Deep Learning for Reducing Redundancy in Madrid's Traffic Sensor Network

Leyuan Ding, Praboda Rajapaksha, Roberto Minerva, Noel Crespi

Deep Learning for Reducing Redundancy in Madrid's Traffic Sensor Network

Leyuan Ding[1], Praboda Rajapaksha[1,2], Roberto Minerva[1], Noel Crespi[1]
[1]Samovar, Telecom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France.
[2]Department of Computer Science, Aberystwyth University, SY23 3DB Ceredigion, UK.
Email: {leyuan.ding, praboda_rajapaksha, roberto.minerva, noel.crespi}@telecom-sudparis.eu

***Abstract*—Redundancy reduction plays a critical role in optimizing sensor network performance. This research proposes a deep-learning approach to identify and eliminate redundant sensors in a traffic network. This strategy aims to create a more cost-effective, efficient and reliable traffic monitoring system, ultimately leading to improvements in the transportation infrastructure. Leveraging traffic data from the Madrid Open Data Portal (focusing on 'District 19'), we employed sensor correlation (cosine) and similarity analysis (VGG16-based model) to identify significant correlations among sensors. This allows for accurate prediction (using Long Short-Term Memory(LSTM)-based models) of values from highly correlated sensors, leading to a potential reduction in District 19's sensor nodes by 43% (from 32 to 18) and connectivity edges by 82% (from 106 to 19). Notably, the predictive accuracy for 'highly similar' sensors achieved an average R-squared score of 0.82, validating the reliability of LSTM model predictions. These initial results encourage a larger analysis of the methodology to better prove the potential of our deep learning approach in optimizing and streamlining smart city infrastructure. This promising approach can be extended to analyze districts with higher sensor density and be adapted for application in other cities. We aim to utilize deep learning algorithms to optimize future sensor deployment planning.**

I. Introduction

Madrid, the capital of Spain, has emerged as a leading smart city, leveraging advanced technologies to enhance urban living and sustainability. With approximately 7 million residents and a rapidly growing economy, Madrid faces significant challenges in traffic management, energy consumption, and resource optimization. Rapid urbanization and technological advancement present both challenges and opportunities in managing city infrastructures. As detailed on the Madrid open data portal [1], a comprehensive network of traffic sensors serves as the cornerstone of its IoT infrastructure. This network encompasses more than 7,000 strategically positioned vehicle detectors across the city, operating at over 4,000 measurement points. These sensors capture diverse traffic-related parameters at 15-minute intervals, resulting in an annual accumulation of approximately 145 million data points [2]. However, the current deployment of traffic sensors reveals redundancy in the collected data, as observed from the assimilation of traffic data [3].

Based on our exploratory analysis, we observed that a few redundant sensors are available within the Madrid traffic sensor network. This redundancy leads to wasted resources, as data from these redundant sensors is processed unnecessarily, hindering overall network efficiency. Detecting redundant sensors is crucial for optimizing the cost-effectiveness of traffic monitoring systems through streamlined data processing, thereby enhancing network efficiency. This research aims to evaluate if some sensors can be deactivated or out into idle mode without losing the capability to derive their information content from other strongly related sensors, thereby optimizing the sensor network. The identified redundant sensors that are not considered could be put idle, but eventually reactivated in case of malfunctioning of some of the core units. This study aims to minimize the required number of sensors while preserving global information coverage. The majority of the previous research on optimizing the Madrid traffic network primarily focused on traffic network simulation [4], monitoring city traffic [2], and leveraging air pollution and atmospheric data to enhance road traffic forecasting [5]. Hence, it is important to establish a methodology for enhancing the sensor network by identifying redundant sensors and strategically deploying new ones in alternative areas to minimize their count while preserving essential data. As a result, our research proposes a two-step approach to improve the efficient **"measurability"** of future smart cities. Firstly, we employ Pearson correlation coefficient analysis and VGG16-based [6] similarity calculations to identify 'highly-correlated' traffic sensors by understanding the similarity and correlation of real traffic data. Secondly, we utilize an LSTM [7] model pre-trained with data from one traffic sensor to predict or infer traffic data from other sensors, thereby reducing redundancy within the traffic sensor network.

Our contributions can be summarized as follows:

- Raw traffic data collection from Madrid City Open Data Portal [8] and data preprocessing.
- Proposing an approach to detect redundant traffic

sensor nodes in Madrid city.
- Applying deep learning technologies to real IoT sensors data analysis, specifically utilizing VGG16 and LSTM models.
- Implementing an algorithm to predict traffic data for highly correlated traffic sensors using a pre-trained LSTM model.

