Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06872-0

ORIGINAL ARTICLE

Deep learning and Internet of Things for tourist attraction recommendations in smart cities

Juan Carlos Cepeda-Pacheco[1] · Mari Carmen Domingo[1]

Received: 12 April 2021 / Accepted: 15 December 2021

Abstract
We propose a tourist attraction IoT-enabled deep learning-based recommendation system to enhance tourist experience in a smart city. Travelers will enter details about their travels (traveling alone or with a companion, type of companion such as partner or family with kids, traveling for business or leisure, etc.) as well as user side information (age of the traveler/s, hobbies, etc.) into the smart city app/website. Our proposed deep learning-based recommendation system will process this personal set of input features to recommend the tourist activities/attractions that best fit his/her profile. Furthermore, when the tourists are in the smart city, content-based information (already visited attractions) and context-related information (location, weather, time of day, etc.) are obtained in real time using IoT devices; this information will allow our proposed deep learning-based tourist attraction recommendation system to suggest additional activities and/or attractions in real time. Our proposed multi-label deep learning classifier outperforms other models (decision tree, extra tree, k-nearest neighbor and random forest) and can successfully recommend tourist attractions for the first case [(a) searching for and planning activities before traveling] with the loss, accuracy, precision, recall and F1-score of 0.5%, 99.7%, 99.9%, 99.9% and 99.8%, respectively. It can also successfully recommend tourist attractions for the second case [(b) looking for activities within the smart city] with the loss, accuracy, precision, recall and F1-score of 3.7%, 99.5%, 99.8%, 99.7% and 99.8%, respectively.

Keywords Deep learning · Deep neural networks · Multi-label classification · IoT architecture

1 Introduction

The Internet of Things (IoT) is a unique dynamic "network" of smart interconnected objects [1] with self-configuration capabilities, based on standards and communication protocols, whose objective is to provide more control and comfort to everybody as well as resource optimization. The IoT architecture allows interaction and communication between humans and objects anytime, anywhere and with anybody or anything. The IoT can generate an unprecedented amount of events and data (Big Data), which can be sent to the cloud for further analysis and discovery of connections and patterns using Artificial Intelligence (AI) methods. AI is a system based on experience, where machine-learning algorithms are able to learn what are the best actions to facilitate daily tasks or improve productivity without being explicitly programmed.

Smart cities are urban areas that integrate Information and Communication (ICT) technologies; they can significantly benefit from IoT [2] and AI to improve the basic needs of citizens, businesses and institutions. They use technological innovation to address urban challenges, such as waste management, noise control, air quality control, safety, traffic congestion, citizen participation and tourism [3]. *Smart tourism* focuses on the development of innovative tools for the acquisition and adjustment of real-time tourism information through mobile Internet or Internet terminal equipment [4]. It relies on four essential information and communication technologies: IoT, mobile communication, cloud computing and artificial intelligence

✉ Juan Carlos Cepeda-Pacheco
juan.carlos.cepeda@upc.edu

Mari Carmen Domingo
cdomingo@entel.upc.edu

[1] Department of Network Engineering (ENTEL), Universitat Politècnica de Catalunya - BarcelonaTech (UPC), Esteve Terradas, 7, 08860 Castelldefels (Barcelona), Spain

Published online: 11 January 2022

[4]. Smart cities transform into *smart tourist* destinations [5] when they enhance the tourists' travel experience, provide intelligent platforms to gather and distribute information, facilitate efficient allocation of tourism resources and integrate tourism suppliers at both macro- and micro-level. Furthermore, in order to enrich the tourists' experience, it is recommended that all stakeholders are also involved in the co-creation process to increase destination competitiveness.

Nowadays, when a person wants to take a trip to a specific city, the tourism companies offer package tours that include transport and accommodation. They can also provide additional services such as activities or outings during the holiday. These offers are adjusted to a heterogeneous group of people, although each tourist has different needs and characteristics. On the other hand, tourists who travel independently look for activities and attractions that best align with their interests. Therefore, before the trip, they need to search the Internet for possible options and their alternatives (opening hours of museums, ticket prices, weather forecasts, itineraries and transportation, etc.). In so doing, they find very extensive, dispersed and disorganized information. This process represents long hours in front of a computer and causes an unnecessary loss of time. This problem can be solved using recommender systems.

Recommender systems [6] suggest the most appropriate products or services for a particular user. A user's preference for an item is predicted based on information collected from different sources such as: the user's inclination for particular products (songs, movies, etc.), the user's social information (ratings, followed, followers, etc.), his/her demographic features (age, gender, etc.), or the behavioral data from the Internet of Things (e.g., GPS, RFID tags, sensors, etc.).

Collaborative filtering [7] is the most commonly implemented technique in recommender systems. It generates recommendations by identifying peer users with a taste/profile similar to the current user's. Collaborative filtering (CF) methods are classified as memory-based and model-based. Memory-based methods [6] usually use similarity metrics to obtain the distance between two users, or two items, based on each of their ratios. Model-based techniques incorporate machine-learning to learn a model about a user.

On the other hand, content-based recommendation systems suggest similar items based on a domain-specific notion of item content and a user profile. These algorithms try to recommend items similar to those that a user liked in the past or is examining in the present. In addition, context-based recommendation systems apply sensing and analysis of user context (location, user activity, etc.) in order to provide personalized recommendations [8].

When there is not enough information to make a recommendation that results in the 'cold start problem,' the 'cold start problem' refers to the entry of a new user into the recommender system. No recommendations can be made to this new user due to the lack of previous data [9].

Artificial neural networks (ANNs) [10] have gained in popularity as an excellent tool for classification, clustering, pattern recognition and prediction in many areas. Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning, where feature or representation learning refers to a set of techniques that allows a system to automatically discover the representations needed for feature detection or classification from raw data. Deep-learning architectures such as deep neural networks have produced results comparable to and in some cases surpassing human expert performance [11].

Recently, deep learning-based recommender systems have also obtained very promising results [12, 13]. Furthermore, although IoT is considered a key concept in smart tourism, it is very rarely applied in smart tourism recommendation systems [14]. To fill this gap, we propose a tourist attraction IoT-enabled deep learning-based recommendation system to enhance the tourists' experience in a smart city. Travelers will enter the particular circumstances of a travel, such as traveling alone or with a companion, type of companion (partner, family with kids), traveling for professional or vacation purposes, and user side information (age of the traveler/s, hobbies, etc.) into the smart city app/website. This information is useful to improve the accuracy of the tourist attraction recommendations. For example, tourists who travel with children will avoid visiting many museums, and will consider practicing outdoor activities, such as going to beaches or parks. Our proposed deep learning-based recommender system will process this personal set of input features to recommend the activities or attractions that best fit the tourist's profile. Furthermore, when the tourists are in the smart city, content-based information (already visited attractions) and context-related information (tourists' location, weather, time of day, etc.) are obtained in real time using IoT devices. This information will allow our proposed deep learning-based tourist attraction recommender system to suggest additional tourism activities and/or attractions in real time based on the tourist's own choices, the current time and how far the new places are.

Therefore, we distinguish between two different cases (a) searching and planning activities before traveling and (b) looking for activities within the smart city.

In the first case (a), recommendations will be generated using model-based collaborative filtering and multi-label classification. In the second case (b), recommendations will

be generated using a hybrid-based (model-based collaborative filtering, content and context) approach and IoT context-related data (e.g., location, weather). The contributions of this paper are summarized as follows:

- We propose a tourist attraction recommender system for smart cities based on deep learning.
- It alleviates the cold start problem by integrating side information about users into a deep neural network.
- It collects IoT data about the tourist's visit in the smart city to do real-time attraction recommendations.
- It includes the already visited tourist attractions to improve the accuracy of the recommendations with the tourist's own choices.
- We have evaluated several multi-label classification techniques (decision tree, extra tree, k-nearest neighbor (kNN) and random forest) for tourist attraction recommender systems to determine the best overall.
- Deep neural network is the top performing multi-label classification algorithm. Our proposed multi-label deep learning classifier outperforms the other models and can successfully recommend tourist attractions for the first case ((a) searching and planning activities before traveling) with the loss, accuracy, precision, recall and F1-score of 0.5%, 99.7%, 99.9%, 99.9% and 99.8%, respectively. It can also successfully recommend tourist attractions for the second case ((b) looking for activities within the smart city) with the loss, accuracy, precision, recall and F1-score of 3.7%, 99.5%, 99.8%, 99.7% and 99.8%, respectively.

Our paper provides an approach to address the gaps in real-time IoT-enabled tourist attraction recommender systems. We address different aspects in building a more accurate recommender system using deep learning. Our tourist attraction recommender system takes into account the particular circumstances of a travel (traveling alone or with a companion, type of companion such as partner, family with kids, etc.) as well as user side information (age of the traveler/s, hobbies, etc.) to alleviate the cold start problem and improve the accuracy of the recommendations. Content-based information (already visited attractions) and context-related information (tourists' location, weather, time of day, etc.) are obtained in real time to improve its effectiveness. The grid search technique was used to obtain, after a long training, the optimal values for the correct performance of the deep learning algorithm. To the best of our knowledge, this is the first recommender system that does real-time tourist attraction recommendations based on IoT context-related data and deep learning.

This paper is structured as follows. Section 2 discusses the related work. Section 3 introduces the proposed IoT smart tourism architecture. Section 4 covers the proposed deep learning algorithm for smart city tourism. The experiments carried out and results obtained are introduced in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Related work

Tourism has become a sector that contributes significantly to the economy of many cities. Cities have become more and more competitive and seek to expand their offerings and provide a better experience in order to attract more tourists. By becoming a smart city, allows cities to provide a more immersive, richer, and more personalized user experience. IoT and ICT are essential building blocks of a smart city. They can be used by smart cities to create new and exciting products for tourists [15]. IoT also enables the development of a monitoring and control system for the protection and preservation of cultural heritage [16–18]. It can be used to build an early warning system for disasters [19]. In [20], a mobile application detects cultural objects equipped with a Bluetooth low energy (BLE) sensor nodes. The cultural objects are ranked depending on the user preferences, and multimedia content related to the objects is recommended. Similarly, the authors in [21] propose an indoor location-aware architecture that uses image recognition and a localization technique based on BLE to provide the users with cultural contents related to the observed artworks. Both proposals [20, 21] aim to improve the user's experience indoors (museum, art gallery, etc.). Furthermore, in [22], the authors propose a Deep learning (DL) methodology that enables the prediction of visitors' paths within a museum with two purposes: (1) to acquire more information about visitors' behaviors and (2) to predict the occupancy of the available rooms within a cultural space.

