

Available online at <http://www.mecspress.net/ijwmt>

# Predictive Analysis of RFID Supply Chain Path Using Long Short Term Memory (LSTM): Recurrent Neural Networks

Meghna Sharma<sup>a\*</sup>, Manjeet Singh Tomer<sup>b</sup>

<sup>a</sup>The NorthCap University, Gurugram, Haryana, 122017, India

<sup>b</sup>YMCA University, Faridabad, Haryana, 121006

Received: 13 October 2017; Accepted: 17 January 2018; Published: 08 July 2018

---

## Abstract

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. The paper defines the architecture for the outlier detection in RFID enabled Supply Chain Path based on historical datasets. Given the training datasets, different classification models are compared for the accurate prediction of the outlieriness 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), Recurrent Neural Network(RNN) and state of art technique in RNN known as Long Short Term Memory (LSTM) is done. To our knowledge LSTM has never been used for this application area for outlier prediction. 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.

**Index Terms:** Outlier prediction, Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM), Supply Chain, Radio Frequency Identification (RFID).

© 2018 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science

---

## 1. Introduction

The domain area discussed in this paper is related to Radio Frequency Identification (RFID). **Radio-frequency identification (RFID)** is a technology to detect the presence of an object with radio waves as medium. Unlike bar code reading mechanism, line of sight is not required in RFID system and thus can be

\* Corresponding author.

E-mail address: meghnasharma@ncuindia.edu

really helpful for long distance automatic sensing/reading/identification of any tagged object, person or package within readers 'range. This is done using RFID tags. These are small *transponders* which consist of radio receiver and transmitter to transmit identity information over a distance, when asked. RFID tag reader is used to read the tag using radio waves. The readers are connected to computer systems to record the data read from the tagged object. Most RFID tags contain at least two parts. One is an integrated circuit for storing and processing information, modulating and de-modulating a *radio-frequency* (RF) signal, and other specialized functions. The second is an antenna for receiving and transmitting the signal.

This Technology can be used in various applications[21][22] which involve tracking and monitoring of objects provided with unique identification number e.g. patient monitoring, vehicle tracking in toll path, objects tracking in supply chain etc. In this paper tracking and monitoring of RFID tagged objects or group of objects with unique electronic product code (EPC) in supply chain path is discussed. It starts with the brief introduction RFID enabled supply chain process, then explanation of the pre processing and feature extraction from the raw data read by the readers. Various classification techniques are compared for the predictive modelling.

### 1.1. RFID Enabled Supply Chain Process

In the whole process of the RFID-based supply chain, all readings from RFID readers are automatically captured. In the complete process, each product item is tagged with an electronic product code (EPC) in the production line and related product specifications are written into tags. Then the tagged products are packed into cases in supplier warehouses, where the EPC tags of both the cases and the containing pallets are scanned by RFID readers when the objects come in the range of readers and then the pallets are loaded into trucks, to be distributed to dealers. Unloading of the pallets from the trucks is done at the retail stores to be finally purchased by consumers. The position of the objects is read in form of latitude and longitude by the readers, GPS sensors are attached on EPC tags. The process of reading tags by RFID readers at various positions in the supply chain process can be visualized as in Figure 1:

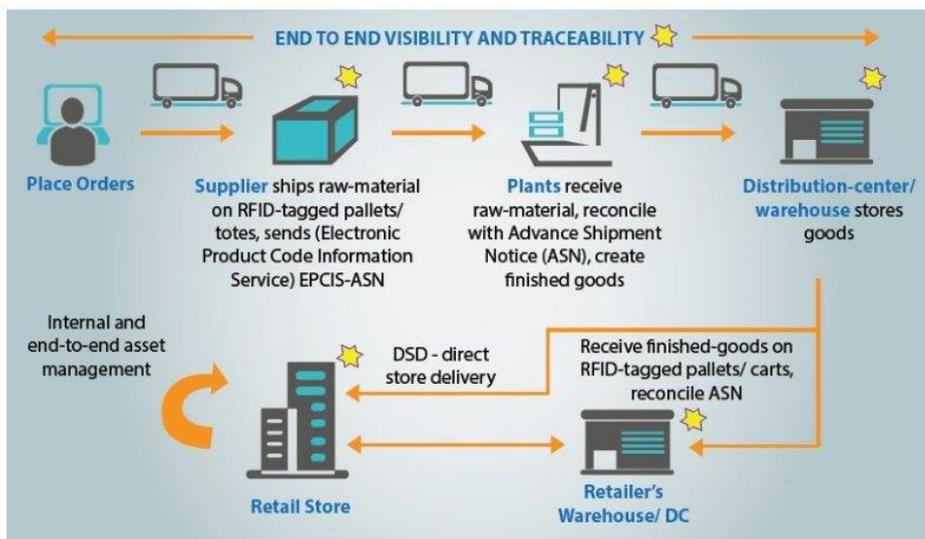


Fig.1. RFID Supply chain Process [1]

1.2. Data Model

Data read by readers from suppliers to consumers in raw form is: Tag id, Reader id, Timestamp. RFID readers keep on continuously reading the same tag id multiple numbers of times till the time any object is in the range of the reader which leads to generation of redundant data. Compression of such data can be done by dividing the timestamp into time in and time out .So after compression the data format is in the form of Tag Id, Reader id, In time, Out Time as shown in figure 2.

	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

Fig.2. Compressed raw Data Format

Here Tag Id represents unique EPC of the tag on the object, Reader id represents the reader’s Location which is mapped to longitude and latitude for further processing, In time is the time when a tagged object comes in range of reader and Out Time is the time when a tagged object goes out of the range of reader. Data is integrated from the local servers attached to the readers in the relational data format .This data 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 with the information about outlier positions with the details of latitude, longitude, path/trackid, timestamp and a class label showing whether it’s an outlier position in the previous history or not in form of 0/1.Sample from the datasets 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 .The schema for the table used is as shown in Figure 3:

	A	B	C	D	E
1	latitude	longitude	track_id	time	outlier
2	-10.93934139	-37.06274211	1	13-09-2014 07:24	0 s1
3	-10.93834606	-37.06258756	1	13-09-2014 08:25	0 m1
4	-10.94117939	-37.05763973	1	13-09-2014 09:26	1 d1
5	-10.9414799	-37.05739138	1	13-09-2014 10:12	0 d2
6	-10.94153251	-37.05734578	1	13-09-2014 10:26	0 d3
7	-10.94154221	-37.05733795	1	13-09-2014 13:00	1 d3
8	-10.94154221	-37.05733795	1	13-09-2014 21:27	1 d3

Fig.3. Supply Chain Path data

The data shown in figure 3 is converted into data in form of (latitude, longitude, track\_id, time, outlier) as feature vector for each time step in the sequence in a trajectory. Each track\_id represents a single trajectory.

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 threshold up to 10m 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) [19] 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  $n(s) = (\text{longitude}(s), \text{latitude}(s))$  and the end node  $n(e) = (\text{longitude}(e), \text{latitude}(e))$ . The point  $p(p) = (\text{longitude}(p), \text{Latitude}(p))$  of projection of longitude and latitude on road is calculated by equation (1) for any point  $n(i) = (\text{longitude}(i), \text{latitude}(i))$  on road with  $k$  as slope of the line of the road. This is a feature extraction step extracting discrete points [22] on the road so that it's much easier to develop predictive model after that. Equations (1) and (2) define the conversions.

$$\text{longitude}(p) = \frac{k \times \text{longitude}(s) + \frac{\text{longitude}(i)}{k} + \text{latitude}(i) - \text{latitude}(s)}{\frac{1}{k} + k} \quad (1)$$

$$\text{latitude}(p) = -\left(\frac{1}{k}\right) \times (\text{longitude}(p) - \text{longitude}(i)) + \text{latitude}(i) \quad (2)$$