Our experimental results demonstrate that the proposed 2-step algorithm achieves promising outcomes. Within the selected region of Madrid city, consisting of a total of 45 traffic sensors, our data preprocessing phase revealed that 34 out of 45 sensors were functioning correctly as of October 2023. Subsequently, upon applying the implemented algorithm, we identified 106 instances of high-similarity relations among the 34 operational traffic sensors. Additionally, we detected 18 redundant sensors, from which traffic data can be extrapolated or predicted using the pre-trained LSTM model. The average squared R score obtained from this prediction process is 0.82. In further research, we aim to apply the proposed approach to larger regions or other cities comprising a larger number of traffic sensors. We plan to: i. Identify redundant traffic sensors and reduce redundancy in this study. ii. Determine the best positions for future sensor deployment plans based on correlation analysis. iii. Explore other deep learning methods that can be applied for real-time series data analysis to predict traffic patterns. These efforts will enhance our understanding and optimization of urban traffic management systems.

The rest of the paper is structured as follows: Section II reviews related work on the application of deep learning technologies in IoT. Section III details our methodology and system architecture, which includes the dataset description and the model implementation pipeline comprising correlation analysis, similarity analysis, and prediction of traffic data using **LSTM**. Section IV discusses the conclusions drawn from our study, interprets our findings, and suggests potential future research directions. Our source code and additional resources are available.[1]

II. Related work

Various approaches of machine learning and deep learning algorithms for optimizing IoT sensor networks have been explored in the literature, focusing on the enhancement of network performance, energy efficiency and the accuracy of data collection and processing.

The application of machine learning algorithms in wireless sensor networks (WSNs), detailing the use of supervised learning methods for solving challenges in WSNs, addresses challenges such as localization, event detection, media access control, and security. The paper [9] highlights several supervised learning algorithms like K-Nearest Neighbors (K-NN), decision trees [11], neural networks, support vector machines (SVMs), and Bayesian statistics, each with its specific applications and benefits for WSN optimization. Ahmad R. et al. [14] focused on the security challenges in WSNs, noting how machine learning can address these by monitoring and making intelligent decisions. It covers the hurdles ML algorithms face in training and data requirements, proposing solutions to improve sensor's abilities to identify threats and maintain security efficiently. These studies illustrate the diverse applications of advanced learning algorithms in improving IoT networks' deployment and operation.

In a previous study, a novel data mining (NDM) strategy [15] for the elimination of data redundancy in the Internet of Things (IoT)-based Wireless Sensor Networks (WSNs) was proposed. This strategy effectively evaluates collected information to decisively eliminate duplicate and redundant packets, thereby reducing the redundancy within the WSN. Similarly, S. D. Padiya et al. [16] analyzed sixteen different methods to evaluate their redundancy, latency, computational overhead, and data accuracy. These methods included the Hierarchy Data Aggregation(HDA) [17], Opportunistic Data Aggregation(OPAG) [18], and Greedy Aggregation (GA) [19] etc, aiming to enhance the efficiency of IoT sensor networks by minimizing data redundancy. Adawy et al. introduce a method to reduce data redundancy in wireless sensor networks (WSNs) in their study "Data Redundancy Reduction in Wireless Sensor Network" [20]. They tackle excessive network traffic in Dust IoT systems through sensor clustering, transmitting representative data values instead of data from all sensors. This approach significantly cuts down on transmission data size, as demonstrated by their experimental results, offering an efficient solution to the bottleneck issues in Relay Dust Device data transfer to Smart Dust IoT Servers. Their findings enhance WSN efficiency, contributing to ongoing research efforts like those by Padiya et al. [16], who explore data aggregation techniques to reduce redundancy and boost network performance. These examples illustrate how ML and DL methodologies are being leveraged to tackle the complex challenges of IoT sensor network deployment, including efficient management of computational resources and optimization of network infrastructure to meet the increasing demands of IoT applications. C. Lanza [21] and Vélez-Serrano [22] focused on traffic data prediction in Madrid city, but there is limited research on detecting sensor similarity for redundancy reduction in the network, which is the focus of our work.

[1]Madrid's Traffic Sensor Redundancy Analysis

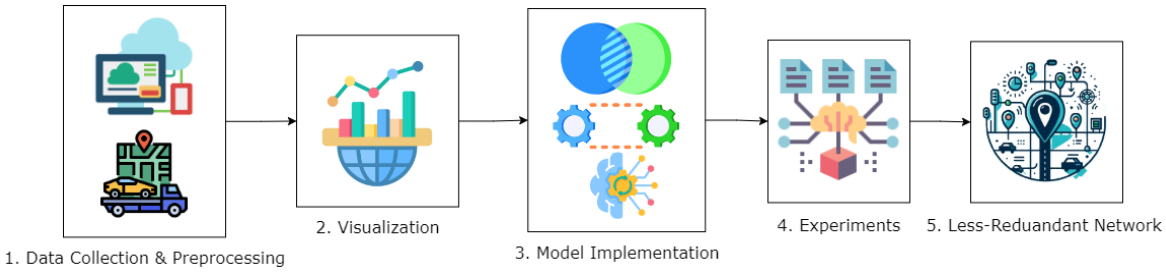


Fig. 1: **Overview of the System Architecture**

TABLE I: Traffic Dataset Description

Name	Type	Description
id	Integer	Identification of the measurement point. It is sequential, unique, and unchangeable.
fecha	Date	Official date and time of Madrid in the format dd/mm/yyyy hh:mm:ss.
tipo_elem	Text	Name of the type of measurement point: Urban or M30.
intensidad	Integer	Number of vehicles in 15 minutes, expressed in vehicles/hour. A negative value implies missing data.