Due to the large number of attractions and activities that can be carried out while visiting a city, making a suitable itinerary can be a burdensome task for the tourist. This problem can be solved by recommender systems.

Recommender systems (RS) are models that let users find content or alternatives that may be of interest to him/her [23, 24]. RS must be relevant, novel, or diverse to ensure that users receive precise information according to their requirements. RS have been deployed in entertainment, e-commerce, services, social media, etc. The authors in [14] present a systematic review covering all major aspects of e-tourism management systems applied to smart tourism during the last few years. They also classify the techniques used by recommender systems. The most common are collaborative filtering, content, context and hybrid-based models.

CF-based models employ user–user or item–item similarity to generate recommendations [25, 26]. This is the most common method used by classic recommender systems, but it suffers from the cold start problem.

A content-based recommender system works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link).

The implicit preferences of the user for certain points of interest (POI) such as restaurants, museums, parks can be obtained from three main sources:

- Geotagged photos from social media [27–33].
- Location-based social networks [34, 35].

Context-based models add information related to the user's context (location, user activity, etc.) for its application to provide personalized recommendations [36].

In [27–35], user information is obtained from social networks, photo-sharing websites and the location of the users without any personal identifiable information (PII). In contrast, we include information about the particular circumstances of a user's trip to the smart city to improve the accuracy of our tourist attraction recommender system. In our research, we rely on a hybrid model-based TRS, because each technique, when used separately, has its strengths and weaknesses. Therefore, by combining these techniques with deep learning techniques, we are able to complement them to generate a recommendation system that satisfies a broader range of users' needs.

In [37], the use of machine learning in recommender methods can achieve higher predictive accuracy for online reviews (TripAdvisor). In addition, due to the significant achievements of deep learning in many fields such as Natural Language Processing (NLP) and image processing, deep learning methods have been used for recommender systems [38, 39]. Cultural heritage recommendation systems [40–43] evaluate and classify the visitors' behavior during a cultural event, provide information that matches the preferences of visitors, recommend artwork that could be of interest to the user, or even suggest routes. Our study goes beyond these cases and is not limited to recommendations in the same environment/event.

In [44], the authors present an architecture and a conceptual framework for a hybrid tourism recommender system based on big data and artificial intelligence. In [45], the authors propose a hybrid recommendation system to combine the predictions from the content-based filtering (CB), collaborative filtering (CF) and demographic filtering (DF) approaches using the neural network model; they compare their results with each of the traditional approaches individually, providing a better focus for recommending tourist sites in Taiwan. They use a neural network to recommend tourist attractions based on the user's profile. In this paper, we also use this approach when the tourist is searching for, and planning activities before traveling to the smart city. However, we also provide real-time tourist attraction recommendations using IoT context-related information (location and weather forecast) once the tourist is looking for activities within the smart city. Additionally, we present in detail the deep learning techniques used for tourist attraction recommendations.

3 IoT smart tourism architecture and methodology

When the tourists are in the target smart city, content-based information (already visited attractions) and context-related information (tourists' location, weather, time of day, etc.) are obtained in real time using the IoT. This information allows our proposed deep learning-based tourist attraction recommender system to suggest additional tourism activities and/or attractions in real time based on the tourist's own choices, the current time and distance from the new places.

3.1 IoT smart tourism architecture

Our flexible IoT smart tourism architecture is shown in Fig. 1. A three-layer proposal is presented as detailed below:

Device layer: this layer is responsible for identifying objects and for receiving information through sensors and monitoring the environment.
Fog layer: this layer allows the transmission and processing of sensor data in a distributed manner for those services that are sensitive to latency.
Cloud layer: this layer provides intelligent services that generate a global repository of relevant information and provides recognition and learning patterns, which are fed from the other layers.

Next, we will go through the details of each layer.

3.1.1 Device layer

As shown in Fig. 1, this layer is composed of physical devices such as sensors and actuators whose main function is to collect and process information. This broad set of intelligent IoT devices allows the system to monitor the user's movement within a smart city any time, from any computer or mobile device. The collected information is securely stored in the cloud, analyzed, and processed by our machine-learning algorithm to provide user specific recommendations.

The suggested IoT smart tourist devices are the *Global Positioning System (GPS)*, *temperature sensors*, *RFID, sensors* and *video cameras*. When tourists go sightseeing in the smart city, the attractions that they visit are registered using *GPS*. This way, we determine which recommendations the users acted upon and recommend new similar

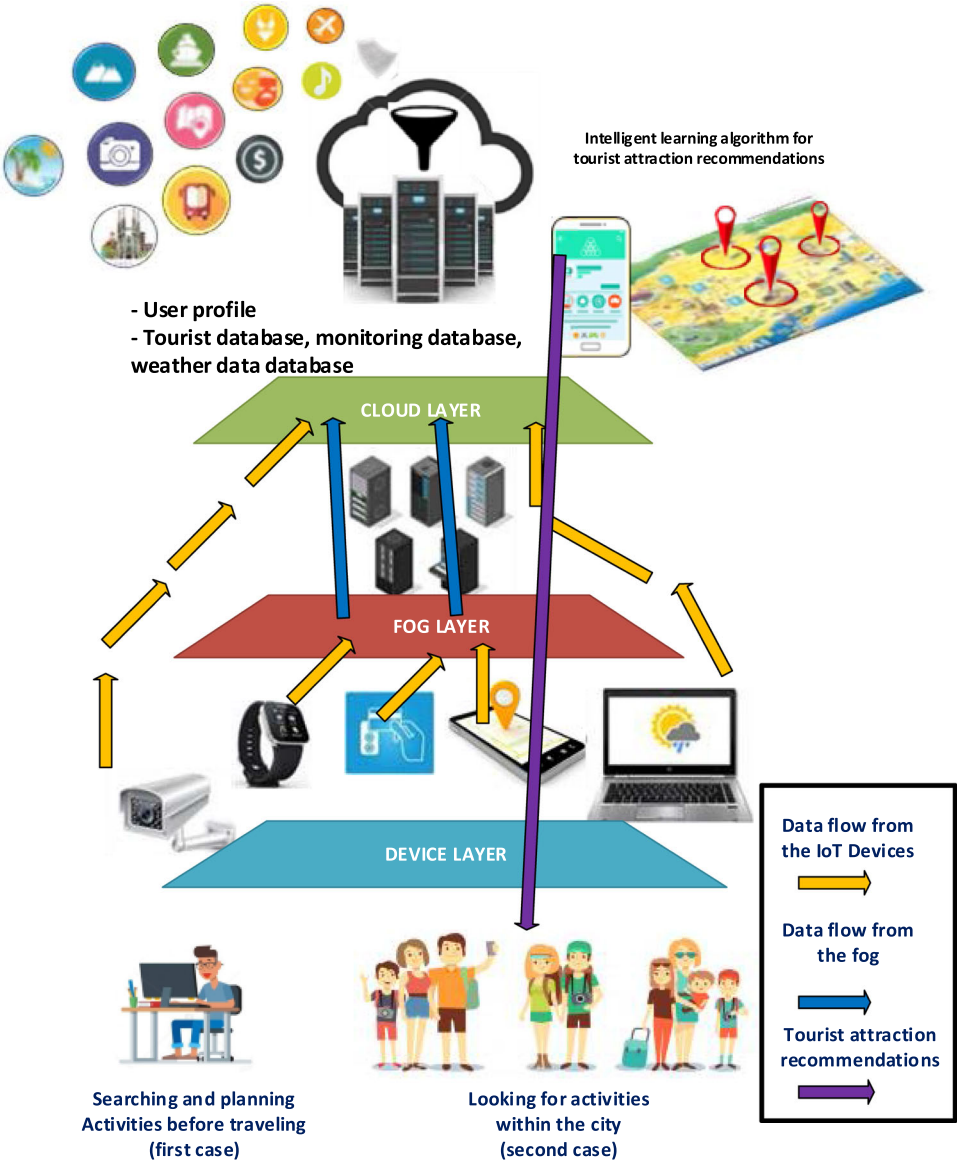


Fig. 1 Schematic representation of the IoT-based smart tourism architecture

attractions still based on their own preferences, their current location and the current time of day.

One of the main elements to consider before a trip or sightseeing tour is the weather; by means of *temperature sensors,* the weather conditions are predicted to define possible routes and activities that can be carried out outdoors (e.g., parks) or indoors (e.g., museum visits) depending on the weather.

Tourists' behaviors can be captured more precisely by RFID, *video cameras* and other sensors located in strategic places, such as stores, museums, churches or entertainment places; this constant monitoring allows to update the tourists' profile and to improve future recommendations [46].

Furthermore, photographs uploaded on the web by tourists are a very powerful tool to obtain additional information about the user (granted that access is allowed by the user). They are processed through image recognition to obtain behavioral patterns and make even more appropriate recommendations.

3.1.2 Fog layer

The *fog layer* is shown in Fig. 1 as the middle level in the current architecture; it offers real-time analysis and preprocessing services; afterward, the main information is transferred upwards to the servers in the cloud for storage and further treatment.

The sensors capture information from the *device layer* to send to the network, which can be congested with the huge amount of data. The *fog layer* should be understood as a part of the distributed architecture that expands between

the cloud and the edge networks. This improves efficiency and reduces the volume of data transferred, because critical tasks (such as computing, communication, storage and decision-making) are distributed and closer to where the data are generated [47]. Therefore, the workload of the cloud and user devices is reduced. This is particularly important for time-sensitive IoT applications that require very low latency.

3.1.3 Cloud layer

This layer is responsible for the delivery of user-specific application services. It allows processes to be transferred to central servers distributed throughout the world, connected to the Internet through a high data throughput connection. It also supervises the applications of Internet of Things, implementing decision-making processes based on Big Data analysis [48].

This layer receives all the data (which do not require immediate processing) generated by the different layers. It then carries out the functions of processing, analysis and storage, becoming a global repository of relevant information to be used for automation or decision making at a later time.

3.2 Proposed methodology

At this level, our machine-learning algorithm operates to recommend tourists in smart cities the best attractions/sights to visit according to their information profile and their IoT data. These suggestions are adjusted to each user with the aim of obtaining a better experience when visiting a certain smart city. Since the raw data generated from sensors, GPS, etc., can be voluminous, rather than forwarding it to the cloud for processing, the idea is to do as much processing as possible in the fog layer using computing units. That way, processed rather than raw data is forwarded to the cloud (or the devices in the device layer if necessary). A tourist database located in the cloud stores the dataset related to the tourist profile. A monitoring database located in the cloud stores the real-time location of tourists in the smart city and the places that they have already visited. The mobile device of the tourist monitors in real-time the tourist's location with the GPS and forwards this data to the monitoring database. Another database located in the cloud stores weather data. The weather sensors located in the device layer sense and forward the temperature to this database. The weather forecasts are also stored.

