Traffic Prediction in Telecom Systems Using Deep Learning

Prashant Kaushik[1], Sajal Singh[2], Pankaj Yadav[3]

[1]*Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida, India*
[2]*Jaypee Institute of Information Technology, Noida, India*
[3]*Department of Electronics and Communication Engineering, Jaypee Institute of Information Technology, Noida, India*
[1]*prashant.kaushik@jiit.ac.in,* [2]*sajal.domain666@gmail.com;* [3]*pankaj.yadav@jiit.ac.in*

Abstract: **The deep neural network implementation in this work analyses, evaluates and generates predictions based on the open source big data of telecommunications activity released by Telecom Italia. The deep learning library used for the neural network implementation is Tensorflow which contains many high and mid-level APIs to achieve the functionality. The model uses random data from the test dataset for generating predictions and Estimator API of Tensorflow for building the neural network. Also Adam optimizer is used for optimizing the loss function with the model's resulting efficiency to be around 98.6-99.8%.**

Keywords: **Deep learning, neural networks, big data, Tensorflow, Kibana.**

I. INTRODUCTION

With the current cellular technologies rapidly advancing towards 5G-NR with IoT and self-optimization capabilities, the future networks will optimize themselves efficiently and effectively using deep learning models and algorithms. The self-organizing network (SON) should have adaptability and compatibility for scenarios with dynamic usage patterns and take pre-emptive and cognitive actions [1]. As there will be network softwarization and virtualization based on software defined networking protocols, 5G networks will be optimized and developed to deal with network traffic congestion using self-optimization schemes based on deep learning models.

Deep learning approaches and algorithms can extract features from the telecommunication activity dataset and find correlations among them. Deep neural network (DNN) is composed of hidden layers with neurons or perceptronsin each layer for feature extraction and using Adam optimizer to optimize the loss function while training and evaluating the dataset for supervised prediction [14].

The aggregate mobile internet traffic activity and load, is an important type of core network and channel state information. It contains complex patterns which need an optimised approach to find and relate those patterns for better traffic level prediction and congestion avoidance [15].

Big data analytics which consists of data pre-processing, exploratory data analysis (EDA) and optimization is important before applying deep learning models for evaluation and prediction [2]. Mobile traffic activity have irregular patterns with fluctuating and complex variations. Even so, radio access networks have become more advanced and complex, with substantial real time traffic load, resource allocation and usage patterns, etc every moment.

Various algorithms and implementations have been applied in telecom traffic systems for pattern recognition and correlation to effectively predict congestion scenarios. In [10], the cellular traffic was categorized according to voice, sms and internet data and random and conditional entropy of every type of data is investigated using temporal, spatial and service related information. In [11], a dictionary learning-based alternating direction method was used for application-level traffic learning and prediction in cellular networks. In [12], a deep belief network with multitask learning was employed for unsupervised machine learning in traffic flow prediction of transportation systems. The deep neural network approach employed in this work focuses towards optimizing the cellular network traffic and is very effective and efficient compared to other approaches.

In this implementation, a deep neural network (DNN) model is proposed for evaluating and predicting traffic activity levels based on the telecommunication activity of calls, sms and internet [3]. The deep neural network extracts features, labels and trains the data to efficiently evaluate and predict with an accuracy of 98.6 - 99.6 %. DNN is a better implementation over the machine learning algorithms such as auto-regressive integrated moving average (ARIMA) and other non-deep machine learning algorithms.

The structure of this paper is as follows. Section II describes and presents the telecommunication activity big data. Section III details the cleaning of the dataset. Section IV discusses the filling and optimising the dataset for prediction. Prediction model is described in Section V. Results and concluding remarks are described in Section VI and Section VII.

