Smart Human Mobility in Smart Cities

Luís Rosa[1[0000−0001−9967−2680]], Fábio Silva[1,2[0000−0001−9872−7117]], and Cesar Analide[1[0000−0002−7796−664X]]

[1] Department of Informatics, ALGORITMI Center, University of Minho, Braga, Portugal
[2] CIICESI, ESTG, Polytechnic Institute of Porto, Felgueiras, Portugal
id8123@alunos.uminho.pt, fabiosilva@di.uminho.pt, analide@di.uminho.pt

Abstract. Nowadays, society has challenged the scientific community to find solutions able to use technology to solve the gentrification[3] of city centers. Within this context, smart cities have had an important role because they view each citizen as a data source. In the same way, the Internet of Things network increases the number of physical devices generating peta-bytes of information into a Smart city architecture. Thus an appropriate Machine Learning approach is required to process and analyze collected data. In this paper, we apply three different Machine Learning techniques such as Convolutional Neural Network (CNN), Long-Short Term Memory (LSTM), and a combined architecture, which we call CNN-LSTM, to the data generated by LinkNYC Kiosks devices — based on the city of New York —, and come to the conclusion the combined model gets better results in predicting human mobility.

Keywords: Smart Cities · Internet of Things · Machine Learning · Mobility.

1 Introduction

Historically, cities are defined as the highest forms of economic and sociocultural achievements in human civilization [13]. This fact is challenging if we analyze the growth of urban population. Today, approximately 359 million Europeans—72% of the total EU population—live in cities, towns and suburbs, and there are 26 cities with more than 1 million inhabitants. In 2050, more than 6.3B people, 60% of the population, will be living in cities [5].

Simultaneously, the gentrification of cities has led the authorities to adopt laws and policies to address problems like the planned and sustainable development and efficient management of cities. For example, with the aim to increase the efficiency of public transportation, accessibility and mobility of people through intelligent traffic management.

In the same way, a smart city is a high-speed communication hub with a strong and modern infrastructure making use of information and communication technologies (ICTs) that connects it with other cities all over the world in real time [7]. Quite simply, smart cities use Internet of Things (IoT) devices (e.g., connected sensors, lights and meters) to collect and analyze data and then use this data to improve the environmental efficiency and intelligent employment of resources. Thus, the IoT enables the city infrastructure to be intelligent.

The previous topics (i.e., Smart Cities and IoT) present several challenges and they are the reason why we provide a new approach to human mobility within cities. We propose to address the aforementioned problems by selecting a prediction model and examining the role and potential of sensing devices as one of the key pillars towards smart mobility management. By describing the overall human mobility pattern, it is possible to compare the outcomes from many locations, cities or even countries and therefore to analyze possible differences in trends of human displacements.

[3] result of wealthier people moving into a specific area

In this paper, we demonstrate how collected data can be useful to solve the mobility issues of citizens within smart cities. We do an extensive analysis of the influence of many data features to evaluate the performance of mobility prediction algorithms. Also, we develop socially responsible and efficient mobility solutions for towns and their inhabitants, particularly in the face of growing population pressure.

To illustrate a solution, we describe the use of LinkNYC Kiosks developed at CityBridge and used by millions of users to collect device data in the city of New York, United States. Although in a modest way, in this particular case we present insights into lessons learned, ways in which collected data was used by different stakeholders, and identify existing gaps and future research needs in this field.

The rest of this paper is organized as follows. Section 2 provides concepts and lessons from the literature that was used to develop the models, both to point out the many contributions of previous researchers and to place our contributions in the proper context. Section 3 presents data regarding the LinkNYC Kiosks, focusing on exploiting the patterns in the historical data, addresses prediction methods and a formal domain that supports models. Section 4 aims to narrate findings. Finally, the article presents the conclusions drawn and the lines of future work.

2 Background and Related Work

In this section, we provide background information about smart city and neural network models used in this survey, followed by a discussion of previous research related to our own.

2.1 System Overview and Deep Learning Architecture

As mentioned, the main focus of this article is to use the data for mobility solutions from a Smart City (SC) architecture. In other words, the word "smart" encompasses different adjectives, such as technological and interconnected, but also sustainable, comfortable, attractive, safe [3]. Fig. 1 presents a generic Smart City architecture with integration of LinkNYC Kiosks.

