UNIVERSITÀ
DEGLI STUDI
FIRENZE

Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione
Curriculum: Informatica

Human Centered Big Data Analytics for Smart Applications

Candidate
Luciano Alessandro Ipsaro Palesi

Supervisors
Prof. Paolo Nesi

Prof. Pierfrancesco Bellini

PhD Coordinator
Prof. Fabio Schoen

ciclo XXXIV, 2018-2021

Università degli Studi di Firenze, Dipartimento di Ingegneria
dell'Informazione (DINFO).

To Capri Leone and Mirto, the villages where I grew up

Acknowledgments

I would like to acknowledge the support, effort and input of my supervisor, Prof. Paolo Nesi. In particular, my thanks go to all my colleagues of the DISIT Lab who were of great help during my research.
I would always be grateful to my father, my mother, my brother and my grandparents. Thanks also to Gioele, Leonardo, Milena, Irene, Marta and Gaetano for the affection they show me.
Finally, I would like to thank my friends in Florence and in Sicily for supporting me through these PhD years.

Abstract

This thesis is concerned with Smart Applications.
Smart applications are all those applications that incorporate data-driven, actionable insights in the user experience, and they allow in different contexts users to complete actions or make decisions efficiently. The differences between smart applications and traditional applications are mainly that the former are dynamic and evolve on the basis of intuition, user feedback or new data. Moreover, smart applications are data-driven and linked to the context of use. There are several aspects to be considered in the development of intelligent applications, such as machine learning algorithms for producing insights, privacy, data security and ethics.
The purpose of this thesis is to study and develop human centered algorithms and systems in different contexts (retail, industry, environment and smart city) with particular attention to big data analysis and prediction techniques.
The second purpose of this thesis is to study and develop techniques for the interpretation of results in order to make artificial intelligence algorithms "explainable".
Finally, the third and last purpose is to develop solutions in GDPR compliant environments and then secure systems that respect user privacy.

Contents

Chapter 1

Introduction

This thesis describes the PhD activity carried out at the Distributed Data Intelligence and Technology (DISIT) Lab of the Department of Information Engineering (DINFO) at the University of Florence. The work involves area of studies in Computer Science such as Computer Application (Artificial intelligence) and it is composed of five parts, each dealing with aspects of the Smart Applications such as data acquisition, data analytics, prediction and Explainable AI.

Smart applications are those types of applications that dynamically use data to create actionable insights into the user experience to complete actions or make decisions efficiently.

Smart applications, therefore, differ from traditional applications since they are data-driven and use machine learning algorithms to obtain actionable insights. At the architectural level, they are based on microservices as they must adapt and evolve continuously based on new data, insights or user feedback. Finally, compared to traditional applications they must provide a personalized experience.

For these applications it is important to perform Human-centered design, that is an approach to system design and development that aims to make systems interactive and more usable [3]. This type of approach increases efficiency, improves user satisfaction, accessibility and sustainability.

Thus, there are several aspects to consider when developing Human-Centered Smart Applications. In addition to machine learning algorithms for producing insights, other aspects to consider are privacy and thus data security. Another important aspect is ethics and thus the explainability of

the results of a smart application. In this thesis, we will analyze these aspects in different contexts.

Solutions presented in this thesis have been implemented on Snap4City infrastructure. Snap4City (Smart aNalytic APp builder for sentient Cities and IoT) is an open source IOT/IOE platform implemented in response of a research challenge launched by Select4Cities PCP (Pre Commercial Procurement) H2020 research and development project of the European Commission (`https://www.select4cities.eu`). The main tools composing the Snap4City solution are data ingestion and aggregation tools, data management tools, data processing tools, APIs system, data indexing system, development tools, and tools for final users. Snap4City provides flexible solutions to quickly create smart and sentient applications to improve city services, security and safety, attracting industries and stakeholders, and offering the city or industries a sustainable solution.

The research activity presented in this thesis focused on the back office parts and therefore on technical aspects of data analytics that did not involve the usability of the smart applications.

The chapter 2 of this thesis describes an original solution for a recommendation system in the fashion retail domain, based on a multi clustering approach of items and users' profiles in online or physical stores. The proposed solution relies on mining techniques, allowing to predict the purchase behaviour of newly acquired customers, thus solving the cold start problems which is typical of the systems at the state of the art. The presented work has been developed in the context of the Feedback project partially funded by Regione Toscana, and it has been conducted on real retail company Tessilform, Patrizia Pepe mark. The recommendation system has been validated in store, as well as online.

In chapter 3, a framework for predictive maintenance is presented. It has been built upon a deep learning model based on Long-Short Term Memory Neural Networks, LSTM and Convolutional LSTM. The proposed model provides a one-hour prediction of the plant status and indications on the areas in which the intervention should be performed by using explainable LSTM technique. The solution has been validated against real data of ALTAIR chemical plant, demonstrating an average accuracy of 91.8% and an average F1-score of 90%, with the capability of being executed in real-time in a production operative scenario. The chapter also introduces business intelligence tools on maintenance data, and the architectural infrastructure

for the integration of predictive maintenance approach into the whole control and management systems of ALTAIR industry 4.0 plant.

In chapter 4, the state of the art on landslide prediction for early warning has been carefully reviewed. To find a better solution, a number of machine learning solutions have been implemented and tested (e.g., random forest (RF), extreme gradient boosting (XGBoost), convolutional neural networks (CNN) and autoencoders (AE)). These models have been trained, validated and compared each other and with the SIGMA approach from the literature. The validation has been performed in the context of the Metropolitan City of Florence from 2013 to 2019. The method based on XGBoost achieved better results, demonstrating that it is most reliable and robust to false alarms. Finally, we applied explainable artificial intelligence techniques to the XGBoost model (locally and globally), to provide a deeper understanding of the predictive model outputs and the feature relevance.

In chapter 5, a what-if analysis solution has been described and validated with the major focus on traffic flow which has a strong impact since most of the simulations in the context of the cities are based traffic flow, including: parking, pollutant, people flow, accidents, commercial sites, tourism, etc. The contributions of the chapter are: (i) the definition and formalization of a what-if analysis framework, including the formalization of what-if scenarios with multiple connected areas, (ii) the definition and implementation of large traffic flow reconstruction and simulation against what-if scenarios, (iii) the validation of the traffic flow reconstruction against complex scenarios, (iv) the high performance obtained in the what-if analysis providing traffic flow predictions on large changes on city road traffic. Point (ii) extended the solution for traffic flow reconstruction at the state of the art by (a) dynamically reshaping the road graph network on the basis of the scenarios with multiply connected critical areas, (b) computing multiple reconstructions in consecutive time slots taking into account the evolution of road graph, junction redistribution and of traffic flow data, (c) computing and comparing traffic flow KPI at the support of the what-if analysis, considering the complexity of their comparison since sensors and roads may be involved into the blocked areas as well.

In Chapter 6, an innovative data acquisition solution through the Snap4city platform is proposed. The domain of the application is sleep quality. Metrics have been studied and calculated to establish its quality in accordance with medical lines. It is a multi-user and GDPR compliant solution. Finally, a

data visualization solution is illustrated. The research activity in this chapter has laid the foundations for future experimentation in the medical field by solving complex privacy and GDPR problems using the Snap4City tool.

Finally in 7 a summary of this thesis and the conclusions.

Chapter 2

Multi Clustering Recommendation System for Fashion Retail

Fashion retail has a large and ever-increasing popularity and relevance, allowing customers to buy anytime by finding the best offers and providing satisfactory experiences in the shops. Consequently, Customer Relationship Management solutions have been enhanced by means of several technologies to better understand the behaviour and requirements of customers, engaging and influencing them to improve their shopping experience, as well as increasing the retailers' profitability. Current solutions on marketing provide a too general approach, pushing and suggesting on most cases, the popular or most purchased items, losing the focus on the customer centricity and personality. In this chapter, a recommendation system for fashion retail shops is described, wich is based on a multi clustering approach of items and users' profiles in online and/or physical stores. This solution relies on mining techniques, allowing to predict the purchase behaviour of newly acquired customers, thus solving the cold start problem which is typical of the systems at the state of the art. The presented work has been developed in the context of the Feedback project partially funded by Regione Toscana, and it has been conducted on real retail company Tessilform, Patrizia Pepe mark. The recommen-

dation system has been validated in store, as well as online [1][2].

2.1 Introduction

The competitiveness of retailers strongly depends on the conquered reputation, brand relevance and on the marketing activities they carry out. The latter aspect is exploited to increase sales and thus a retailer, through marketing, should be capable to stimulate customers to buy more items or more valuable items. Today, consumers tend to buy more on ecommerce and the COVID-19 situation also stressed this condition. Online shopping offers the possibility to buy at any time of the day; customers buy where they find the best offer, online as well as offline, and they are also influenced by an increasing amount of information from blogs, communities, and social networks. To retain a customer is, therefore, an extremely difficult achievement, and in some measure, it can get easily out of control.

Currently, ICT (Information and Communication Technologies) offers Customer Relationship Management (CRM) solutions that are capable to construct and manage user data profiles, from customer information to product details, to sales transactions. CRM systems comprise a set of processes to support business strategies to build long term profitable relationships with customers [124]. Customer data and information technology (IT) tools form the foundation on which successful CRM strategies are built. Swift in [146] defined CRM as an enterprise approach to understand and influence customers' behaviour through meaningful communications in order to improve customer acquisition, retention, loyalty, and profitability. However, CRM solutions on the market use approaches suggesting the most popular items, bundled offers, similar items or featured items and therefore they often neglect the relevance of customer personal preferences in their marketing strategies. In addition, there are IoT Devices offered by big vendors, promising an evolved engagement at various levels [76], interacting with fewer queues, promotions, more involvement, assistance, although they are hardly triggered within companies, especially on retail, which needs more flexible solutions.

[1] Part of the work presented in this chapter has been published as "Multi Clustering Recommendation System for Fashion Retail" in *Multimedia Tools and Applications* [37] 2022.

