region-based and trajectory-based clustering. In Proceedings of the VLDB Endowment, 1(1), 1081-1094.

[28] Ghosh, J. (2007). A probabilistic framework for mining distributed sensory data under data sharing constraints. In First International Workshop on Knowledge Discovery from Sensor Data.

[29] George, B., Kang, J. M., & Shekhar, S. (2009). Spatio-temporal sensor graphs (stsg): A data model for the discovery of spatio-temporal patterns. Intelligent Data Analysis, 13(3), 457-475.

[30] P. Rashidi and D. J. Cook, “An Adaptive Sensor Mining Framework for Pervasive Computing Applications,”In International Workshop on Knowledge Discovery from Sensor Data, Heidelberg, 2010.

[31] Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015, April). Long short term memory networks for anomaly detection in time series. In Proceedings Presses Universitaires de Louvain, 89.

[32] Lv, S., & Yu, S. A. (2012). Middleware-Based Algorithm for Redundant Reader Elimination in RFID Systems [J]. ACTA ELECTRONICA SINICA, 40(5), 965-970.

[33] Ziekow, H., Ivantysynova, L., & Günter, O. (2011). RFID Data Cleaning for Shop Floor Applications. In Unique Radio Innovation for the 21st Century, 143-160.

[34] He, X. U., Jie, D. I. N. G., Peng, L. I., & Wei, L. I. (2016, November). A Review on Data Cleaning Technology for RFID Network. In International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 373-382.

[35] Bai, Y., Wang, F., & Liu, P. (2006, September). Efficiently Filtering RFID Data Streams. In Proceedings of the First International VLDB Workshop on Clean Databases, 163-174.

[36] Gonzalez, H., Han, J., & Shen, X. (2007, April). Cost-conscious cleaning of massive RFID data sets. In Proceedings of 23rd IEEE International Conference on Data Engineering, 1268-1272.

[37] Shen, H., & Zhang, Y. (2008). Improved approximate detection of duplicates for data streams over sliding windows. Journal of Computer Science and Technology, 23(6), 973-987.

[38] Fan, H., Wu, Q., Lin, Y., & Zhang, J. (2013). A split-path schema-based RFID data storage model in supply chain management. Journal of Sensors, 13(5), 5757-5776.

[39] Jeffery, S. R., Garofalakis, M., & Franklin, M. J. (2006, September). Adaptive cleaning for RFID data streams. In Proceedings of the 32nd international conference on Very large data bases, 163-174.

[40] Xu, H., Shen, W., Li, P., Sgandurra, D., & Wang, R. (2017). VSMURF: A novel sliding window cleaning algorithm for RFID networks. Journal of Sensors, 1-11.

[41] Wang, Y., Song, B. Y., Fu, H., & Li, X. G. (2011). Cleaning method of RFID data stream based on Kalman filter. Journal of Chinese Computer Systems, 32(9), 1794-1799.

[42] Y. L. Wang, C. Wang and X. H. Jiang.RFID duplicate removing algorithm based on temporal-spatial Bloom Filter. Journal of Nanjing University of Science and Technology, 39(3), 253-259.

[43] Massawe, L. V., Kinyua, J. D., & Vermaak, H. (2012). Reducing false negative reads in RFID data streams using an adaptive sliding-window approach. Journal of Sensors, 12(4), 4187-4212.

[44] Zhao Q, Fränti P (2014) WB-index: A sum-of-squares based index for cluster validity. Data &Knowledge Engineering, 92(1), 77–89.

[45] Cui, D., & Zhang, Q. (2017). The RFID data clustering algorithm for improving indoor network positioning based on LANDMARC technology. Cluster Computing, 1-8.

[46] Tan, P. N., Steinbach, M., & Kumar, V. (2016). Data mining cluster analysis: basic concepts and algorithms. Introduction to Data Mining. Pearson India Publishers.

[47] Wang, H., Su, H., Zheng, K., Sadiq, S., & Zhou, X. (2013, January). An effectiveness study on trajectory similarity measures. In Proceedings of the Twenty-Fourth Australasian Database Conference-Volume, 137,13-22.