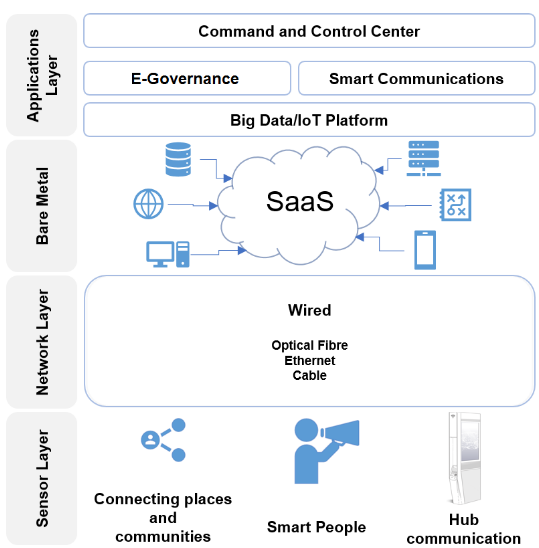

Fig. 1. Smart City multi-layered architecture (adapted from [20]).

The connectivity of a multi-layered architecture above demonstrates that cities should employ resources in an intelligent and environmentally efficient way. The Internet of Things (IoT) theory is used to manage the city's infrastructure in an intelligent way. The IoT is on the verge of bringing an immense revolution in the way we interact with "things" and all of this is going to happen using the large amounts of data that many LinkNYC Kiosks devices (New York City) are going to produce. In this phase, we can use Machine Learning (ML) to analyze the real data from a historical listing of them. But in ML if we train dependencies in data for long periods of time to predict human mobility when using big data, we should use two Deep Learning (DL) techniques: Long Short-Term Memory Network (LSTM) and Convolutional Neural Network (CNN).

LSTM (Eq. 1) is a type of Recurrent Neural Network (RNN) capable of remembering past information and taking this past information into account in the prediction of future values [14]. The internal structure of an LSTM cell is shown in Fig. 2.

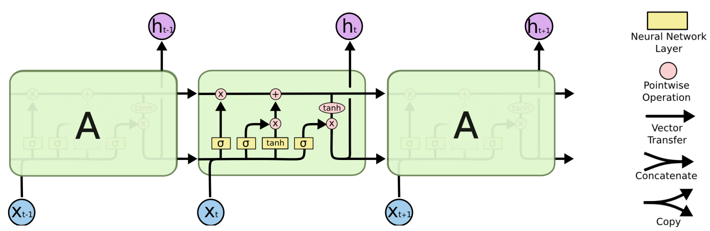

Fig. 2. The internal structure and notation of an LSTM (adapted from [2]).

Each LSTM is a set of cells where data is captured and stored [15]. The cells connect out of one module to another one conveying data from past and gathering them for the present one. Due to the use of some gates in each cell, data in each cell can be disposed of, filtered, or added to the next cells. Hence, the gates, which are based on a sigmoidal neural network layer, enable the cells to optionally let data pass through or dispose of it. Each sigmoid layer yields numbers in the range of zero and one, depicting the amount of every segment of data ought to be let through in each cell. To be more precise, an estimation of zero value implies "let nothing pass through"; whereas; an estimation of one indicates "let everything pass through". Additionally, LSTMs are also relatively insensitive to gaps (i.e., time lags between input data points) compared to other RNN models.

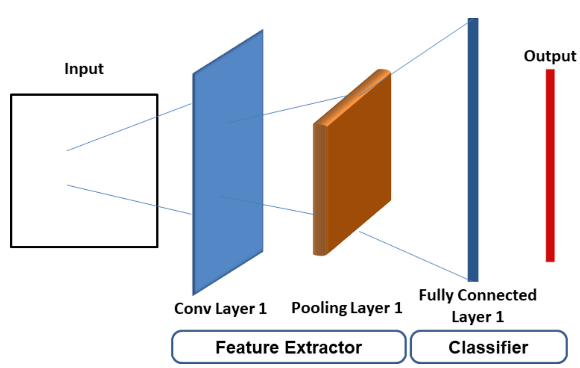

Fig. 3. The internal structure and notation of an CNN (adapted from [2]).

On the other hand, CNNs are inspired on the way the human brain responds to visual stimuli, with individual neurons responding to stimuli only in a restricted region of the visual field known as Receptive Field. Basically, they are a sequence of stacked layers [8]. We use three types of layers: Convolutional (Eq. 2), Pooling (Eq. 4) and Fully-Connected layer.

Fig. 3 is a schema of a typical CNN. The first part consists of convolutional and max-pooling layers which act as the feature extractor. However, Zero padding (Eq. 3) is a technique that allows us to preserve the original input size. This is something that we specify on a per-convolutional layer basis. The second part consists of the fully connected layer which performs non-linear transformations of the extracted features and acts as the classifier. Finally, just like LSTM, the output indicates the prediction of human mobility.