The key attributes of the traffic dataset utilized in our study: the 'id' column denotes the unique identifier assigned to each traffic sensor, the 'fecha' column records the date and time of each traffic data capture and the 'intensidad' column quantifies the traffic intensity measured at 15-minute intervals.

III. Methodology and System Architecture

A. System Architecture

Figure 1 depicts the overall systems architecture of our analysis as a step-by-step process for ease of understanding the procedure and to replicate some of the steps in the generalization. It outlines the systematic approach, broken down into sequential steps: **(1)** Collection of Traffic and Geometrical Data from Madrid City Open Data Portal; **(2)** Advanced Visualization Techniques employing PythonAnywhere for 2D and 3D representations; **(3)** Cosine Similarity Analysis of traffic flows enhanced with VGG16 [24]; **(4)** LSTM [25] Model Implementation for predictive traffic patterns analysis; **(5)** Strategically minimizing redundancy within the sensor network in Figure 7b to streamline the sensor network without losing critical traffic data integrity.

B. Dataset Description

Since 2015, Madrid has been releasing public sector data on the digital platform Madrid City Council's Open Data Portal as the first European city to sign the International Open Data Charter. The platform datos.gob.es hosts an extensive catalog of data related to mobility, which currently contains 1,820 datasets grouped under a category called "Transport." In this study, we collected data from these open data platforms to analyse traffic patterns. Table I contains the dataset description and key attributes we selected in our analyses. We have chosen the traffic data in Madrid city for October 2023, which contains 12,946,685 data items. Additionally, there are 4,892 data items relevant to the geometric localization of all traffic sensors, as presented in table II.

TABLE II: Description of Geometric Data

Name	Type	Description
district	Integer	Identifier for the district where the point is located.
id	Integer	Unique and unchanging identifier for the point.
name	Text	Measurement point name, identified by street or access details for Madrid.
longitude	Real	Longitude in WGS 84 system (EPSG: 4326).
latitude	Real	Latitude in WGS 84 system (EPSG: 4326).

Geometric data analysis involved key features: 'id' and 'district' to assess the current deployment of traffic sensors, while 'longitude' and 'latitude' were employed to pinpoint the precise localization of each sensor. We have utilized this geospatial data to accurately pinpoint the real-world locations on the visualization map in Figure 2.

After analyzing the current deployment of traffic sensors, we selected *District 19* to implement the proposed approach, which includes 45 traffic sensors. This decision aims to reduce computational time and hardware resource demands, initiating a compact deep model that will subsequently transfer its learning to other regions and cities. Furthermore, to enhance our understanding of the existing sensor deployment, we have developed a map[2] for visual references that include all geometric data of the traffic sensors II. This map, as depicted in Figure 2, provides a comprehensive overview of the traffic sensor deployments across the city of Madrid. All currently deployed traffic sensors are represented by

[2]PythonAnywhere:https://www.pythonanywhere.com/

circular nodes, with different colors signifying different districts.

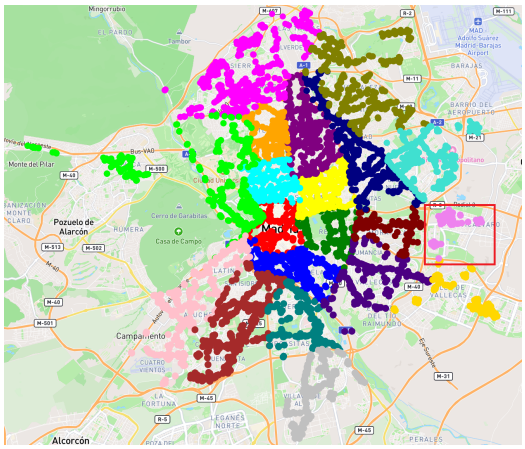


Fig. 2: Traffic Sensors' Locations
All currently deployed traffic sensors are visualized on the map, with District 19 selected for further analysis. This district is highlighted by a red rectangle on the right side of the figure. The visualization is accessible directly via Map.

We have preprocessed the dataset by splitting the data column *fecha* into two columns *date* and *time*.This helps us to extract all traffic data for a certain period and region. We used MinMaxScaler from scikit-learn for normalization [3], which is shown in Equation 1.

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

where X is the original value, X_{min} is the minimum value, X_{max} is the maximum value and X_{scaled} is the scaled value of the feature.

Figure 3 presents an example of normalized traffic data for the sensor with *id* '3695' recorded on 2023-10-02. The data comprises 96 traffic intensity values captured every 15 minutes over 24 hours. These values have been normalized to the interval (0,1), where higher values indicate greater traffic intensity.