The length of the road  $L$  can be calculated by equation (3) where  $\emptyset_1$  is the radian of latitude(s),  $\emptyset_2$  is the radian of latitude(e),  $\Delta\emptyset$  the radian of (latitude(s)-latitude(e)) and Similarly  $\phi_1$  is longitude<sub>s</sub> and  $\phi_2$  is longitude(e) and  $\Delta\phi$  is the radian of (longitude(s)-longitude(e)).  $R$  is the radius of the earth taken equal to 6371km. The fixed points in the trajectory on road can be calculated by  $n(s), n(e), k$  and the projected points.

$$L = 2 \times R \times \arcsin\left(\sqrt{\left(\sin^2\left(\frac{\Delta\emptyset}{2}\right) + \cos(\emptyset_1) * \cos(\emptyset_2) * \sin^2\left(\frac{\Delta\phi}{2}\right)\right)}\right) \quad (3)$$

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 discretised [22] by assigning time ids by diving time into intervals. The size of time is taken as 15 minutes. It creates four time ids in an hour and total 144 timeids 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.

### 1.3. Objective

The main objective of the research is to predict the outlieriness 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 recurrent neural network and Long Short term Memory (LSTM) technique is used. This approach has never been used earlier as per our knowledge in the domain of supply chain process.

Traditional models like Hidden Markov Model and Recurrent Neural Networks can manage sequences or trajectories with short lengths but for longer sequence of points in the chain, performance of traditional models may deteriorate. LSTM [2], 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. The historical data already contain a class label for checking whether a particular point is an outlier or not in terms of 0 or 1 where 0 stands for no outlier and 1 stands for an outlier position. Different traditional techniques as well as LSTM technique are compared to check the

accuracy of predicting the outlierness of any point in the trajectory path.

## 2. 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 [3] [4] [5] 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 [3] studied taxi trajectories and found their moving pattern and behavior. The trajectories here need to be converted into cells first. Jeung [4] used Apriori algorithm to extract movement patterns. An improvement of Apriori algorithm is used by Yuvus [5]. Co-occurrences of the locations can be extracted from the frequent patterns generated. A modified Prefix Span algorithm was developed by Morzy[6] as both temporal and spatial features were taken into account .So sequential pattern method was used and gave better results. I [7] [8] 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[9] proposed multi centre Gaussian model for finding the distance between the patterns generated but in this method the sequencing or ordering is not considered.. Mathew [10] proposed a hybrid hidden markov model .Jeung [11] 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 [12]. When the RNN feed forward network is unfolded, the different layers representing different time steps can be expressed... Multiple hidden layers in RNN [14] can adjust dynamically with the input of behavioral history; therefore, an RNN is suitable for modeling temporal sequence. Liu [13] 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 [15] 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 [16], machine translation [17], automatic question answering systems [18]. 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.

## 3. Proposed Architecture

The proposed architecture follows LSTM networks with input vector consisting of discretized longitude, latitude, timeid 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 sent to the stakeholders like supplier, manufacturer, distributor 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

Figure 4 shows the complete process for the prediction of the outlier point in any trajectory path of supply chain.