[48] Santini, S., & Jain, R. (1999). Similarity measures. IEEE Transactions on pattern analysis and machine Intelligence, 21(9), 871-883.

[49] Rucklidge, W. J. (1997). Efficiently locating objects using the Hausdorff distance. International Journal of computer vision, 24(3), 251-270.

[50] Ristad, E. S., & Yianilos, P. N. (1998). Learning string-edit distance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(5), 522-532.

[51] Eiter, T., & Mannila, H. (1994). Computing discrete Fréchet distance. Tech. Report CD-TR 94/64, Information Systems Department, Technical University of Vienna.

[52] Paterson, M., & Dančík, V. (1994, August). Longest common subsequences. In Proceedings of International Symposium on Mathematical Foundations of Computer Science,127-142.

[53] Berndt, D. J., & Clifford, J. (1994, July). Using dynamic time warping to find patterns in time series. In KDD Workshop, 10(16), 359-370.

[54] Buchin, K., Buchin, M., & Wenk, C. (2008). Computing the Fréchet distance between simple polygons. Computational Geometry, 41(1-2), 2-20.

[55] Wu, F., Fu, K., Wang, Y., Xiao, Z., & Fu, X. (2017). A spatial-temporal-semantic neural network algorithm for location prediction on moving objects. Journal of Algorithms, 10(2), 37-62.

[56] Monreale, A., Pinelli, F., Trasarti, R., & Giannotti, F. (2009, June). Wherenext: a location predictor on trajectory pattern mining. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 637-646.

[57] Inokuchi, A., Washio, T., & Motoda, H. (2000, September). An apriori-based algorithm for mining frequent substructures from graph data. In Proceedings of European conference on Principles of Data Mining and Knowledge Discovery,13-23.

[58] Erman, J., Arlitt, M., & Mahanti, A. (2006, September). Traffic classification using clustering algorithms. In Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data, 281-286.

[59] Lee, C. H., & Chung, C. W. (2010). RFID data processing in supply chain management using a path encoding scheme. IEEE Transactions on Knowledge and Data Engineering, 23(5), 742-758.

[60] Masciari, E. (2012). SMART: stream monitoring enterprise activities by RFID tags. Journal of Information Sciences, 195, 25-44.

[61] Chen, H., Zhu, Y., & Hu, K. (2010). Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning. Applied Soft Computing, 10(2), 539-547.

[62] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100-108.

[63] Cheng, Y. (1995). Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(8), 790-799.

[64] Liu, X., Shannon, J., Voun, H., Truijens, M., Chi, H. L., & Wang, X. (2014). Spatial and temporal analysis on the distribution of active radio-frequency identification (RFID) tracking accuracy with the kriging method. Journal of Sensors, 14(11), 20451-20467.

[65] Cressie, N. (1990). The origins of kriging. Journal of Mathematical geology, 22(3), 239-252.

[66] Kwon, K., Kang, D., Yoon, Y., Sohn, J. S., & Chung, I. J. (2014). A real time process management system using RFID data mining. Computers in Industry, 65(4), 721-732.

[67] Giannotti, F., Nanni, M., Pinelli, F., & Pedreschi, D. (2007, August). Trajectory pattern mining. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 330-339.

[68] Cao, H., Mamoulis, N., & Cheung, D. W. (2005, November). Mining frequent spatio-temporal sequential patterns. In Fifth IEEE International Conference on Data Mining, 8-16.

[69] Van der Aalst, W. M., Schonenberg, M. H., & Song, M. (2011). Time prediction based on process mining. Information systems, 36(2), 450-475.

[70] Li, Q., Zeng, Z., Zhang, T., Li, J., & Wu, Z. (2011). Path-finding through flexible hierarchical road networks: An experiential approach using taxi trajectory data. International Journal of Applied Earth Observation and Geoinformation, 13(1), 110-119.

[71] Yavaş, G., Katsaros, D., Ulusoy, Ö., & Manolopoulos, Y. (2005). A data mining approach for location prediction in mobile environments. Journal of Data & Knowledge Engineering, 54(2), 121-146.