II. DATASET

The big data used in the deep neural network implementation is the open source telecommunication activity dataset released by Telecom Italia consisting of call in and out activity, sms in and out activity and internet traffic activity records and logs[1]. The dataset is spatially and geographically aggregated in

978-1-5386-4692-2/18/$31.00 ©2018 IEEE

Authorized licensed use limited to: ULAKBIM UASL - Bahcesehir University. Downloaded on July 14,2023 at 07:16:13 UTC from IEEE Xplore. Restrictions apply.

accordance with the WGS84 (EPSG:4326) standard. Whenever the user is involved in a telecommunication activity, an eNodeB handles the interaction and communication via the access network and delivers the signal information (control plane) and data (data plane) to the core network [6]. All this information and communication are recorded in the form of call detail records (CDR). The following activities are present in the dataset:

- Incoming sms activity (feature)
- Outgoing smsactivity (feature)
- Incoming call activity (feature)
- Outgoing call activity (feature)
- Internet traffic activity (feature)
- Total activity (feature)
- Activity (label)

The internet or data activity occurs and is recorded whenever the user connects or disconnects to the internet network. A CDR is generated and aggregated every 60 minutes for all the mentioned activities if the interaction or the activity occurs for more than 15 minutes or more than 5 MB internet data is consumed.

III. CLEANING THE DATASET

The dataset to be pre-processed contains CDR records generated by the Telecom Italia cellular network over the city of Milano. The telecommunication activity big data contains information logs and records for 10,000 grids about incoming and outgoing sms activity, incoming and outgoing call activity, and internet data traffic activity[13]. The dataset contains 6 numerical features about incoming and outgoing sms activity, incoming and outgoing call activity, and internet data traffic activity and the total traffic activity which is the sum of all the activities and 1 label about traffic activity level.

Data pre-processing involves data cleansing, data type conversion, and wrangling [4]. To pre-process data, following steps were performed:

- Conversion of all the feature columns into float type as part of type conversion.
- Conversion of the label column into integer type.
- Omitting ‘na’ field rows from the dataset.
- Creation of new derived column, total activity, which is the sum of incoming and outgoing sms activity, incoming and outgoing call activity, and internet data traffic activity.

IV. FILLING THE DATASET

Exploratory data analysis is the process of analyzing the data visually. It involves outlier detection, anomaly detection, missing values detection, aggregating the values, filling with stats model which use mean, medium and producing the meaningful insights using appropriate data visualization tools [7]. The platform used here to analyse and visualize data is Elasticstack with Kibana, Elasticsearch and Logstash.

Kibana is one of the UI platforms of Elasticstack software developed by Elastic and it acts as a tool to visualize, analyse, search and interact with big data which is stored in Elasticsearch indices and is integrated with Kibana. Kibana also facilitates advanced data analytics, optimization and visualization via charts, tables and heat maps.

Kibana makes it simple to comprehend and understand big data. Its web based interface operates on the localhost address and also enables creation and management of dashboards that contain all the data visualizations and search queries in near real time.

Elasticsearch is the platform of Elasticstack which acts as a analytics and search engine for the big data that is uploaded to it. It uses indices to store the uploaded data and queries to search the data via Kibana which is its UI platform. It enables analysis of complex big data and its features with ease via its simplifies yet effective API functionalities.

Logstash is the platform of Elasticstack which enables data collection in any format (csv, json, etc) and transfers this data into the Elasticsearch engine. Logstash can import data from a variety of sources such as Facebook, Twitter trends, Google, etc via editing its configuration file and enables data normalization and optimization before transferring it to Elasticsearch. The structure of Logstash uses input, filter and output plugins to input or import data in any format and hence is an effective, fast and efficient data collection engine.

Fig. 1 shows the average internet traffic activity aggregated against the total activity.

