Received April 7, 2022, accepted April 20, 2022, date of publication April 22, 2022, date of current version May 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3169890

Graph Based Learning for Building Prediction in Smart Cities

ASIER GARMENDIA-ORBEGOZO[1], (Member, IEEE), SARAH NOYE[2], (Member, IEEE), MIGUEL ANGEL ANTON[2], (Senior Member, IEEE), AND J. DAVID NUÑEZ-GONZALEZ[1], (Senior Member, IEEE)

[1]Department of Applied Mathematics, University of the Basque Country, 48940 Leioa, Spain
[2]TECNALIA, Basque Research and Technology Alliance (BRTA), 20018 Donostia-San Sebastian, Spain

Corresponding author: Asier Garmendia-Orbegozo (asier.garmendiao@ehu.eus)

This work was supported by the TECNALIA Member of Basque Research and Tecnology Alliance.

ABSTRACT Anticipating pedestrians' activity is a necessary task for providing a safe and energy efficient environment in an urban area. By locating strategically sensors throughout the city useful information could be obtained. By knowing the average activity of those throughout different days of the week we could identify the typology of the buildings neighboring those sensors. For these type of purposes, clustering methods show great capability forming groups of items that have great similarity intra clusters and dissimilarity inter cluster. Different approaches are made to classify sensors depending on the typology of buildings surrounding them and the mean pedestrians' counts for different time intervals. By this way, sensors could be classified in different groups according to their activation patterns and the environment in which they are located through clustering processes and using graph convolutional networks. This study reveals that there is a close relationship between the activity pattern of the pedestrians' and the type of environment sensors that collect pedestrians' data are located. By this way, institutions could alleviate a great amount of effort needed to ensure safe and energy efficient urban areas, only knowing the typology of buildings of an urban zone.

INDEX TERMS Building prediction, clustering, graph networks, smart city, sensors.

I. INTRODUCTION

The increasing number of Internet of Things (IoT) devices spread throughout cities has evolved in a scenario in which different public services has developed positively, in the way that dynamically information is provided and decision are made in real-time. As a consequence, citizen's lifestyle has become safer, more convenient and environmental issues could be faced up more efficiently.

In a smart city, sensors play the role of collectors, obtaining a huge number of data by sensing different parameters of the environment and different events, such as traffic incidents or pedestrians' mobility. The elevate activity of these devices carries with it a proper maintenance and an intelligent distribution so as to avoid different problems related to safety and energy consumption and to tackle security issues. According to the International Energy Agency (2015), the implementation of a correct control illumination system could save energy by up to 35%. These numbers are behind the increasing use of more advanced control illumination systems mainly in the commercial and public sectors where lighting represents the highest energy consumption.[1]

The positioning of sensors and the distributed management of them is determining. Thus, it is important to make a correct classification of the building typologies, if those could give us the information of the environment they are placed or the pedestrians' tendencies or patterns of mobility.

In this work, which is an extension of "Building typology prediction in Smart Cities" presented in the CIB W78 Information Technology for Construction 39th Conference WBC 2022 [20], two clusterings of sensors were developed. One of them was performed based on the typology of the buildings near-by the sensors, and the second based on the average counts that sensors had made during different time slots throughout different days of the week. After applying Principal Component Analysis (PCA) in order to reduce the

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

[1] Design of a Smart and Compact Illumination System. Available online at https://www.redalyc.org/jatsRepo/5722/572261854020/html/index.htmlredalyc.org

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

dimensionality of the problem, a clustering was performed and it was created a pair of new datasets adding to each sensor the cluster it belongs to in each case. Later on, supervised Machine Learning (ML) learning algorithms were applied to the latter datasets to validate the clusterings carried out previously.

Finally, a graph convolutional network (GCN) was performed in order to enhance the results obtained from the clustering based on the average counts that sensors had made. By this way, it was feasible to raise different performance metrics of the classification of sensors based on their activity.

This is a novel research in this field due to the lack of attempts to classify sensors based on their characteristics. A similar approach has been carried out in [5] classifying buildings depending on human interaction. In this case, the classification of buildings was done attending the interaction of people and spatio-temporal population density. The main contributions of this work are the following. On the one hand, processing of data to know the activity of sensors and the typology of buildings surrounding them is performed so that sensors are characterized. Next, clustering based on the information obtained in the previous phase is carried out grouping sensors. By this way, pedestrian activity would be predicted giving the opportunity to anticipate in different ways providing an eco-friendly and safe urban environment. Finally, an alternative approach driven by a GCN is carried out.

The remainder of this work is organized as follows. Section 2 reviews the literature. In section 3 the fundamental concepts about PCA, clustering and supervised machine learning are described and the proposed methodology is reviewed. The materials used in this work are described in Section 4. The experimental work is presented in Section 5. The experimental results and analysis are presented in Section 6. Conclusions of this work and outlines of some potential directions for further investigation are made in Section 7. The abbreviations cited in this paper are summarized in Table 12.

II. LITERATURE REVIEW

In the way of achieving a smart distribution and maintenance of cities different researches have been made recently in order to tackle a wide variety of issues, such as building functionality identification, pedestrians detection, traffic prediction or other type of detections. Different skills have been used in those works enabling optimum solutions to the mentioned tasks.

Different techniques have been developed to address tasks related with traffic. In [2] a camera based system was used, even though bad visibility conditions caused by bad weather or insufficient lighting were limiting factors. The same problem was limiting their performance in infrared sensor based system [3]. Similarly, in [4] a sparse coverage of video cameras in the public space was performed, acceptance levels amongst citizens being too low, though.

Identifying the buildings' typology has been useful in a wide range of applications. In [5] a new method to identify building functions from the perspective of the spatial distribution and spatial interactions of human activities was proposed. First, taxi data were used to acquire the spatiotemporal interaction characteristics among buildings with different functions. Then, the spatiotemporal population density distribution was adopted to depict the building vitality. Finally, an iterative clustering method was introduced to identify the building functions.

The correlation between space and time in smart cities has emerged the need of different research lines so as to solve the lack of works bridging this two variables. Heterogeneity related to the distribution within space and the dynamism of data through time has led to the partitioning of space in regions and time intervals. In [9] they provide a method for combining both spatial and temporal factors in predicting pedestrian flow within city centres. The model utilizes sensor data over an extended time period that allows seasonality and time of day factors to be incorporated as well as actual walking distances to points of interest and transport terminals in contrast to Euclidean distances to identify influencing factors.

Different approaches has been made in order to solve issues related to traffic incidents, by using binomial logistic regression and space–time cube model [6] or geographic information system (GIS) to visualize the distribution of pedestrian crashes in cities to explore the relationships between pedestrian crashes and the population, road network, land use and social services and activities and to analyze the impacts of the building environment and road characteristics on the severity of pedestrian crashes by combining the binary logistic regression and tree-based models [7], among others. Nowadays, the complex spatial dependency of road networks, non-linear temporal dynamics with changing road conditions, and the inherent difficulty of long-term forecasting is challenging. In [8] there is a presented deep learning framework titled, "Diffusion Convolutional Recurrent Neural Network (DCRNN)" for traffic forecasting. With the aim of making traffic prediction and the incorporation of spatial and temporal dependency in the traffic flow, DCRNN also integrates the encoder-decoder architecture with a scheduled sampling technique to improve performance and long-term forecast of traffic.

With the advancement of AI techniques, new methods have emerged to perform clustering tasks. In that sense, graph clustering has become one of the most popular and widely adopted methods. There are an increasing number of applications that use graphs to represent data. For example, in e-commerce, a graph-based learning system can provide extremely accurate suggestions by using the interactions between customers and products, recommender system, social networks, biological protein-protein networks [10]. In chemistry, molecules are represented as graphs, and their bioactivity must be determined in order to develop new drugs. Whereas citation network; traffic forecasting, taxi demand prediction are all used to predict the concentrations of a wide variability of air pollutants. By forecasting the crowd flows to predict urban traffic flow, management of tourism flows can be predicted. Finding the way to incorporate graph structure information into a machine learning model is the core problem in graph machine learning.