An app related to our proposed tourist attraction recommender system is installed on the mobile devices of the tourists. The proposed DNN with multi-class classification algorithm in the cloud will access the data of the different databases and use it as input data to return the most appropriate tourist attractions according to the user's profile. Figure 1 illustrates the different data flows. It can be observed that the raw data are transferred to the fog or to the cloud. The machine learning algorithm in the cloud returns an appropriate attraction recommendation to the device (mobile app) of the tourist based on predictions. In the following section, we will explain in detail our deep neural network algorithm tailored to our proposed approach.

4 The proposed deep learning algorithm for smart city tourism

We have developed a tourist attraction recommendation system based on deep learning. We distinguish between two cases; the first one is when the tourist plans his/her trip and the second one, when the tourist is already in the smart city and is looking for new alternatives during his/her visit.

Figure 2 shows the block diagram of our proposal. A traveler will enter the particular circumstances of his/her trip (traveling alone or with a companion, type of companion such as partner, family with kids.) as well as user side information (age of the traveler/s, hobbies, etc.) into the smart city app/website. Our proposed deep learning algorithm will process this personal set of input features to predict and recommend the tourist activities or attractions that best fit his/her profile. For this purpose, the system must be trained with previously collected data, which consists of information about the most relevant tourist attractions in the city and the information provided by tourists who have previously visited the city. The information provided by tourists will be collected through surveys, which will contain information about the attributes of the tourists (tourists' profile) (dataset). Furthermore, for the second case, the system will pick up information (location, weather forecast and already visited places) from the database collected by the IoT devices in real time. These attributes will be the input data that will be normalized and processed by our deep learning algorithm. Nowadays, there are different deep learning techniques that have been created to facilitate the learning process in specific tasks. These techniques include:

- *The Multilayer Perceptron (MLP)* consists of a neural network with multiple hidden layers (deep neural network). The network is trained allowing the perceptron to adjust its parameters so the algorithm learns in order to achieve high accuracy for the task to be accomplished. This network is known for its wide range of application in different areas of knowledge and is also considered one of the most versatile architectures thanks to its applicability.

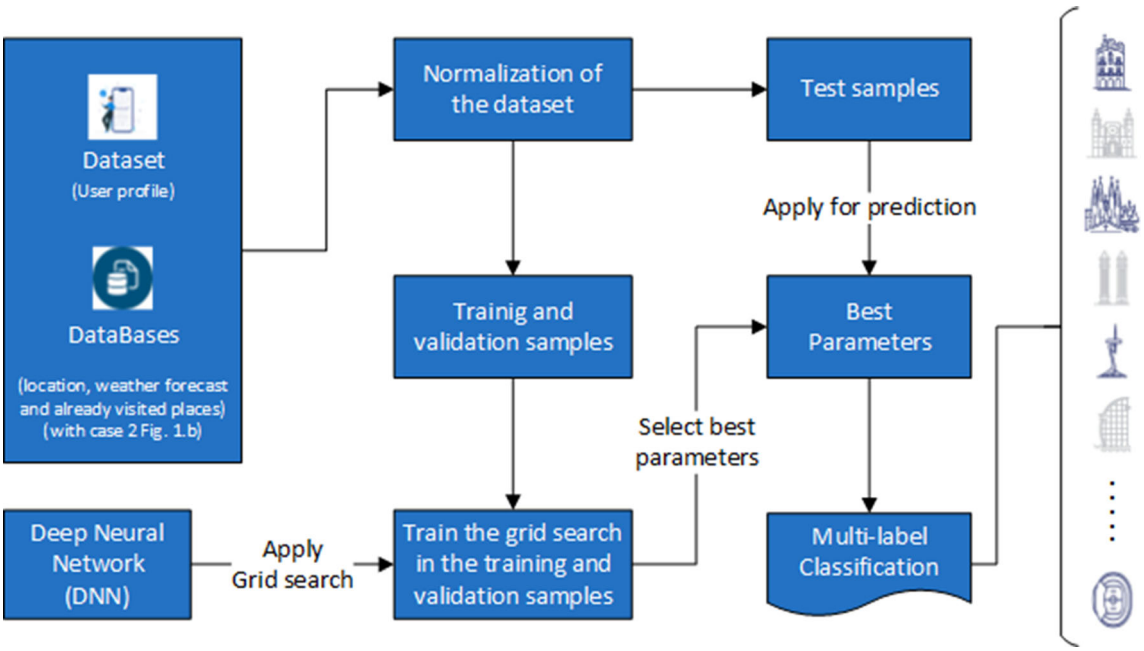


Fig. 2 Block diagram and workflow of the proposed method

- *An autoencoder (AE)* is an unsupervised neural network model trained to attempt to copy its input to its output. Traditionally, autoencoders were used for dimensionality reduction or feature learning [49].
- *The Convolutional Neural Networks (CNN)* is a specialized kind of neural network for processing data that has a known, grid-like topology. Because of the fact that its application is performed on two-dimensional arrays, this variation of a multilayer perceptron, is very effective for computer vision tasks, such as image classification and segmentation, and other applications [50].
- *The Recurrent Neural Network (RNN)* is a neuronal network specialized in the processing of data sequences. It incorporates Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU) type layers that allow backpropagation through time; by connecting events that appear far apart in the input data since if the sequence is too long, the vanishing gradient problem may appear [51].
- The *Restricted Boltzmann Machine (RBM)* is two-layer surface neural networks that are the building blocks of deep-belief networks. The first layer of the RBM is called the visible or input layer, and the second is the hidden layer. The restriction in a constrained Boltzmann machine is that there is no communication between layers. Each node is a computational place that processes the input and starts making stochastic decisions about whether to transmit that input or not [52].
- The *Adversarial Network (AN)* is a type of convolutional neural network that generates images or songs, that are as realistic as possible, from a generator and a discriminator [53].
- *Deep reinforcement learning (DRL)* combines reinforcement learning (RL) and deep learning (DL). RL addresses the problem of automatically learning optimal decisions over time, i.e., the problem of a computational agent learning to make decisions by trial and error. It consists of the following components: agents, environments, states, actions, and rewards. Deep reinforcement learning has been deployed in a diverse set of applications, including (but not limited to) robotics, video games, natural language processing, computer vision, education, transportation, finance and healthcare [54].

The use of different techniques for the same application provides better results in some cases than others. Therefore, the MLP technique is used in this research to achieve the proposed objective, which is to develop a tourist attraction recommender system. After having reviewed the different deep learning techniques and taking into account that we are doing supervised learning, we have decided to use the multilayer perceptron (MLP) technique which consists of a neural network with multiple hidden layers (deep neural network). The network is trained by the perceptron adjusting its parameters. This way, the algorithm learns with the aim of achieving a high accuracy for the task to be accomplished. The output provides

recommendations for tourist attractions according to the user's profile.

4.1 Proposed DNN

The proposed deep neural network consists of a number of hidden layers with a required number of neurons. A vectorized representation of neurons in a hidden layer is given by

$$a^{[l]} = f\left(W^{[l]T}a^{[l-1]} + b^{[l]}\right). \tag{1}$$

where $a^{[l]}$. is a vector and each element represents a nron in layer l, $W^{[l]}$ is the weight matrix in layer l. such that $W^{[l]} \in \mathcal{R}^{i\times j}$, where i is the number of nodes in the hidden layer l. and j is the number of nodes in the previous layer (including the bias term $b^{[l]}$). Each neuron in the network is a nonlinear combination of inputs $a^{[l-1]}$.ted by the parameters w_i. f is the activation function. The proposed model implements two activation functions for the hidden layers and the output layer.

Tectified linear unit (ReLU) has been used as activation function for the hidden layers:

$$f(x) = \max(0, x) \tag{2}$$

The sigmoid function has been used as activation function for the output layer:

$$f(x) = \left(\frac{1}{1+\exp^{-x}}\right) \tag{3}$$

For predicting the different tourist sites, we consider four possible events, according to the number of days of the stay. Thus, when the tourist spends more days in the city, we can recommend a greater number of tourist attractions (see Table 2 in Sect. 5.3). For example, for tourists that stay one to three nights in the smart city, the system will recommend 8 out of 40 possible tourist attractions.

The output layer is composed of 40 neurons, one for each of the tourist sites that the system can recommend; the neurons are activated according to the number of recommendations made to each user depending on his/her profile (see Fig. 3). The output of the last feed-forward layer is passed through a sigmoid activation function to scale each of its element values in the range [0,1]. The model is trained using binary cross-entropy as the loss function, which is defined as

$$\mathcal{L} = -\frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{U}(y_{ij}\cdot\log(\sigma(z_{ij})) + (1-y_{ij})\log(1-\sigma(z_{ij})) \tag{4}$$

where σ denotes the sigmoid function, z_{ij} is the jth element in z_i, and y_{ij} is the ground truth value foR jth tourist (label) and ith tourist attraction. U is the number of tourists and T is the number of tourist attractions.

Gradient descent finds the minimum of an objective function by taking steps proportional to the negative of the gradient at the current point. A learning model estimates the weights by computing partial derivatives of the weight vector at each point and stopping when the minimum of the error function is reached.

$$w_{k+1} = w_k - \alpha\frac{\partial J(w,b)}{\partial w} \tag{5}$$

$$b_{k+1} = b_k - \alpha\frac{\partial J(w,b)}{\partial b} \tag{6}$$

where α is the learning rate, $J(w,b)$ is the cost function.

In order to compute the derivatives for a neural network, we apply the backpropagation technique to minimize the cost function using gradient descent. From (1), let $z^{[l]}$ represent the linear combination of weights and inputs in layer l, such that $z^{[l]} \in \mathcal{R}^i$ where i is the number of nodes in layer l.

$$z^{[l]} = W^{[l]T}a^{[l-1]} + b^{[l]} \tag{7}$$

In addition, to fine-tune our system, each layer will be followed by Batch Normalization and Dropout. The selection of each parameter that composes this deep neural network will be explained later after the implementation of the grid search method, which is detailed in the next section.

5 Experiments and results

Next, we report the results of the experimental evaluation of the proposed tourist attraction recommendation system.