[2] *Acknowledgments:* Our thanks goes to FEEDBACK project and partners for which we have developed a part of the solutions described in this chapter, and Regione Toscana for the partial founding POR FESR 2020 Phase 2 `https://www.vargroup.it/progetti-rd/`

Therefore, market solutions are unable to build actual profiles by exploiting users' historical, social, and behavioural activities. Through transactions, retailers can generate knowledge about their consumer's behaviour. In this context, one of the techniques receiving more attention from researchers to generate consumer knowledge, is machine learning, specifically clustering techniques. Clustering techniques are used to group customers by similarity. So that, retailers can tailor marketing actions more effectively with respect to the above-mentioned generic marketing actions. Understanding the reasons why consumers choose a specific item within the store is of extreme relevance for the retailer. In addition, knowing the consumer's needs through the factors that influence shopper's decision-making process is important for the business of every single store. This is what recommendation systems are all about. Recommendation systems are applications that assist users in finding items (products, services and information) that should match their preferences/needs [110]. The generated recommendations are considered (i) personalized, in the sense that they have been generated for a user or a group of users, or, in the opposite to (ii) non-personalized recommendations (e.g., best-selling items, or selection of items), which are typically not addressed by research.

Recommendation systems at the state of the art do not solve typical retail problems. Most of the retail companies today have both online and physical store customers who are assisted in purchasing by shop assistants. With the GDPR (General Data Protection Regulation) rules [24], often the customer demographics are differently collected in different areas and shops, where different regulations are adopted. Deep learning methods, which are typically employed to improve accuracy, are hard to be adopted for the scarcity of data. For example, in fashion retail shops most of the transactions are anonymous and related to a single item; moreover, periodic acquisitions are performed every 8-12 months. This behaviour is mainly due to the high costs of the items and to seasonality aspects of most of the products. Regarding classification methods, the multichannel nature of retailers allows to provide data with different features and with many incomplete records, which are difficult to be exploited on most of the classic methods for recommendations. As for clustering methods, we registered the usage of RFM (Recency - Frequency - Monetary Value) [51], and LTV (Life-Time-Value) [97], where demographic values are taken as input without taking advantage of the typical intuition of deep learning about customer behaviour with re-

spect to items. Another problem related to the fashion retail industry is related to the seasonality of most of the items. Their commercial life ranges from 6 months to 1 year.

In this chapter, a recommendation solution in the context of fashion retail is proposed. The aim has been to solve the above-mentioned problems of cold start, computational complexity, low number of returns in the shops of fashion retails and long period for returning, the needs of more mediated interactions in the shops and more direct interactions online, and the effects of the seasonality of products. To this end, we realized a multi clustering approach by taking as input the RFM value of online and physical stores separately. To solve the problem of the products' seasonality the items have been clustered taking into account multiple seasons. In addition, input data have been enriched with the customer behaviour towards the items. In order to solve the cold start problem of cluster-based recommendation systems, the association rules mining technique has been used to predict the purchase behaviour of newly acquired customers. The work presented in this chapter has been developed in the context of Feedback research and development project co-funded by Regione Toscana, Italy, and by partners. Partners of the project have been VAR Group, University of Florence (DISIT lab, DINFO dept.), TESSIFORM (Patrizia Pepe trademark), SICETELECOM, 3F CONSULTING and CONAD (External partner). The studies illustrated in this chapter have been conducted on retail company Tessilform: which is a fashion retailer owning online sales and many different stores in the world, mainly in Italy, the owner of Patrizia Pepe trademark.

The chapter is structured as follows. In Section 2.2, related work on recommendation systems is presented. The section also includes a comparative table. Section 2.3 describes the system architecture adopted in Feedback solution. In Section 2.4, the proposed recommender systems based on multi clustering is presented. The solution allowed to prepare the recommendations in advance and consume them in real time when the conditions occur, or for stimulating the customer to return in the shop via email and when they access on Web. In Section 2.5, the assessment and validation are reported. Final considerations are drawn in Section 2.6.

2.2 Related work

In this section, in the first part, the types of recommendation systems in the retail context are presented. Next, we analyze the Machine Learning techniques used in this context.

Recommendation techniques can be classified into six categories according to the sources of knowledge they use: content, collaborative filters, demographic, knowledge, community, and hybrid [137], [136].

The **content-based** approaches recommend items by computing similarities among items and users through a set of features associated to them [32], [34]. For example, for a clothing item, the considered features can be the group (shirt, sweater, T-shirt, etc.), colour, popularity, etc.; while for the users: demographic aspects, surveys answers, etc. In [156] a content-based recommender system has been presented. It suggests the most suitable items after the creation and first login of a new user, taking into account the similarity with other users and the popularity of the items. This solution showed also how to solve cold start problems for new users.

The **collaborative filtering-based** approach is based on the historical data of the user's interactions with the items, either explicit (e.g., user's ratings) or implicit feedback (e.g., purchase, visit, tests). The mathematical techniques used are the neighbourhood method and the latent factor model [99]. The neighbourhood method identifies relationships among elements or, alternatively, among users. The latent factor model sets a number of evaluation methods to characterize both items and users and it is mainly based on the matrix factorization (for example the ratings-matrix). These kinds of approaches do not need a representation of the items, as they are based only on ratings, so they are the best recommendation systems in terms of scalability since they act on rules or patterns instead of the entire dataset. The accuracy of recommendations increases as the user interactions increase. They have cold start problems for both new items and new users.

The **Demographic-based** approaches generate recommendations on the basis of the user's demographic profile (age, gender, education, etc.). They do not require a user ratings history, and they have cold start problems for new items. In [70], demographic information has been used to predict the number of products sold in a store and as a recommendation system. The experimental predictive accuracy was 1.5-5 times greater for the items of interest, as measured by r-squared error statistics.

The **knowledge-based approaches** are based on the knowledge of item

features that meets the users' needs. They do not have cold start problems; however, they require a broad knowledge of the domain and, in case of many items, they are very difficult to implement. In [85] a knowledge-based recommendation system has been implemented starting from the logs (purchase times and choices) of an ecommerce. The obtained results confirm an optimization of purchase times of purchase for customers.

The **community-based** systems make recommendations through the preferences of users' friends in contexts of social networks or communities. The basic concept is that a user tends to rely on recommendations from their friends instead of those of similar but unknown users. This approach is very useful for cold-start recommendations. In [65], a method is proposed to solve cold start problems in a recommendation system to suggest movies, which exploits the implicit relationships among the items derived from the direct interactions of the users with them.

The **hybrid-based** recommender systems combine two or more of the above listed recommendation approaches in different ways. Usually, considering two different approaches, the advantages of the former are used to mitigate the weakness of the latter. In [58], it was shown how a hybrid approach (demographic and collaborative filtering) improves the accuracy of item evaluation predictions compared to individual approaches.

The sources of knowledge are usually represented by three types of descriptors for items, users and transactions (relations between user and item). Modern recommendation systems also use textual reviews [56], images [151], web page sequences [163], user emotion (Facial Expression Recognition [113] or even text reviews [55]), images and web page sequences, and processed through data mining or deep learning methods, to generate recommendations.

In Table 2.1, a comparative overview of the reviewed methods for recommendation systems in literature is reported.

Paper	Features	Data	Domain	Technology	Results
A Two Phase Clustering Method for Intelligent Customer Segmentation [123]	Demographic variables, RFM and LTV	Iranian Bank, 38254 ratings, 491 customers with 25 attributes	Bank industry	Clustering K-means. Neural networks classification	The accuracy has not been quantitatively evaluated, but based on the results, the solution allows to make better decisions to create suggestions and to create marketing strategies
Product Recommendation based on Shared Customer's Behavior [139]	RFM and LTV Shopping Basket	Chain of perfumeries (2012-2013); 3245 customers, 11000 products	Retail	Clustering - Association rule mining	Increases 96% the average value of the sales when compared with based recommendation
Recommender Systems Using Support Vector Machines [119]	Ratings	EachMovie 1000 users with more than 100 movie ratings	Simulated	Support Vector Machines (SVM) - Genetic Algorithm	Using McNemar's test, Support Vector Machines - Genetic Algorithm model shows better performance than the SVM and Traditional models

Collaborative Filtering Recommendation Algorithm Based on Item Clustering and Global Similarity [162]	Items, Users, Ratings	MovieLens 100K ratings of 943 users on 1038 movies.	Simulated	Clustering K-means	Can improve the accuracy of the prediction and enhance the recommendation quality
Collaborative filtering with the simple bayesian classifier [120]	Ratings	EachMovie (movie ratings): ratings of 2000 users on 1410 movies and JesterData: ratings of 3000 users per 100 jokes.	Simulated	Bayesian Classifiers	F-measure 70.02%.
Learning and Revising User Profiles: The Identification of Interesting Web Sites. [129]	words in web pages	40 tests	Different domains	Decision trees classifier	Benefit in selecting features that are relevant to the classification task
Fast algorithms for mining association rules in large databases. [15]	transactions	100K transactions	Different domains	Association rule mining	Good computational performance

Image-Based Fashion Product Recommendation with Deep Learning [151]	Descriptive metadata or user reviews, visual information	Fashion dataset. dataset contains 11851 images, and the texture attributes dataset contains 7342 images.	Simulated retail	kNN - CNN - AlexNet [100] and batch-normalized Inception	not available
Personal recommendation using deep recurrent neural networks in NetEase [163]	Web page sequence, purchase history	e-commerce website www.kaola.com	retail	RNN	It extracts the common purchase patterns and shorten the purchase path for future users, reaching a compression ratio of 0.724127 and an accuracy of 0.331312
Sentiment-Aware Deep Recommender System With Neural Attention Networks [56]	Item textual reviews	Amazon Product Reviews (#Users: 784926, #Items: 141786, #Reviews: 1064767), Yelp 2017 Datasets Challenge #Users: 169257, #Items: 63300, #Reviews: 1659678)	Simulated retail	Attention Network	This approach in terms of MSE outperforms other algorithms with a value of 0.850 compared to values in the range from 0.901 to 1.884 of the other algorithms.