Traffic forecasting is essential for guidance and traffic control. In [11] there is a proposed model based on Spatio-Temporal Graph Convolutional Networks (STGCN) for traffic prediction. STGCN seeks to predict in the traffic domain by integrating convolution with graphs and space-time convolution in blocks so that the training is faster with a smaller number of parameters. In [12] a proposed Origin – Destination based Temporal Graph Attention Network (OD-TGAT) framework is used for taxi demand forecasting. This model has two main building blocks: a graph network and a neural network. This is the first representation of a model employed graphing network used for taxi demand prediction. In [13] there is a proposed hybrid model based on deep learning methods. This hybrid model integrates Graph Convolutional networks and Long Short-Term Memory networks (GC-LSTM) to establish and predict the spatiotemporal variation in space-time of $PM_{2.5}$ concentrations by applying the graph convolutional networks (GCN) to extract the spatial dependency between different stations, as well as the Long Short-Term Memory (LSTM) to capture the temporal dependency between observations at different times.

Forecasting the crowd flows in each and every part of a city, especially in irregular regions, is very important for the following reasons: traffic control, risk assessment, and public safety. Nevertheless, it is very challenging because of the interactions and spatial correlations between different regions. In [14], the proposed multi-view graph convolutional network (MVGCN) is used to predict the inflow and outflow in each and every irregular region of a city to integrate the geospatial position via spatial graph convolutions. In [15] the proposed attention-based deep spatio-temporal network, with multi-task learning (ADST-Net) at a citywide level, creates a goal to predict urban traffic flow. ADST-Net furthermore introduces an outside embedding mechanism to extricate the impact of external factors on flow prediction, such as weather conditions.

III. FUNDAMENTAL CONCEPTS AND PROPOSED METHODOLOGY

A. PROPOSED METHODOLOGY

This research has followed the approach described in this section. As it is shown in Fig. 1 one can distinguish 3 main phases. The second one, could be divided into 2 subphases depending the architecture used to tackle the problem in question. Both of them, the GCN and the clustering method are based on the previous processing of data obtained from open data source from the city of Melbourne. In this first part of the research new datasets derived from the ones obtained from the open sources were calculated followed by a dimensionality reduction technique. The three dimensionality reduction techniques that we compared were Principal Component Analysis (PCA), Unifold Manifold Approximation and Projection (UMAP) and t-Distributed Stochastic Neighbor Embedding (tSNE). The activity throughout different time intervals of days of the sensors was calculated as well as the number of buildings of each type of their surroundings as explained in section IV, where materials are described in depth. After this, reduction techniques were carried out to reduce the dimensionality of the issue, and use fewer variables in the next phase raising the efficiency of these.

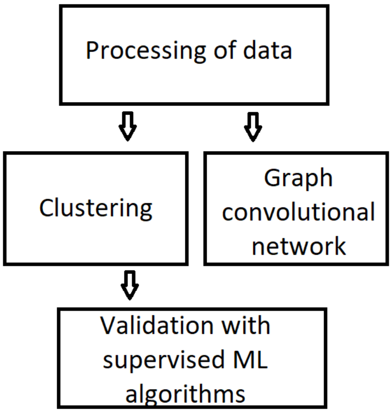

Processing of data
Clustering
Graph convolutional network
Validation with supervised ML algorithms

FIGURE 1. Diagram of the proposed methodology.

In the second phase with the information obtained in the previous one a clustering process or a graph convolutional network is carried out to distinguish groups of sensors depending on their activity throughout the day or the type of environment(building) they are located. In case of the clustering processes a posterior validation using supervised Machine Learning algorithms was performed to show the effectiveness of the previous approach. For both datasets, 3 clusterings were done to determine which of the dimensionality reduction techniques suits best for this problem. After all, we verified that PCA was the most adequate method to use in this case, continuing the rest of the work applying this technique.

Within this section a more detailed explanation of each of the concepts of the phases mentioned above is given.

B. PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis is the process of computing the principal components of a collection of points, that are sequence of p unit vectors where the i-th vector is the direction of a line that best fits the data while being orthogonal to the first i-1 vectors, and using them to perform a change of basis on the data. One of the objectives of using this method is to carry out the dimensionality-reduction of a data set with a large number of connected variables while maintaining as much variance as feasible in the data set. This is accomplished by converting to a new set of uncorrelated variables known as principle components (PCs), which are sorted so that the first few keep the majority of the variance existing in a dataset [16]. Among other dimensionality reduction techniques PCA offers lowest computational cost compared to tSNE or UMAP. To decide which of them fits best our problem, we developed part of the methodology with each of the 3 dimensionality reduction techniques and after all, we saw that clustering of sensors provides a better accuracy with PCA than with the other two methods. In fact, PCA outperformed in a range of 5% the performance of UMAP and in 10% the performance of tSME. Further details are contained in section 6. Consequently, during the rest of the work we adopted PCA as the dimensionality reduction technique.

Generally, a reduction in the number of variables in a data set carries a reduction in accuracy. Nevertheless, there is a trade-off between accuracy and simplicity. The reason for this reduction is not only that smaller data sets are easier to study and display, but also the machine learning algorithms can analyze data more easily and quickly without having to deal with superfluous factors.[2]

In this case, adopting this technique is the optimum solution in order to lower the number of dimensions of the problem in question.

C. CLUSTERING

Clustering is a type of unsupervised learning method. Unsupervised learning is a technique for extracting references from datasets that contain input data but no labelled answers, and self-discovering naturally occurring patterns. It is a method for identifying significant structure, explaining underlying processes, generating traits, and groups in a set of samples.

Clustering is the process of partitioning a population or set of data points into several groups so that the similarity of points within a group is high and dissimilarity between points from different groups is high, as well. It is essentially a grouping of items based on their similarity and dissimilarity.

This method is critical since it determines the inherent grouping among the unlabeled data. There are no requirements for a successful clustering. It is up to the user to determine what criteria employ to satisfy its needs. For instance, we might be interested in locating representations for homogeneous groups (data reduction), locating ‘‘natural clusters’’ and describing their unknown qualities (‘‘natural’’ data types), locating useful and appropriate groupings (‘‘useful’’ data classes), or locating odd data objects (outlier detection).

It is worth differentiating between fuzzy clustering and hard/crisp clustering. The former one gives the degree of belonging to each cluster for each item, whereas the latter one classifies each item to an unique cluster.

Attending the criteria used for dividing the clusters by the algorithm, different clustering methods can be defined. These are the type of methods and the most important examples of them:

- Density-Based Methods: k-Means, Partitioning Around Medoids (PAM), Clustering Large Applications(CLARA), k-Prototypes, K-Mode.
- Hierarchical Based Methods: Sequential Agglomerative Hierarchical Non-overlapping(SAHN), Balanced Iterative Reducing and Clustering Using Hierarchies(BIRCH), Clustering Using Representatives(CURE), Robust Clustering using Links(ROCK).
- Partitioning Methods: Density-based spatial clustering of applications with noise(DBSCAN), Density-based Clustering(DENCLUE).
- Grid-based Methods: Statistical Information Grid(STING), Wavecluster.

The simplest and most satisfactory unsupervised machine learning approach for solving the clustering problem in many cases is the K-means clustering algorithm. The K-means algorithm divides n observations into k clusters, with each observation belonging to a cluster. The centroids of each cluster are initialized randomly from the initial observation set, and the rest of the items are assigned to the nearest centroid's cluster. After each assignation the centroids are recalculated and identical process is repeated until there are no remaining observations to classify.[3]

D. SUPERVISED MACHINE LEARNING METHODS

The supervised machine learning techniques aim to classify a set of items based on their features and other set of pre-classified items with the same features. This techniques infer a function from a training data set to use it to classify other instances from a test data set. Each instance of a training set consists of a set of features seen as an input vector and the desired output, which is the remaining feature for the instances of the testing data set.

After differentiating those two data sets, the inferred function is used for predicting the output for the instances of the test data set (known beforehand). By this way, the accuracy of the algorithm could be stated comparing the results obtained from the classification process and the ground-truth. As well as that, those algorithms could be used for validating the results of a clustering process.