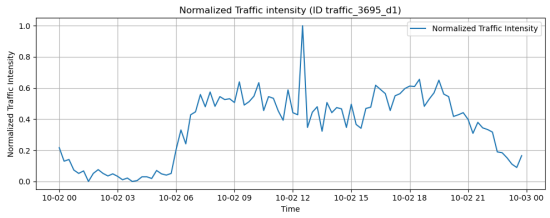


Fig. 3: Traffic data for 2023-10-02 (Sensor ID: 3695)

[3]https://scikit-learn.org

C. Model implementation Pipeline

Our model implementation pipeline comprises two primary steps aimed at identifying the similarity among different sensors in the network, thereby facilitating the identification of redundant sensors within the network. We conduct correlation analysis and similarity analysis on the traffic intensity values extracted from 45 traffic sensors located at the '*district*' '19'. We have first exported all normalized traffic data figures for one day, one week, and one month and various methods have been applied to compute traffic flow similarity.

1) Correlation Analysis: To identify traffic sensors with high similarities, we began by computing the Pearson correlation coefficient [26] of the normalized traffic data from all traffic sensors in District 19 using the following formula:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

where:

r_{xy} is the Pearson correlation coefficient between variables x and y,

x_i and y_i are individual data points,

$\bar{x}$ and $\bar{y}$ are the means of x and y, respectively,

n is the number of data points.

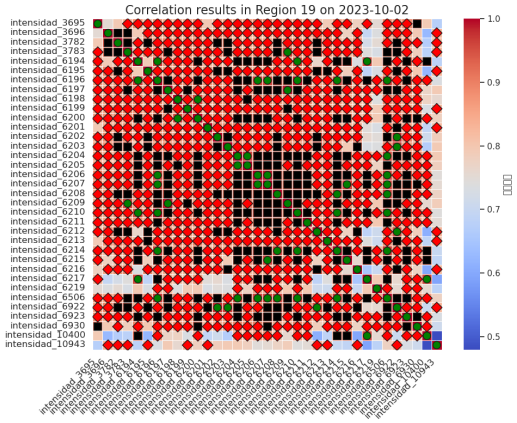


Fig. 4: Correlation Matrix results for 2nd Oct. 2023
The markers denote the correlation values of traffic sensor intensities in District 19. Red markers represent correlations ranging from 0.8 to 0.9, black markers indicate correlations between 0.9 and 0.95, and green markers signify sensor pairs with computed Pearson correlation coefficients exceeding 0.95.

As depicted in figure 4, the traffic data collected by sensors in the selected district on 2nd October 2023 exhibit a high correlation. We noted similarities in the

traffic patterns throughout a week's analysis with an average correlation of 0.77 and 0.80 during workdays.

2) Similarity analysis: We utilized two distinct similarity measures to identify sensors with similar behavioral patterns: cosine similarity and deep learning-based similarity measure.

i) Cosine Similarity

Cosine similarity [27] measures the similarity between two vectors of an inner product space. The cosine similarity between two vectors $\mathbf{A}$ and $\mathbf{B}$ can be computed using the formula 3:

$$\left(\frac{\sum_{i=1}^{n}(X_i - X_{\min}) \times (Y_i - Y_{\min})}{\sqrt{\sum_{i=1}^{n}(X_i - X_{\min})^2} \times \sqrt{\sum_{i=1}^{n}(Y_i - Y_{\min})^2}}\right. \quad (3)$$

where:

X is the original value,

$X_{\min}$ is the minimum value of the feature,

$X_{\max}$ is the maximum value of the feature,

X_{scaled} is the scaled value.

The two different vectors for traffic intensity when applying the formula3 are the normalized histograms of pixel intensities from each traffic image. Each vector represents the normalized distribution of pixel values for a single image, which is used for comparing traffic patterns through cosine similarity calculations. As shown in the figure 5, the cosine similarity values for the entire month which range from 0.98 to 1.00. These values indicate minimal differences, making it challenging to discern significant differences. Hence, we tried to explore different deep learning methods and algorithms to understand the similarity and we achieved the best performances with VGG16 model. Details of its architecture are explained below.

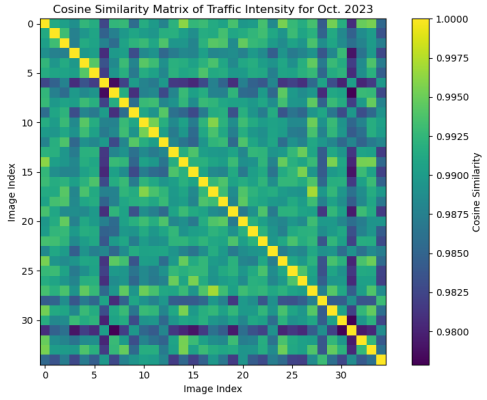


Fig. 5: Cosine Similarity Matrix results for Oct. 2023 The computed cosine similarity values for the entire month range from 0.97 to 1.00 using the formula 3.

ii) Deep learning-based similarity measure

To scale up the similarity results, we deployed a Visual Geometry Group 16 (VGG16) [24] deep neural network which comprised 13 convolutional layers and 5 Max Pooling layers, followed by convolutional layers to reduce the dimension and the number of parameters of the feature maps created by each convolution step. This model was initially trained on the ImageNet dataset, enabling it to extract high-level features from images. Leveraging its exceptional performance across diverse image classification benchmarks, VGG16 often serves as a preferred pre-trained model for transfer learning in computer vision tasks. In our study, aiming to understand the correlation and similarity among traffic data from various sensors in a better way, we employ the pre-trained VGG16 model loaded via the Keras library.