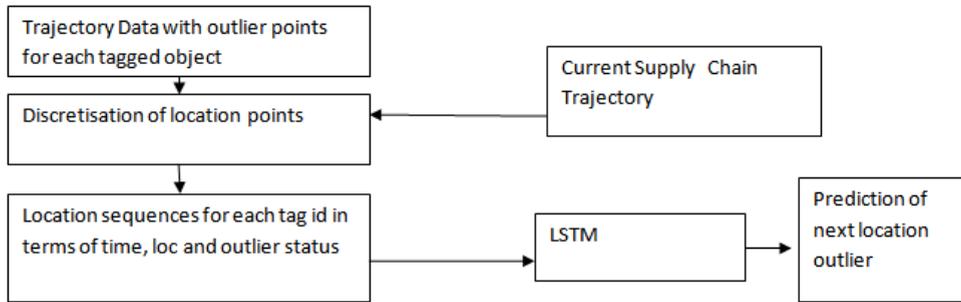


Fig.4. Outlier Prediction Analysis Process of Trajectory Point

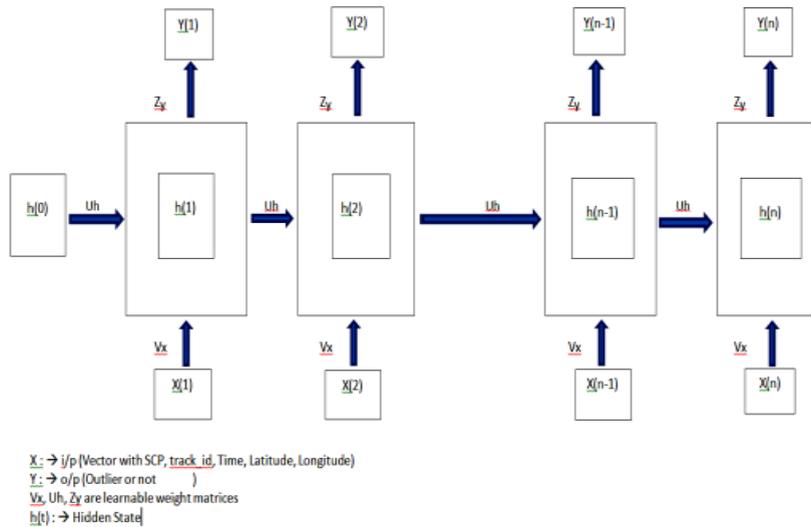


Fig.5. Recurrent Neural Network Model for RFID Supply Chain Trajectory

In figure 5, X is input vector and X(i) stands for input at time i. Its a vector with latitude, longitude as projected positions on road ,timeid 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.

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.

As discussed in previous section, 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.

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,  $g^{(t)}$ , an input gate,  $i^{(t)}$ , a forget gate,  $f^{(t)}$ , output gates,  $o^{(t)}$ , and internal state,  $s^{(t)}$ . 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.

$$g^{(t)} = \tanh \left( W^{gh} x^{(t)} + W^{gh} h^{(t-1)} + b_g \right) \quad (4)$$

$$i^{(t)} = \text{sigmoid} \left( W^{ix} x^{(t)} + W^{ih} h^{(t-1)} + b_i \right) \quad (5)$$

$$f^{(t)} = \text{sigmoid} \left( W^{fx} x^{(t)} + W^{fh} h^{(t-1)} + b_f \right) \quad (6)$$

$$o^{(t)} = \text{sigmoid} \left( W^{ox} x^{(t)} + W^{oh} h^{(t-1)} + b_o \right) \quad (7)$$

$$s^{(t)} = g^{(t)} \Theta i^{(t)} + S^{(t-1)} \Theta f^{(t)} \quad (8)$$

$$h^{(t)} = \tanh \left( S^{(t)} \right) \Theta O^{(t)} \quad (9)$$

Where  $\Theta$  is point wise multiplication and  $W^{gh}$ ,  $W^{ix}$ ,  $W^{fx}$  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, but other activation functions like ReLu and sigmoid function can be used as well. The internal state,  $s^{(t)}$  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

$s(t)$ . If  $i^{(t)}$  consists of 0s, then we completely disregard the current input. If it consists of 1s, we utilize the whole current input. Similarly, forget gates,  $f^{(t)}$ , 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 [6]. Recently, LSTMs have started to become start of the art in speech translation [1], machine translation [7], image captioning [8], and question answering systems [9]. 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. Figure 6 defines a single LSTM cell with input gate( $i_t$ ), Forget Gate( $f_t$ ) and Output Gate( $o_t$ )

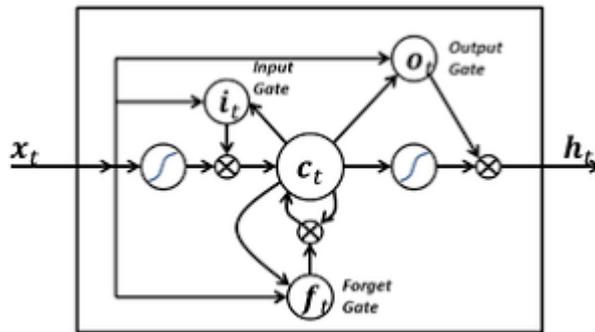


Fig.6. LSTM Cell

#### 4. Evaluation Setup

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 1 , learning rate taken as 0.01 and maximum iterations =100.