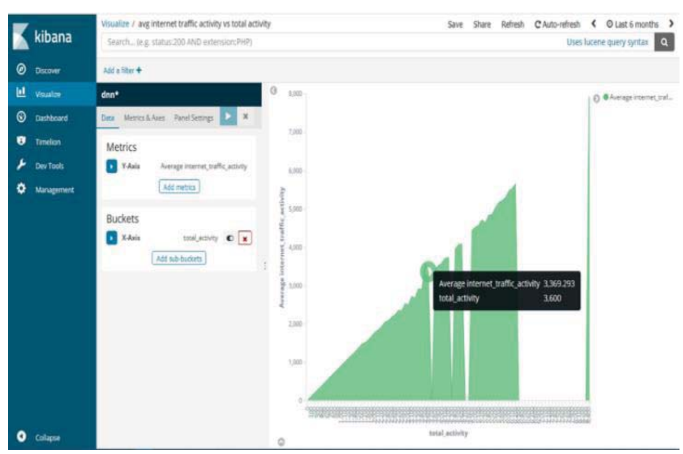

Fig. 1. Average internet traffic activity vs total activity

The average internet traffic activity constitutes a significant part of total activity compared to incoming and outgoing sms activity and incoming and outgoing call activity as evident from Fig. 1. Also the variation of internet traffic activity with

Authorized licensed use limited to: ULAKBIM UASL - Bahcesehir University. Downloaded on July 14,2023 at 07:16:13 UTC from IEEE Xplore. Restrictions apply.

total activity has sharp drops depicting the irregular patterns in the dataset.

Fig. 2 shows the average sms in activity aggregated against the sms out activity.

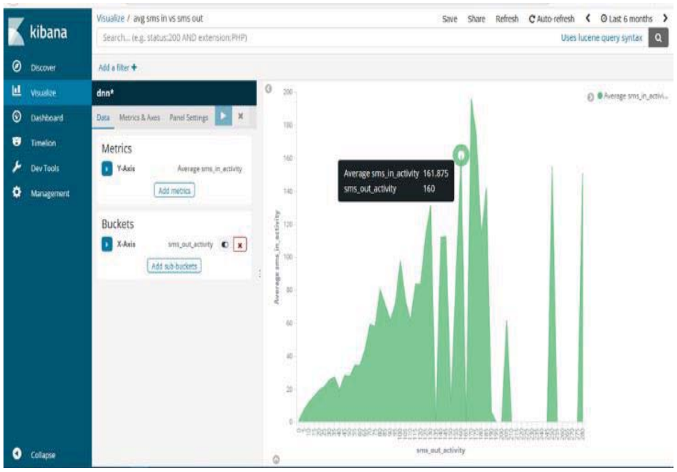

Fig. 2. Average sms in activity vs sms out activity

Clearly, it is evident from Fig. 2 that the average incoming sms activity varies non-linearly with outgoing sms activity with many sharp drops showing the irregular patterns exhibited by these two features in the big data suggesting the need of an optimized deep neural network to analyse the complex and irregular patterns among these features.

Fig.3 shows the average call in activity aggregated against the call out activity.

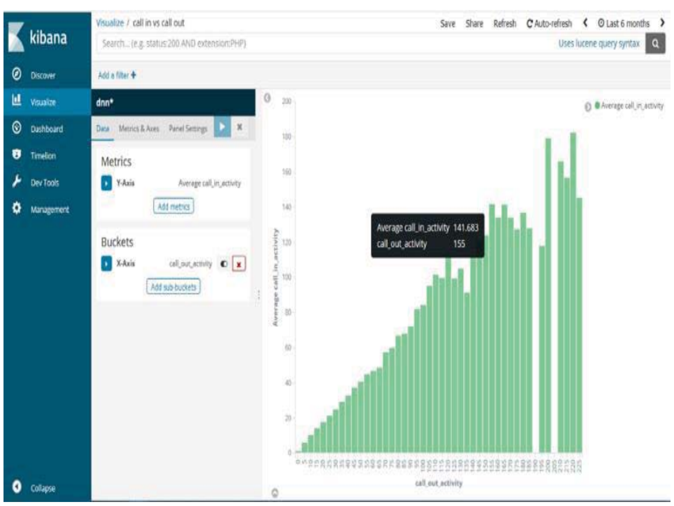

Fig. 3. Average call in activity vs call out activity

From Fig. 3 it is evident that though the variations and irregularities in the average incoming call activity with outgoing call activity is less than that of average incoming sms activity with outgoing sms activity, still there are sharp drops which need to be analysed and the corresponding relationship between these features needs to be evaluated efficiently by the deep neural network model thus illustrating the importance of feature extraction stage during the neural network implementation phase [9].