RecSys-DAN: Discriminative Adversarial Networks for Cross-Domain Recommender Systems. [158]	review	Amazon dataset in five categories	Simulated retail	AN	Through the calculation of MAE and RMSE it has been demonstrated that the proposed technique performs better than other algorithms (such as KNN or SVD)
Deep Reinforcement Learning for List-wise Recommendations [169]	Recommendations	ecommerce database of 100000 recommendation sessions (1,156,675 item)	Simulated retail	Markov Decision Process + DRL	Through the calculation of Normalized Discounted Cumulative Gain metrics, demonstrating that it outperforms long-term recommendations.
Personalized Recommendation for Online Retail Applications Based on Ontology Evolution [16]	User profile	No validation	Different domains	ontology-based	The ontology evolution technique in recommendation detects the behaviour of the user and give accurate and updated recommendations to each user

Enhancing the Role of Large-Scale Recommendation Systems in the IoT Context [93]	ratings	Simulated in different dataset	Different domains	k-means, FCM, SLINK, SOM	The results show that clustering improves the recommendation accuracy.
Online discrete choice models: Applications in personalized recommendations, Decision Support Systems [57]	User profile	Swissmetro data set	Transportation	Discrete choice models	Computationally efficient and empirically accurate

Table 2.1: Comparative overview of the main related works on recommendation systems for retail.

The data mining methods for recommender systems can be summarized in three types of algorithms, as follows.

Classification. For example, the kNN classifier finds the closest k points (closest neighbours) from the training records. In [39], kNN has been implemented to suggest short-term news to users. with very good results in terms of precision. Decision Trees classifier works well when objects have a limited number of features. In [129] and [54], it has been shown that this technique can have low performances, since small changes involve recalculating all distances between items or customers. In [53], the classification approach has been used for the identification of target customers minimizing the recommendation errors, by selecting users to whom the recommendations should be addressed, according to which categories of purchases they have made in a selected period of time. In [39], a Naive Bayes classifier has been used to predict the user's long-term preferences in the news domain, with excellent results in accuracy. Support Vector Machines (SVM) classifier is used to find a linear hyperplane (decision boundary) that separates input data in such a way that the distance among data groups is maximized [119].

Cluster Analysis has been used for segmenting a heterogeneous population into a number of subgroups [108], [31]. Through the Clustering Analysis, it is possible to explore the data set and to organize the data for creating recommendations. For example, variables used in the clusters may be: demographic [70], RFM [51], LTV [97], demographic + RFM [117], demographic + LTV [87], LTV+RFM [50]. The commonly used clustering algorithms are: K-means (each cluster is represented by the geometric centre of the data points belonging the cluster, supposing the feature on some numerical space); K-Medoids (each cluster is represented by the most representative element of the cluster); Clara (it is an extension to Partitioning Around Medoids, PAM, adapted to large data sets); Self-Organizing Map (SOM, it is based on artificial neurons clustering technique) [123], [162]. About the Internet of Things (IoT) context in [93] k-means, fuzzy c-means (FCM), Single-Linkage (SLINK), and Self-Organizing-Maps (SOM) techniques are used to manage sparsity, scalability, and diversity of data in different domains. The results show that clustering improves the recommendation accuracy.

Association Rules aim at finding rules in the dataset that satisfy some minimum support and minimum confidence constraints. An association rule is an expression $X \implies Y$, where X and Y are item sets (e.g.,

$Milk, Cookies \implies Sugar$). Given a set of transactions T, and denoting MinSup and MinConf the minimum support and the minimum confidence constraint values, the goal of association rule mining is to find all rules having support greater than or equal to MinSup, and confidence greater than or equal to MinConf. The most common algorithms used for implementing association rule mining are apriori [14], FP-Growth (Frequent Pattern Growth) [79], SSFIM (Single Scan for Frequent Itemsets Mining) [62], and SETM (Set-oriented Mining) [83].

In [139], a **hybrid recommendation** system combining content-based, collaborative filtering and data mining techniques has been proposed. The recommendation algorithm makes similar groups of customers using LTV value, for this the segmentation of customers based on costumer behaviour through RFM attributes has been performed.

Discrete choice models are used to personalize recommendations as in [57] where a framework for estimating and updating user preferences in a recommendation system is presented. The authors demonstrated that the framework is computationally efficient and empirically accurate, however, parameter estimation can be inaccurate in the presence of non-heterogeneous data.

With the growing volume of data acquisition, the possibility of using **deep learning** in recommendation systems has been also considered, in order to overcome the obstacles of conventional models listed above, achieving higher accuracy of recommendation. Through deep learning, it is possible to detect non-linear and non-trivial relationships among users and items from contextual, textual and visual inputs [121]. The main limitations of deep learning-based recommendation systems are represented by the fact that there are often privacy issues in the collection of information for content-based systems, while for collaborative filtering the acquisition of data from different sources often results in incomplete information that greatly affects the accuracy of recommendations. The main deep learning algorithms for recommender systems are described as follows. Multilayer Perceptron (MLP) is a class of feedforward artificial neural network with multiple hidden layers between the input and the output layer. In [81], a standard MLP approach to learn interaction among user and item latent features has been used by providing the model with flexibility and non-linearity. Autoencoders (AE) represent an unsupervised model that generates an output by compressing the input in a space of latent variables. There are many variants of autoen-

coders; the most common are denoising autoencoder, marginalized denoising autoencoder, sparse autoencoder, contractive autoencoder and variational autoencoder [143]. Convolutional Neural Networks (CNN) are feedforward neural networks that use convolution in place of general matrix multiplication in at least one of their layers. They can capture the global and local features and improve the efficiency and accuracy [151]. They have been used in several implementations, such as AlexNet [100] and batch-normalized Inception [89]. Recurrent Neural Networks (RNN) are typically employed to trace dynamic temporal behaviour, actually in this kind of neural network the connections among the nodes form a direct graph along a temporal sequence [163]. Other fields of research have achieved an improvement by exploiting Long-Short Term Memory networks (LSTM) that minimize RNN problems regarding the gradient vanishing/exploding. LSTM has been applied in [126] to a movie recommendation system, in order to take into account users' dynamic and time varying behaviour, and not only their static preferences. Adversary Network (AN) is a generative model where two neural networks are trained simultaneously within a minimax game framework [158]. Deep reinforcement learning (DRL) combines deep learning and reinforcement learning that enables to learn the best possible actions to attain the expected goals [169].

An **ontology-based** recommendation system has been proposed in [16]. The proposed architecture sends semantic recommendations for each user profile by applying content-based filtering and collaborative filtering techniques.

Compared to the previously discussed data mining techniques, all deep learning algorithms have cold start problems and require a considerable amount of data to improve performance. Open problems in the literature for deep learning-based recommendation systems concern the frameworks' scalability and the explicability of generated recommendations. On the other hand, deep learning solutions are not applicable in this case, in which the number of acquisitions per user is low, which is one of the most critical problems of fashion retail.

2.3 System architecture

In the context of fashion retail, the shops are typically small in size (they are also known as boutiques), and the customers in the shops are directly

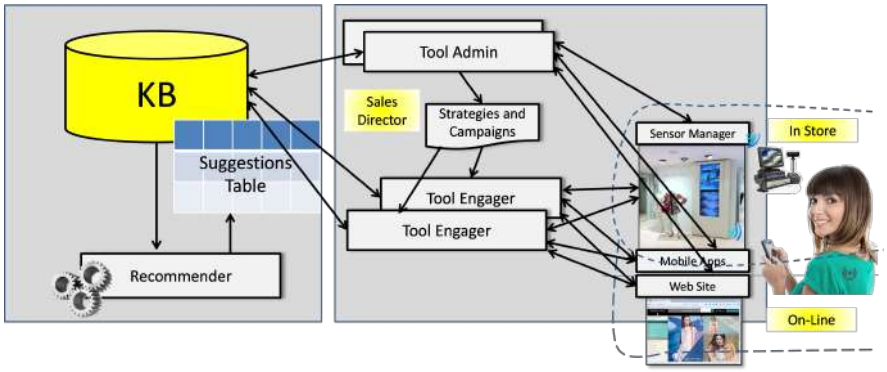


Figure 2.1: General architecture.

followed step by step by the attendees who provide suggestions and are ready to support them on every aspect. A similar scenario may occur on online shopping, in which an online assistant is ready to follow the customer, while the customer can more easily ignore the pressions of the attendee. In both cases, the user profiles are improved with new data in the few occasions in which the customers interact, and thus the customers might be continuously engaged with suggestions. This is obviously not possible since it cannot be acceptable by all the customers. So that, a moderated engagement tool has to be provided that may consume the possible recommendations by proposing them directly to the customers (via some devices into the shops or online) or via the assistant. Thus, the suggestions can be provided only a limited number of times per experience, and in specific conditions to avoid annoying/irritating the customer.

The architecture of the proposed system is reported in Figure 2.1. In compliance with GDPR rules, the Tool Admin stores the details of customers' profile, items and transactions on stores and on the ecommerce website in a centralized database based on MS Azure. The Recommender reads the information from the KB (Knowledge Base) and generates the recommendations which in turn are stored in the Suggestion Table. The Tool Engager is the only responsible for sending recommendations to the customers, directly or via the shop assistant within the store. After that a recommendation has been sent, the Tool Engager has the duty to records the customer's interaction/reaction with respect to the recommendations (e.g.: detect and track if the customer reads the recommendation, accepts the suggestion, test the

product and eventually buy it). The Tool Engager are instantiated one for each shop or group of shops. The Recommender creates a list of suggestions taking into account users' profiles and items' descriptions, as described in the following. The recommendations have to be carefully provided, since suggested items should not have been purchased by the customer recently, neither already proposed by the human Assistant. All the suggestions need to be generated on the basis of purchases made by the customer in the last few experiences and months, when possible. These last rules for filtering are applied directly at the final stage by the Tool Engager, and this means that the provided suggestions have to be abundant with respect to those strictly needed to be consumed in short time, to be sure that the Tool Engager would have always new suggestions to be spent when it can be in conditions to deliver one. For example, when a newsletter is sent, the customer arrives on web shops, goes in the test room, etc. The Sensor Manager is capable to manage and collect data and events from sensors in the shops, such as those in the fitting rooms, close to totems in stores, RFID technology on items for proximity and customers' interactions with products. All these data and events are identified and stored, and also trajectories performed by customers into the shop, which are tracked by using a Wi-Fi network of sensors. A rule system is capable to identify specific conditions to be sent to the Tool Engager, which enters in action providing or not a recommendation on the basis of discourse in place with the shop assistants and former suggestions. This chapter is focussed on the Recommender.