There are many algorithms that can address this task, and variations of them can be found in the literature. These are some of the most commonly used ones: Naive-Bayes, Decision Tree, Supported Vector Machine, Artificial Neural Networks, Boosting methods, Bagging methods, etc.

[2] A Step-by-Step Explanation of Principal Component Analysis (PCA). Available online at https://builtin.com/data-science/step-step-explanation-principal-component-analysisbuiltin.com

[3] Clustering in Machine Learning. Available online at https://www.geeksforgeeks.org/clustering-in-machine-learning/geeksforgeeks.org

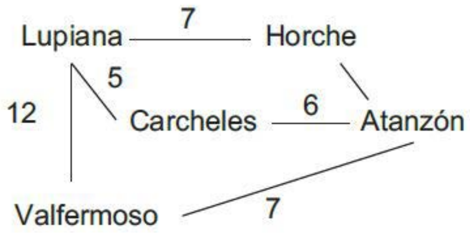


FIGURE 2. **Valued undirected graph.**

E. BASICS OF NEURAL NETWORKS

An artificial neural network is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain These values can be integer, real, or binary. Based on the inputs and the weights, the weighting function produces a weighted sum that is passed through an activation function to produce an output.

F. CONVOLUTIONAL NEURAL NETWORKS (CNNS)

The higher performance of convolutional neural networks or ConvNet with picture, speech, or audio signal inputs sets them apart from other artificial neural networks [18].

They are divided into three sorts of layers:

- Convolutional layer. The convolutional layer is the central component of a CNN, and it is here where the majority of the computation takes place. Input data, a filter, and a feature map are components required.
- Pooling layer. Downsampling, also known as pooling layers, is a dimensionality reduction technique that reduces the number of factors in the input. The pooling process sweeps a filter across the entire input, similar to the convolutional layer, however this filter does not have any weights.
- Fully-connected (FC) layer. The full-connected layer's name is self-explanatory. In partially linked layers, the pixel values of the input image are not directly connected to the output layer, as previously stated.

G. GRAPH THEORY

1) BASIC CONCEPTS

A graph is represented by the pair $G = (V, A)$. The V represents the set of vertices or nodes and the A the set of edges (or arcs).

The nodes represent the elements of the system and edges the interrelationships between them. If all the edges can be traversed in both directions, the graph is known as undirected. In the case of directed graphs, each edge has a direction, generally represented by its origin node and its destination node.

Valued graph: is a graph G together with a function W_E that assigns a numerical weight W_{ij} to each edge (i, j). Eventually it can also coexist with a W_V function that assigns a W_i value to each i node.

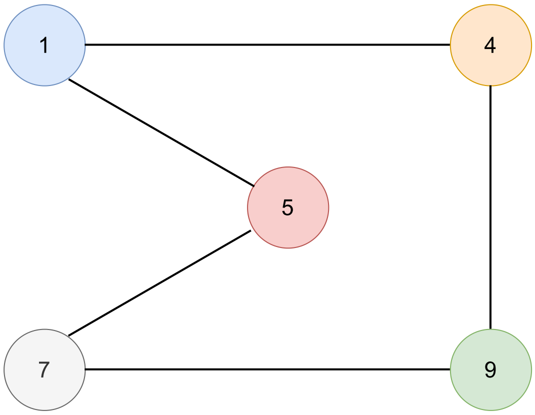


FIGURE 3. **Undirected graph.**

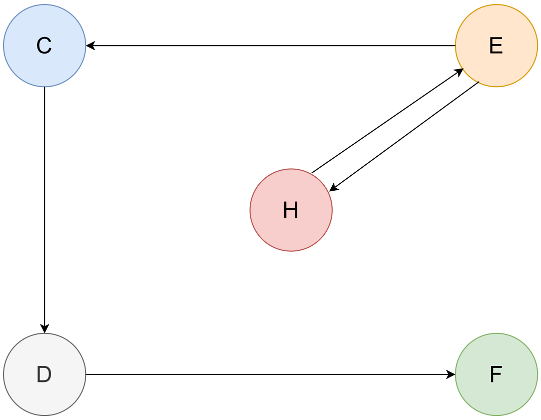


FIGURE 4. **Directed graph.**

As can be seen in the Fig. 2 a valued graph is shown in which each arc has an associated weight that is the length between two nodes [19].

The graph is undirected if the arcs are formed by pairs of unordered vertices, not pointed.

As can be seen in the Fig. 3 an undirected graph is shown formed by the vertices $V = \{1, 4, 5, 7, 9\}$ and the set of arcs A ={(1,4), (4,1), (5,1), (1, 5), (7,9), (9.7), (7.5), (5.7), (4.9), (9.4) }.

When a graph is directed, it is also known as diagraph. In this type of graph the pairs of nodes that form the edges are ordered and are represented by an arrow indicating the direction of the relationship $u \rightarrow v$.

As can be seen in the Fig. 4 a directed graph is shown formed by the vertices $V = C, D, E, F, H$, and the arcs $A = \{(C, D,), (D, F), (E, H), (H, E), (E, C)\}$ form the directed graph $G = V, A$.

2) GRAPH REPRESENTATION

- List of neighbors: associates to each node the list of its neighbors, that is, neighbors $(i) \ = j : (i, j) \in E$
- Adjacency matrix: of Boolean values 0, 1 such that $M(i, j) = 1 \Leftrightarrow (i, j) \in E$

The graphs can be applied to different tasks such as the following:

- Node Classification: this categorization, which is founded exclusively on non-attribute graphs, is based on the graph's structure and the class of the known nodes in our experiment.
- Link prediction: here the nodes' classes aren't taken into account.
- Node clustering: node clustering can be applied to a group items by their proximity to each other.

H. GRAPH NEURAL NETWORKS (GNNS) AND GRAPH CONVOLUTIONAL NETWORKS(GCNS)

Graphs are a type of data structure that represents a collection of items (nodes) and their connections (edges). A Graph Neural Network is a sort of Neural Network that works with the graph structure directly. Node categorization is a common use of GNN. Every node in the network has a label, and predictions of the labels of the nodes using ground-truth data is being made. Convolutional networks multiply the input neurons with a set of weights that are commonly known as filters or kernels. The filters act as a sliding window across the whole image and enable CNNs to learn features from neighboring cells. GCNs perform similar operations where the model learns the features by inspecting neighboring nodes. The major difference between CNNs and GNNs is that CNNs are specially built to operate on regular (Euclidean) structured data, while GNNs are the generalized version of CNNs where the numbers of nodes connections vary and the nodes are unordered (irregular on non-Euclidean structured data).[4]

In our experiment, sensors represent the nodes of the graph, so that node classification is performed to obtain different sensor groups. These nodes' features represent the mean activity of sensors. The number of features was reduced by applying PCA, finally obtaining only 5 features per node for the clustering based in the activity of sensors, and 14 features for the clustering based in the buildings' typology. On the other hand, edges are the invert of the relative distances between sensors.

IV. MATERIALS AND METHODS

In this section, we describe the materials used in this work. We introduce a description of the datasets used.

A. DATASETS

The data that has been used in this work is partly from the city of Melbourne.[5] In this portal we can find an open dataset: Pedestrian Counting System - Monthly (counts per hour). This dataset contains hourly pedestrian counts since 2009 from pedestrian sensor devices located across the city. In the same way we can find Pedestrian Counting System - Sensor Locations. This dataset contains status, location and directional information for each pedestrian sensor device installed throughout the city transportation. Finally, Buildings with name dataset contains the information about the typology of the buildings and their locations.

In order to expand the amount of data used to train the classifiers described in section V we made use of the pedestrian mobility information provided by the city hall of Madrid.[6] There were counts of pedestrian and cyclists from 2019 to 2021. Although there were counts available of 2019, due to several break downs of sensors there were various missing values. Furthermore, we considered only data between 2020 and 2021.