Algorithm 1 Traffic Flow Preparation

procedure VISUALIZETRAFFICINTENSITY
  Prepare save directory ('traffic_day1').
  Retrieve unique sensor IDs.
  for each sensor ID in unique IDs **do**
    Prepare sensor data for visualization:
      Combine date and time into datetime.
      Normalize intensity: $\frac{\text{intensity} - \min \text{intensity}}{\max \text{intensity} - \min \text{intensity}}$
    Plot normalized intensity versus datetime.
    Save and close the plot.
  end for
end procedure

To prepare input for the VGG model, figures representing normalized traffic intensity for each sensor on a specific day were generated using a custom Python script. In total 102 traffic flow figures are generated from *'district'* '19' sensor data, 34 for single-day(2nd Oct. 2023) data, 34 for 5 workdays in the first week(2nd Oct. - 7th Oct.) and 34 for the entire month. As stated in Algorithm 11, the process involved creating a directory to store the figures, extracting traffic data for each sensor, and converting date and time information into a DateTime format for time-series analysis. The traffic intensity values were then normalized, and plots were created to visually represent these values over time. Each plot was saved as a PNG file, which not only facilitates easy visualization but also serves as input for the VGG model. These images allow the VGG model to potentially learn and recognize patterns or anomalies in traffic conditions based on visual data, illustrating an innovative approach to applying deep learning in traffic management and analysis. Figure 6 depicts the distribution of the similarity of the sensors after employing the deep learning-based approach.

In comparison to the outcomes presented in Figure 5, the range of cosine similarity values depicted in Figure 6

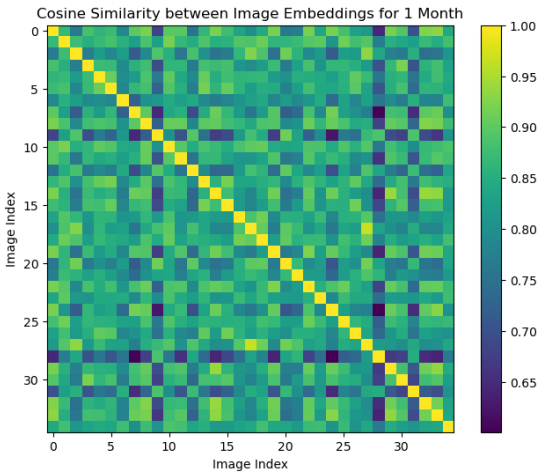


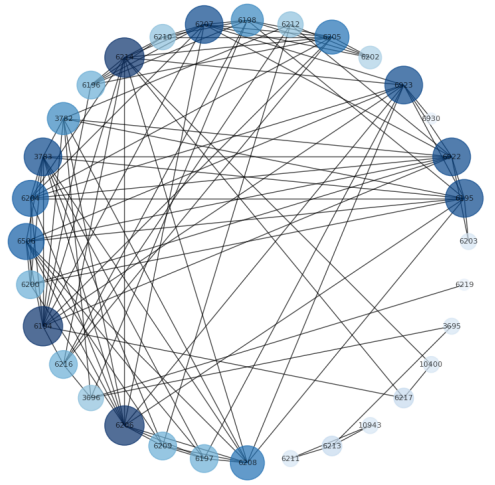
Fig. 6: Cosine Similarity Matrix results with VGG16
The figure displays the cosine similarity computed for all functioning traffic sensors within the selected district, from October 2, 2023, to October 31, 2023.

has been notably expanded from a very limited interval of (0.97, 1.00) to a more inclusive range of (0.60, 1.00). This expansion, enabled by the incorporation of the VGG16 deep learning model, significantly enhances the utility of the cosine similarity metrics for redundancy analysis. Previously constrained interpretability due to the narrow value range is now alleviated, rendering the results both more precise and substantially more informative for subsequent analytical processes.

The distribution of similarity values, derived using the VGG16-based similarity measure, demonstrates a distinct pattern: approximately half of the values surpass the established threshold of 0.85, while the remainder fall below this benchmark. This threshold of 0.85 is strategically chosen based on the dataset characteristics. Consequently, traffic sensor pairs that consistently exhibit a cosine similarity exceeding **0.85** across one-day, one-week, and one-month intervals are considered as 'highly similar sensors'.