2.2 Summary of Related Work

In [10,12] neural network models are considered to exploit human mobility and improve citizens lives. Some research focuses on exploiting popularity to enhance predicting in CNN with the aim of discovering different mobility modes existing in history trajectories with the combination of route patterns and travel activity information [6]. Other works focus on how a LSTM-based method generates synthetic traffic by learning the innate structure of human mobility [11]. Based on social information contribution, in [9] human mobility is predicted by a latent allocation model which describes the probabilistic relationships between the observed trajectory data and the latent (unobserved) mobility profiles and their parameters. Or based on LinkNYC dataset [18] studies the impact of LinkNYC kiosks, on the local commercial activity in the affected neighborhoods of New York City. Also, other projects such as [16,19] that combine popularity with cooperation have also been considered.

However, the major issues with these approaches are i) most of the surveys are only based on one neural network model, ii) projects that apply to two neural networks model do not use the hybrid model to enrich their conclusions, as well as iii) the few LinkNYC kiosk projects known do not involve machine learning techniques, especially neural networks models, to present their results. For these reasons our work aims to apply deep learning techniques to LinkNYC datasets to predict human mobility, such as Convolutional Neural Network (CNN), Long-Short Term Memory (LSTM), and Hybrid model (CNN-LSTM). Hence, we constitute one of first attempts to apply the hybrid neural network model to smart human mobility.

3 Materials and Methods

This stage is focused on exploiting the patterns in the historical data and predicting performance based on LinkNYC Kiosk attributes.

3.1 Dataset

In this study we used data collected by LinkNYC Kiosks, distributed by five boroughs of New York City. These kiosks stored a historical list of their devices, location, and the status of the Link's Wi-Fi, tablets, and phones. LinkNYC is a first-of-its-kind communication network that will bring the fastest available free public Wi-Fi to millions of New Yorkers, small businesses, and visitors [4].

By mapping LinkNYC Wi-Fi data, we are able to quantify details of Wi-Fi-based location tracking, which are usually not available to the general public. For example, we enumerated value of 'up' or 'down', 1 and 2, respectively, indicating the last status of Link's dialer. The value 'null' will be assigned if Link is not activated. In the same way, we indicated date and time for last tablet status metric. The value 'no poll' will be assigned if Link is not activated or no status has been retrieved within the last hour. These are some examples of how we address the original dataset, other decisions can be checked in section 3.2.

We use a sample of the last 3 years, between 2016 and 2018, with a 1 month interval. In original data, for each day of the month we had the number of people registered at specific kiosks in a 12-hour interval. As a result we had a set of 1739 links measured between 01-2016 and 12-2018 located in different locations of New York City (see Table 1).

Table 1. LinkNYC Kiosks Locations dataset.

Borough	Date	Location	Census	Temp (°C)	Humidity (%)	Precip	Press (Pa)	Dew-point (°F)	Wind (MPH)
1	2018-02-12	(40.71, -73.95)	514260	12	50.3	0	30.3	13	2
4	2017-01-01	(40.56, -74.11)	4071876	-1	61.3	0	29.9	14	2
3	2018-07-21	(40.59, -74.10)	7984662	20	67.7	1	30.1	11	3
2	2018-05-17	(40.85, -74.10)	2208318	14	59.8	0	29.9	12	2
1	2018-03-08	(40.71, -73.95)	3387972	13	61.2	0	30.6	11	3

Note: We ignored Wi-Fi, Tablet and Phone features because they present constant values.

Data collected will allow us to learn a lot of information. Besides being able to detect devices in areas, we are also able to monitor human activity. Furthermore, we can control the people who have permission to connect to these devices, and with multiple Wi-Fi sensors, we can track their movement in real time. Additionally, these findings are deepened through preprocessing techniques.

3.2 Data Preprocessing

Data pre-processing is an important step in the data mining process. Data gathering methods are often loosely controlled, resulting in missing attribute values (e.g., install date, active date) or certain attributes of interest (e.g., Wi-Fi status, Tablet status, Phone status), impossible data combinations (e.g., install date: yes; activate date: yes), or containing only aggregate data. Thus, running an analysis should always be done before the representation and quality of data.

For preprocessing we are encouraged to use a list of tools but, assuming data preprocessing has three main components like extraction, transformation and loading, we chose the Knime [17]. Written in Java, this open source data analytics, reporting and integration platform gives us a graphical user interface to allow for the assembly of nodes for data processing. Also, it presents a modular data pipe lining, leveraging machine learning, and data mining concepts liberally for building business intelligence reports.