1) PEDESTRIAN COUNTING DATASETS OF MELBOURNE

The Pedestrian Counting System - Monthly (counts per hour) dataset[7] contains hourly pedestrian counts (since 2009) using pedestrian sensor devices located across the city. The data is updated on a monthly basis. This dataset contains 3,482,938 records in all, and it has been collected from May 1st, 2009 to December 31th, 2020.

In order to avoid analysing instances from sensors that were not working or were damaged, we used another dataset named Pedestrian Counting System - Sensor Locations dataset.[8] This dataset contains information about status, location and direction for each pedestrian sensor device installed throughout the city. It is made available by the City of Melbourne with a Creative Commons Attribution 4.0 International license.[9]

We conducted to the exploratory analysis of the data from the previous data set in order to know if all the sensors have enough information to include them later in a PCA analysis and clustering. It is important to know the activity of the sensors and see in which period there was no record of pedestrian crossings as part of the exploratory analysis. After this, we removed the instances from sensors which had not been operating.

After all, we decided to establish two hour time intervals for each day and separate weekdays from Saturdays and Sundays. By this way, we generated a new dataset (named Melbourne pedestrians with mean hourly count datafinal all and both hourly counts, for both cities' mean values) that

[4] Understanding Graph Convolutional Networks for Node Classification. Available online at https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classification-a2bfdb7aba7btowardssciencedata.com

[5] City of Melbourne Open Data, available at https://data.melbourne.vic.gov.au/stories/s/data-principles/9f8u-v2fn?src=hdrCity of Melbourne Open Data

[6] City of Madrid Open Data, available at https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=695cd64d6f9b9610VgnVCM1000001d4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default#City of Madrid Open Data

[7] Pedestrian Counting System - Monthly (counts per hour). Available online at https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-Monthly-counts-per-hour/b2ak-trbp?src=featured_bannerdata.melbourne.vic.gov.au

[8] Pedestrian Counting System - Sensor Locations. Available online at https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-Sensor-Locations/h57g-5234data.melbourne.vic.gov.au

[9] Creative Commons Attribution 4.0 International license, available at https://creativecommons.org/licenses/by/4.0/legalcodecreativecommons.org

included the mean hourly count value for each sensor in different time intervals of different days (weekday, Saturday, Sunday).

2) BUILDINGS WITH NAME DATASET OF MELBOURNE

This dataset contains the typology of buildings that will be used to calculate the geodesic distance that exists from a pedestrian sensor to a type of building. By this way, we registered the buildings that were under a previously defined neighboring distance (300 m) from each sensor (obtaining their location from Pedestrian Counting System - Sensor Locations dataset) and these values were used later to perform a PCA and posterior clustering of sensors depending on the typology of the buildings near-by. Thus, the predominant types of buildings surrounding each sensor could be known. These results were summarised in a new dataset (named Melbourne count pedestrians buildings) that had been used in the posterior phases.

3) MADRID PEDESTRIAN DATASETS

These datasets contain the hourly pedestrian counts of every sensor spread throughout the city of Madrid for each year. They contain information about each sensor, its latitude and longitude, information of its address, and the typology of the address (pedestrian street, sidewalk, etc.).

V. EXPERIMENTAL WORK

After obtaining the desired data, we sought out to divide the sensor into different groups or clusters. For this purpose it is advisable to reduce the number of features of each dataset, not only for enhancing the interpretability of the data, but also for improving the results of clustering, because the machine learning algorithms can analyze data more easily and quickly without having to deal with superfluous factors. After this reduction had been made, we present the techniques used to divide the sensors based on different criteria, depending on the building typology surrounding them and the number of the activations that they had in different time intervals throughout the week. Finally, we validate each division made in the previous phase by using supervised machine learning algorithms.

A. PRINCIPAL COMPONENT ANALYSIS (PCA)

First, we analysed the cumulative explained variance of different numbers of components for each dataset, after scaling the data with mean 0 and standard deviation of 1 for each attribute for optimizing the results. There is no single answer or method to identify the optimal number of main components to use. A very widespread way of proceeding consists of evaluating the proportion of accumulated explained variance and selecting the minimum number of components from which the increase is no longer substantial. Depending on the degree of accuracy required this proportion varies. In our case, we established a 95 % of variation to be explained by the amount of components selected.

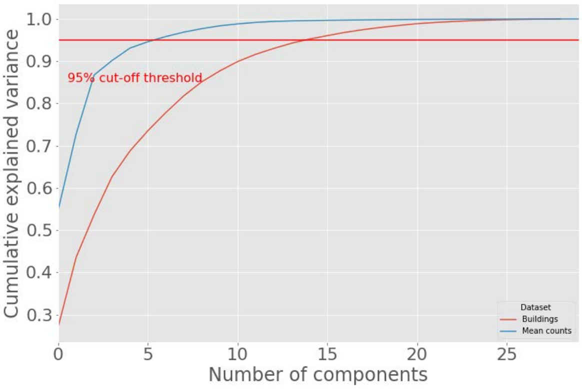


FIGURE 5. Cumulative explained variance for the Melbourne count pedestrians buildings and Melbourne pedestrians with mean hourly count datafinal all datasets.

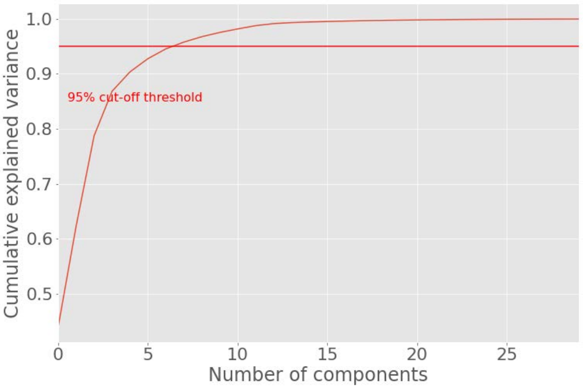


FIGURE 6. Cumulative explained variance for the Melbourne & Madrid mean hourly count dataset.

As it can be seen in the Fig. 5, with 14 components the desired variance is achieved for the Melbourne count pedestrians buildings dataset. In the same way, 5 components were sufficient to satisfy the requirement mentioned above for the Melbourne pedestrians with mean hourly count datafinal all dataset. However, 6 was the optimum number of components to satisfy the mentioned requirement in case of both cities' pedestrians activity pattern dataset, as it can be seen in Fig. 6.

B. CLUSTERING

1) SELECTING OPTIMAL NUMBER OF CLUSTERS

To select the optimal number of clusters there are different traditional methods. For the problem in question, we are going to use the elbow method. The function Inertia simply computes the squared distance of each sample in a cluster to its cluster center and sums them up. The smaller the Inertia value, the more coherent are the different clusters. When as many clusters are added as there are samples in the data set, then the Inertia value would be zero. After obtaining principal components, we proceed to apply the K-Means algorithm after evaluating the optimal number of clusters according to the elbow method. The point in which the graph flattens will indicate the optimal number of clusters, just enough to achieve a desired difference and coherence between clusters.

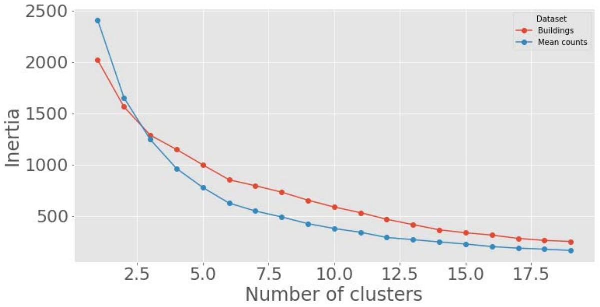


FIGURE 7. Elbow method to obtain the optimal number of clusters for Melbourne pedestrians with mean hourly count datafinal all dataset.

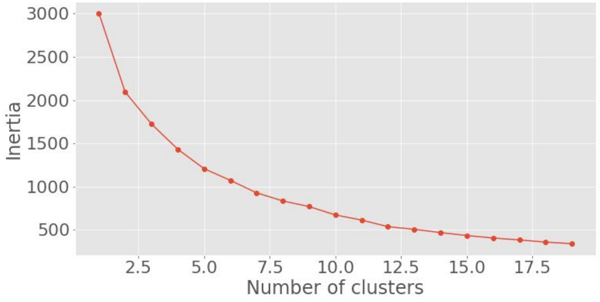


FIGURE 8. Elbow method to obtain the optimal number of clusters for Melbourne & Madrid pedestrians with mean hourly count dataset.