Fig. 4 shows the aggregated traffic activity level variation.

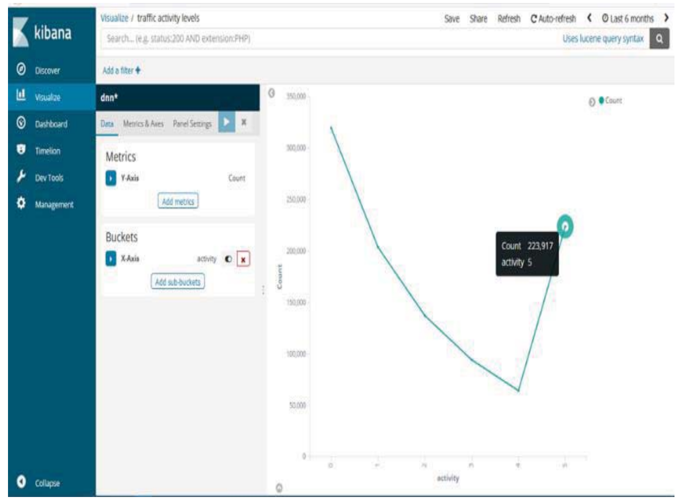

Fig. 4. Traffic level variation

Finally, the aggregated traffic level variation in Fig. 4 which is the label in the telecommunication activity big data depends on the range in which the total activity falls in. In this implementation, 6 traffic levels have been represented as labels with range interval of 20 from 0 to 100. If the total activity exceeds 100, the traffic activity level is classified as 6.

V. PREDICTION MODEL

The deep learning model used to train, evaluate andgenerate predictionsis a fully connected deep neural network implemented using the tensorflow machine learning library. Tensorflow facilitates a programming stack containing various API layers. The following two APIs have been used in this implementation:

- Estimators, which is a high-level API consisting of a variety of different functions to train and evaluate the model efficiently, accurately and effectively and generate predictions thus modelling a complete deep neural network.
- Datasets, which is another high-level API with optimized functions for data processing which is then imported by the deep neural network model via Estimators API.

The model classifies total activity based on incoming and outgoing sms activity, incoming and outgoing call activity, and internet data traffic activity, into six different traffic levels. The open source telecommunication activity big data used in this implementation comprises of 6 features, consisting of the mentioned activities and 1 label, namely the traffic activity level: level 1 traffic (0), level 2 traffic (1), level 3 traffic (2), level 4 traffic (3), level 5 traffic (4), level 6 traffic (5). The

Authorized licensed use limited to: ULAKBIM UASL - Bahcesehir University. Downloaded on July 14,2023 at 07:16:13 UTC from IEEE Xplore. Restrictions apply.

model uses float32 datatype to represent these features and int32 datatype to categorize the label.

The deep learning algorithm used in this implementation uses Tensorflow's high-level Estimator and Dataset API to train a deep neural network classifier model with the following properties:

- 7 abstract and hidden layers with ReLU activation.
- 200 multi-layer perceptrons (MLP) in each layer.

The following steps illustrate the algorithm structure and scheme:

- Data processing (Importing and reading the dataset).
- Describing the data by categorizing features and labels.
- Model training via feature extraction.
- Evaluating the model's accuracy and generating predictions.

Neural network model used in this implementation is built via feature extraction and then optimizing the model by tuning some parameters such as the number of hidden layers, the number of perceptrons in each hidden layer, the activation function used and also the optimizer used to optimize the loss function [5]. In this implementation, Adam optimizer with learning rate of 0.01 has been used and the loss function has been calculated via softmax cross entropy [8]. Fig. 5 shows this mechanism.