2.4 The recommender system

One of the main goals consisted in increasing the customer recency, and thus to increase the number of times user contacts and sales may occur. For this purpose, the computational workflow reported in Figure 2.2 has been adopted. The data are continuously collected by the Tool Admin and Tool Engager (sales in shops: online and onsite) into Knowledge Base (see Figure 2.1); then a periodic clustering on items is performed. The results are taken into account in the computation of an integrated clustering driven by the user profiles and additional features to finally provide a set of suggestions of different kinds. The main steps of the workflow are described in the following subsections.

The production of recommendations and their submission are asynchronous

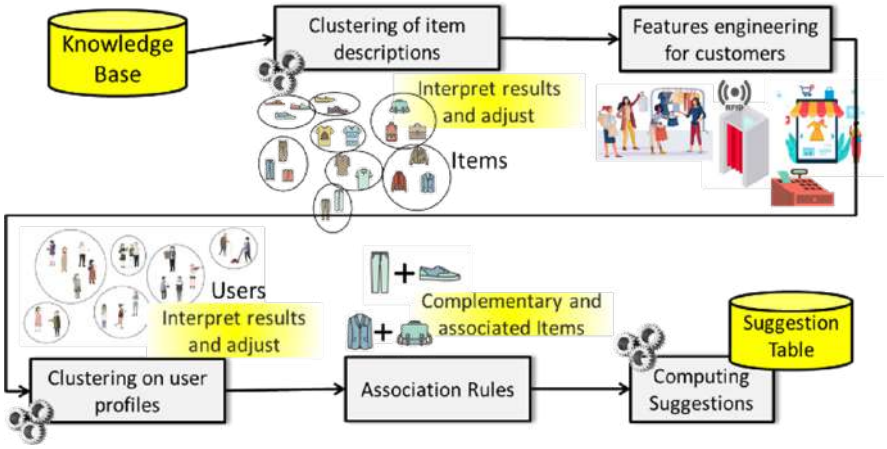


Figure 2.2: General data computing workflow.

since recommendations can be: (i) mediated by the assistant that may decide or not to accept and pass them to the customer, (ii) filtered by the Tool Engager according to the last actions performed by the customer, (iii) decided to be spent by sending them online via email when the time passed since the last contact with the users is greater than a reference value, when the new products which fit with the user preference would be available, etc. The produced pool of recommendations (for each potential returning user, and user kind) is generated on the Suggestion Table, which is refilled on demand of the Tool Engager or periodically with a high rate. The Suggestion Table includes a programmed mix of suggestions computed by *customer similarity, items similarity,* and *serendipity* (randomly produced).

2.4.1 Clustering of Item Descriptions

As described above, the first analysis has been performed to clusterize the item domain on the basis of their descriptions. This allows to reduce the space of all combinations and to weight the relevance of item categories. In the case of fashion retail, typically the number of products is not huge, differently to what one may have on supermarkets, in which a huge number of products is active on marketing at the same time. In our case of fashion retail, the database contained about 50000 items which have been classified according to the fields reported in Table 2.2, and which may belong to more

than one season.

Field ID	Item Description	Example
TYPE	Type	“1A0145”, “1A0333”, ...
CONFIGURATION	Configuration	“DRESS” ,“JACKET”, ...
PATTERN	Color	“White”, “Red”, “Navy blue”, ...
MODEL	Alphanumeric code model	“1A0145”, “1A0333”, ...
PACKAGING_TYPE	Type packaging	“Packaging Basic PE”, “Packaging Basic-Contin.”, “Women’s Packaging A/I”, ...
PRODUCTION_CATEGORY	Production category	“Accessories”, “Clothing”, “Jeans”, ...
MERCHANDISE_MCR_TYPE	Merchandise type	“Basic, Preview”, “Women”, “Main Women”, ...
MERCHANDISE_TYPOLOGY	Merchandise typology	“Preview Women SS”, “Main Women AI”, “Women PE”, ...
MERCHANDISE_MCR_FAMILY	Merchandise family	“Coat”, “Bag”, “Dress”, ...
MERCHANDISE_GROUP	Merchandise group	“Jewellery”, “Dress”, “Shirt”, ...
GENDER	Gender	“Accessories Women”, “Child”, “Women”, ...
BRAND	Brand	“VA”, “GM”, “PW”, ...
STYLE_GROUP	Style	“P”, “C”,...

BIRTH_SEASON	Season	“20201”, “20062”, “20071”, ...
PERIODICITY	Periodicity	“C”, “S”, ...
IS_CLOTHING_ITEM	Marking if the item belongs to a clothing category	1,0 (Yes/No)
NRM_CAT_LVL_1	Code normalized business classification level 1	“Accessories”, “Clothing”, “Jeans”, ...
NRM_CAT_LVL_2	Code normalized business classification level 2	“Bag”, “Clothing”, “Coat”, ...
NRM_CAT_LVL_3	Code normalized business classification level 3	“Shopping”, “Dress”, “Jacket”, ...
NET_SOLD_PRICE	Price	1580.00
IN_STOCK	Whether an item is available or not	1,0 (Yes/No)
132 X Hashtag tasche, abalze,...	Hashtag website	1,0 (Yes/No)

Table 2.2: Product item descriptions fields.

Most of the fields are textual descriptions, thus they are strings coding the description; then, only a few of them provide numeric or Boolean. Therefore, the clustering cannot be based on Euclidean space. For this reason, the clustering has been carried out by using K-medoids [94], which is a classical clustering technique that partitions a dataset of n objects into k a priori known clusters. A number of techniques to identify the best compromise on the value of K can be used [116]. To calculate the distance among items we used the Gower distance [74], which is computed as the average of partial dissimilarities across individuals. Each partial dissimilarity (and thus the Gower distance) ranges in [0,1]. In particular, the Gower distance is defined as follows:

$$d(i,j) = \frac{1}{p} \sum_{i=1}^{p} d_{ij}^{(f)}$$

where: $d_{ij}^{(f)}$ is the partial dissimilarity computation which depends on the type of variable being evaluated. For a qualitative assessment, the partial

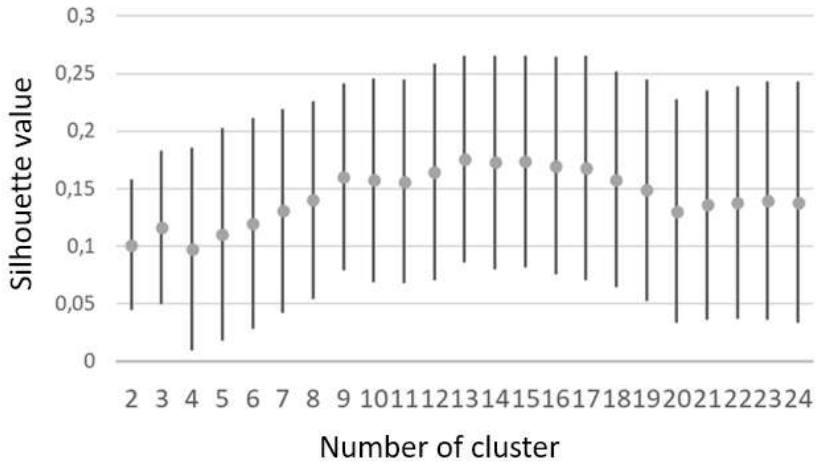


Figure 2.3: Trend of the silhouette value as a function of the number cluster K for item dataset.

dissimilarity is 1 only if observations x_i and x_j have different values, and 0 otherwise. Through the silhouette method, we determine the optimal number of clusters. The silhouette method calculates the average silhouette of observations for different values of K [116]. The optimal number of clusters K is the one that maximizes the silhouette over a range of possible values for K. In Figure 2.3, the trend of silhouette index with its standard deviation as a function of K, is reported. From the trend, the value of K=13 corresponds to the maximum of the averaged silhouette. It has been estimated as a compromise since the standard deviation is quite large, and a smaller number of clusters would provide too large sets for making selections.

Figure 2.4 shows the distribution of clusters' size for $K = 13$.

In Table 2.3, the descriptions of the identified clusters, and the corresponding sales are reported. The main descriptions have been identified by cluster analysis. The main drivers for clustering have been CONFIGURATION, MERCHANDISE_TYPOLOGY, BRAND and NRM_CAT_LVL_1.

Cluster	Derived descriptions of the item clusters	# items	# sales
2	DRESS, PE, clothing	6074	1171

1	BAG, AI, Accessory	6801	969
7	SHIRT, SS, Clothing	4346	838
3	TROUSERS, PE, Clothing	5786	794
4	KNIT, FW, knitwear	5222	678
5	T-SHIRT, PE, clothing	5100	674
6	ACCESSORIES (HAT - FOULARD - SCARF - NECKLACE - GLOVES - BRACELET), AI, Accessories	4479	596
10	SKIRT, PE, Clothing	2374	530
8	COAT, AI, Clothing	3133	388
9	SHOES, AI, Shoes	2835	341
11	JACKET, AI, Clothing	2365	292
12	BELT, AI, Accessory	2025	237
13	CHILDREN'S CLOTHING, Outlet PE. Cloting	1220	126

Table 2.3: Main description of products' clusters.