TABLE 1. Sensors and clusters based on buildings' typology.

Cluster Namel	Sensors
Office	45, 63, 66, 15, 47, 3, 21, 30, 56, 52, 2, 1, 20, 19
Rest of sensors	7, 14, 25, 64, 17, 29, 23, 33, 13, 12, 9, 6, 42, 5, 35, 34, 57, 40, 58, 8, 11, 18, 24, 28, 44, 10, 43, 73, 75
Residential Apartment	55, 61, 62, 54, 46, 59, 49, 26, 51
House-Community use	70, 27, 48
Office-Retail	72, 41, 60, 39, 65, 67, 68, 69, 71, 13, 53, 38, 32, 4, 50, 40, 31, 37, 36, 16, 22

As it can be seen in the Fig. 7, 6 clusters would be a reasonable option for the Melbourne pedestrians with mean hourly count datafinal all dataset, but 4 also could be considered. At the beginning we decided to choose 6 clusters and similarly for the Melbourne count pedestrians buildings dataset, with 6 clusters as well.

When considering data from both cities we concluded that 5 would be the optimum number of clusters as it is shown in Fig. 8

2) CLUSTERING BASED ON BUILDINGS' TYPOLOGY

The Fig. 9 shows the clusters based on the typology of buildings. We can differ some clear groups analysing only the two first principal components. But as we will see in the section 6 the fusion of the clusters 4 and 6 improved the supervised classifiers performance.

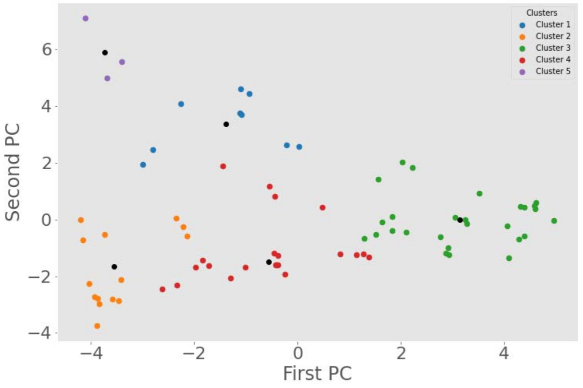


FIGURE 9. Clustering based on the typology of buildings. Black points indicate the centroids of each cluster.

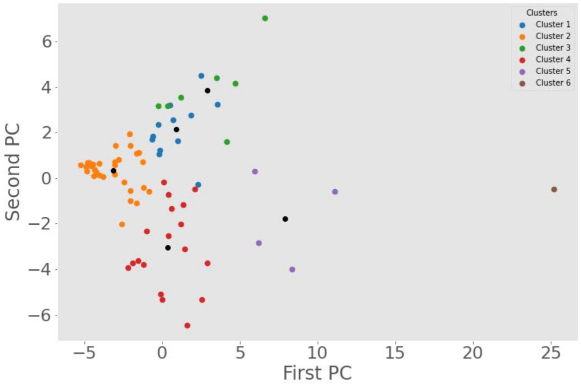


FIGURE 10. Clusters based on pedestrians' activity. Black points indicate the centroids of each cluster.

Table 1 shows the cluster's name based on the main type of buildings that have near-by the sensors included on it. The sensors of the cluster named ''Rest of sensors'' do not have a clear predominance of typology of any building. Thus, was made its nominalisation.

3) CLUSTERING BASED ON THE ACTIVATION OF THE SENSORS

In the clustering based on the mean activations of the sensor per time interval throughout the week we obtained the clusters shown in the Fig. 10.

Clearly, there is a sensor that differs completely from the rest observing the two first principal components. In the results and analysis section (section 6) we could see that taking out this data the latter classifier outperformed the former one. At the same time, it is noticeable the similarity between the clusters 1 and 3, and for this reason we saw that the accuracy of the classifier after fusing clusters 1 and 3 was enhanced.

Table 2 shows the name of the clusters and sensors based on their activity. Depending whether their sensors' main activations were concentrated in weekdays or weekends and their frequency of activations was the nominalisation of clusters made. The sensors that have a very high activity pattern

TABLE 2. Sensors and clusters based on the pedestrians’ activity of Melbourne.

Cluster Namel	Sensors
Weekday	39, 37, 40, 43, 44, 48, 56, 17, 53, 10, 12, 11, 14, 26, 23, 25, 19, 27, 31, 33, 63, 64, 65, 66, 67, 68, 69, 70, 71, 73
Weekend+	29, 46, 50, 51, 62, 8, 30, 34, 36, 42, 43, 49, 52, 54, 56, 57, 59, 61, 18, 7, 20
Weekday++	35, 45, 47, 55, 58, 2, 1, 3, 15, 9, 5, 6, 24, 22, 21, 28, 13, 16
Weekday-weekend++	4, 41, 32, 60

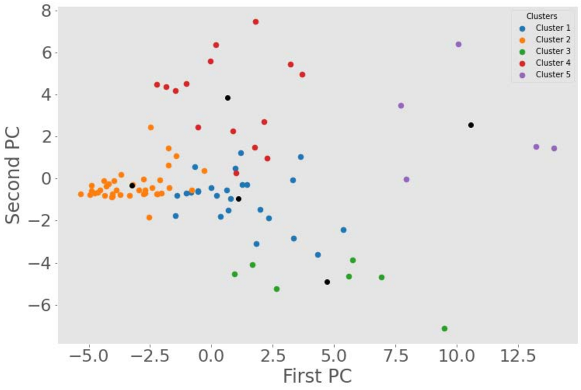


FIGURE 11. Clusters based on pedestrians’ activity from Madrid & Melbourne. Black points indicate the centroids of each cluster.

TABLE 3. Sensors and clusters based on the pedestrians’ activity of Melbourne & Madrid.

Cluster Namel	Sensors
Weekend	34, 40, 36, 42, 43, 49, 52, 54, 56, 57, 58, 59, 61, 17, 18, 9, 7, 20, 27, 16, 66, 77, 78, 85, 88
Weekday	39, 37, 44, 48, 53, 10, 12, 11, 14, 26, 23, 25, 19, 31, 33, 63, 64, 65, 67, 68, 69, 70, 71, 73, 74, 75, 76, 77, 78, 81, 82, 83, 84, 85, 86, 88, 89, 91
Weekend+	5, 11, 15, 16, 25, 38, 54
Weekday+	9, 10, 12, 19, 30, 31, 32, 33, 40, 41, 44, 47, 50, 52, 56
Weekday-weekend+	26, 35, 58, 60, 71

on weekdays were grouped in weekday++ cluster, while the ones that were mostly activated in weekdays but with a minor activity were grouped in weekday cluster. Sensors whose activity was concentrated in weekends were grouped in weekend cluster and the ones that follow a similar activity pattern throughout all the week were grouped in weekday-weekend++ cluster, whose activity was also very high.

In order to conclude more accurate results, as we mentioned above we merged pedestrians’ activity data from Madrid and Melbourne. We repeated the process of clustering and we obtained the clusters shown in Fig. 11.

C. VALIDATION USING SUPERVISED ML ALGORITHMS

Next step was to validate the goodness of the clusterings performed in the previous phase by applying supervised machine learning algorithms in both cases. With the obtained clusters we modified the former datasets and included the cluster each sensor belongs to as an extra feature, that was to be used as the output of the machine learning algorithms. By this way, observing different performance metrics of the algorithms we can deduce the goodness of the clusterings.

TABLE 4. The optimal parameter values for different machine learning algorithms.