5.1 Smart city selection

We have selected Barcelona (Spain) as reference for the implementation of our research, since it is considered within the world ranking among the 20 most visited cities by foreign tourists (see Fig. 4) [55] and among the 10 most visited cities in Europe (see Fig. 5) [56].

Moreover, Barcelona is considered worldwide as a reference of smart city in mobility, transport, urban planning, governance, technology, etc. [3, 57]. Due to the great number of tourists that visit this city every year, we can count on a great variety of tourist profiles [58].

According to the data published [58], the profiles of tourists arriving to this tourist destination from 2014 to 2018 have the following characteristics: 40.44% of tourists are women and 59.56% are men, 65.98% travel for vacations and 49.54% visit this city for the first time. The

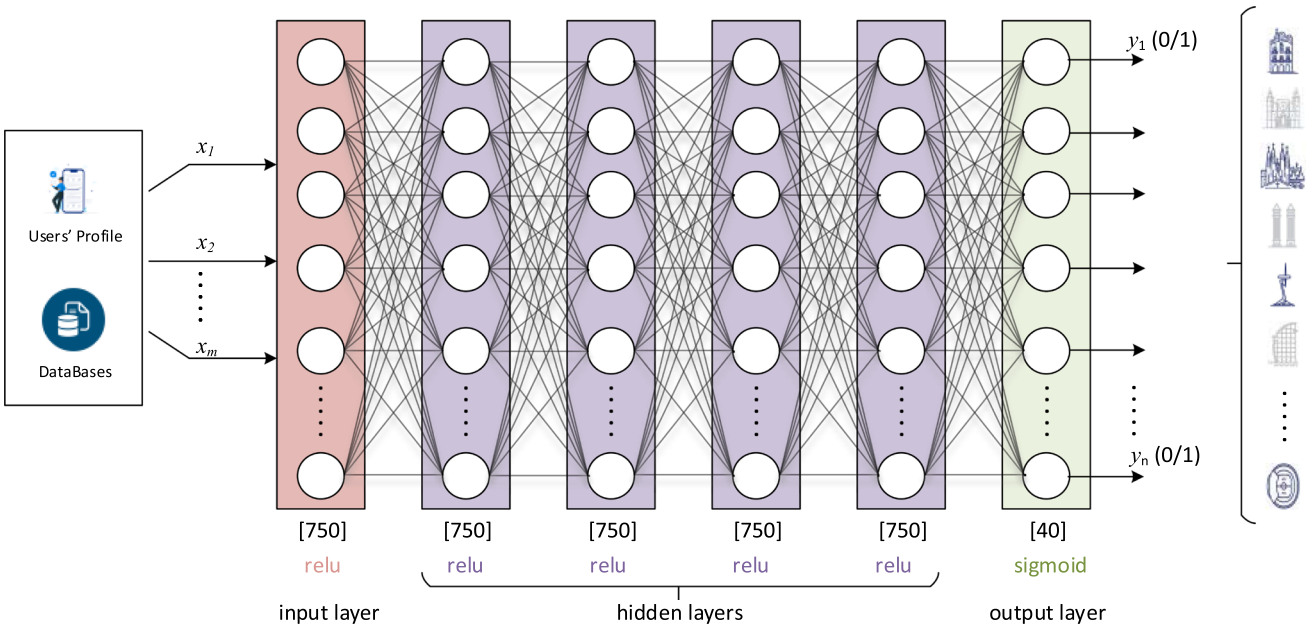


Fig. 3 DNN structure created for the proposed recommender system with multi-label output

average age of visitors is 37 years, and 95% use various mobile applications to search for information on the Internet. In addition, the average stay is one week.

The city of Barcelona has many and varied tourist attractions to meet the requirements and preferences of the visitors. Many of these visitors come to the city with the purpose of changing their daily routines, alone, in groups of friends or with the family.

Furthermore, the city of Barcelona keeps accurate records of the most visited places by tourists in recent years (see Table 1) [59].

5.2 Input data

Next, we explain in detail our dataset as well as the IoT devices database.

5.2.1 User profile

Our dataset is a collection of data acquired through surveys designed to profile each tourist, in order to generate predictions and recommendations. These recommendations are based on the preferences of other tourists with similar profiles.

The survey was carried out in the municipality of Barcelona during summer 2019. 1000 surveys were conducted, which represents 1000 different profiles of tourists or groups of tourists. The interviews were done on paper or through digital media. The surveys were conducted close to tourist attractions or in their vicinity (museums, monuments, etc.), city access points (airport, train station, bus station and cruise terminal) and hotels. For each of these places, the interviewees were chosen randomly throughout the day during working days, weekends and holidays.

For the preparation of the survey, certain characteristics of the tourist's profile were considered; these attributes (X_i) were associated with the data samples and are grouped into the following categories:

(1) *Members Attribute:* One attribute with five categorical data that contain the characteristics of the traveler, whether he/she travels alone, as a couple, in family with or without children or with a group of friends.
(2) *Reason for the Trip Attribute:* One attribute with four categorical data denotes the characteristics that motivate the trip, which range from vacation to professional.
(3) *Age Attributes:* Six attributes which group the ages of the members of the travel group.
(4) *Activities Attributes:* Nine attributes with the different activities that can be carried out by the travel group during the stay. This makes it possible to identify the different activities associated with the respective ages of the travel group members.
(5) *Sports Activities Attribute:* One attribute with three categorical data that identify the types of sports activities, ranging from the most traditional to those of an extreme nature.
(6) *Number of Days of Stay Attribute:* One attribute with five categorical data that indicate the number of days available for visiting the city.

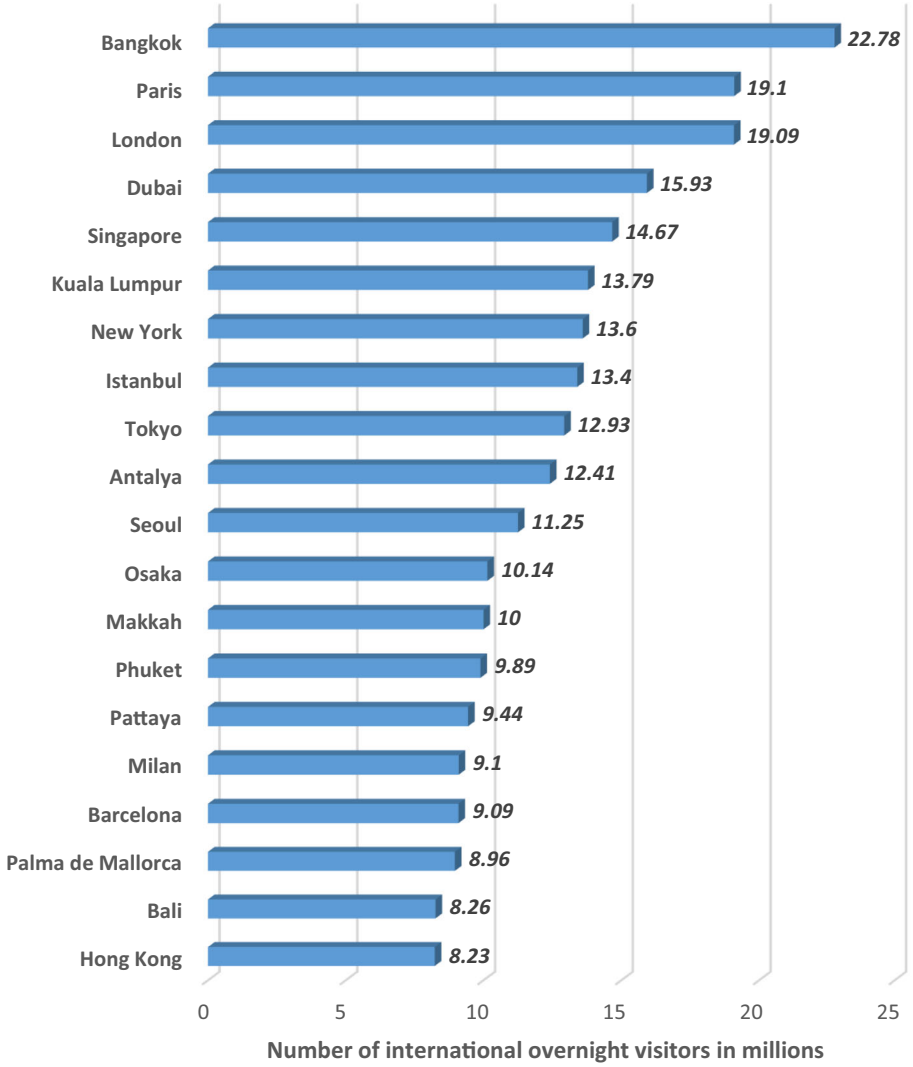


Fig. 4 International overnight visitors in the most popular city destinations worldwide 2018 (in millions)

(7) *Time of the Year / Dates of Stay Attribute:* The dataset also has one attribute with four categorical data that indicate the season of the year or month during the trip.

5.2.2 Database from IoT devices

Once the tourist is already in the city of Barcelona, the system uses GPS to detect his/her location and distances from other landmarks (thirteen attributes), as well as sensors to detect the temperature (an attribute with six categorical data), weather forecast (one attribute with three categorical data) and hour (one attribute with twelve categorical data) to recommend places according to his/her location. Furthermore, the already visited tourist attractions will be detected by GPS. A database of these places will be maintained. After finishing the visit to a tourist attraction, the tourists will rate the attraction based on his/her individual taste. These data are fed back into the system for future recommendations based on the tourist's own preferences.

5.3 Output data

We have grouped stays in four categories depending on the number of days of the stay. Thus, when the tourist spends more days in the city, we can recommend a greater number of tourist attractions to visit as shown in Table 2.