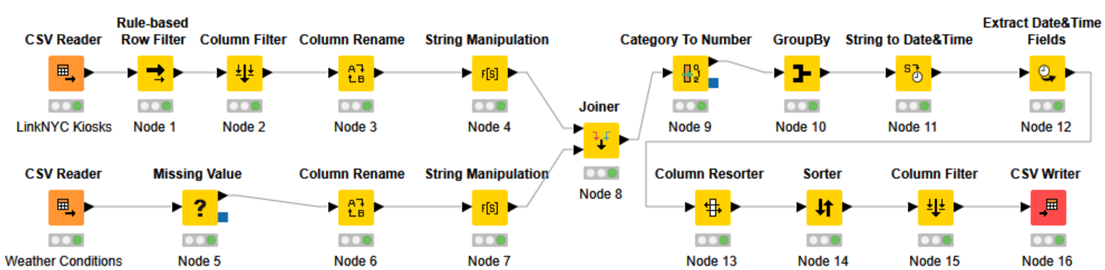

Fig. 4. The process of data preprocessing.

In Fig. 4 the first nodes import the original data from the LinkNYC Kiosks and Weather Conditions CSV files. Since there are missing values in the imported data, these are excluded in node 1. In Node 5, we fill the lack of weather data with the most frequent value. In Node 2, we only include the most important columns (i.e., Borough, Date, Location and Census). Node 3 and Node 6 modify the column names to uniformize the feature names between two datasets. In Node 4 and Node 7, the process changes the date format for easy data aggregation. Node 8 computes the importance of every course. In this phase, we merge the datasets by date. At the same time it filters the unnecessary columns to reduce the noise. In Node 10, the outcome includes the calculation of the number of people living in a particular place by data aggregation such as region, geo-coordinates and date, calculating the mean for all numeric features (e.g., temperature, humidity), but only after the category to number conversion process (e.g., weather description), in Node 9. Node 11 converts the date string to datetime format. In Node 12, the process extracts the datetime fields to *Year* and *Month (of Year)*. Before data export, we resort the columns, and filter unnecessary features of merged data. Node 16 writes the samples into a CSV file in order to use the Deep Learning (DL) algorithms. Topic 3.3 explains some scenarios where it can be applied.

3.3 Experimental Protocol

Now, to learn user habits we analyze the change in the data trend over a period of time. For example, in Table 1 the number of people who access the kiosk located at lat 40.71 and long -73.95 varies in different time series. The time series are recommended for the prediction of the future value of an item based on its past values with the help of DL algorithms [15].

We will be predicting the future number of people in New York City based on the number of tracked people via LinkNYC kiosks over the past 3 years. The calculations were conducted in the Python environment using the main packages pandas, sklearn, numpy, keras, matplotlib, seaborn. Fig. 5 shows typical time series for the number of people accessing kiosks and values are normalized using $\log_{10}$ units.

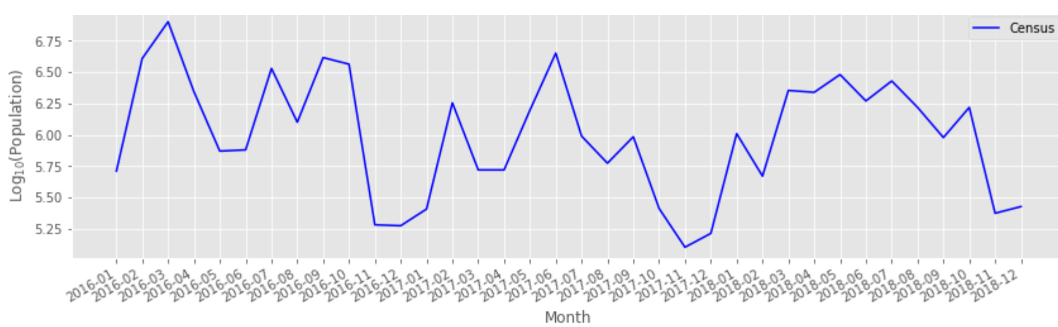

Fig. 5. Number of tracked people via LinkNYC kiosks over the past 3 years.

We can see that the trend is highly non-linear, there is a multiscalar and long-range dependence and it is very difficult to capture the trend using this information. Therefore, we need an appropriate model for time series forecasting. This is where the combined power of CNN and LSTM (or CNN-LSTM model) can be useful because the first is a forward looking technique, self learning method and predictions are more realistic while, based on past data, the second allows for the easy "memorization" of the number of tracked people for an extended number of timesteps [1].