Algorithm	Param. 1	Param. 2	Param. 3	Param. 4
SVM	kernel=’poly’ [’linear’, ’poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’]	degree of polynomial function=2 [2,3,4,5,6]	kernel gamma coefficient=’auto’ (’scale’, ’auto’)	
RF	n estimators= 130 [30, 250]	criterion= ’entropy’ [’gini’, ’entropy’]	max depth=4 [2, 7]	
DT	criterion = ’entropy’ [’entropy’, ’gini’]	min samples to split a node = 5 [2,7]	min samples per leaf = 3 [1,6]	max depth = 10 [2,20]
MLP	hidden layer sizes = 250 [30,3000]	max iterations = 700 [150,2000]		
Adaboost	base estimator = RF [RF, SVM, GNB]	n estimators = 10 [5,20]		
Bagging	base estimator = RF [RF, SVM, GNB]	n estimators = 10 [5,20]		

For all datasets different machine learning algorithms were applied. Those were: Supported Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), Multilayer Perceptron (MLP), Gaussian Naive-Bayes (GNB), Adaboost (base classifier: RF for the Buildings’ dataset and DT for the mean activation’s dataset) and Bagging (base classifier: SVM). For them, an optimization of the parameters was done, in order to enhance their performance. The optimal hyperparameters and the range of parameters used to find the optimal ones (between parentheses) are given in Table 4.

1) MODEL EVALUATION METRICS

The performance of our model was evaluated using the following metrics:

- Confusion matrix: It is a specific table structure that shows the performance of an algorithm class by class. The examples of an actual class are represented by each row of the matrix, whereas the instances in a predicted class are represented by each column.[10]

[10] Confusion matrix, Available online at https://en.wikipedia.org/wiki/Confusion_matrixen.wikipedia.org

- Accuracy: It is the proportion of correct predictions among the total number of cases being examined.

$$Accuracy(Acc) = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- Precision: It gives the probability of an instance predicted of one class being of such class in reality.

$$Precision(Pr) = \frac{TP}{TP + FP} \tag{2}$$

- Recall: It gives the probability of an instance belonging to a class being predicted of such class.

$$Recall(Re) = \frac{TP}{TP + FN} \tag{3}$$

- F-Score: A performance metric that takes into account the trade-off between Precision and Recall.

$$F - Score(F) = \frac{2 * Pr * Re}{Pr + Re} \tag{4}$$

Notation: TP = True positive; FP = False positive; TN = True negative; FN = False negative

D. GRAPH CONVOLUTIONAL NETWORK

After obtaining a ground-truth data from the clustering based on the activity patterns of sensors, this was used to build the convolutional graph. The nodes of the graph were the sensors themselves and the edges were weigthed by the invert of the distance used as their feature. Those all distances were obtained from the Pedestrian Counting System - Sensor Locations dataset.

We performed a two convolutional layer network. The first layer has as input features the number of components(5) and 22 features as output. The second layer has these 22 features as input and the number of classes as output features. The number of features was adjusted manually to optimize the accuracy of the network. Simpler layers could not achieve the desired performance, and with more features the structure was too complex for the learning process of this problem. The activation function used was Relu. For training the network 0.001 learning rate was used, and the optimization algorithm chosen was Adam. These last parameters were adjusted manually to reach the highest performance of the network as well. A too low learning rate derives in a computationally too long learning process, while a higher one would not produce optimal results.

The feature matrix was calculated using the components obtained in PCA for the Melbourne pedestrians with mean hourly count datafinall all dataset derived from Pedestrian Couunting Datasets of Melbourne(2009-2020). The edges of the graph were the inverse of relative distances between sensors (nodes), thus strengthening the links between proximal sensors.

The division of the sensor belonging to each cluster was done in the following way: 80% was used for training process, 10% was used for validation phase and the rest 10% for testing process. In order to emulate a cross-validation process, we shuffled 10 times the sensors belonging to each clusters, so that 10 different divisions were carried out to train the network in 10 different ways.

TABLE 5. Accuracies for building typology (B) and time-interval mean activation (A) datasets based clustering after 3 different dimensionality reduction techniques.

Classifier	B(PCA)	B(tSNE)	B(UMAP)	A(PCA)	A(tSNE)	A(UMAP)
SVM	93.39%	81.25%	85.36%	87.14%	70.54%	59.46%
RF	94.14%	84.88%	89.77%	85.07%	78.99%	82.90%
DT	88.86%	68.73%	71.54%	87.43%	61.33%	59.92%
MLP	81.86%	79.23%	80.56%	66.71%	57.33%	63.08%
GNB	91.96%	71.61%	86.96%	85.71%	82.14%	80.54%
Adaboost	95.15%	85.24%	89.52%	81.85%	77.17%	79.70%
Bagging	91.54%	80.87%	85.55%	86.10%	72.49%	66.02%

VI. RESULTS AND ANALYSIS

In this section the results obtained after applying supervised ML algorithms to all datasets are presented. As we mentioned beforehand, new datasets had been made after obtaining the cluster predicted for each sensor, adding as an extra feature the cluster each sensor belongs to. This feature was used as the output of the supervised algorithms.

As mentioned in section 5.2.1 for both datasets from Melbourne the number of clusters was decided to fix at 6 clusters, following the elbow method after applying dimensionality reduction techniques.

To decide which of the dimensionality reduction techniques between UMAP, PCA nad tSNE, fits best with our data, we followed in parallel three clustering processes for both datasets after applying these techniques. As it could be seen in Table 5 the best results were achieved after applying PCA as the dimensionality reduction method for both clusterings, so we followed the rest of the work applying this technique.

In the same way, the clustering made attending the mean activations of the sensors in different time intervals showed that a sensor was clearly different attending the first two principal components, as it is shown in Fig. 10. Thus, after analysing different confusion matrices we saw that it was classified in a random cluster, so we decided to remove it from the dataset. Moreover, after examining the activations of the sensor belonging to cluster 1 and 3 we decided to merge them, assuming that they had a closely similar activity throughout different days. After applying those changes we observed an improvement of the performance metrics of a significance of 5-6% on average.

For the clustering based on buildings' typology we saw in the confusion matrix calculated for each classification algorithm that the sensors of cluster 6 were almost always classified as part of the cluster 4, so we decided to merge those two clusters. After this change had been made, the performance metrics outperformed the former ones by a significance of 1-1.5% on average.

In Table 6 different scores for different algorithms applied to the Buildings' typology dataset based clustering are shown.

TABLE 6. Metrics for building typology dataset based clustering.

Classifier	Accuracy	Precision	Recall	F-Score	Confidence interval
SVM	93.39%	92.99%	93.24%	92.93%	92.64-94.15%
RF	94.14%	93.42%	93.92%	93.08%	93.42-95.07%
DT	88.86%	89.44%	89.05%	88.90%	87.06-90.21%
MLP	81.86%	82.50%	81.80%	81.45%	79.48-84.23%
GNB	91.96%	92.52%	91.89%	91.39%	90.74-93.19%
Adaboost	95.15%	94.28%	93.55%	93.79%	94.41-95.90%
Bagging	91.54%	90.49%	90.59%	90.17%	90.51-92.56%

TABLE 7. Metrics for time-interval mean activation dataset based clustering.

Classifier	Accuracy	Precision	Recall	F-Score	Confidence interval
SVM	87.14%	88.03%	87.32%	87.02%	85.45-88.84%
RF	85.07%	85.03%	84.93%	84.86%	83.66-86.48%
DT	87.43%	87.70%	87.32%	87.02%	85.74-89.12%
MLP	66.71%	66.92%	66.71%	66.31%	64.66-78.66%
GNB	85.71%	88.02%	85.92%	86.40%	83.93-87.50%
Adaboost	81.85%	82.09%	81.78%	81.76%	80.10-83.60%
Bagging	86.10%	86.98%	86.24%	85.83%	84.48-87.71%

The scorings are the mean values of 10-fold cross validations for 30 different seeds.

In the same way, Table 7 shows the scores for different algorithms applied to the time-interval mean activation dataset based clustering. The proceeding of obtaining the scores was identical to the previous case.