Our proposed DNN is trained for the tourist attraction classification task. Classification is the goal to learn a mapping from inputs x to outputs y, where $y \in \{1, \ldots, C\}$, with C being the number of classes. If $C > 2$ and the class labels are not mutually exclusive (e.g., somebody is classified as tall and strong), it is called multi-label classification [60]. The output of the proposed DNN generates a multi-label classification with the number of sites predicted according to the number of days of the stay. Forty tourist

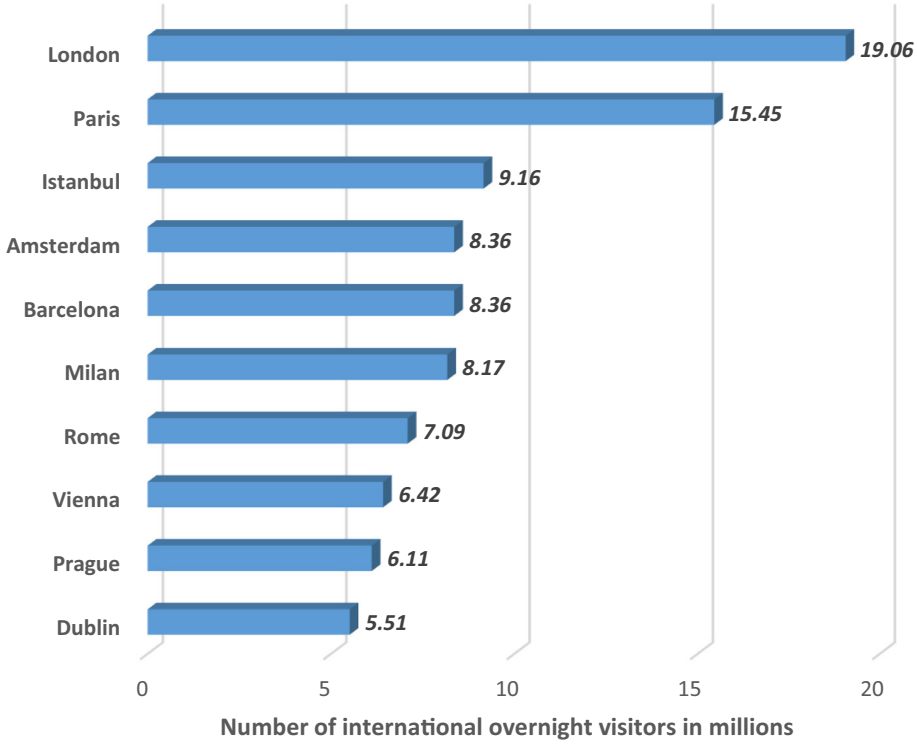


Fig. 5 Number of international overnight visitors in the most popular European city destinations in 2016 (in millions)

Table 1 Number of visitors of the tourist attractions in Barcelona for the period 2014–2018

	2014	2015	2016	2017	2018
Expiratory temple of the Sagrada Familia	3.260.880	3.722.540	4.561.848	4.527.427	4.661.770
Park Güell	2.598.732	2.761.436	2.958.901	3.120.733	3.136.973
FC Barcelona Museum President Núñez	1.530.484	1.785.903	1.947.014	1.848.198	1.730.335
The Barcelona Aquarium	1.590.420	1.549.480	1.587.828	1.626.193	1.631.108
Poble Espanyol de Montjuïc	1.236.664	1.221.647	1.299.376	1.299.386	–
The Born Cultural Center	1.894.400	1.486.228	1.306.230	1.190.762	1.080.079
Casa Batlló	930.000	992.126	–	1.136.000	1.062.863
CosmoCaixa Barcelona	739.649	733.778	757.245	884.636	1.045.961
Picasso Museum	919.814	1.008.125	954.895	1.046.190	978.483
Palau Robert	810.000	715.000	827.957	865.776	976.276
Catalonia Foundation. Stone mine	932.356	990.112	1.207.087	972.508	934.524
National Art Museum of Catalonia (MNAC)	718.230	717.211	820.516	866.271	891.346
CaixaFòrum Barcelona	775.068	775.020	753.944	748.140	863.605
Montjuïc Castle	577.639	670.526	734.460	761.729	831.210
Barcelona History Museum	973.034	916.517	926.571	926.184	816.989
Barcelona Zoo	1.057.188	1.004.069	965.292	834.885	785.992

sites can be recommended according to the profile of each tourist.

5.4 Experimental settings

For training, from the surveys, we get the user's profile information with his/her likes and preferences. We also use the information from the database of IoT devices, thus generating an attribute matrix of 1000 × 20 (1000 samples, 20 attributes) (X_i) for the first case and 1000 × 36 (1000 samples, 36 attributes) for the second case, also from the survey we get the information of the places he/she has visited and that have been most liked; considering the days of his/her visit getting a 1000 × 40 label matrix (Y_i) with

Table 2 Recommended tourist attractions based on the total visit duration

Number of days to stay	Number of tourist attractions
1–3 days	Eight sites
1 week	Twelve sites
2 weeks	Sixteen sites
More than 3 weeks	Twenty-four sites

eight, twelve, sixteen or twenty-four sites. The experiments were performed with 1000 samples, which were divided into two parts keeping ratio 7:2:1, for training, validation, and testing, respectively. The experiments were carried out on Dell computer 2.5 GHz Intel (R) Core(TM) processor with 16 GB RAM. The algorithms were implemented in several Jupyter Notebooks in version 6.0.3 installed with anaconda programs suite, developed by Python. We have implemented the algorithms using Python Keras library, and accuracy, loss, F1-score, recall and precision have been selected during the training process as metrics to evaluate the performance of the algorithms.

5.5 Neural network modeling and optimization

Machine Learning models have several parameters to adjust their behavior in order to reduce overfitting. The most common alternative is to use dropout, which has a parameter that must be tuned for optimal performance [61]. An alternative to reduce overfitting is to optimize the set of parameters through a process known as *grid search* and try to find the most appropriate combination that provides greater precision. The approach used by *grid search* is an exhaustive search by the brute force paradigm in which a list of values for different parameters is specified, and the computer evaluates the model's performance for each combination of these parameters to obtain the optimal set that gives us the highest performance [62]. The result is the values of the adjusted parameters such as the number of hidden layers, the number of neurons in each layer, the number of epochs, the size of the batch of the neural network, as well as the hyperparameters such as the dropout value. In this paper, the grid search technique was used; after a long training, it allows to obtain the optimal values for the correct performance of the algorithm.

5.5.1 Network dimensions and tuning

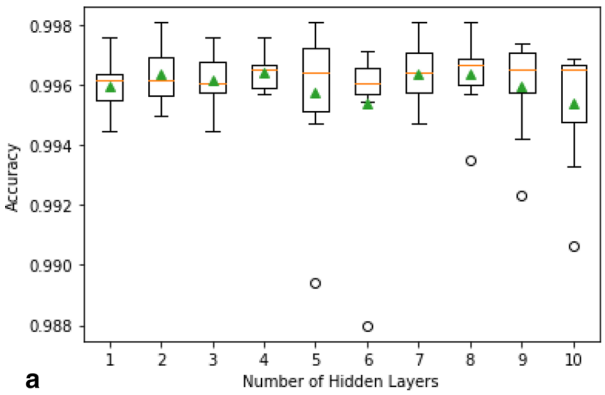
Two very important parameters for deploying a DNN are the depth and width of the network, i.e., the number of hidden layers and the number of neurons per hidden layer. On the one hand, the design of a shallow neural network can lead to the fact that in training it does not collect the correct information for an appropriate prediction. On the other hand, a very deep neural network may fall into overfitting in training, so the system learns the training data memorized, becoming improper to make future predictions. Accordingly, different topologies of neural networks are tested with network depth varying from 1 to 10 hidden layers and from 200 to 800 neurons for each layer.

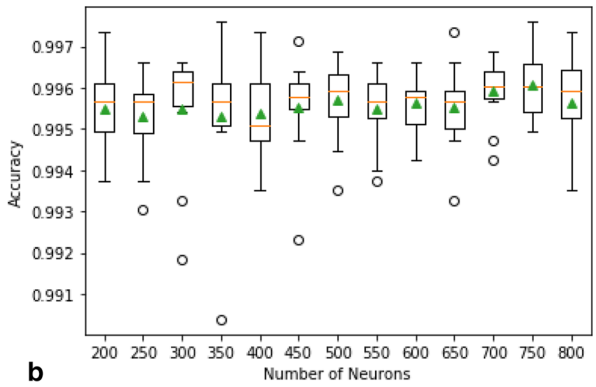
(1) *Network depth*: Improvement in network performance is based on the depth of the network. In Fig. 6.a, box and whisker plots are shown; the performance of the DNN improves with a higher number of hidden layers. In Fig. 6a, the average accuracy shows that the system obtains better performance when it reaches four hidden layers with an accuracy of 99.65%. Although in each case, the system achieves a considerable value in most of the samples, we can notice the system with four hidden layers achieves a better stability on training and it deteriorates when the number of hidden layers continues increasing, meaning that the training may become inadequate.

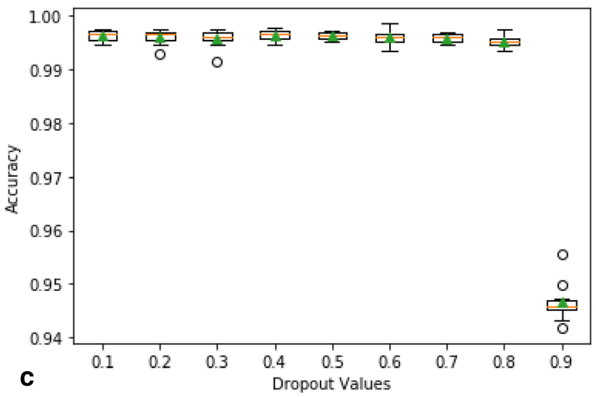
(2) *Network width*: This parameter measures the impact of the number of neurons per hidden layer on the performance of the overall system. Figure 6.b shows the results with 4 hidden layers when the number of neurons varies between 200 and 800 neurons. We conclude that the performance of the neural network improves with a higher number of neurons per layer. The average accuracy shows that the system obtains a better performance when it reaches 750 neurons, achieving 1% higher performance than the case with 400 neurons.

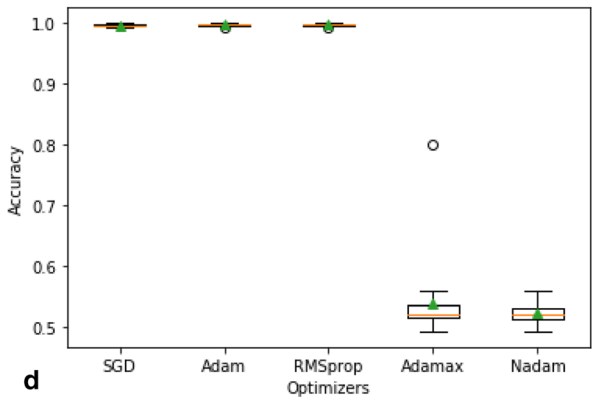
(3) *Network tuning*: To avoid overfitting, we have considered the dropout technique. Figure 6.c shows the most suitable value for an optimal performance of the neural network. The average accuracy shows that the system performs best when it reaches a value of 0.4 in a range of 0.1 and 0.9, reaching an optimal value of 99.8%. Note that the system achieves a remarkably high accuracy.