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

1. Build the computation graph for defining calculations and functions executed in runtime.
2. A Tensor Flow session is created and execution of the graph/network created in the previous step is done with the supplied data.
3. Calculation of the probability scores of each output.
4. 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.
5. Last step is of optimization. Tensorflow has optimization functions like AdamOptimizer, RMSPropOptimizer.We have chosen AdamOptimizer .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 layers is 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 and Figure 6 represents the evolution of loss minimization and classification accuracy on the training data. To make the plot readable, tensor flow plot starts from step 300. Initial loss value at step 0 as anything more than 15.0. The batch size is taken as 100. These figures run for 140, 000 iterations. Adam optimizer function with learning rate 0.0001 is chosen.

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.

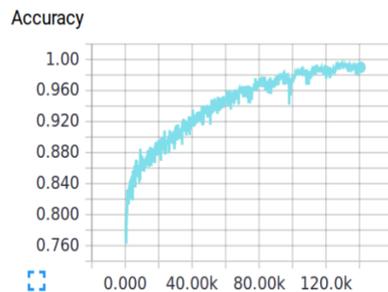


Fig.7. Accuracy Function

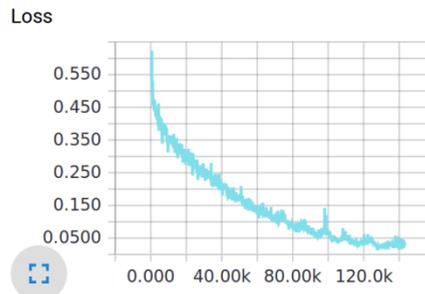


Fig.8. Loss Function

Table 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 and table 2 shows the results. 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 2.8GHz i7 CPU and 4GB RAM.

Table 2. Comparison between Different Models

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

From the table 2, we can conclude LSTM and RNN outperformed others and then the comparison of RNN and LSTM is done for the precision when the length of the trajectory is increasing, beyond 40 points, the precision starts decreasing for RNN model and is almost constant for LSTM without any effect on precision.

## 5. Conclusions and Future Scope

Performance of LSTM is best among all. Due to sequential nature of the data, all the other approaches are also compared. Though training time of LSTM is more but prediction time is not differing much. The datasets generated here is synthetically generated due to the lack of the availability of the data for privacy reasons by the supply chain stakeholders but the same approach can be applied to the real datasets in future and is expected to perform in the similar way.

## References

- [1] <https://www.linkedin.com/pulse/how-epcis-rfid-changing-e2e-visibility-traceability-supply-gaethofs>
- [2] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory, *Neural Comput.*, 9(8):1735-1780, November 1997.
- [3] Jiang, B.; Yin, J.; Zhao, S. Characterizing the human mobility pattern in a large street network. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* 2009, 80, 1711–1715. [CrossRef] [PubMed].
- [4] Jeung, H.; Liu, Q.; Shen, H.T.; Zhou, X. A hybrid prediction model for moving objects. In *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE 2008)*, Cancun, Mexico, 7–12 April 2008; pp. 70–79.
- [5] Yava s, G.; Katsaros, D.; Ulusoy, Ö.; Manolopoulos, Y. A data mining approach for location prediction in mobile environments. *Data Knowl. Eng.* 2005, 54, 121–146. [CrossRef]
- [6] Morzy, M. Mining frequent trajectories of moving objects for location prediction. In *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM 2007)*, Leipzig, Germany, 18–20 July 2007; pp. 667–680.
- [7] Li, Z.; Ding, B.; Han, J.; Kays, R.; Nye, P. Mining periodic behaviors for moving objects. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 25–28 July 2010; pp. 1099–1108.
- [8] Li, Z.; Han, J.; Ji, M.; Tang, L.A.; Yu, Y.; Ding, B.; Lee, J.G.; Kays, R. Movemine: Mining moving object data for discovery of animal movement patterns. *ACM Trans. Intell. Syst. Technol.* 2011, 2, 37.