The RNN model (i.e., LSTM) will find the patterns in the whole set of data whereas CNN also focuses on local patterns. In combination they become a switched non-linear system. The overall system learns a fixed set of rules (filters) and applies the rule to the portion of the data that it sees fit best. An overview representation of CNN-LSTM model is depicted in the following algorithm:

Algorithm 1: CNN-LSTM mobility human prediction algorithm

Input: Historical number of tracked people data
Output: Prediction for mobility human based on the number of tracked people variation

1. Start
2. Transform LinkNYC kiosks dataset to make it suitable for the model, including:
   – Transforming the data to a supervised learning problem;
   – Splitting the data into train and test sets;
   – Transforming the data so that it has the scale -1 to 1.
3. Evaluating the static model on the test data. An architecture used in our experiments is given by the LSTM (left) and CNN (right) equations:

$$\begin{gathered}\text{LSTM} : h_t^{l-1}, h_{t-1}^l, c_{t-1}^l \to h_t^l, c_t^l \\ \begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} T_{2n,4n} \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix} \\ c_t^l = f \odot c_{t-1}^l + i \odot g \\ h_t^l = o \odot \tanh(c_t^l) \end{gathered} \tag{1}$$

In these equations, sigm and tanh are applied element-wise.

$$O = \frac{W - K + 2P}{S} + 1 \tag{2}$$

$$Zero - Padding = \frac{K-1}{2} \tag{3}$$

$$\begin{gathered} W_2 = \frac{W_1 - K}{S} + 1 \\ H_2 = \frac{H_1 - K}{S} + 1 \\ C_2 = C_1 \\ O = W_2 \times H_2 \times C_2 \end{gathered} \tag{4}$$

where O is the output length, W is the input length, K is the filter size, P is the padding, and S is the stride.

4. Evaluating the static model on the test data.
5. Repeat steps 3 and 4 until optimal convergence is reached.
6. Obtain predictions by providing test data as input to the network.
7. Provide statistics of the collected RMSE (Root Mean Squared Error) scores:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\chi_i - f_i\right)^2} \tag{5}$$

Where N is the total number of observations, χ is the actual value and f_i is the predicted value.

8. End

Initially, we prepare the input data again (e.g., decrease the quantity of data, interval time series). We process the data to be used in different scenarios such as building heatmaps based on geolocation; or predicting values by Manhattan city. Nevertheless, independently of starting conditions and once the CNN-LSTM model is fit to the training data, it should provide the best accuracy of human mobility.

The accuracy of the prediction model can be estimated robustly using the RMSE metric (Eq. 5). The main benefit of using RMSE is that it penalizes large errors. It also scales the scores in the same units as the forecast values (i.e., per month for this study). In algorithm 1, we repeat the model construction and prediction several times. The average RMSE is an indication of how well our configuration would be expected to perform on unseen real-world LinkNYC kiosks data. Finally, we compare our predictions with actual trends in human mobility that can be inferred from historical data.

However, in Topic 4 we begin to explain separating the models, first LSTM model, then CNN model, and finalize with CNN- LSTM model approach. The goal is to provide the best model for time series forecasting of our survey.

4 Results

In this section, we will evaluate the prediction accuracy of data in study. In 2014, the first kiosks were installed but only started operating in late 2015. As of 2017, LinkNYC kiosks were made up of 7500 Links installed in the New York metropolitan area. In addition, there are currently 1736 active Link kiosks [4]. The output looks like this:

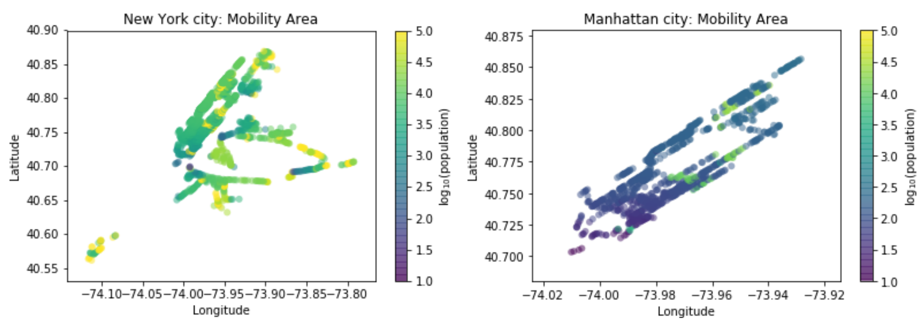

Fig. 6. Density of tracked people in New York City and the city of Manhattan via LinkNYC kiosks over the past 3 years.

4.1 Five boroughs of New York City

The used outcome for proposed algorithms is drawn in a map represented in Fig. 6. During feature extraction, we sum the census by the same time interval, and the distance threshold is the max length of each borough for better visualization. Points mapped to the same street will be represented by one density color and the location is visualized by the density color segment. For better and concise illustration, after the feature extraction, the location sequence is represented between the blue points and green points.

4.2 Manhattan city

From here, to evaluate human mobility predictions, we consider only filtered data from the city of Manhattan (Fig. 6). As detailed in section 3.3, we transform LinkNYC kiosks dataset to make it suitable for the LSTM, CNN and CNN-LSTM models. Firstly, we split supervised data into two matrices: training matrix X contains the first 24 months and testing matrix Y the rest. Matrix X is used to train and Y is used to evaluate and validate its accuracy. As we did for the training set, we need to scale our test data. Finally, we use Keras library to build and train our model.