We have evaluated the SGD, RMSprop, Adam, Adamax and Nadam optimizers using the grid search technique. Grid search is a tuning technique that computes the optimal values for the hyperparameters. Figure 6d shows the box and whisker plot of the accuracies of each optimizer. The SGD optimizer achieves an accuracy of 99.55%, the Adam 99.62%, RMSprop 99.61%, Adamax 53.70% and Nadam 52.23%. In all cases, the learning rate is 0.001. The Adaptative Moment Estimation (Adam) optimizer has been selected to train the deep neural network because it obtains the best results. It minimizes the loss function and speeds up the training process.









◀**Fig. 6** DNN's performance. **a** The box-and-whisker plot of accuracy versus number of hidden layers. **b** The box-and-whisker plot of accuracy versus number of neurons per hidden layer. **c** The box-and-whisker plot of accuracy versus dropout values. **d** The box-and-whisker plot of accuracy versus most popular optimizers

Table 3 Best hyperparameter values found after the grid search process

	Number of Hidden Layers	Number of Neurons/ Hidden layer	Dropout	Learning rate	Optimizer
DNN	4	750	0.4	0.001	Adam

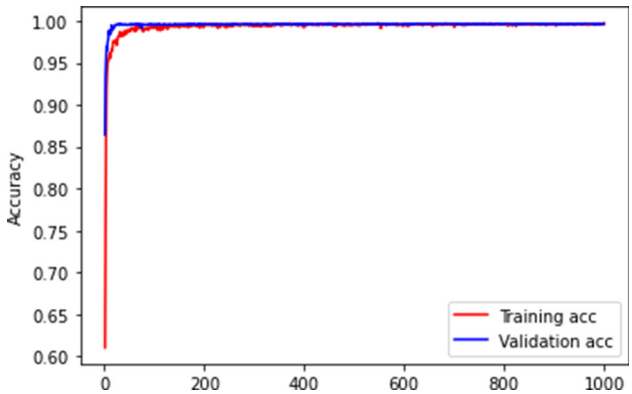


Fig. 7 DNN model training and validation accuracies versus training epochs (first case)

5.6 Training model

The deep neural network consists of 4 hidden layers, each with 750 neurons according to the above paragraph findings. The deep learning algorithm implementation was performed using *keras* library with the results provided by grid search technique implementation. Table 3 summarizes the parameter values that should be adjusted for the optimal performance of the proposed algorithm. The Adaptive Moment Estimation (Adam) optimizer has been selected to minimize the loss function and speed up the training process.

Adam optimizer is one of the most popular gradient descent optimization algorithms because it is computationally efficient and has very little memory requirements. This method calculates the individual adaptive learning rate for each parameter from estimates of first and second moments of the gradients.

Algorithm 1: Adam, our proposed algorithm for the training process.

1: Declare the parameters Objective function $f(\theta)$, hyperparameter learning rate α, exponential decay rates β_1, β_2 for moment estimates, tolerance parameter $\epsilon > 0$ for numerical stability

2: Initialize first moment vector $m_0 = 0$, second moment vector $v_0 = 0$ and timestep $t = 0$

3: **while** θ_t has not converged **do**

3.1 update timestep $t = t + 1$

3.2 compute gradient of objective using $g_t = \nabla_\theta f_t(\theta_t - 1)$

3.3 update first moment estimate and second moment estimate using eq. (8) and eq. (9), respectively.

3.4 compute unbiased first and second moment estimate using eq. (10) and eq. (11), respectively.

3.5 update objective parameters using eq. (12).

end while

4: return final parameter θ_t

Adam algorithm first updates the exponential moving averages of the gradient (m_t) and the squared gradient (v_t) which is the estimates of the first and second moment. The hyperparameters β_1, $\beta_2 \in [0, 1)$ control the exponential decay rates of these moving averages as shown in the following equations:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{8}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{9}$$

where g is the current gradient value of the error function for the neural network training.

Moving averages are initialized as 0. The moment estimates are biased around 0 especially during the initial timesteps. This initialization bias can easily be counteracted resulting in bias-corrected estimate.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{10}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{11}$$

Finally, we update the parameter as shown below

$$\theta_t = \theta_{t-1} - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{12}$$

We have used in our experiments for the Adam optimizer a learning rate $\alpha = 10^{-3}$ and two decay parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ [63].

5.7 Experimental results

The recommendation of different tourist attractions is a very complex task because it must link the input attributes with the possible recommendations generated by the DNN algorithm. The two already mentioned stages or cases related to the smart city trip were tested, that is, (a) searching and planning activities before traveling and (b) looking for activities within the smart city Barcelona.

To evaluate the effectiveness of this approach, the main indicators in the field of multi-label classification [64] were applied. Let D be a multi-label evaluation data set, consisting of $|D|$ multi-label examples $(X_i, Y_i), i = 1..|D|$. Let H be a multi-label classifier and $Z_i = H(X_i)$ be the set of labels predicted by H for example x_i.

$$\text{Accuracy}(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_{Xi} \cap Z_{Xi}|}{|Y_{Xi} \cup Z_{Xi}|} \tag{13}$$

$$\text{Precision}(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_{Xi} \cap Z_{Xi}|}{|Z_{Xi}|} \tag{14}$$

$$\text{Recall}(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_{Xi} \cap Z_{Xi}|}{|Y_{Xi}|} \tag{15}$$

$$F1 - \text{score}(H, D) = 2 \frac{\sum_i |Y_{Xi} \cap Z_{Xi}|}{2 * \sum_i |Y_{Xi} \cap Z_{Xi}| + \sum_i |Y_{Xi} - Z_{Xi}| + \sum_i |Z_{Xi} - Y_{Xi}|} \tag{16}$$

5.7.1 First case

The first case refers to searching and planning before travelling. The tourist attraction recommender system suggests activities in a generic mode according to the data previously entered by the user based on his/her profile and particular circumstances of the trip, without taking into account distances, weather or rain predictions.

For this first case, several tests have been performed to measure the accuracy of our model. Figures 7 and 8 show the results for accuracy and loss, respectively, for training and testing data; our deep neural network on the training data reaches an accuracy, precision, recall and F1-score of

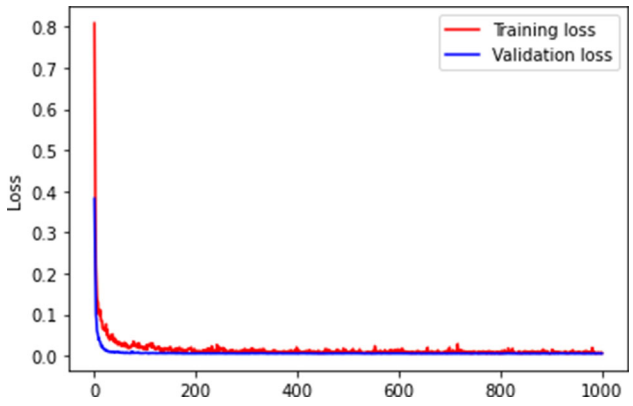


Fig. 8 DNN model training and validation losses versus training epochs (first case)

Table 4 Testing versus training accuracies, losses, F1 scores, recalls and precisions for our DNN first case

Data	Accuracy	Loss	F1 score	Recall	Precision
Training	0.996	0.004	0.998	0.999	0.999
Testing	0.997	0.005	0.998	0.999	0.999

99.6%, 99.9%, 99.9% and 99.8%, respectively, with the loss of 0.4%. For the testing data, it reaches an accuracy, precision, recall and F1-score of 99.7%, 99.9%, 99.9% and 99.8%, respectively, with the loss of 0.5%. Table 4 shows the accuracy, recall and precision values during training and testing. We conclude that these results confirm the effectiveness of our proposed tourism recommendation system for this first case.

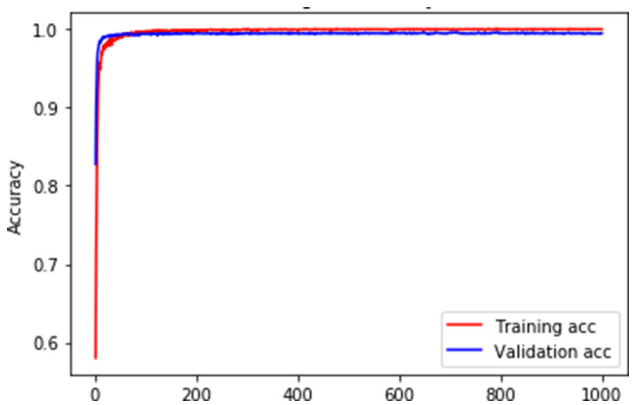


Fig. 9 DNN model training and validation accuracies versus training epochs (second case)

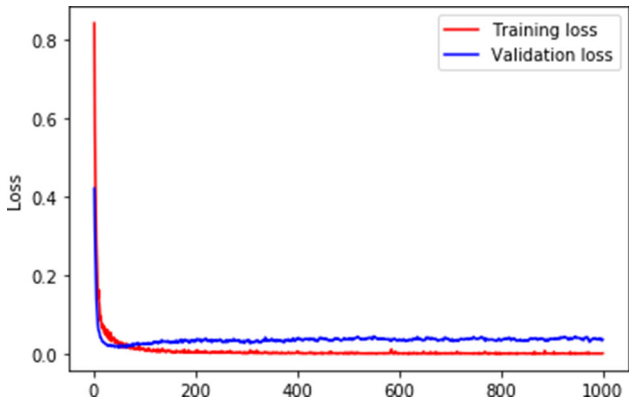


Fig. 10 DNN model training and validation losses versus training epochs (second case)

5.7.2 Second case

In addition to the information previously entered by the user (user profile and particular circumstances of the trip), in this case, the tourism recommender system also uses information collected from various IoT devices: corresponding location, temperature and weather forecasts. With all this information, the system recommends the tourist attractions that best suit the requirements of the user based on the weather forecast and the nearest locations.

For this second case, several tests were performed to measure the accuracy of our model. Figures 9 and 10 show the results for accuracy and loss, respectively, for training and testing data; after the tests were performed, our deep neural network on the training data reaches an accuracy, precision, recall and F1-score of 99.7%, 99.7%, 99.8% and 99.8%, respectively, with the loss of 0.1%. For the testing data reaches an accuracy, precision, recall and F1-score of 99.5%, 99.8%, 99.7% and 99.8%, respectively, with the loss of 3.7%. From Table 5, we conclude that these results confirm the effectiveness of our proposed tourism recommender system for our second case.