[CrossRef]

- [9] Cheng, C.; Yang, H.; King, I.; Lyu, M.R. Fused matrix factorization with geographical and social influence in location-based social networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012.
- [10] Mathew, W.; Raposo, R.; Martins, B. Predicting future locations with hidden markov models. In Proceedings of the ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012; pp. 911–918.
- [11] Jeung, H.; Shen, H.T.; Zhou, X. Mining Trajectory Patterns Using Hidden Markov Models; Springer: Berlin/Heidelberg, Germany, 2007; pp. 470–480.
- [12] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In Tony Jebara and Eric P. Xing, editors, Proceedings of the 31<sup>st</sup> International Conference on Machine Learning (ICML-14), pages 1764-1772. JMLR Workshop and Conference Proceedings, 2014.
- [13] Liu, Q.; Wu, S.; Wang, L.; Tan, T. Predicting the next location: A recurrent model with spatial and temporal contexts. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 194–200.
- [14] Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* 1997, 9, 1735–1780. [CrossRef] [PubMed]
- [15] Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* 1997, 9, 1735–1780. [CrossRef][PubMed]
- [16] Asif, Zaheeruddin. "Integrating the supply chain with RFID: A technical and business analysis." *Communications of the Association for Information Systems* 15.1 (2005): 24.
- [17] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. CoRR, abs/1409.3215, 2014.
- [18] Mengye Ren, Ryan Kiros, and Richard S. Zemel. Image question answering: a visual semantic embedding model and a new dataset. CoRR, abs/1505.02074, 2015.
- [19] <https://www.openstreetmap.org/>
- [20] S.Valarmathy, R.Ramani, Fahim Akhtar, S.Selvaraju, G.Ramachandran, "Automatic Ration Material Distributions Based on GSM and RFID Technology", *IJISA*, vol.5, no.11, pp.47-54, 2013. DOI: 10.5815/ijisa.2013.11.05
- [21] Arshia Zaffar, Mahnoor Gul, Rida Mumtaz, Aliya Ashraf Khan, "Development of Decision Support System for Health Care Consultation using RFID Based NFC Tags for Patient Identification", *International Journal of Information Engineering and Electronic Business (IJIEEB)*, Vol.8, No.3, pp.20-30, 2016. DOI: 10.5815/ijieeb.2016.03.03
- [22] Wu, Fan, et al. "A Spatial-Temporal-Semantic Neural Network Algorithm for Location Prediction on Moving Objects." *Algorithms* 10.2 (2017): 37.

## Authors' Profiles



**Ms Meghna Sharma** is currently working as Assistant Professor (Selection Grade) in The NorthCap University, Gurgaon. She is B.E (CSE) and M.Tech (CSE) and is currently pursuing her PhD from YMCA University, Faridabad. She is a recipient of the Science Award conferred by the Government of Haryana in 2007. Her area of interests are Data Mining, Machine Learning, and Data Analytics.



**Dr Manjeet Singh Tomer** is currently working as a Professor in Department of Computer Science and Engineering in YMCA University Faridabad. His areas of interest are Artificial Intelligence, Data Mining, Soft Computing, Natural Language Processing, Computer Networks, AdHoc Networks, and Information Retrieval, Semantic web Compiler Design, and Data Structures and Algorithm Design.

**How to cite this paper:** Meghna Sharma, Manjeet Singh Tomer, " Predictive Analysis of RFID Supply Chain Path Using Long Short Term Memory (LSTM): Recurrent Neural Networks", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.8, No.4, pp. 66-77, 2018.DOI: 10.5815/ijwmt.2018.04.05