All traffic sensors are represented by blue and orange nodes. The size of each node indicates the number of similar sensors connected to it; larger nodes have more connections. On the map in Figure 7a, isolated orange nodes can be indicative of either anomaly detection or sensors that have no similar traffic sensors during the chosen period, which could be one day, one week, or one month. Corresponding to Figure 7a, as illustrated in Figure 7b, nodes represent individual traffic sensors within the specified area, with edges connecting those deemed as "similar sensors". Larger nodes, characterized by darker colors such as '6194', '6216', and '6506', etc., denote sensors with a higher number of "similar

(a) Visualization of similar sensors in district 19



(b) Network Graph of similar sensors

Fig. 7: Visualization and Network Graph of highly similar sensors

sensors". On the contrary, smaller nodes with lighter colors, such as 6219 and 6203, represent sensors that do not exhibit significant overlap in captured traffic with other sensors.

*D. Prediction of Traffic Data using **LSTM***

By predicting future traffic patterns, the network can intelligently allocate resources, such as sensor activations/deactivations, to areas where they are most needed. This allows for efficient utilization of resources while ensuring adequate coverage and data accuracy. Predictive models can dynamically adjust the activation and deactivation of sensors based on anticipated changes in traffic patterns. This ensures that sensors are active when they are most likely to capture valuable data and can be made idle during periods of low activity to conserve energy and resources. In addition, predictive models can

Algorithm 2 LSTM-based Traffic Prediction

procedure TRAFFICPREDICTION(raw_data)
    Read the raw traffic data.
    Preprocess the data:
        a. Convert sensor IDs to string type.
        b. Select the specified sensor and target sensors.
        c. Extract relevant columns from the raw data.
        d. Drop rows with missing values.
        e. Merge date and time columns into a datetime column.
        f. Set the datetime column as the index.
        g. Split the data into features (X) and target variables (y).
        h. Scale the features using MinMaxScaler.
    Construct an LSTM model:
        a. Add an LSTM layer with specified units and activation function.
        b. Add a dense layer with specified units.
        c. Compile the model using the specified optimizer and loss function.
    Train the LSTM model:
        a. Use a different number of epochs and batch size.
        b. Monitor training loss and validation loss.
    Evaluate the model's performance:
        a. Calculate the squared R score and mean absolute error (MAE).
end procedure

be used to identify periods or specific locations where redundant sensors are less necessary due to low traffic volume or predictable patterns. Therefore, we explore sensor prediction deep learning algorithm based on Long Short-Term Memory (LSTM) [25] targeting the biggest node (sensor *'6506'*) identified in the network graph in Figure 7b, as the pilot study.

Algorithm 21 outlines the training procedure of the LSTM model utilizing traffic intensity data from the sensor *'6506'*, identified as a key node within the network topology, as depicted in Figure 7b. The model employs traffic intensity values (*'intensidad'*) as the input to predict subsequent traffic intensity values, aiming to capture short-term fluctuations in traffic flow. Specifically, the model inputs consist of current traffic intensity values, and its output is the predicted intensity for the next time step, facilitating dynamic traffic management for similar sensors. After the training phase, the LSTM model is optimized for deployment to predict traffic data for sensors depicted by smaller, lighter-colored nodes in Figure 7b, specifically *['6208', '6922', '6200', '6206', '6195', '6209', '3783', '6197', '6923', '6194']*, which demonstrate high similarity with sensor '6506'. As indicated in Table III, sensor '6506' served as the primary source of training data for the LSTM model, which is tasked with forecasting the traffic patterns of all target sensors listed.

The model's training and validation loss showed stabilization and convergence at approximately the 230th epoch, achieving minimal loss and indicating readiness for real-world deployment to assist in proactive traffic management.

TABLE III: Performance Evaluation of LSTM Model Predictions using Sensor '6506'

Selected Sensor	Target Sensors	R^2 Score	MAE
6506	6208	0.891	49.41
	6922	0.830	
	6200	0.867	
	6206	0.900	
	6195	0.774	
	6209	0.881	
	3783	0.880	
	6197	0.862	
	6923	0.831	
	6194	0.879	

The LSTM model achieved an average Squared R Score of **0.820**, demonstrating its capability to capture and explain variance in the observations from target sensors. This score reflects the model's consistent performance across different sensors and its ability to generalize effectively to unseen data. The model was trained using data from prominently featured sensors in the network graph 7b, which are depicted as larger, darker nodes, to forecast traffic for sensors represented as smaller, lighter nodes, indicative of their high similarity. The predictive accuracy for these "highly similar sensors," as listed in the 'Target Sensors' column, validates the effectiveness of the approach. This method could potentially reduce the number of necessary traffic sensors, thus optimizing the deployment of sensor networks and enhancing traffic management efficiency.

Figure 8 illustrates the comparison between actual and model-predicted traffic data for sensors ['6208', '6922', '6200', '6206', '6195', '6209', '3783', '6197', '6923', '6194'], utilizing data from sensor '6506'. The precision of these predictions confirms the model's ability to accurately forecast traffic patterns throughout the network.