5.8 Comparison to other models

Python libraries such as scikit-learn enable the deployment of many traditional models such as Support Vector Machines (SVM), k-nearest neighbors algorithm, random

Table 5 Testing versus training accuracies, losses, F1 scores, recalls and precisions for our DNN second case

Data	Accuracy	Loss	F1-score	Recall	Precision
Training	0.997	0.001	0.998	0.998	0.997
Testing	0.995	0.037	0.998	0.997	0.998

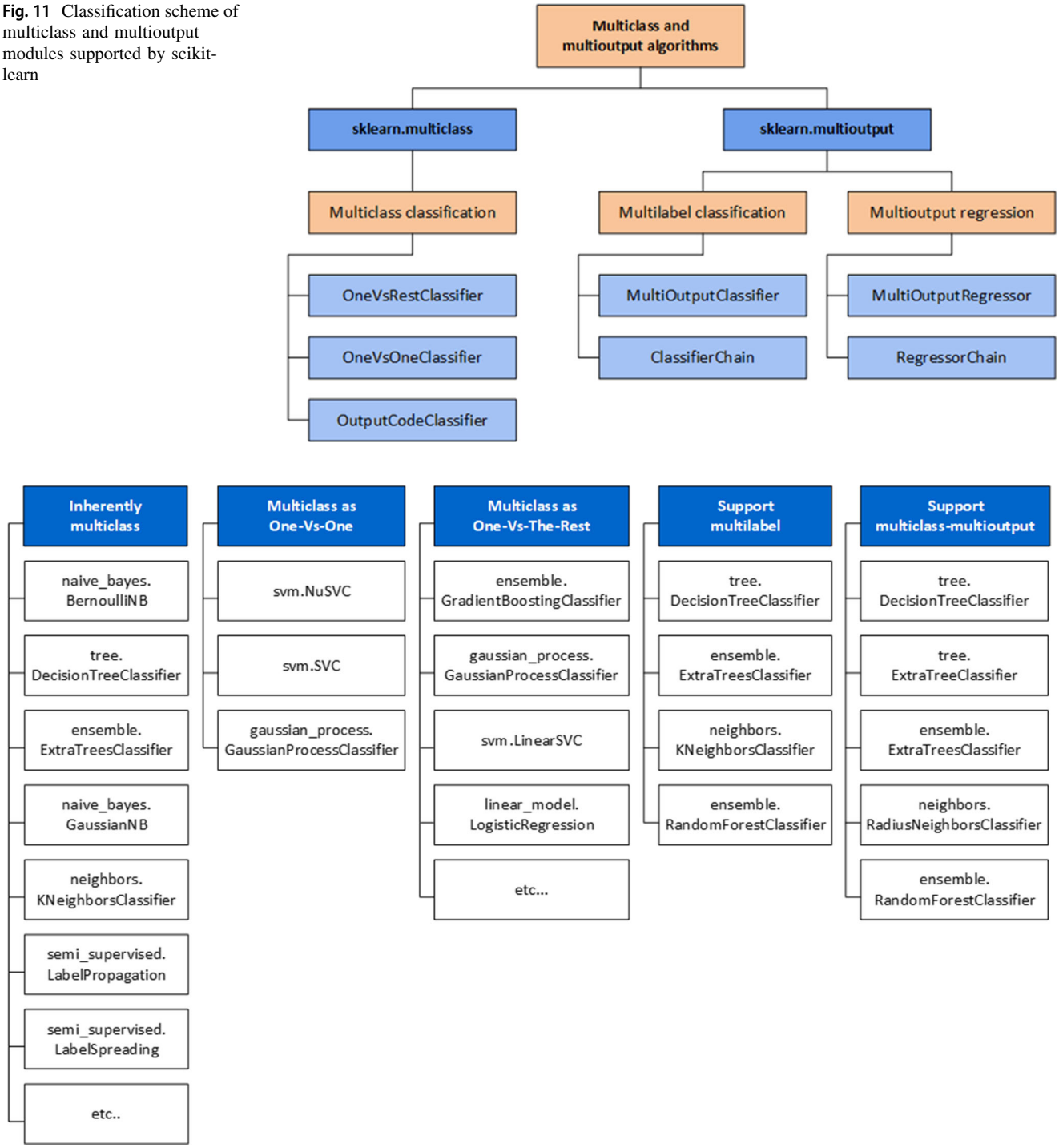


Fig. 11 Classification scheme of multiclass and multioutput modules supported by scikit-learn

Fig. 12 Overview of scikit-learn estimators that integrate multi-learning, organized by strategy

forest classifier and so on. Figures 11 and 12 show the classification algorithms and models available for implementation in different use cases [65].

For our analysis, we compared our proposal with the models provided by the scikit-learn library that support multi-label classification. These classifiers are decision tree, extra tree, k-nearest neighbor (kNN) and random forest. We have compared our proposal with all of these classifiers. Tests were also performed for the two proposed cases.

We also employ cross-validation as a statistical method for evaluation and comparison of learning algorithms. Table 6 shows the results achieved by our deep neural

Table 6 Comparison of accuracy, F1-score, recall and precision for our proposed DNN and other traditional machine learning algorithms (first case)

	Accuracy	F1-score	Recall	Precision
Our Neural Network	0.997	0.998	0.999	0.999
Decision tree	0.892	0.92	0.90	0.98
Extra tree	0.891	0.91	0.91	0.90
K-nearest neighbor	0.905	0.92	0.89	0.89
Random forest	0.384	0.52	0.40	0.62

Table 7 Comparison of accuracy, F1-score, recall and precision for our proposed DNN and other traditional machine learning algorithms (second case)

	Accuracy	F1-score	Recall	Precision
Our Neural Network	0.995	0.998	0.997	0.998
Decision tree	0.909	0.91	0.88	0.87
Extra tree	0.886	0.89	0.87	0.85
K-nearest neighbor	0.584	0.60	0.61	0.56
Random forest	0.370	0.69	0.65	0.68

network model and the traditional models implemented for the first case.

The analysis shows that our proposed DNN model achieves the highest accuracy, precision, recall and F1-score of 99.7%, 99.9%, 99.9% and 99.8%, respectively, compared to traditional machine learning models. We can also observe that from the traditional models, k-nearest neighbor achieves the highest accuracy, precision, recall and F1-score of 90.5%, 89%, 89% and 92%, respectively, followed by decision tree with accuracy, precision, recall and F1-score of 89.2%, 98%, 90% and 92%, respectively.

Table 7 shows a summary of the scores achieved by our deep neural network model and each of the traditional models implemented for the second case.

The analysis shows that our proposed DNN model in this second case achieves the highest accuracy, precision, recall and F1-score of 99.5%, 99.8%, 99.7% and 99,8%, respectively, compared to traditional machine learning models. We can also observe that from the traditional models, decision tree achieves the highest accuracy, precision, recall and F1-score of 90.9%, 87%, 88% and 91%, respectively, followed by extra tree with accuracy, precision, recall and F1-score of 88.6%, 85%, 87% and 89%, respectively.

In both cases, our algorithm has a better performance, so with these values, we can appreciate that our algorithm is able to perform future predictions adequately.

5.8.1 Discussion

The proposal has been carried out in the Barcelona city, taking into account the most emblematic places of the city. All the information collected for the realization of this proposal is based on real data (location, temperature, weather forecasts and profile of each user). The implementation of our model using the Keras library allows incorporating and adjusting more parameters (batch normalization, dropout, etc.). We have obtained better results in comparison with other traditional models provided by the scikit-learn library. In addition, due to the implementation of the grid search technique, we can get the most suitable configuration of these parameters for the optimal performance of the system. The results show that our algorithm is able to make future predictions to obtain optimal recommendations according to the information provided by the tourist profile as well as the information provided by the IoT devices.

6 Conclusions

In this paper, a tourist attraction recommendation system based on IoT, and deep learning is proposed.

This proposal investigates the impact of a deep neural network (DNN) topology for tourist attraction recommendations with multi-label classification under two use cases (a) searching and planning activities before traveling and (b) looking for activities within the smart city. The results show that the deeper the neural network, the better the performance of the algorithm is. Grid search technique has led to the selection of four hidden layers with seven hundred and fifty neurons in each layer as the DNN topology of choice with a dropout value of 0.4 to avoid overfitting during training.

In addition, a comparison of our DNN classifier is made with traditional models; obtaining the best results in two significant cases. The results are the highest for the first case with an accuracy, precision, recall and F1-score of 99.7%, 99.9%, 99.9% and 99.8%, respectively. For the second case, our DNN classifier achieves the highest accuracy, precision, recall and F1-score of 99.5%, 99.8%, 99.7% and 99.8%, respectively.