2.4.2 Features engineering for customers

The data collected by the administrations and the retail shops refer to the user behaviour, which is associated with the user profile. The user profile is enriched with information regarding customer behaviour such as: (i) fields about the customer's maximum interest for an item within the cluster, such as: Interest (Yes/No), Observed (Totem, Online, etc.), Tried, purchased item; (ii) fields describing the items purchased within the cluster. Point (i) is a vector of 13 elements (one for each item cluster) where 0 identifies no interaction for the client with items in the cluster; 1 if at least one item in the cluster was observed by the client; 2 if at least one cluster item has been tried or placed in the shopping cart; 3 if at least one cluster item has been purchased. Point (ii) is a vector of 13 integers (one for each item cluster)

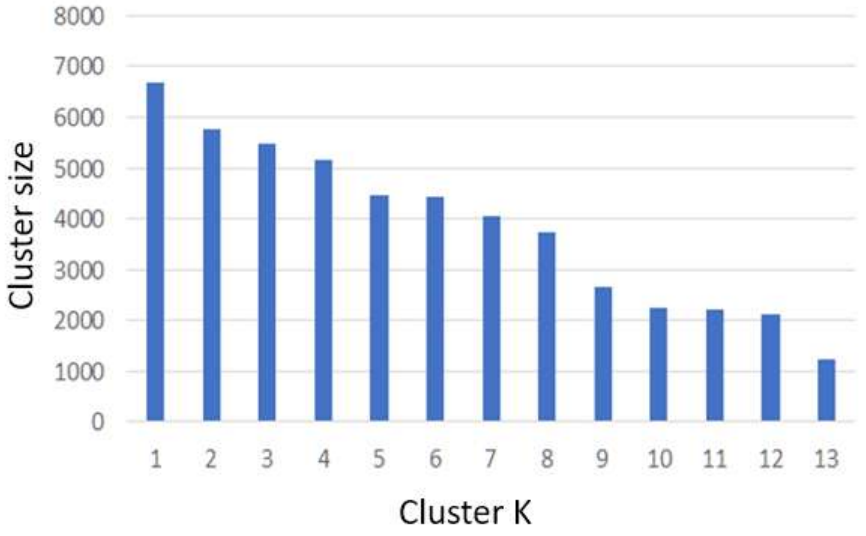


Figure 2.4: Distribution of size of the clusters in terms of items for K=13.

which represents the number of items purchased by the client within the cluster, 0 if no item has been purchased.

In addition, a number of features (which in some sense are KPI, (Key Performance Indicators) have been also computed, and assessed by taking into account the experience of business developers. Among them: recency, frequency, and average spending. Recency is defined as the number of days passed since the last visit or access in a store or online; Frequency represents the frequency of purchase in terms of the number of days; Average spending is the average value of a single ticket for the customer (estimated on the basis of the admin track record). In addition, in order to distinguish between online and in-store behaviour, online and in-store frequency and recency are separately computed.

2.4.3 Clustering on user profiling

In the considered scenario, the number of user profiles has been 608447, of which about 27300 have been acquired in the 2016-2019 temporal range. The user profile includes the features listed in Table 2.4. The features are the following: RFM_TRN_DaysFrequency is the frequency transaction, more pre-

cisely, how often the customer makes a transaction; RFM_TRN_DaysRecency is recency transaction, more precisely, how many days have passed since the customer's last transaction; RFM_TRN_AvgAmount is the average spending in a single transaction; RFM_PRS_ONLINE_DaysFrequency is the frequency presence online; RFM_PRS_ONLINE_DaysRecency is the recency presence online; RFM_PRS_ONPREM_ DaysFrequency is the frequency presence in a store; RFM_PRS_ONPREM_ DaysRecency is the recency presence in a store, FidelityUsageRange is fidelity card use, ranging from 0 (lowest usage frequency) to 3 (highest usage frequency); CUS_FIDELITY_CARD_LEVEL_CD is the fidelity card level based on fidelity points accumulated according to the spending (0 is the lowest level, 3 is the highest); Cluster_k_Interest size [13] is the maximum interest in an item within the cluster; Cluster_k_Purchased size [13] is the number of items purchased within the cluster. Other features such as Gender, Age, Family Status, Fidelity card level, family status, country, city were not considered because, due to different sources of profile collection, they had incomplete or missing data, or they are constant in almost all records.

Name profile feature	Description
RFM_TRN_DaysFrequency	Frequency transaction
RFM_TRN_DaysRecency	Recency transaction
RFM_TRN_AvgAmount	Average spending transaction
RFM_PRS_ONLINE_DaysFrequency	Frequency presence online
RFM_PRS_ONLINE_DaysRecency	Recency presence online
RFM_PRS_ONPREM_DaysFrequency	Frequency presence store
RFM_PRS_ONPREM_DaysRecency	Recency presence store
FidelityUsageRange	Fidelity card use
CUS_FIDELITY_CARD_LEVEL_CD	Fidelity card level
Cluster_k_Interest size[13]	Max interest for each cluster
Cluster_k_Purchased size[13]	Number of items purchased

Table 2.4: User customer features (all numbers).

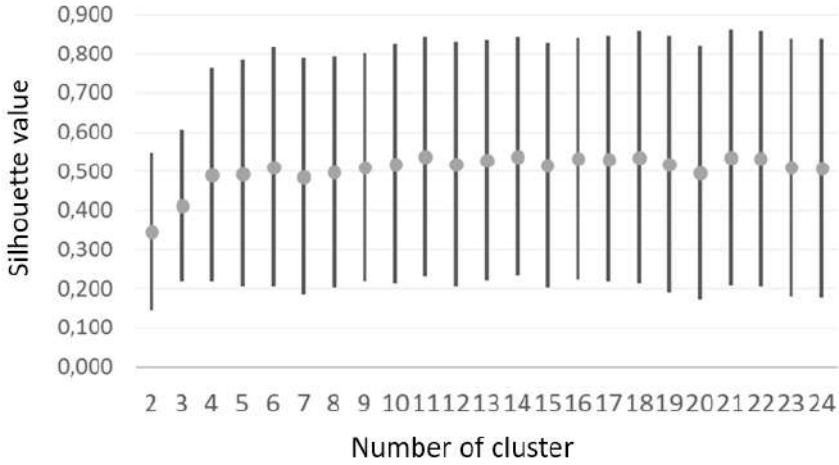


Figure 2.5: Average silhouette and its standard deviation vs number of clusters K.

On the basis of the user profile features, which include two arrays of user's preferences for items clusters identified in the first phase, a clustering has been carried out through the K-means method since the domain was Euclidean in this case. The Silhouette method has been used to determine the optimal number of clusters, in this case $K = 14$ (see Figure 2.5), taking the maximum of the average silhouette. The standard deviation is very large, while taking a smaller number of clusters would result in having too large clusters on which to make selections.

In Table 2.5, the derived descriptions of customers/user clusters and they corresponding size are reported. Please note that the main features characterizing the clusters have been: average amount of spending, frequency and recency.

Cluster	Derived Description from Customer cluster analysis	# Total customer
1	Customers with average spending amount not defined; the frequency is not defined neither in store neither online; day of the last purchase not defined	9195

2	Customers with low average spending amount, mainly online with undefined frequency and last purchase older than two years	3158
3	Customers with undefined average spending amount, mainly in store, with undefined frequency and last purchase older than two years mainly online	2433
4	Customers with low average spending amount, last purchase older than one year.	2302
5	Customers with low average spending amount in store, with frequency of about 4 months in store; last purchase has been made within one year. often using the fidelity card	2302
6	Customers with low average spending amount, more frequent in store with annual frequency; last purchase older than one year.	1657
7	Customer with low average spending amount, more frequently online, but also buying in store with a frequency of about 2 months online and about 6 months in store; last purchase older than one year, use fidelity card	1493
8	Customer with average spending amount not defined, mainly online; last purchase midterm days	1186
9	Customer with very high average spending amount in store	887
10	Customer with medium average spending amount more frequent in store but also buys in store with frequency about 230 days; last purchase about 262 days, use fidelity card	819
11	Customer with average spending medium amount in store; last purchase one year ago; frequency is not defined	797

12	Customer with average spending amount not defined, mainly online, with a frequency of about 270 days; last purchase one year	717
13	Customer with medium average spending amount, mainly in store, with not defined frequency and last purchase older than one year	391
14	Online customers with annual frequency	90

Table 2.5: Description of users' clusters.

According to the obtained results, cluster #1 was actually very large. For this reason, a second level clustering has been performed to split user cluster #1 into subclusters based on the same features. The Silhouette method has been used to determine the optimal number of clusters. The clustering result was initially highly unbalanced regarding the customers' distribution, therefore a further analysis on distributions of customers at varying cluster sizes led to take $K = 5$, with the aim of having maximum classifications and expression, as shown in Table 2.6. The distribution of clusters has been reported in Figure 2.6.

Cluster	Derived Description from Customer cluster analysis	# total customer
1.1	Customers with average spending amount undefined; the frequency is undefined neither in store nor online; day of the last purchase undefined	5133
1.2	Customers with low average spending amount. They mainly buy in the product cluster #12	2411
1.3	Customers with very low average spending amount, mainly in the product clusters: #2, #10 and #12	1330
1.4	Customers with recency of about 23 days, frequency of about 18 days	173

1.5	Customers with average spending amount of about 150 Euro; mainly buying in the product cluster #1	148

Table 2.6: Description of second level cluster of cluster #1.

The final distribution of clusters has been reported in Figure 2.7. In which the first level clusters are numbered from 2 to 14, and those of the second level clustering decomposing cluster 1 (of 9195 units) has been decomposed in clusters from 1.1 to 1.5.

2.4.4 Computing Suggestions

As described above, the identified solution produces a number of recommendations for each user. Each possible suggestion is labelled with the kind, the date of emission, and a deadline. The Engager Tool also marks those that have been spent with the date and time of emission, the channel adopted (shopID, mobileApp, website, shopID. totemID, etc.), the ID of the assistant, etc. This information is useful for the assessment of the acceptance level at follow up, and thus for the validation, as described in the next section. Therefore, the database with the suggestions is never discharged since the recommender must take into account the already spent suggestions.