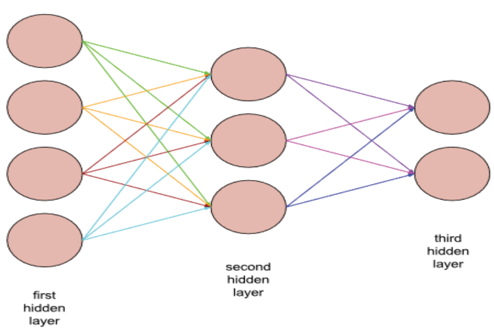

Fig. 5. A Neural network with three hidden layers

The model was tested with the following optimizers among which Adam optimizer gave the best test set accuracy:

AdaGrad: AdaGrad tunesthe learning rate η at step t for each argumentθ(i) according to previously evaluated gradientsfor θ(i). It is useful with sparse and scattered data.

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} . g_{t,i} \quad (1)$$

The main advantage of AdaGrad is that the learning rate doesn't require manual tuning with a default value of 0.01. Its disadvantage is that its learning rate-η is always decreasing and decaying.

AdaDelta: AdaDelta is an optimized version of AdaGrad and eliminates the learning rate decay encountered in AdaGrad. AdaDeltaoptimizes by minimizing and fixing the window size of previously evaluated gradients to w.

The sum of gradients is defined recursively as a decaying average of all previously squared gradients. The running average $E[g^2](t)$ at time step t then varies according to the previous mean and the present gradient: $E[g^2](t)=\gamma.E[g^2](t-1)+(1-\gamma).g^2(t)$. γ is fixed to a value as the momentum term, around 0.9.

$$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} . g_t \quad (2)$$

Adam: Adaptive Moment Estimation (Adam) evaluates adaptive and modifying learning rates for every argument. Apart from accumulating and storing an exponentially decaying average of past squared gradients like AdaDelta. Adam also keeps an exponentially decaying mean of previous gradients M(t), as in the case of momentum. M(t) and V(t) are values of the first moment which is the average and the second moment which is the uncentered variance of the gradients respectively.

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (3)$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (4)$$

The parameter update can be calculated as

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} . \widehat{m}_t \quad (5)$$

Authorized licensed use limited to: ULAKBIM UASL - Bahcesehir University. Downloaded on July 14,2023 at 07:16:13 UTC from IEEE Xplore. Restrictions apply.

VI. RESULTS

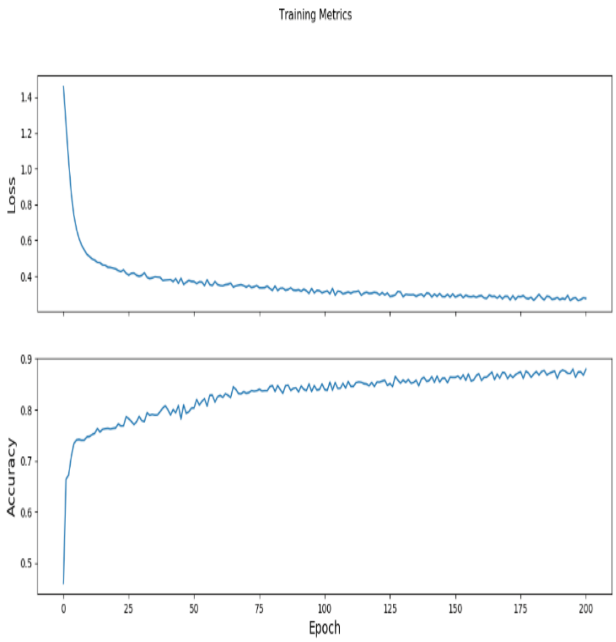

Fig. 6. Loss and accuracy with Adam optimizer

After the initial training of the model, few activity measurements are passed to the neural network model, and the model predicts the traffic level accordingly. The predicted traffic activity levels with their probabilities as computed by the model are compared with the expected traffic activity levels.The model trained in this work uses Adam optimizer to optimise the loss and gives a test set accuracy of 98-99.6 percent.

The values for $\beta 1$ is 0.9, 0.999 for $\beta 2$, and (10 x exp(-8)) for ϵ. Adam converges quickly and also enhances the learning speed of the model thus making it fast, effective and accurate and also improvises over other optimizers thereby optimizing the varying loss function.Fig. 6 shows the plot of accuracy and loss when the model uses Adam optimizer.