LSTM model The model was trained with a 80 batch size, 100 nodes and 200 epochs and Adam optimization. Moreover, to develop a robust result, we repeated the experiment 20 times as that is sufficient to provide a good distribution of Root Mean Square Error (RMSE) scores. In the output, a line plot of the test data (blue), another of the predicted values (red), providing context for the model skill, and a box and whisker plot were created from the distribution shown below.

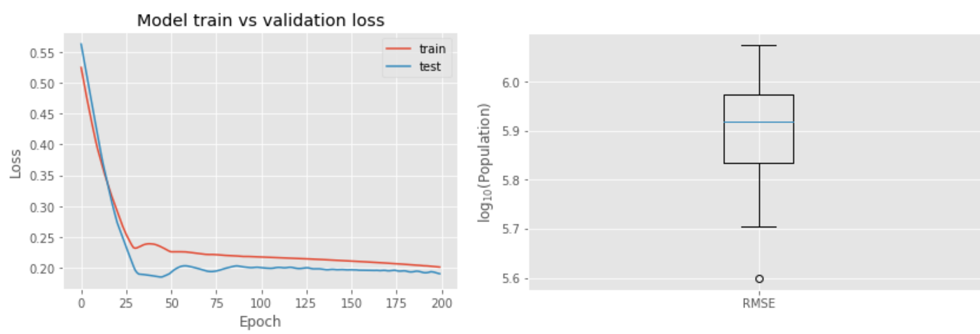

Fig. 7. Plot of Loss on the Train and Test Datasets and RMSE score in a box and whisker plot.

As observed in Fig. 7, the left side shows the train and test loss are printed at the end of each training epoch; the right side summarizes the distribution of RMSE scores. The first chart is important for predicting human mobility based on the train-line chart and test-line chart results. It shows that test loss drops below training loss, and the model is overfitting the training data. To be more precise, we can see that in the 30 epochs, the chart achieves a respectable decrease exponencial but after 50 epoch the lines are parallel.

In the case of measuring and plotting RMSE during training, and based in Table 2, we can see that the mean and standard deviation is $10^{6.16}$ and $10^{0.04}$ for the monthly number of connected people respectively, and RMSE value is $10^{0.18}$. Furthermore, it shows that this model is the second lowest among the human mobility RMSE chart images.

CNN model As has been said previously, the LSTM algorithm has advantages on predicting human mobility in the city, but the CNN model can also be used for time series forecasting problems.

In this algorithm, we develop a suite of CNN model with layers represented in Fig. 3. In other words, we choose one-dimensional CNN that has a convolutional hidden layer that operates over a 1D sequence. This is followed by perhaps a pooling layer whose job it is to distill the output of the convolutional layer to the main elements. The convolutional and pooling layers are followed by a dense fully connected layer that interprets the features extracted by the convolutional part of the model. A flatten layer is used between the convolutional layers and the dense layer to reduce the feature maps to a single one-dimensional vector.

The data is trained with 64 filters, 5 kernel size, 80 batch size and 200 epochs. To find a good result we repeat the same experiment number such as previous model. The out-of-sample results are also judged based on the $10^{0.23}$ RMSE, $10^{6.02}$ mean and $10^{0.05}$ standard deviation. As a result, we see in Fig. 8 that the model chart has a worse performance than those of the

LSTM chart. Thus, the LSTM model is more useful than the CNN model for predicting human mobility.

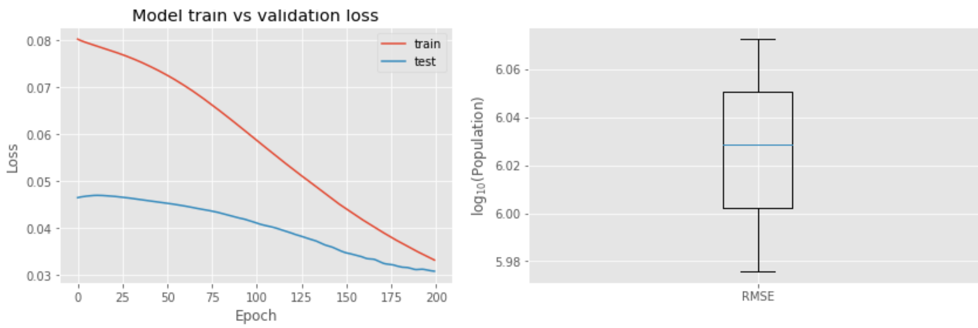

Fig. 8. Plot of Loss on the Train and Test Datasets and RMSE score in a box and whisker plot.