The recommendations are generated according to different kinds (as described in the following list) and they are consumed in different contexts by the Engager Tool. Thus, the rate and the percentage of their exploitation/-consumption depend on the decisions of the Engager and on the number of occasions in which the recommendations can be provided. Moreover, since the number of suggestions is abundant, they are also substituted with new ones if not consumed in a reasonable time. The different kinds of recommendations are computed according to:

- **customer similarity**. For each customer cluster the most representative items are computed. They are identified among the most purchased items within the users' ones belonging to the same item cluster (they can be selected by using other criteria, for example: because they are the most frequently asked, or the company would like to push them, or they are closer to the cluster centroid, or to maximize the revenue or

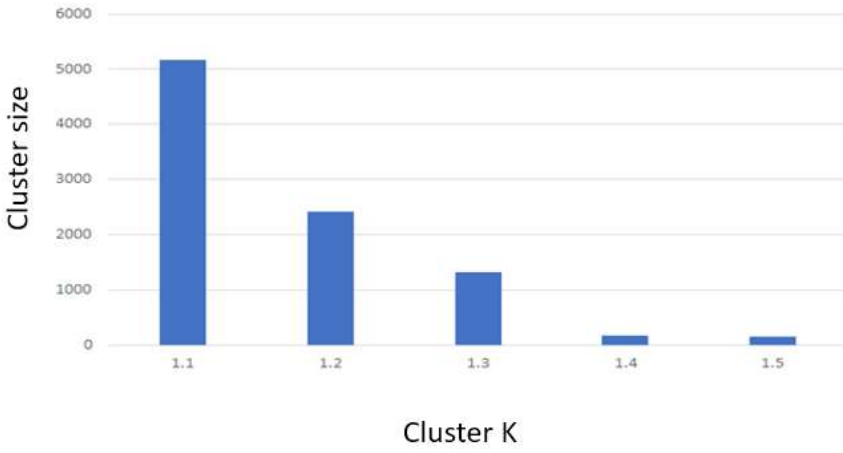


Figure 2.6: Distribution of customers along the resulting 5 clusters.

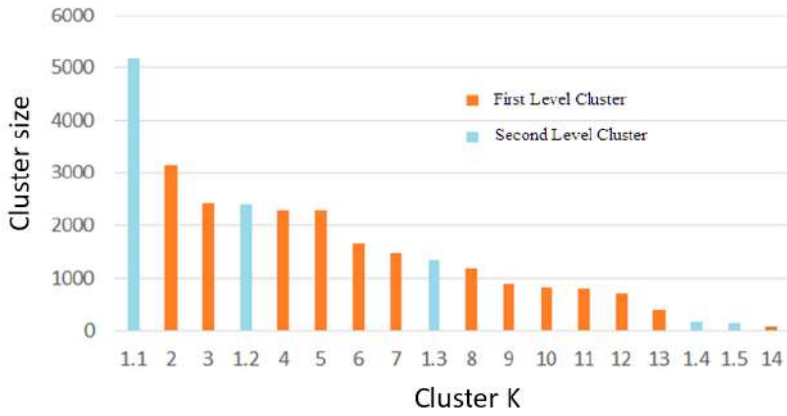


Figure 2.7: Distribution of 18 clusters for the number of customers, including first level cluster from 2-14 and second level from 1.1 to 1.5.

minimize the stock, etc.). In addition, the suggested item should have not been already purchased or proposed/suggested to same customer in the same season.

- **item similarity**. Considering the last items purchased by the customer according to the information contained in its profile, and randomly selecting items in the same item clusters, avoiding proposing items which have been already bought or proposed. Also in this case, the items can be filtered/selected by using additional criteria, for example: because they are the most frequently asked, or the company would like to push them, or they are closer to cluster centroid, or to maximize the revenue or minimize the stock, etc.

- **item complementary**. Considering items that may complement the last items that have been bought by the customer according to a table of complementary items; for example: a belt in combo with a bag. Please note that some of the item clusters are complementary each other, see the above descriptions - e.g., #1 and #2 of Table 2.3. To this end, through association rules using apriori algorithm [14] for each transaction in the dataset a set of metrics have been calculated; some examples are reported in Table 2.7, for the first 5 clusters.

Cluster	Complementary clusters				
	Cluster	Support	Confidence	Lift	Count
1	2	0.26486066	0.6069351	1.106003	12935
	7	0.24864345	0.5697729	1.253423	12143
	3	0.24465057	0.5606231	1.213722	11948
	8	0.24336057	0.5576670	1.277549	11885
	4	0.22298667	0.5109797	1.282096	10890
2	4	0.29840080	0.5437687	1.364367	214573
	3	0.34351004	0.6259701	1.355196	16776
	7	0.32391425	0.5902612	1.298495	15819
	8	0.31392182	0.5720522	1.310504	15331
	11	0.26490161	0.7762511	1.414544	12937
3	2	0.34351004	0.7436830	1.355196	16776
	7	0.30397035	0.6580814	1.447690	14845
	8	0.29868747	0.6466442	1.481385	14587
	4	0.27753548	0.6008511	1.507592	13554
	1	0.24465057	0.5296569	1.213722	11948
4	2	0.29840080	0.7487156	1.364367	214573
	3	0.27753548	0.6963625	1.507592	13554
	7	0.26578209	0.6668722	1.467029	12980
	8	0.27260069	0.6839807	1.566918	13313
	1	0.22298667	0.5594945	1.282096	10890
5	2	0.13366914	0.7559931	1.377628	6528
	8	0.12396339	0.7011002	1.606137	6054
	7	0.12224338	0.6913723	1.520926	5970
	3	0.12199767	0.6899826	1.493780	5958
	4	0.12158814	0.6876665	1.725420	5938

Table 2.7: Example of complementary clusters assessment by using metrics: support, confidence, lift and count (part).

The used metrics are $Support, Confidence, Lift, Count$, and are defined as follows. Let N and M be two clusters. $Support(N \rightarrow M)$ is the ratio of the number of transactions/tickets including N and M with respect to the total number of transactions. $Confidence(N \rightarrow M)$ is the ratio of the number of transactions containing N and $M$with respect to the total number of transactions containing N. $Lift(N \rightarrow M)$ is

the ratio of confidence of N with respect to the total number of transactions containing M. $Count(N \rightarrow M)$ is the number of transactions containing N or M. To generate the recommendations, we considered the top 5 clusters with highest *Support* and suggested one of the best-selling items (*Count*) within the cluster.

- **item associated**: in order to increase the customer's purchase frequency, we generated suggestions on the basis of what has been purchased in the last three months. For the generation we have proceeded as follows: through association rules using apriori algorithm [14] we have defined pairs of items (i, j) with $Support >= 0.001$ and $Confidence >= 0.01$. If a customer buys item i then item j will be suggested. This is the typical suggestion which can be delivered for stimulating the return on the shop. In order to take into account the evolution of the market and transactions the computation of Table 2.7 data, and thus of the association rules, is updated periodically. The periodic assessment has to take into account at least the last 6 months according to Table 2.5, which provides the evidence of transaction frequency/recency of customers.

- **suggestions for serendipity**: randomly selecting items to be suggested from the whole present collection, taking also into account what is available in the physical shop.

2.4.5 Considerations on Functional Dependencies

The above-described techniques for producing the suggestions are covering almost all cases. Recently, we have also analysed the usage of Functional Dependencies and their imprecise/relaxed and/or precise approaches [18], [48], [47], [75]. Those approaches are mainly focussed on identifying the complexity of relationships on data models. And thus, they can be profitably used to identify association rules, extracting the possible dependencies among the different fields related to user, products and transactions in a recommendation system. According to Section 2.4.4, for suggestions produced as item associated, specific products have to be suggested to the users; thus, the metrics of Table 2.7 identify the typical relationships among items bought by users and belonging to different clusters. Then, on the basis of the last item bought by a specific user it is possible to land on specific products by using the clusters identified by the association.

We explored the possibility of using Relaxed Functional Dependencies, RFD [48] in the context of recommendation. The first exploration has been on using RFD on a large data set with all features listed in the above clusters, using different time ranges (taking into account that in retails the returning period of user is large). Thus, incremental changes must be performed on long time range. An interesting result has been obtained by using RFD on the list of transactions annotated with the user information including: RFM_TRN_AvgAmount, RFM_TRN_DaysFrequency and RFM_TRN_DaysRecency, and by using product CONFIGURATION, MERCHANDISE_TYPOLOGY, and BRAND. The analysis has been limited to these features since all the other features have been discovered to be strongly dependent on them. The RFD has been produced by using the RFD-Discovery tool [7] which is referring to [48], using a bottom-up approach. With this approach it has been possible to identify relevant dependencies, for example, on the basis of a given distance or similarity on:

- Average Spending, which may help to identify the products the user could be interested to. This is also modelled by the customer similarity approach, observing that users tend to spend the same budget in average.

- MERCHANDISE_TYPOLOGY (using a semantic distance) which may help to identify typical user's average spending, frequency, and recency. This last relationship could be used to identify a potentially similar cluster of users that could be prone to buy certain kinds of products.

The distance adopted worked on numbers and on strings. For strings the tool exploited the lexical database WordNet for computing the semantic distance.

Therefore, the RFD techniques could be used for computing association rules among the fields. Most of the associations are straightforward, and in most cases, they produce similar results than those of the above presented multi clustering approach, which also depend on multidimensional distances. The techniques for differentials or incremental estimation of FD can be also used to detect changes [44], [69]. In our approach, the evolution of relationships on feature/clustering is progressively adapted by periodically recomputing the clusters and metrics of Table 2.7.

2.4.6 Consuming Suggestions

The suggestions are provided to the Suggestion Table, which is structured as described in Table 2.8. With this table structure, it is possible to save both generalized suggestions (e.g., by customer age or gender) and customer-identified suggestions.

Field name	Description
Recomm_id	Id suggestion
Customer_id	customer identification. -1 for generalised suggestions
Type	contains information about the type of suggestion
Suggestion	item identifier to suggest (id or list)
Preferences	pattern of the item to be suggested (color, size, etc)
Created	Suggestion creation date
Duration	Number of days the suggestion is valid.
Date_Issued	Date in which it has been spent
How	Contains information on how the suggestion was submitted (via web or in-store).
Feedback	On Likert scale 1-5 (1 is good), or NULL
Interest	interest shown by the customer (tried on in the fitting room, purchased etc)

Table 2.8: Suggestion table schema description.

Please note that the table does not provide all the information since the identified item ID allows to recover the description from the catalogue, and similarly the customer ID allows to recover the current status on the shop to avoid proposing multiple similar suggestions. Identical suggestions are also avoided since the Date_Issued marks when the suggestion has been spent. A segment of the Suggestion Table with instances is reported in Table 2.9.