[72] Morzy, M. (2007, July). Mining frequent trajectories of moving objects for location prediction. In International Workshop on Machine Learning and Data Mining in Pattern Recognition, 667-680.

[73] Cheng, C., Yang, H., King, I., & Lyu, M. R. (2012, July). Fused matrix factorization with geographical and social influence in location-based social networks. In Proceedings of 26th AAAI Conference on Artificial Intelligence, 17-25.

[74] Mathew, W., Raposo, R., & Martins, B. (2012, September). Predicting future locations with hidden Markov models. In Proceedings of the 2012 ACM conference on ubiquitous computing, 911-918.

[75] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Proceedings of Springer International Conference of Advances in Neural Information Processing Systems, 3104-3112.

[76] Aggarwal, C. C., Bhuiyan, M. A., & Al Hasan, M. (2014). Frequent pattern mining algorithms: A survey. In Proceedings of Springer International Conference of Frequent pattern mining, 19-64.

[77] Kim, M. C., Kim, C. O., Hong, S. R., & Kwon, I. H. (2008). Forward–backward analysis of RFID-enabled supply chain using fuzzy cognitive map and genetic algorithm. Expert Systems with Applications, 35(3), 1166-1176.

[78] Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In Proceedings of 2013 IEEE international conference on acoustics, speech and signal processing, 6645-6649.

[79] Trotter, M. S., & Durgin, G. D. (2010, April). Survey of range improvement of commercial RFID tags with power optimized waveforms. In Proceedings of 2010 IEEE International Conference on RFID, 195-202.

[80] Rao, J., Doraiswamy, S., Thakkar, H., & Colby, L. S. (2006, September). A deferred cleansing method for RFID data analytics. In Proceedings of the 32nd international conference on Very large data bases, 175-186.

[81] Hongsheng, Z., Jie, T., & Zhiyuan, Z. (2013, September). Limitation of RFID data cleaning method—SMURF. In Proceedings of 2013 IEEE International Conference on RFID-Technologies and Applications, 1-4.

[82] Özelkan, E. C., & Galambosi, A. (2010). Analysis of Financial Returns and Risks of Implementing RFID for Supply Chains. In Proceedings of International

Conference on Innovations in Supply Chain Management for Information Systems: Novel Approaches, 89-124.

[83] Khan, K., Rehman, S. U., Aziz, K., Fong, S., & Sarasvady, S. (2014, February). DBSCAN: Past, present and future. In Proceedings of The Fifth International Conference on the Applications of Digital Information and Web Technologies, 232-238.

[84] Danielsson, P. E. (1980). Euclidean distance mapping. Computer Graphics and Image Processing, 14(3), 227-248.

[85] Janakiram, D., Reddy, V. A., & Kumar, A. P. (2006, January). Outlier detection in wireless sensor networks using Bayesian belief networks. In Proceedings of 1st International Conference on Communication Systems Software & Middleware, 1-6.

[86] Toohey, K., & Duckham, M. (2015). Trajectory similarity measures. Sigspatial Special, 7(1), 43-50.

[87] Varatharajan, R., Manogaran, G., Priyan, M. K., & Sundarasekar, R. (2017). Wearable sensor devices for early detection of Alzheimer disease using dynamic time warping algorithm. Cluster Computing, 1-10.

[88] Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., & Keogh, E. (2008). Querying and mining of time series data: experimental comparison of representations and distance measures. In Proceedings of the VLDB Endowment, 1(2), 1542-1552.

[89] Paterson, M., & Dančík, V. (1994, August). Longest common subsequences. In International Symposium on Mathematical Foundations of Computer Science, 127-142.

[90] Li, Z., Ding, B., Han, J., Kays, R., & Nye, P. (2010, July). Mining periodic behaviors for moving objects. In Proceedings of the 16th ACM SIGKDD International conference on Knowledge Discovery and data Mining, 1099-1108.

[91] Sim, D. G., Kwon, O. K., & Park, R. H. (1999). Object matching algorithms using robust Hausdorff distance measures. IEEE Transactions on Image Processing, 8(3), 425-429.