In the revised network configuration depicted in figure 9, sensor '6506' emerges as a distinct, isolated node, marked in red with a reduced size, illustrating its streamlined role within the optimized network. This contrasts with the original network layout, as seen in figure 7b, where the refinement process has notably diminished the network's complexity, scaling down the sensor nodes from 32 to 18 and the connectivity edges from 106 to

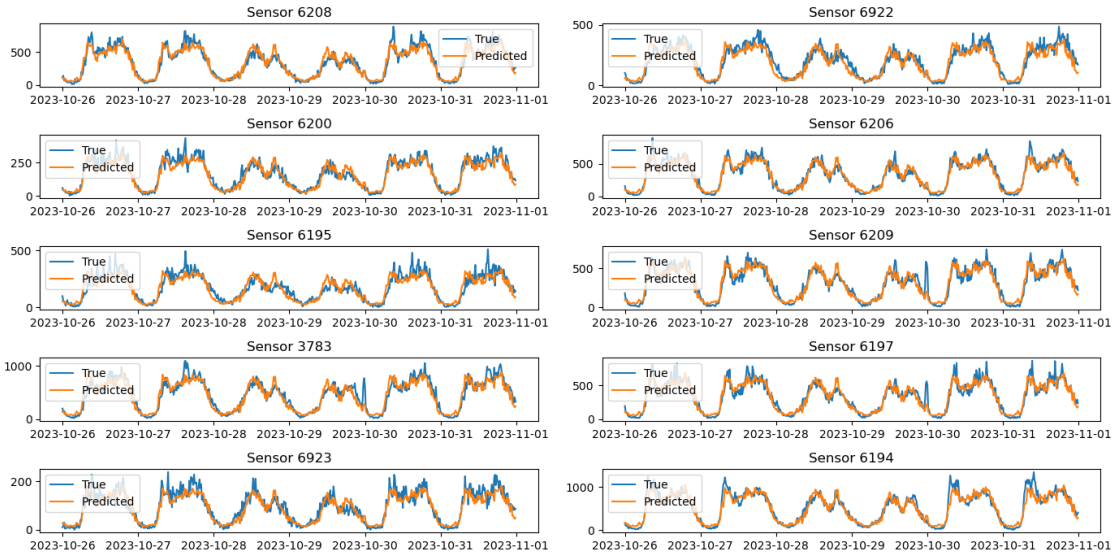


Fig. 8: Comparison of True and Predicted Traffic Data for Multiple Sensors.
This figure illustrates the actual versus predicted vehicle counts for a selection of traffic sensors: 6208, 6922, 6200, 6206, 6195, 6209, 3783, 6197, 6923, and 6194. The predictions are generated based on data from sensor 6506. The x-axis denotes the date, while the y-axis represents the number of vehicles detected by each sensor.

19. This substantial reduction underscores the efficacy of the optimization strategy.

The outcomes of this optimization affirm the pre-trained LSTM model's proficiency in accurately forecasting traffic data for sensors exhibiting high similarity. In future, we aim to implement our automated sensor scheduling algorithm to strategically deactivate certain nodes based on traffic prediction patterns and coverage redundancy within the designated region.

IV. Conlcusion

In this research, we propose an approach to identify redundant sensors in a traffic sensor network by analyzing correlations and similarity patterns among sensors. Our results demonstrate that the method significantly refines the network structure while preserving crucial traffic data, thereby enhancing the efficiency of the sensor deployment without compromising data quality. This balance between optimization and data retention provides valuable insights for smart city infrastructure development.

For future studies, we aim to: 1. Utilize traffic data from Madrid spanning a longer duration to better demonstrate the stability of our redundancy results. 2. First apply the proposed approach to other districts beyond the selected 'District 19' to analyze the correlation among a larger number of traffic sensors, and then extend this application to smart cities other than Madrid as an automated approach. This will also help identify challenges with the proposed algorithm and facilitate its

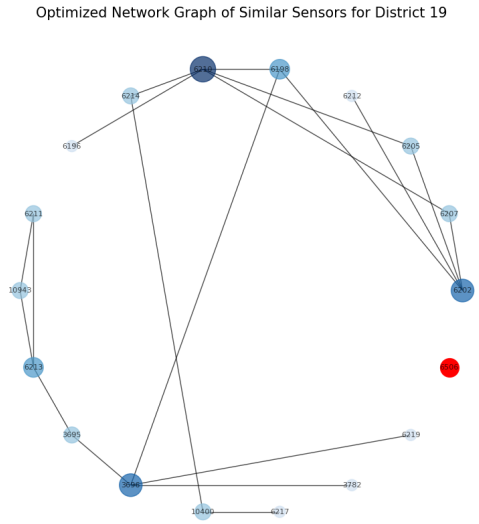


Fig. 9: Optimized Network Graph of similar sensors

improvement. 3. Select more strategic locations for future traffic sensor deployment by applying the proposed two-step algorithm.

It is important to note that our initial experiments used normal traffic data to establish a baseline performance. Future work will incorporate abnormal traffic data, such as incidents and road closures, to improve the predictive capabilities and robustness of our models in real-world scenarios.

REFERENCES

[1] Traffic location of traffic measurement points in Madrid Madrid open data portal.