Recomm id	Customer id	Type	Suggestion	Preferences	Created	Duration	Date Issued	How	Feedback	Interest
...	...	C_SEREND	30624	C=4, T=48	2020-09-18	60	2020-11-10	email	NULL	NULL
...	...	C_SEREND	10799	...	2020-09-18	60	2020-10-02	WEB	1	Purchased
...	...	C_CORR	22389	...	2020-09-18	60	2020-10-03	InStore	4	FittingRoom
...	...	C_COMPL	19149	...	2020-09-18	60	2020-10-04	email	NULL	NULL
...	...	...	...	...	...	...	...	...	...	...

Table 2.9: An example of the suggestion table status.

2.5 Assessment and validation

The recommendation system has been validated in a store located in Florence and on the online store as follows. We have exploited the data collected until December 2019 to test and tune the solution, verifying if the suggestions produced were also provided by the Assistant in shops and finally acquired by the customers. The algorithm updates the clusters monthly and generates the new suggestions daily. Considering the generated suggestions, without stimulating customers, we verified if there was a match among suggestions and items purchased by customers in the period January - June 2020, by checking transactions and verifying the shop assistants (which are the reference experts). This analysis showed that on about 400 customers who bought, about 10000 suggestions were generated. On suggestions generated, the 6.36% items were purchased. This was considered the minimum level of reaching with the efficiency since resulted to be possible without the tool. Then, the recommendation system was tuned on operative modality from July 2020 until December 2020, to stimulate a certain class of users, entering in the store, using the totem in the store and by mail for ecommerce. This analysis with the stimulated customers showed that on 67 selected customers in the trial, 3050 suggestions have been generated, while only about 20% has been actually sent to the customers (on shops and/or email). On the items suggested, 9.84% of them were actually acquired or tested. Therefore, using the stimulus of the recommendation system, we have increased the customers' attention of 3.48%. The period for the assessment and validation was also complicated by the COVID-19 pandemic which strongly limited the access to the stores, and the validation via the e-commerce without the effective verification of the shop assistant is not comparable with the conditions of the 2019.

2.6 Final Considerations

In this chapter, a recommendation system in the context of fashion retail has been proposed and described, relying on a multi-level clustering approach of items and users' profiles in online and physical stores. The solution has been developed in the context of the Feedback project funded by Regione Toscana, and has been conducted on real retail company Tessilform, and it has been validated against real data from December 2019 to December 2020, showing

that the use of the proposed recommendation tool generated stimulus to the customers which brought to an increase of buyers' attention and purchase increase of 3.48%. The solution proposed has demonstrated to be functional also in the presence of low number of customers and items (as happens in retail shops, in which the items are of high value), and when suggestions are mediated by the assistants, as happens in the fashion retail shops. Moreover, the proposed solution addresses and solved lacks and issues which are present in current state of the art tools, such as also the cold start problems in generating recommendations for newly acquired customers, since it relies on rules mining techniques, allowing to predict the purchase behaviour of new users. Our solution is also GDPR compliant, addressing the current strict policies for users' data privacy, solving one of the main issues for managing users' demographic details.

Chapter 3

A Deep Learning Approach for Short Term Prediction of Industrial Plant Working Status

Predictive Maintenance has gained more and more research and commercial interests, being a pivotal topic for improving the efficiency of many production industrial plants to minimize downtimes, as well as to reduce operational costs for interventions. Solutions reviewed in the literature are increasingly based on machine learning and deep learning methods for the prediction of fault proneness with respect to normal working conditions. Solutions often do not consider or properly model temporal information, which is usually an important aspect to be considered for modelling industrial systems degradations. Many state of the art solutions are not actually applied in real scenarios and have restrictions to be executed in real-time in the production environment. In this chapter, a framework for predictive maintenance is presented. It has been built upon a deep learning model based on Long-Short Term Memory Neural Networks, LSTM and Convolutional LSTM. The proposed model provides a one-hour prediction of the plant status and indications on the areas in which the intervention should be performed by using explainable LSTM technique. The solution has been validated against real data of ALTAIR chemical plant, demonstrating an average accuracy of

91.8% and an average F1-score of 90%, with the capability of being executed in real-time in a production operative scenario. The chapter also introduced business intelligence tools on maintenance data and the architectural infrastructure for the integration of a predictive maintenance approach into the whole control and management systems of ALTAIR industry 4.0 plant [1][2].

3.1 Introduction

In real world Industry 4.0 scenarios, it is necessary to maximize the efficiency and productivity of plants, in order to improve competitiveness in the market. To this end, a crucial role is played by the production plant maintenance. In addition to efficiency and productivity, good maintenance reduces operative costs, improves product quality, and rationalizes resources. Typical kinds of maintenance policies are Corrective Maintenance (CM) and Preventive Maintenance (PM). The CM [42] or run-to-failure is quite expensive, it consists of the intervention after a failure in the production cycle that in most cases leads to the production plant stop. The PM is defined as maintenance carried out according to predetermined technical criteria. For equipment, it may be indicated in the instructions for use or manufacturers' technical documentation, with the intention of reducing the likelihood of equipment failure or degradation of a service rendered [68]. PM can reduce the number of failures/stops and can also be cyclical (time-based maintenance, TBM) and predictive (condition-based maintenance, CBM). In TBM the decisional process is determined on the basis of failure time analyses [165]. Instead, in CBM, it is based on information related to the condition of the plant [90]. In complex production plants, different kinds of maintenance strategies may be adopted at the same time for different parts and production lines. For PM, solutions and techniques proposed in research literature can be classified in three approaches, based on: physical, data-driven

[1] Part of the work presented in this chapter has been published as "A Deep Learning Approach for Short Term Prediction of Industrial Plant Working Status" in *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)* [35].

[2] *Acknowledgments:* Our thanks goes to ALTAIR for which we have developed the solutions described in this chapter, SODA project partners and Regione Toscana for cofounding of ALTAIR. Former technologies as Snap4City/Industry are 100% open source by DISIT Lab and can be accessed at `https://www.snap4city.org`

and hybrid [106].

Physical methods use domain knowledge of the physical system to be maintained to estimate a mathematical function or an analytical description of system degradation, anomalies and failures. They provide a clear understanding of physical meanings of the dysfunctions, and thus, they are very difficult to be implemented in complex systems. Data-driven methods aim at predicting systems' state by monitoring conditions as learned from historical data. They are suitable for complex systems since they do not require understanding the inner detail of plant operating conditions. On the other hand, it is more difficult to relate their output to a physical meaning, and this makes difficult to plan the intervention. Finally, hybrid methods are a mix of both the above-mentioned approaches.

Recently, the development of advanced sensor technology and computing systems has brought more and more interest towards PM based on data-driven solutions. In addition, there is a growing push in the industries to monitor production lines via IoT (Internet of Things) devices/sensors. The new IoT trend is making possible to collect a huge volume of data related to the operation of a single machine or of an entire production line/cycle. The challenge of the scientific community is to improve predictive maintenance systems by exploiting this amount of data in order to recognize anomalies and avoid downtime that still may frequently occur even with good cyclical maintenance plans.

In this chapter, an integrated solution for predictive maintenance in a chemical plant is presented. Most of the chemical plants are critical infrastructures that present a production process never stopping and running 24H/7D per week. In addition, the case taken into account presents a production process including chemical products which have to be carefully treated for their potential impact on the environment in case of an accident. This implies that early warning and an efficient corrective maintenance are mandatory policies to be established to become operative. The aspects addressed in this chapter are: (i) the usage of deep learning techniques for predictive maintenance, specifically Long-Short Term Memory Neural Networks, LSTM and Convolutional LSTM, with some technique for explaining the prediction which can be used to help the maintenance teams; (ii) the integration of workflow management system for maintenance with general control systems and data flow (also developing Node-Red library for integrating data flow and workflow ticketing system); and (iii) a business intelligence tool for

maintenance. The solution has been developed exploiting the IoT Industry 4.0 development environment and framework called Snap4Industry, which in turn is based on Snap4City which is 100% open source (and licence free) and it is available at [https://www.snap4city.org], [24], [22]. The new capabilities have been exploited to implement the higher-level control in the large chemical plant of ALTAIR. The Altair Chemical has been the first European KOH producer mercury-free chlor-potash plant. ALTAIR SODA-4.0 project aimed to develop and implement an integrated management system and optimization of the production and consumption processes of hydrogen and chlorine. It has been based on a new production structure based on a new NaOH (caustic soda) production line on the plant.

This chapter is structured as follows: in Section 3.2, a review of related work in the context of Predictive Maintenance is reported. In Section 3.3, the general architecture of the solution is presented, where the action to put in place a predictive maintenance aspects to work in real time are evident. Section 3.3.2 describes the Business Intelligence for the analysis of the maintenance data. In Section 3.4, an early version of the Predictive Maintenance Model based on LSTM is described with its assessment. Section 3.5 presents an advanced Predictive Maintenance Model based on CNN-LSTM and its validation results. All the validations have been performed by taking into account data of ALTAIR chemical plant. In section 3.5.3, an approach for explaining the results in real time and thus for exploiting the maintenance predictions for the identification of the area in which to operate has been reported. Finally, Section 3.6 reports final considerations.

3.2 Related Work

The data-driven techniques used for the prediction of industry plant failures can be used as early warning, and firing activities of maintenance, as in CM. This approach aims to avoid/reduce the occurrence of major disasters that may happen on chemical plants. A typical approach consists of the following major steps: data acquisition, feature extraction, feature engineering, model training, finalizing a predictive model, and then perform predictive model validation/testing [168]. Predictive and/or early monitoring results obtained from the resulting model may finally support the decision-making process in the development of predictive maintenance policies. In critical infrastructures, one has to set up any kind of planned maintenance, and

resilience guidelines also impose to be ready for the unexpected unknowns [30]. Data-driven based methods can be divided into two groups, based on two main approaches: traditional machine learning techniques (e.g., logistic regression, eXtreme Gradient Boosting (XGBoost) and Random Forest (RF) for supervised classification) and deep learning techniques.

In [41], logistic regression, XGBoost and RF techniques have been used on environment variables and machine temperature variables to predict 30-minute machine failure in real time. From the evaluation of the Receiver Operator Characteristic (ROC) curves, all three algorithms perform significantly better than a random classifier. In [153], k-means clustering technique was used to illustrate normal machine operation and three failure conditions, exploiting temperature and pressure sensor data. [115] demonstrated how Support-Vector Machine (SVM) techniques obtained a Root Mean Square Error (RMSE) of 0.732 on Gas turbine engine time series sensors. In [92], a model based on Auto Regressive Integrated Moving Average, (ARIMA), has been proposed to predict production values, which are feed to different supervised models (Classification and Regression Trees, SVM, Naïve Bayes and Deep Neural Networks).