[92] Mueen, A., & Keogh, E. (2016, August). Extracting optimal performance from dynamic time warping. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2129-2130.

[93] Beckmann, N., Kriegel, H. P., Schneider, R., & Seeger, B. (1990, May). The R*-tree: an efficient and robust access method for points and rectangles. In Proceedings of Conference ACM Sigmoid Record,19(2), 322-331.

[94] Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. Data mining and knowledge discovery, 2(2), 169-194.

[95] Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Transactions on Database Systems (TODS), 42(3), 19:1-19:21

[96] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.

[97] Khammari, A., Nashashibi, F., Abramson, Y., & Laurgeau, C. (2005, September). Vehicle detection combining gradient analysis and AdaBoost classification. In Proceedings of IEEE Intelligent Transportation Systems, 66-71.

[98] Jiang, B., Yin, J., & Zhao, S. (2009). Characterizing the human mobility pattern in a large street network. Physical Review E, 80(2), 021136:1-021136:7.

[99] Weiss, D. J., Nelson, A., Gibson, H. S., Temperley, W., Peedell, S., Lieber, A. & Mappin, B. (2018). A global map of travel time to cities to assess inequalities in accessibility in 2015. Nature, 553(7688), 333-340.

[100] Saaty, T. L. (1961). Elements of queueing theory: with applications, 34203. New York: McGraw-Hill.

[101] Labus, N., & Gajšek, B. (2018). Use of ergonomic principles in manual order picking systems. Logistics & Sustainable Transport, 9(1), 11-22.

BRIEF PROFILE OF RESEARCH SCHOLAR

Meghna Sharma is pursuing her Ph.D. in Computer Engineering from J C BOSE University of Science & Technology YMCA, Faridabad. She did her M.Tech (Computer Engineering) from Guru Jambeshwer University, Hisar and B.E. (CSE) from Chotu Ram State College of Engineering (now university), Murthal, Sonipat. She has more than 40 papers published in various reputed journals and conferences and total 40 citations in her name. She is currently working as Assistant Professor (Selection Grade) in The NorthCap University, Gurugram, Haryana, India. Prior to this, she also worked as Scientist/Engineer 'SC' in ISRO satellite centre, Bangalore. She is a recipient of Award for Science in 2007. Her areas of interests are Data Mining, Machine Learning, Database Management and Big-Data Analytics.

LIST OF PUBLICATIONS

S.No	Title of Paper	Journal/Conference/Chapter	Year Month Vol. (Issue)	Page no.	Indexing/Listing
1	Outlier Detection in RFID Datasets in Supply Chain Process: - A Review	International Journal of Computer Applications. (ISSN: 0975-8887)	2013 March 65(25)	47-51	UGC Journals
2	An Improved Algorithm for Reducing False and duplicate readings in RFID data stream based on an adaptive data cleaning scheme	International Journal of Computer Trends and Technology. (ISSN: 2231-2803)	2013 April 4(4)	944-950	UGC Journals
3	Outlier Detection in a RFID-Enabled Supply Chain Process Trajectory TRAJODBSCAN	International Journal of Data Mining and Emerging Technologies (ISSN: 2249-3212)	2018 May 8(1)	10-17	UGC Journals
4	Predictive Analysis of RFID Supply Chain Path Using Long Short-Term Memory (LSTM): Recurrent Neural Networks	International Journal of Wireless and Microwave Technologies (ISSN: 2076-1449)	2018 July 4(4)	66-77	Google Scholar

5	Performance Analysis of Clustering Algorithms of RFID datasets in SCM Process.	AICTE & CSI sponsored Int. Conf. on Advance Comm. & Tech., TITS Bhiwani, Proceedings	2013 Nov.	149-151	Google Scholar
6	Comparative study of clustering Algorithms on RFID Datasets using WEKA	National Conference on Science in Media, YMCA University of Science and Technology Faridabad.	2012 Dec.		
7	Inventory Control and Big Data	Optimal Inventory Control and Management Techniques, Business Science Reference (an imprint of IGI Global) (DOI:10.4018/978-1-4666-9888-8)	2016 June	222-235	Google Scholar