CNN-LSTM model For a mixed CNN-LSTM model we define the CNN model first, then add it to the LSTM model by wrapping the entire sequence of CNN layers in a TimeDistributed layer. The idea is to merge the two models, creating a third model, and with the experiments of this fusion model we compare the performances. The construction of model is based on the assumption that the used parameters are the same from the previous models.

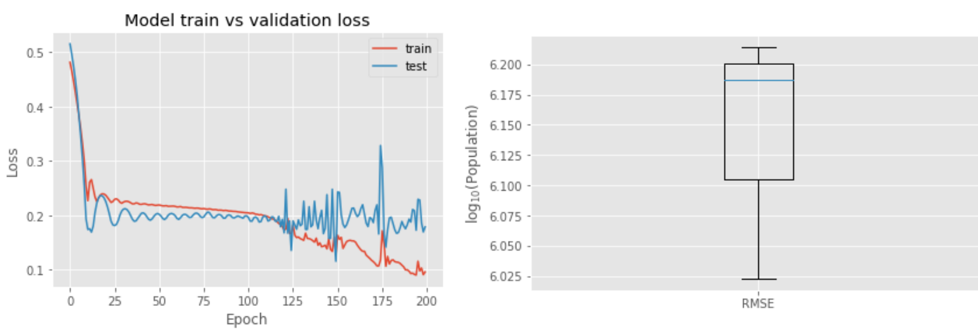

Fig. 9. Plot of Loss on the Train and Test Datasets and RMSE score in a box and whisker plot.

The loss values represented in Fig. 9 show that there is a good overfit model to 125 epochs, presenting a good performance on the train set and it keeps improving and, consequently, the RMSE decreases, while the performance on the validation set improves to a point and then begins to degrade. Next, we check which model is more appropriate for predicting human mobility.

Results Comparison To conclude, we can summarize our survey in Table 2. It shows the out-of-sample results of experiments with density human mobility in the city of Manhattan based on three different models. First, we determined the RMSE LSTM using the LSTM model to predict human mobility. Second, we showed that the CNN model performs worse than LSTM and CNN-LSTM models. Lastly, we needed to make sure that our proposed merged CNN-LSTM model is meaningful. Thus, similarly to previous results, we resort again to the RMSE equation but add other performance metrics.

Table 2. The human mobility modeling results (best values in bold).

	Model		
	LSTM	CNN	CNN-LSTM
Mean	6.16	6.02	6.14
Standard deviation	0.04	0.05	0.10
Minimum	6.07	5.93	6.02
Maximum	6.29	6.07	6.21
Accuracy$^*_{T=25}$ (%)	6.00	5.97	6.10
Accuracy$^*_{T=50}$ (%)	6.02	6.02	6.18
Accuracy$^*_{T=75}$ (%)	6.05	6.01	6.20
RMSE	**0.18**	**0.23**	**0.14**

* Train set size percentage.

We can infer from the results that, in general, the prediction accuracy of the three models improves with an increase in the size of the dataset. In its turn, comparing the models, analyzing only metrics, the out of sample results show that the CNN-LSTM model is an improvement in terms of the accuracy and variation between maximum and minimum values, respectively. We confirm that our proposed model, the CNN-LSTM model is more accurate than other models.

5 Conclusion and Future Work

In conclusion, this paper helped us understand that the Internet of Things (IoT) has an important role in the evolution of smart cities. It also aims to explain from a practical standpoint the wide applicability of Machine Learning (ML). Moreover, it shows how these three concepts can work together.

We find that the device networks from Wi-Fi Kiosks could boost efficacy in other data collection contexts. In other hand, our findings have significant privacy implications, indicating that for practical purposes WiFi data should be considered location data. That is the reason why raw data is kept anonymous and is aggregated before it can be used. Furthermore, data is usually analysed in a highly secure and protected environment. The good news is that we demonstrated that it is possible to both analyse human mobility patterns and preserve privacy.

As has been said previously, we proposed a human mobility prediction method combined with Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN) and a mixed model CNN-LSTM. We compared the loss of the observation data with the predicted results for disturbance detection. Using the number of tracked people as an example, the validation results indicated the effectiveness of the proposed approachs. The prediction based on CNN-LSTM had better performance, and is thus recommended.

There are many machine learning algorithms that can be used to learn from big data collected through a smart city's infrastructure. However, in future work, we plan to develop further in-depth studies on the LSTM to capture both short and long-term horizon forecasts. Finally, we will articulate several challenges and trending research directions for incorporating this model to adopt it in new smart city services.