VII. CONCLUSION

The work done here demonstrates the use of deep neural network to predict the network traffic to help optimize the network congestion which is the basis for self-organizing networks which will be seen in 5G.

The implementation here is a deep neural network model using Tensorflow deep learning library based on telecommunication activity open source big data from 2013 released by Telecom Milano. The deep neural network implementation here is done in Python environment and the big data pre-processing is done in R programming language. Also the Exploratory Data Analysis (EDA) is done using Elasticsearch, Kibana and Logstash which is an open source software developed by Elastic for big data analysis and visualization.

The approach of this implementation uses the concept of perceptrons, neurons and hidden layers leveraging various deep learning libraries of Tensorflow like Keras, and high-level APIs like Estimator and Dataset to implement the deep neural network. Also the loss function is calculated using softmax entropy function and optimized using Adam optimizer which yield the best results. After training and evaluating the model, random activity values from the test dataset are fed into the model and the model is run. The model generates predictions for the traffic activity levels along with their probabilities which match the expected traffic levels with an efficiency of 98.6 – 99.8 percent and average loss of 0.02 – 0.03.The dataset and source code for this work can be found at https://github.com/prashant343/telcotraffic.

REFERENCES

[1] C. W. Huang, C. T. Chiang and Q. Li, "A study of deep learning networks on mobile traffic forecasting," *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Montreal, QC, 2017, pp. 1-6.

[2] J. Wang *et al.*, "Spatiotemporal modeling and prediction in

Authorized licensed use limited to: ULAKBIM UASL - Bahcesehir University. Downloaded on July 14,2023 at 07:16:13 UTC from IEEE Xplore. Restrictions apply.

cellular networks: A big data enabled deep learning approach," *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, Atlanta, GA, 2017, pp. 1-9.

[3] L. Nie, D. Jiang, L. Guo, S. Yu and H. Song, "Traffic Matrix Prediction and Estimation Based on Deep Learning for Data Center Networks," *2016 IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, 2016, pp. 1-6.

[4] R. Soua, A. Koesdwiady and F. Karray, "Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory," *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, 2016, pp. 3195-3202.

[5] R. Vinayakumar, K. P. Soman and P. Poornachandran, "Applying deep learning approaches for network traffic prediction," *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, 2017, pp. 2353-2358.

[6] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, Atlanta, GA, 2017, pp. 1-9.

[7] Y. Lv, Y. Duan, W. Kang, Z. Li and F. Y. Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865-873, April 2015.

[8] H. Ye, G. Y. Li and B. H. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," in *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114-117, Feb. 2018.

[9] Hongsuk Yi, HeeJin Jung and Sanghoon Bae, "Deep Neural Networks for traffic flow prediction," *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Jeju, 2017, pp. 328-331.

[10] X. Zhou, Z. Zhao, R. Li, Yifan Zhou and H. Zhang, "The predictability of cellular networks traffic," *2012 International Symposium on Communications and Information Technologies (ISCIT)*, Gold Coast, QLD, 2012, pp. 973-978.

[11] R. Li, Z. Zhao, J. Zheng, C. Mei, Y. Cai and H. Zhang, "The Learning and Prediction of Application-Level Traffic Data in Cellular Networks," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3899-3912, June 2017.

[12] W. Huang, G. Song, H. Hong and K. Xie, "Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191-2201, Oct. 2014.

[13] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, “A multi-source dataset of urban life in the city of Milan and the Province of Trentino.” Scientific data, vol. 2, p. 150055, 2015.

[14] P. K. Campbell, A. Christiansen, M. Dale, H. L. Ferra, A. Kowalczyk and J. Szymanski, "Experiments with simple neural networks for real-time control," in *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 2, pp. 165-178, Feb 1997.

[15] M. Fouladgar, M. Parchami, R. Elmasri and A. Ghaderi, "Scalable deep traffic flow neural networks for urban traffic congestion prediction," *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, 2017, pp. 2251-2258.

Authorized licensed use limited to: ULAKBIM UASL - Bahcesehir University. Downloaded on July 14,2023 at 07:16:13 UTC from IEEE Xplore. Restrictions apply.