For the former dataset Adaboost algorithm with RF as base classifier outperformed the rest of the classifiers' performance metrics. In case of the latter dataset, DT was the algorithm that solved the classification issue more adequately. For both we firstly supposed the H0 hypothesis, that says that each of the mean scores obtained previously came from a normal distribution. After applying Kolmogorov-Smirnov test to all the scores obtained in every cross-validation processes (n>50) we concluded that H0 hypothesis was negligible, due to the fact that p-values obtained in both cases (7.66e-188 for buildings dataset and 1.78e-131 for Mean activations dataset) were lower than 0.05. Thus, as H1 hypothesis confirms the means come from a non normal distribution.

Consequently in order to know the statistical significance of the difference of the accuracies' means, we applied the Kruskal-Wallis test. We got the statistics of 8.08504 and 0.85582 and p-values of 0.99996 and 1.0 for the Buildings and Mean activations datasets respectively. Thus, we can not conclude that all the means arise from different distributions.

TABLE 8. Metrics for time-interval mean activation dataset (Madrid & Melbourne) based clustering.

Classifier	Accuracy	Precision	Recall	F-Score	Confidence interval
SVM	88.89%	89.25%	88.89%	88.70%	87.62-90.15%
RF	88.30%	88.49%	88.30%	88.31%	87.32-89.27%
DT	77.37%	77.91%	77.37%	77.36%	76.28-78.47%
MLP	64.00%	64.53%	64.00%	64.03%	62.23-65.77%
GNB	83.33%	86.04%	83.33%	84.00%	82.17-84.50%
Adaboost	88.37%	88.52%	88.37%	88.37%	87.43-89.31%
Bagging	85.93%	87.22%	85.93%	85.48%	84.31-87.54%

In order to figure out the relationship between both clusterings, we analyzed the general trend that followed the sensors of each cluster in the other dataset. We concluded that the sensors belonging to the cluster weekday and weekday++ were mainly related to the Office buildings, the sensors from weekend++ to Residential apartment-House/Townhouse type of buildings and finally the ones from weekday-weekend++ to the Retail type of buildings. Obviously, this makes sense taking into account that pedestrians usually spend most of their time at work during the week and at home at the weekends. At the same time, retails are visited throughout all the week.

Finally, seeking out to emulate the performance of the clustering based on the buildings' typology by the clustering based on the sensors' activity, we enlarged the former dataset by applying the data from the city of Madrid. Table 8 shows the scores for different algorithms applied to the time-interval mean activation based dataset (Madrid & Melbourne) clustering. The proceeding of obtaining the scores was identical to the previous cases.

The best classifier in terms of accuracy, precision, recall and F-score was SVM in this case. We firstly supposed the H0 hypothesis, that says that each of the mean scores obtained previously came from a normal distribution. After applying Kolmogorov-Smirnov test to all the scores obtained in every cross-validation processes for the mentioned classifier (n>50) we concluded that H0 hypothesis was negligible, due to the fact that p-value obtained (1.32e-173) was lower than 0.05. Thus, as H1 hypothesis confirms the means come from a non normal distribution.

Consequently in order to know the statistical significance of the difference of the accuracies' means, we applied the Kruskal-Wallis test. We got the statistic of 0.0 and p-value of 1.0. Thus, we can not conclude that all the means arise from different distributions.

As the results obtained for the clustering process depending on the activity of sensors did not outperform the ones

TABLE 9. Results of accuracy of the GCN.

Results of GCN test	
Accuracy	87.00%
Confidence interval	83.416-90.584%

TABLE 10. Computation times for 10-fold Cross Validation for different algorithms.

Algorithm	Building criteria	Activity criteria
k-Means+SVM	1.41 s	4.8 s
k-Means+RF	5.14 s	115.5 s
k-Means+DT	3.9 s	6.3 s
k-Means+MLP	3.48 s	284.1 s
k-Means+GNB	1.35 s	6 s
k-Means+Adaboost	4.74 s	134.1 s
k-Means+Bagging	36.24 s	1333.8 s
GCN		709.1 s

TABLE 11. Comparative of results with the literature.

Method	Average Accuracy
Top-1 Clustering(Activity)	88.89%
Top-1 Clustering(Building Typology)	95.15%
Top-1 GCN(Activity)	87.00 %
L. Zhao et al.[5]	85.66%

based on the building typology, even if we merge data from Madrid and Melbourne, we opted for designing a graph convolutional network with the ground-truth data obtained from the clustering based on the activity patterns of sensors from Melbourne. After training for 10 different seeds each 10 divisions of the sensors mentioned in section 5.4, we obtained the following accuracies in the testing process that are summarized in Table 11.

We firstly supposed the H0 hypothesis, that says that each of the mean accuracies obtained previously came from a normal distribution. After applying Kolmogorov-Smirnov test to all the accuracies obtained in every cross-validation processes (n>50) we concluded that H0 hypothesis was negligible, due to the fact that p-value obtained(6.26e-49) was lower than 0.05. Thus, as H1 hypothesis confirms the means come from a non normal distribution. Consequently in order to know the statistical significance of the difference of the accuracies' means, we applied the Kruskal-Wallis test. We got the p-value 0.81437. Thus, we can not conclude that all means arise from different distributions.

Attending the time needed by each method to obtain groups of buildings based in both criteria, we can observe that the application of k-Means clustering algorithm and posterior validation using supervised ML algorithms is much faster than obtaining them using a graph convolutional network. The time needed by each method for 10-fold cross validation is shown in Table 10. The environment in which all development of our work had been processed is a x64 Windows 10 Operating System equipped with a Intel Core i5-10210U working at 1,6 GHz (4,2 GHz Turbo frequency, 4 core and 8 subprocesses) and 8 GB DDR-4 RAM.

TABLE 12. Abbreviations used in this paper.

Abbreviation	Whole Phrase/Word
Acc	Accuracy
ADST-Net	Attention-based Deep Spatio-Temporal Network
AI	Artificial Intelligence
BIRCH	Balanced Iterative Recucing and Clustering Using Hierarchies
CIB	International Council for Research and Innovation in Building and Construction
CLARA	Clustering Large Applications
CNN	Convolutional Neural Network
CURE	Clustering Using Representatives
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DCRNN	Diffusion Convolutional Recurrent Neural Network
DDR	Doble Data Rate Synchronous
DENCLUE	Density-based Clustering
DT	Decision Tree
F	F-Score
FC	Fully Connected
FN	False Negative
FP	False Positive
GIS	Graphic Information System
GC-LSTM	Graph Convolutional networks and Long Short-Term Memory networks
GCN	Graph Convolutional Networks
GNB	Gaussian Naive-Bayes
GNN	Graph Neural Network
IoT	Internet of Things
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Mulit-Layer Perceptron
MVGCN	Multi-View Graph Convolutional Network
OD-TGAT	Origin-Destination based Temporal Graph Attention Network
PAM	Partitioning Around Medoids
PC	Principal Component
PCA	Principal Component Analysis
Pr	Precision
RAM	Random Access Memory
Re	Recall
RF	Ranndom Forest
ROCKS	Robust Clustering using Links
SAHN	Sequential Agglomerative Hierarchical Non-overlapping
STGCN	Spatio-Temporal Graph Convolutional Networks
STING	Statistical Information Grid
SVM	Supported Vector Machine
TN	True Negative
TP	True Positive
tSNE	t-Distributed Stochastic Neighbor Embedding
UMAP	Unifold Manifold Approximation and Projection
WBC	World Building Congress

After all, we obtained top accuracies of 88.89% and 95.15% in the validation of the clustering methods of sensor classification attending their main activity and the building typology surrounding them, respectively. Finally, we achieved 87.00% accuracy using GCN. All of our attempts, outperformed the results obtained in [5], where they achieved 85.66% of overall accuracy, as it is shown in Table 11.

VII. CONCLUSION AND FUTURE WORK

This paper shows the potential of clustering methods, specifically K-means algorithm, at identifying groups of sensors following different criteria. First, the typology of the buildings that were within a relatively short distance (<300 m) from each sensor was identified and counted the number of buildings that were within such distance for each type. At the same time, the activation pattern of different sensors was evaluated by obtaining the mean activation in 2-hourly time intervals throughout different days of the week.