References

1. Azzouni, A., Pujolle, G.: A Long Short-Term Memory Recurrent Neural Network Framework for Network Traffic Matrix Prediction (2017), http://arxiv.org/abs/1705.05690

2. blog, C.: Understanding LSTM Networks (2015), http://colah.github.io/posts/2015-08-Understanding-LSTMs/
3. Borsekova, K., Vanova, A., Vitalisova, K.: The Power of Communities in Smart Urban Development. Procedia - Social and Behavioral Sciences **223**, 51–57 (2016). https://doi.org/10.1016/j.sbspro.2016.05.289
4. Bradley, H.: Bringing History to LinkNYC Kiosks (2018), https://medium.com/urban-archive/bringing-history-to-linknyc-kiosks-17f7d63a4129
5. Brockerhoff, M., Nations, U.: World Urbanization Prospects: The 1996 Revision. Tech. Rep. 4, United Nations, New York (2006). https://doi.org/10.2307/2808041, http://www.un.org/en/development/desa/population/publications/pdf/urbanization/WUP2011{_}Report.pdf
6. Chen, R., Chen, M., Li, W., Wang, J., Yao, X.: Mobility Modes Awareness from Trajectories Based on Clustering and a Convolutional Neural Network. ISPRS International Journal of Geo-Information **8**(5), 208 (may 2019). https://doi.org/10.3390/ijgi8050208, https://www.mdpi.com/2220-9964/8/5/208
7. Deloitte: Smart Cities: The importance of a smart ICT infrastructure for smart cities. Tech. rep. (2017), https://www.stokab.se/Documents/Nyheterbilagor/SmartCityInfraEn.pdf
8. Deshpande, A.: A Beginner's Guide To Understanding Convolutional Neural Networks Part 2 (2016), https://adeshpande3.github.io/A-Beginner{%}27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/
9. Gu, Y., Mostafa, H., Prakash, B.A.: Latent Allocation Spatiotemporal Models For Indoor Human Mobility. ACM SIGPLAN Notices (2018). https://doi.org/10.1145/nnnnnnn.nnnnnnn, https://doi.org/10.1145/nnnnnnn.nnnnnnn
10. Jiang, R., Song, X., Fan, Z., Xia, T., Chen, Q., Miyazawa, S., Shibasaki, R.: DeepUrbanMomentum: An online deep-learning system for short-term urban mobility prediction. In: 32nd AAAI Conference on Artificial Intelligence, AAAI 2018 (2018), www.aaai.org
11. Kulkarni, V., Garbinato, B.: Generating synthetic mobility traffic using RNNs. pp. 1–4 (2017). https://doi.org/10.1145/3149808.3149809, https://doi.org/10.1145/3149808.3149809
12. Ouyang, X., Zhang, C., Zhou, P., Jiang, H., Gong, S.: DeepSpace: An Online Deep Learning Framework for Mobile Big Data to Understand Human Mobility Patterns (2016), https://arxiv.org/pdf/1610.07009.pdfhttp://arxiv.org/abs/1610.07009
13. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Sensing as a service model for smart cities supported by Internet of Things. Transactions on Emerging Telecommunications Technologies **25**(1), 81–93 (2014). https://doi.org/10.1002/ett.2704
14. Sepp Hochreiter, T.U.M., Jurgen Schmidhuber, I.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (1997), http://www7.informatik.tu-muenchen.de/{~}hochreithttp://www.idsia.ch/{~}juergen
15. Sherstinsky, A.: Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network p. 39 (aug 2018), http://arxiv.org/abs/1808.03314
16. Sinky, H., Khalfi, B., Hamdaoui, B., Rayes, A.: Responsive Content-Centric Delivery in Large Urban Communication Networks: A LinkNYC Use-Case. IEEE Transactions on Wireless Communications **17**(3), 1688–1699 (2018). https://doi.org/10.1109/TWC.2017.2784433, https://azure.microsoft.com/en-us/documentation/articles/cdn-pop-locationshttps://azure.microsoft.com/en-us/
17. Slah, D.: Data Mining Tools (2017), https://towardsdatascience.com/data-mining-tools-f701645e0f4c
18. Sobolevsky, S., Levitskaya, E., Chan, H., Enaker, S., Bailey, J., Postle, M., Loukachev, Y., Rolfs, M., Kontokosta, C.: Impact Of Urban Technology Deployments On Local Commercial Activity (2017), https://arxiv.org/ftp/arxiv/papers/1712/1712.00659.pdfhttp://arxiv.org/abs/1712.00659
19. Song, X., Hiroshi, K., Ryosuke, S.: DeepTransport: Prediction and simulation of human mobility and transportation mode at a citywide level. IJCAI International Joint Conference on Artificial Intelligence **2016-Janua**, 2618–2624 (2016), https://www.ijcai.org/Proceedings/16/Papers/372.pdf
20. Sterlite Tech: Developing Smart Cities with Smart City Projects - Sterlite Tech (2018), https://www.sterlitetech.com/smart-city-solutions.html