[2] Pilar Rey del Castillo, ANALIZING TRAFFIC FLOWS IN MADRID CITY, 2019.

[3] Sergio Pina Lagunas, Analysis and short-term prediction of urban traffic in Madrid under the influence of multitudinary events using machine learning techniques, 2018.

[4] A. Cuadrado Torre, M. Fiore, C. Casetti, M. Gramaglia, and M. Calderon, Bidirectional Highway Traffic for Network Simulation IEEE VNC, November 2017.

[5] Faraz Malik Awan, Roberto Minerva, and Noel Crespi, Improving Road Traffic Forecasting Using Air Pollution and Atmospheric Data: Experiments Based on LSTM Recurrent Neural Networks Sensors, vol. 20, no. 13, article 3749, 2020.

[6] Tao, Jiahui and Gu, Yuehan and Sun, JiaZheng and Bie, Yuxuan and Wang, HuiResearch on vgg16 convolutional neural network feature classification algorithm based on Transfer Learning 2021 2nd China International SAR Symposium (CISS).

[7] Hochreiter, Sepp and Schmidhuber, JürgenLong Short-Term Memory Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[8] Traffic. Historical traffic data since 2013 Madrid City Open Data Portal

[9] Abu Alsheikh, Mohammad & Lin, Shaowei & Niyato, Dusit & Tan, Hwee Pink.Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications. IEEE Communications Surveys & Tutorials. 16. 2014 10.1109/COMST.2014.2320099.

[10] Pádraig Cunningham and Sarah Jane Delany. 2021. K-Nearest Neighbour Classifiers - A Tutorial ACM Comput. Surv. 54, 6, Article 128 (July 2022), 25 pages.

[11] Izza, Yacine & Ignatiev, Alexey & Marques-Silva, Joao. (2020). On Explaining Decision Trees.

[12] Schmidhuber, Jürgen Deep learning in neural networks: An overviewNeural Networks 61. 10.1016/j.neunet.2014.09.003.

[13] Tang, Yichuan. Deep Learning using Linear Support Vector Machines. (2013).

[14] Ahmad, Rami & Wazirali, Raniyah & Abu-Ain, Tarik. (2022). Machine Learning for Wireless Sensor Networks Security:An Overview of Challenges and Issues. Sensors. 22.

[15] S. Kumar and V. K. Chaurasiya A Strategy for Elimination of Data Redundancy in Internet of Things (IoT) Based Wireless Sensor Network (WSN)IEEE Systems Journal, vol. 13, no. 2, pp. 1650-1657, June 2019

[16] S. D. Padiya, V. S. Gulhane Analysis of Data Aggregation Methods to avoid Data Redundancy in Wireless Sensor Network 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), Bhimtal, India, 2020, pp. 123-129, doi: 10.1109/CICN49253.2020.9242645.

[17] Bin Zhou, Lek Heng Ngoh, Bu Sung Lee, Cheng Peng Fu, HDA: A hierarchical data aggregation scheme for sensor networksComputer Communications,Volume 29, Issue 9,2006,Pages 1292-1299, ISSN 0140-3664

[18] Zhigang Chen and Kang G. Shin OPAG: Opportunistic Data Aggregation in Wireless Sensor Networks 2008 Real-Time Systems Symposium 1052-8725/08

[19] J. Gordevicius, J. Gamper and M. Böhlen A Greedy Approach Towards Parsimonious Temporal Aggregation 2008 15th International Symposium on Temporal Representation and Reasoning, Montreal, QC, Canada, 2008, pp. 88-92, doi: 10.1109/TIME.2008.24

[20] Adawy, Mohammad & Awang Nor, Shahrudin & Mahmuddin, Massudi. Data Redundancy Reduction in Wireless Sensor Network(2018). Journal of Telecommunication. 10. 6.

[21] C. Lanza, E. Angelats, M. Miozzo and P. Dini Urban Traffic Forecasting using Federated and Continual Learning2023 6th Conference on Cloud and Internet of Things (CIoT), Lisbon, Portugal, 2023, pp. 1-8, doi: 10.1109/CIoT57267.2023.10084875.

[22] Vélez-Serrano D, Álvaro-Meca A, Sebastián-Huerta F, Vélez-Serrano J. Spatio-Temporal Traffic Flow Prediction in Madrid: An Application of Residual Convolutional Neural NetworksMathematics. 2021; 9(9):1068.

[23] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. Scikit-learn: Machine Learning in Python 2011. Journal of Machine Learning Research 12, 2825-2830

[24] Simonyan, K., and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition 3rd International Conference on Learning Representations (ICLR 2015) Computational and Biological Learning Society, 2015, pp. 1–14.

[25] Hochreiter, Sepp & Schmidhuber, Jürgen. Long Short-term Memory (1997) Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[26] Freedman, David and Pisani, Robert and Purves, Roger Statistics (international student edition)Pisani, R. Purves, 4th edn. WW Norton & Company, New York 2007

[27] Han, Jiawei and Kamber, Micheline and Pei, Jian Data Mining: Concepts and Techniques 2011.