Acknowledgements This work has been supported by the Agencia Estatal de Investigación of Spain under project PID2019-108713RB-C51/AEI/10.13039/501100011033.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. ITU International Telecommunication Union Internet Reports (2006) The Internet of Things - Executive Summary. http://www.itu.int/publ/S-POL-IR.IT-2005/eS. Accessed 7 Jun 2021
2. Zanella A, Bui N, Castellani A et al (2014) Internet of things for smart cities. IEEE Internet Things J 1:22–32. https://doi.org/10.1109/JIOT.2014.2306328
3. IESE Business School (2020) IESE cities in motion index
4. Guo Y, Liu H, Chai Y (2014) The embedding convergence of smart cities and tourism internet of things in China: an advance perspective. Adv Hosp Tour Res 2:54–69
5. Buhalis D, Amaranggana A (2013) Smart Tourism Destinations. In: Xiang Z, Tussyadiah I (eds) Information and communication technologies in tourism 2014. Springer International Publishing, Cham, pp 553–564
6. Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowl Based Syst 46:109–132. https://doi.org/10.1016/j.knosys.2013.03.012
7. Isinkaye FO, Folajimi YO, Ojokoh BA (2015) Recommendation systems: principles, methods and evaluation. Egypt Inform J 16:261–273. https://doi.org/10.1016/j.eij.2015.06.005
8. Unger M, Tuzhilin A, Livne A (2020) Context-aware recommendations based on deep learning frameworks. ACM Trans Manag Inf Syst. https://doi.org/10.1145/3386243
9. Rohani VA, Kasirun ZM, Kumar S, Shamshirband S (2014) An effective recommender algorithm for cold-start problem in academic social networks. Math Probl Eng. https://doi.org/10.1155/2014/123726
10. Abiodun OI, Jantan A, Omolara AE et al (2018) State-of-the-art in artificial neural network applications: a survey. Heliyon 4:e00938. https://doi.org/10.1016/j.heliyon.2018.e00938
11. Ciregan D, Meier U, Schmidhuber J (2012) Multi-column deep neural networks for image classification. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. https://doi.org/10.1109/CVPR.2012.6248110
12. Sivaramakrishnan N, Subramaniyaswamy V, Viloria A et al (2020) A deep learning-based hybrid model for recommendation generation and ranking. Neural Comput Appl. https://doi.org/10.1007/s00521-020-04844-4
13. Shambour Q (2021) A deep learning based algorithm for multi-criteria recommender systems. Knowl Based Syst 211:106545. https://doi.org/10.1016/j.knosys.2020.106545
14. Hamid RA, Albahri AS, Alwan JK et al (2021) How smart is e-tourism? A systematic review of smart tourism recommendation system applying data management. Comput Sci Rev 39:100337. https://doi.org/10.1016/j.cosrev.2020.100337
15. den Beemt WP, Smith R (2013) Smart tourism tools: linking technology to the touristic resources of a city. pp 1–12
16. González EMA, Municio E, Alemán MN, Marquez-Barja JM (2020) Cultural heritage and Internet of Things. ACM Int Conf Proceeding Ser. https://doi.org/10.1145/3411170.3411267
17. Wang YP, Dai X, Jung JJ, Choi C (2018) Performance analysis of smart cultural heritage protection oriented wireless networks. Futur Gener Comput Syst 81:593–600. https://doi.org/10.1016/j.future.2017.04.007
18. Perles A, Pérez-Marín E, Mercado R et al (2018) An energy-efficient internet of things (IoT) architecture for preventive conservation of cultural heritage. Futur Gener Comput Syst 81:566–581. https://doi.org/10.1016/j.future.2017.06.030
19. Spasova VG, Georgiev BG, Stefanov PD, Stoyanov BP (2021) Prototype of smart monument with IoT-based system of early warning. IOP Conf Ser Mater Sci Eng 1031:1–9. https://doi.org/10.1088/1757-899X/1031/1/012126
20. Piccialli F, Chianese A (2017) A location-based IoT platform supporting the cultural heritage domain. Concurr Comput Pract Exp. https://doi.org/10.1002/cpe.4091
21. Alletto S, Cucchiara R, Del FG et al (2016) An indoor location-aware system for an IoT-based smart museum. IEEEE Internet Things 3:244–253
22. Piccialli F, Giampaolo F, Casolla G et al (2020) A deep learning approach for path prediction in a location-based IoT system. Pervasive Mob Comput 66:101210. https://doi.org/10.1016/j.pmcj.2020.101210
23. Ricci F, Shapira B, Rokach L (2015) Recommender systems handbook, 2nd edn. Springer
24. Aggarwal CC (2016) An introduction to recommender systems. Recomm Syst. https://doi.org/10.1007/978-3-319-29659-3_1
25. Karabadji NEI, Beldjoudi S, Seridi H et al (2018) Improving memory-based user collaborative filtering with evolutionary multi-objective optimization. Expert Syst Appl 98:153–165. https://doi.org/10.1016/j.eswa.2018.01.015
26. Ranjbar Kermany N, Alizadeh SH (2017) A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques. Electron Commer Res Appl 21:50–64. https://doi.org/10.1016/j.elerap.2016.12.005
27. Sertkan M, Neidhardt J, Werthner H (2020) From pictures to travel characteristics: deep learning-based profiling of tourists and tourism destinations. In: Neidhardt J, Wörndl W (eds) Information and Communication Technologies in Tourism 2020. Springer International Publishing, Cham, pp 142–153
28. Kurashima T, Iwata T, Irie G, Fujimura K (2010) Travel route recommendation using geotags in photo sharing sites. Int Conf Inf Knowl Manag Proc. https://doi.org/10.1145/1871437.1871513
29. Xu Z, Chen L, Chen G (2015) Topic based context-aware travel recommendation method exploiting geotagged photos. Neurocomputing 155:99–107. https://doi.org/10.1016/j.neucom.2014.12.043
30. Brilhante IR, Macedo JA, Nardini FM et al (2015) On planning sightseeing tours with TripBuilder. Inf Process Manag 51:1–15. https://doi.org/10.1016/j.ipm.2014.10.003
31. Cai G, Lee K, Lee I (2018) Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos. Expert Syst Appl 94:32–40. https://doi.org/10.1016/j.eswa.2017.10.049
32. Lim KH (2015) Recommending tours and places-of-interest based on user interests from geo-tagged photos. In: Proc ACM

SIGMOD Int Conf Manag Data 31-May-201:33–38. https://doi.org/10.1145/2744680.2744693
33. Lim KH, Chan J, Leckie C, Karunasekera S (2018) Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. Knowl Inf Syst 54:375–406. https://doi.org/10.1007/s10115-017-1056-y
34. Liu Y, Pham TAN, Cong G, Yuan Q (2017) An experimental evaluation of pointofinterest recommendation in location-based social networks. Proc VLDB Endow 10:1010–1021
35. Kesorn K, Juraphanthong W, Salaiwarakul A (2017) Personalized attraction recommendation system for tourists through check-in data. IEEE Access 5:26703–26721. https://doi.org/10.1109/ACCESS.2017.2778293
36. Ben Sassi I, Mellouli S, Ben Yahia S (2017) Context-aware recommender systems in mobile environment: on the road of future research. Inf Syst 72:27–61. https://doi.org/10.1016/j.is.2017.09.001
37. Shen J, Deng C, Gao X (2016) Attraction recommendation: Towards personalized tourism via collective intelligence. Neurocomputing 173:789–798. https://doi.org/10.1016/j.neucom.2015.08.030
38. Da'u A, Salim N, (2020) Recommendation system based on deep learning methods: a systematic review and new directions. Springer, Netherlands
39. Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv. https://doi.org/10.1145/3285029
40. Bartolini I, Moscato V, Pensa RG et al (2016) Recommending multimedia visiting paths in cultural heritage applications. Multimed Tools Appl 75:3813–3842. https://doi.org/10.1007/s11042-014-2062-7
41. Hong M, Jung JJ, Piccialli F, Chianese A (2017) Social recommendation service for cultural heritage. Pers Ubiquitous Comput 21:191–201. https://doi.org/10.1007/s00779-016-0985-x
42. Chianese A, Marulli F, Piccialli F et al (2017) An associative engines based approach supporting collaborative analytics in the Internet of cultural things. Futur Gener Comput Syst 66:187–198. https://doi.org/10.1016/j.future.2016.04.015
43. Cuomo S, De MP, Piccialli F et al (2017) IoT-based collaborative reputation system for associating visitors and artworks in a cultural scenario. Expert Syst Appl 79:101–111. https://doi.org/10.1016/j.eswa.2017.02.034
44. Al Fararni K, Nafis F, Aghoutane B et al (2021) Hybrid recommender system for tourism based on big data and AI: a conceptual framework. Big Data Min Anal 4:47–55. https://doi.org/10.26599/BDMA.2020.9020015
45. Indriana M, Hwang C-S (2014) Applying Neural Network Model to Hybrid Tourist Attraction Recommendations. J Ultim 6:63–69. https://doi.org/10.31937/ti.v6i2.339
46. Nakahara T, Yada K (2012) Analyzing consumers' shopping behavior using RFID data and pattern mining. Adv Data Anal Classif 6:355–365. https://doi.org/10.1007/s11634-012-0117-z
47. Atlam HF, Walters RJ, Wills GB (2018) Fog computing and the internet of things: a review. Big Data Cogn Comput 2:1–18. https://doi.org/10.3390/bdcc2020010
48. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of Things (IoT): a vision, architectural elements, and future directions. Futur Gener Comput Syst 29:1645–1660. https://doi.org/10.1016/j.future.2013.01.010
49. Chen M, Xu Z, Weinberger KQ, Sha F (2012) Marginalized denoising autoencoders for domain adaptation. In: Proc 29th int conf mach learn ICML 2012 1:767–774
50. Woźniak M, Siłka J, Wieczorek M (2021) Deep neural network correlation learning mechanism for CT brain tumor detection. Neural Comput Appl. https://doi.org/10.1007/s00521-021-05841-x
51. Wozniak M, Wieczorek M, Silka J, Polap D (2021) Body pose prediction based on motion sensor data and recurrent neural network. IEEE Trans Ind Inform 17:2101–2111. https://doi.org/10.1109/TII.2020.3015934
52. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press. http://www.deeplearningbook.org
53. Goodfellow IJ, Pouget-Abadie J, Mirza M et al (2014) Generative adversarial networks
54. Li Y (2018) Deep Reinforcement Learning
55. Statista (2020) Leading city destinations worldwide in 2018, by number of overnight visitors. https://www.statista.com/statistics/310355/overnight-visitors-to-top-city-destinations-worldwide/. Accessed 7 Jun 2021
56. Statista (2017) Number of international overnight visitors in the most popular European city destinations in 2016. https://es.statista.com/estadisticas/487720/turistas-internacionales-en-los-principales-destinos-europeos/. Accessed 7 Jun 2021
57. Eden Strategy Institute and ONG&ONG (2018) Smart city governments 1
58. Barcelona municipality (2020) Tourists and overnight stays. https://ajuntament.barcelona.cat/estadistica/castella/index.htm. Accessed 7 Jun 2021
59. Barcelona municipality (2020) Other tourist information. https://www.bcn.cat/estadistica/angles/dades/anuari/cap13/C1306010.htm. Accessed 7 Jun 2021
60. Jackson AH (1988) Machine learning: a probabilistic perspective
61. Srivastava N, Geoffrey H, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfittin. J Mach Learn Res. https://doi.org/10.1016/0370-2693(93)90272-J
62. Liashchynskyi P, Liashchynskyi P (2019) Grid search, random search, genetic algorithm: a big comparison for NAS. 1–11
63. Kingma DP, Ba JL (2015) Adam: a method for stochastic optimization. In: 3rd Int Conf Learn Represent ICLR 2015-Conf Track Proc 1–15
64. Tsoumakas G, Katakis I, Vlahavas I (2006) A review of multi-label classification methods. In: Proc 2nd ADBIS Work Data Min Knowl Discov (ADMKD 2006) 99–109
65. Scikit-learn, Python ML in (2019) Multiclass and multioutput algorithms. https://scikit-learn.org/stable/modules/multiclass.html#multioutputclassifier. Accessed 7 Jun 2021