As it has been presented, the typology of the buildings was the main factor when dividing sensors into different groups. Thus, we can deduce that those sensors were placed in different types of buildings, and so were made the clusters.

On the other hand, the variations of occupants around the sensors should be also a good feature to determine the sensor's typology. After all, the classifications made by different supervised ML algorithms show that this clustering was not as conclusive as the previous one, even if we enlarged the number of sensors collecting data by adding data from the city of Madrid. Trying to improve the classification results based on the pedestrians' activity, a graph convolutional network was performed but there was no significance in improvement of metrics.

However, it was feasible to link clusters based on building typology with the ones based on the pedestrians' activity, revealing that there is a close connection between the activity pattern of the sensors and the type of environment they are located. By this way, it would be possible to tackle different security and energy efficiency tasks by knowing only the building types of an urban zone, not needing any further information. Furthermore, institutions could alleviate a great amount of effort needed to ensure safe and energy efficient urban areas.

Following this research line next step would be forecasting spatiotemporal changes in pedestrians' patterns. By doing so, real time predictions could be made, avoiding different issues and providing cities with a higher security and energy efficiency, among other benefits.

ACKNOWLEDGMENT

The authors would like to thank the TECNALIA Member of Basque Research and Tecnology Alliance for technical support.

REFERENCES

[1] A. Sinaeepourfard, J. Garcia, X. Masip, E. Marin, J. Cirera, G. Grau, and F. Casaus, "Estimating smart city sensors data generation: Current and future data in the city of Barcelona," in *Proc. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net), 15th IFIP MEDHOCNET*, Vilanova i la Geltrú, Spain, Jun. 2016.

[2] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Mar. 2011.

[3] L. Klein, D. Gibson, and M. Mills, *Traffic Detector Handbook*, vol. 1, no. FHWA-HRT-06-108, 3rd ed. McLean, VA, USA: Federal Highway Administration, 2006.

[4] S. Himmel, M. Ziefle, and K. Arning, *From Living Space to Urban Quarter: Acceptance of ICT Monitoring Solutions in an Ageing Society*. Berlin, Germany: Springer, 2013.

[5] L. Zhuo, Q. Shi, C. Zhang, Q. Li, and H. Tao, "Identifying building functions from the spatiotemporal population density and the interactions of people among buildings," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 6, p. 247, May 2019.

[6] J. Yoon and S. Lee, "Spatio-temporal patterns in pedestrian crashes and their determining factors: Application of a space-time cube analysis model," *Accident Anal. Prevention*, vol. 161, Oct. 2021, Art. no. 106291.

[7] L. Hu, X. Wu, J. Huang, Y. Peng, and W. Liu, "Investigation of clusters and injuries in pedestrian crashes using GIS in Changsha, China," *Saf. Sci.*, vol. 127, Jul. 2020, Art. no. 104710.

[8] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," Jul. 2017, *arXiv:1707.01926*.

[9] L. M. Pfiester, R. G. Thompson, and L. Zhang, "Spatiotemporal exploration of Melbourne pedestrian demand," *J. Transp. Geogr.*, vol. 95, Jul. 2021, Art. no. 103151.

[10] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," 2017, *arXiv:1709.05584*.

[11] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.

[12] Y. Xu and D. Li, "Incorporating graph attention and recurrent architectures for city-wide taxi demand prediction," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 9, p. 414, 2019.

[13] Y. Qi, Q. Li, H. Karimian, and D. Liu, "A hybrid model for spatiotemporal forecasting of $PM_{2.5}$ based on graph convolutional neural network and long short-term memory," *Sci. Total Environ.*, vol. 664, pp. 1–10, May 2019.

[14] J. Sun, J. Zhang, Q. Li, X. Yi, Y. Liang, and Y. Zheng, "Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks," vol. 14, no. 8, 2019, pp. 1–12, *arXiv:1903.07789*.

[15] H. Jia, H. Luo, H. Wang, F. Zhao, Q. Ke, M. Wu, and Y. Zhao, "ADST: Forecasting metro flow using attention-based deep spatial-temporal networks with multi-task learning," *Sensors*, vol. 20, no. 16, p. 4574, Aug. 2020.

[16] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY, USA: Springer, 2020.

[17] U. Saralegui, "Occupancy estimation and people flow prediction in smart environments," Facultad de Informática, Universidad del País Vasco/Euskal Herriko Unibertsitatea, San Sebastian, Spain, 2017.

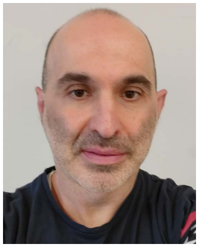
[18] IBM Cloud Education. (Oct. 20, 2020). *Convolutional Neural Networks*. [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neuralnetworks

[19] L. Joyanes, I. Zahonero, M. Fernández, and L. Sánchez, *Estructura de Datos en C++. Libro de Problemas*. New York, NY, USA: McGraw-Hill, 2007.

[20] A. Garmendia-Orbegozo, S. Noye, U. Saralegui, M. A. Anton, and J. D. Nuñez-Gonzalez, "Building typology prediction in smart cities," in Proc. *Proc. CIB W78 Inf. Technol. Construct. 39th Conf. WBC*, 2022.

ASIER GARMENDIA-ORBEGOZO (Member, IEEE) was born in Azpeitia, Gipuzkoa, Basque Country, Spain, in 1996. He received the B.S. degree in physics and electronic engineering and the M.S. degree in embedded systems engineering from the University of the Basque Country (UPV/EHU), whose Director and Co-Director are J. David Nuñez-Gonzalez and Miguel Angel Anton, respectively, where he is currently pursuing the Ph.D. degree in informatics engineering. His research interests include the application of different data mining and machine learning techniques for gaining and generating knowledge and transferring them to edge and end-user devices.

SARAH NOYE (Member, IEEE) received the Industrial Engineering degree from the Écoles Nationale Supérieure des Mines de Nancy, France, and the Ph.D. degree in systems engineering from Imperial College London, U.K. During her Ph.D. degree, she worked with wireless sensors for building commissioning at the Center for Systems Engineering and Innovation, London, providing data analysis solutions to detect deviations between design and actual performance. In 2017, she joined TECNALIA, where she works as a Researcher in the field of artificial intelligence algorithms. She has significant experience in various projects that apply artificial intelligence both in the design and optimization of the operation of smart buildings. She has co-directed several master's theses and student internships on artificial intelligence topics. She currently leads the artificial intelligence strategy focused on graphic systems and semantic models in TECNALIA. Since 2019, she has been co-directing a Ph.D. degree in machine learning metamodels for optimizing the operation and maintenance of buildings.



MIGUEL ANGEL ANTON (Senior Member, IEEE) received the B.S. degree in industrial engineering from the University of the Basque Country, Donostia-San Sebastian, Spain, in 1997, the M.S. degree in automation and industrial electronics from the University of Navarra, Donostia-San Sebastian, in 2004, and the Ph.D. degree in electronics and communications from the University of Navarra, in 2009. From 2003 to 2004, he was a Student Researcher with the Telemedicine Group, Vicomtech Technological Centre. From 2005 to 2010, he was a Ph.D. student and a Junior Researcher in the field of bioinformatics for the development of algorithms for optimization, clustering, and pattern searching using large genomic databases with the CEIT Research Centre. Since 2011, he has been a Senior Researcher with Fundación TECNALIA Research and Innovation, Donostia-San Sebastian. His research interests include the Internet of Things (IoT), embedded intelligence in edge computing, the development of optimization algorithms, and cognitive management of buildings and infrastructures. His scientific and technological production comprises publications on bioinformatics, intelligent building management, energy management, and embedded systems.

J. DAVID NUÑEZ-GONZALEZ (Senior Member, IEEE) received the Ph.D. degree in computer science, in 2016. He is an Associate Professor with the Applied Mathematics Department, University of the Basque Country. He is currently advising undergraduate, master's, and Ph.D. students. He has contributed for several publications (JCR journals, books chapters, and conference papers) and participated in two European projects (SandS from FP7 and CybSpeed from H2020). His research interests include machine learning and artificial intelligence.