With the growing volume of data, the possibility of using deep learning methods in predictive maintenance has been also considered, to overcome the obstacles of conventional models listed above, achieving a higher accuracy of prediction. Many solutions exploiting deep learning models relied on flatten layer architectures, without considering temporal information [170]. However, temporal information is important since machines typically degrade over time from their normal operating state to failure, and the development of faults is also a gradual process [104]. In [166], Long Short-Term Memory (LSTM) neural network has been proposed to detect the system degradation and to predict the remaining useful life. In this context, another issue is represented by the fact that raw sensor data may contain relevant amount of noise, which can badly affect performances of LSTM models. Therefore, Convolutional Neural Networks (CNN) have been combined to LSTM to support the extraction of local features, in addition to temporal information [168]. The main task of convolutional layers is to learn features from data input; actually, CNN was successfully applied for image classification. However, recently CNN has gained an increasing research attention also in the field of time series data analysis and classification. CNN-LSTM models have been successfully applied to time series analysis in different contexts, such as

stock market forecasting [98], photovoltaic power predictions [150], outperforming traditional LSTM models for their robustness to noise. In [144], a model for faults diagnosis in chemical process based on Multi-channel CNN-LSTM architecture is proposed. The model achieved an F1-score of 92.03% only on a simulated scenario, and not in a real test case. In [86], CNN is used to monitor the operation of photovoltaic panels aiming at predicting malfunctions on the panels. The approach outperformed compared approaches based on simple interpolation filters. In [167], a model exploiting a Deep Belief Network (DBN) has been presented for fault detection and diagnosis in a chemical process. The DBN model extracted the features from process data collected over a given time period, to classify the fault status, achieving an average fault diagnosis rate of 82.1%. In [105], a method based on Deep Boltzman machine has been used for learning features and fault classification from vibration measurements of a rotating machinery. The fault classification performances assessed in experiments using this method report a classification rate of 95.17%. Accuracy greater than 90% has been also obtained with the use of Autoencoder and softmax regression in [147], where bearing failures are diagnosed by receiving as input their characteristics.

Even though the performance of deep learning techniques is usually very high, the state of the art does not solve some typical problems in industrial plants. One problem is related to the amount of data describing failures. Most companies collect a huge amount of data related to the normal operating conditions of the plant. Moreover, the use of deep learning techniques creates problems of interpretability of the results. In Table 3.1, a comparative overview of the above mentioned Related Work is presented.

Reference	Techniques	Data	Type of predictions	Real time	Results
[115]	SVM	The method is tested on a simplified simulated time-series data set	Estim. the remaining useful life (RUL) of systems and/or equipment by time series	Simulated	Traditional SVR obtained a RMSE= 0.732
[92]	ARIMA	Data from Slitting machine (Tension, Pressure, Width e Diameter.)	Predicts parameter values for the production cycle.	Yes	accuracy 94.46% for CART to 98,69% for DNN
[144]	Multi-Channel LSTM-CNN	Tennessee Eastman (TE) chemical process simulation	Five-sample time series to predict the next sample	Yes	Average F1-score of 92.03%
[166]	LSTM	NASA's C-MAPSS dataset	Track the system degradation and to predict the remaining useful life	No	RMSE =18.07
[105]	Deep Boltzman machine	Two typical rotating machinery systems	Diagnosing the health of rotating mechanical systems by classification	Yes	Accuracy = 95.17%.

[167]	deep belief Network (DBN)	Tennessee Eastman (TE) chemical process simulation dataset	Extracts the feature from a process data period and classifies the fault status	No	Average fault diagnosis rate of 82.1%.
[147]	Autoencoder - softmax regression	Bearing dataset of Case Western Reserve University	Bearing failures are diagnosed by receiving as input their characteristics	No	Accuracy > 90 %

Table 3.1: Comparative overview of related work implementations.

3.3 Designing a system for chemical plants maintenance

Industry 4.0 approaches expect to have a high level of digitization in the production plant. As a first step, a number of DCS (Distributed control systems) or SCADA (Supervisory Control And Data Acquisition) are devoted to control the production pipelines and machines. On top of them a higher level of control and supervision may be needed. The latter may be also used for telecontrol. All the industry subsystems are typically connected on local area networks, and a number of IoT devices may help to glue and monitor the in/out flows, connections and areas among the machines, and production plants. On this view, the administrative area of the factory has to be connected in order to provide updates of the orders to the production plan, and to the delivering schedule. This implies to communicate the planned production to the control room of the production process. The new planned production may impact on maintenance teams if some changes or settings have to be physically implemented on the production line. On the basis of the foreseen planned production, the acquisitions of raw materials have to be scheduled for minimizing the stock. Thus the transportation of resulting products has to be carefully planned. In H24/7day active plant, any reason to stop production also impacts on delivering, ordering, transportation, etc. In addition, in chemical plants the production lines are in most cases connected each other (see Figure 3.1). Thus, subproducts (liquid, gasses, matter) of a production line, may be a primary matter or source of energy for the functioning of another line/transformation (ISx, input storage/source). This implies nonlinear cause effect and thus relationships among the several phases of the production lines. Moreover, multiple points may have storages and direct feeding with basic (e.g., ISx) and self-produced matter (S2), for example when the internal production is not enough or it is more expensive according to the market (for example transformation A can exploit IS3 and/or S2). In this view, the storages are important, and may create problems when they are full (no more space for producing other results/matter) or empty (not enough matter for the next process phases) since the production cannot continue. In addition, when they are full an immobilization of capital is also realized (see Figure 3.1).

According to this view, an integrated solution for the production plant maintenance has to cope with a number of aspects. It has to present a tick-

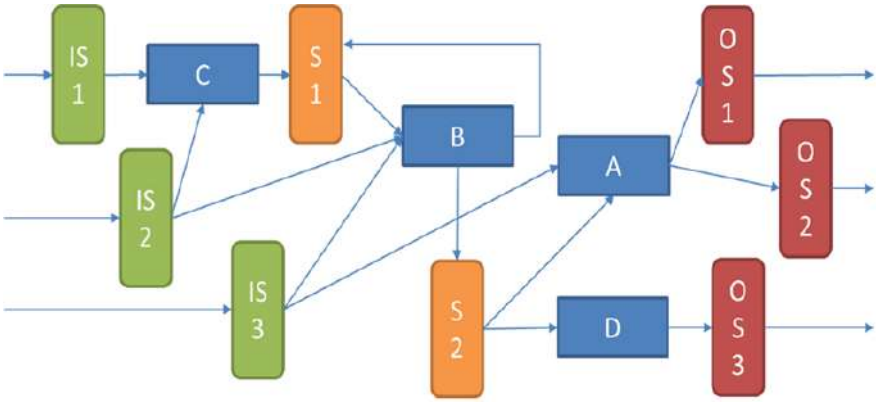


Figure 3.1: Example of chemical production in plant in which storage (round squared) and matter transformations (squared) are connected in a network of consumer/producer.

eting management system addressing workflow for managing maintenance activities with teams of any kind. The teams have to be ready for planned maintenance as well as for corrective maintenance. Predictive maintenance may help them to perform preventive interventions (for example when it is possible to perform some changes into the flow from the control room, putting out of the production pipeline a part of the plant without stopping the rest). For example, to take raw materials from the same or different silos by using a secondary pump and putting a primary on maintenance. In most cases, the predictive maintenance produces a probability of fault once per day/hour, since the stop of chemical plant could be very costly in terms of missed production. Moreover, some of the chemical plants, such as that of ALTAIR, may manage dangerous chemical products and results such as chlore-based products and hydrogen.

The unexpected critical events and alarms that lead to some intervention and plant dysfunction may be detected in control room by the operator, by plant alarms for flooding or any other detection of dysfunction, as well as through the information of some personnel observing the inception of a problem. All these aspects may lead to create a ticket on the Maintenance Management tools. Specific problems may be detected on control room observing DCS data on dashboards. The event produced from data analysis can be automatically generated by means of a direct connection from the

control room to the management department, from data flow to workflow of the maintenance teams.
The whole set of maintenance events have to be collected to be further analysed with some business intelligence tools for decision making and by some machine learning tool to perform some predictive maintenance.

3.3.1 Architecture

In this section, the general framework of the proposed solutions is presented in terms of functional architecture, as shown in Figure 3.2. The functional architecture presents several components which are described in the following by starting from left to right. The plant is controlled by a DCS (Distributed Control System) which produces a new status of the plat in terms of measures every minute. It is controlled on the basis of a number of set points for a large number of machines (pumps, electrolysis, heating, fans, storages, etc.) from a control room. The precise time trend of those setpoints is planned and updated more than once per day, and also changed in real time when critical conditions occur. The values of the setpoints over time are planned on the basis of the production to be performed.

The OpenMaint is an open source ticketing management system that implements: (i) the planned/scheduled maintenance activities, (ii) collect the tickets, (iii) prepare the teams, (iv) send them on the field and (v) collect the results according to a set of specific workflows. The Team Operators have in their hands a tablet to follow the instructions of the maintenance tickets, getting maps, documentations, etc., and a Web App to access at higher level data to see the settings and verify the plant status. Every time a new Ticket for Maintenance is created, a new action is started. Planned Tickets and unexpected Tickets/events are registered and managed. They can be also started from the Team mobile App and interface, and by the Control Supervisor, which may autonomously ask the Maintenance Team to create some Events on the basis of the data collected from DCS and/or on the basis of the Predictive Maintenance results. This last possibility is enabled by: (i) the development of a number of MicroService/Nodes for Node-RED into the Snap4City library for accessing to OpenMaint API, (ii) the Predictive Maintenance solutions presented in this chapter, (iii) and by the whole architecture. Both, the data of the DCS, and those of the Ticketing system have been collected in ALTAIR for several years. This allowed to have a large amount of data for training the Predictive Maintenance solutions

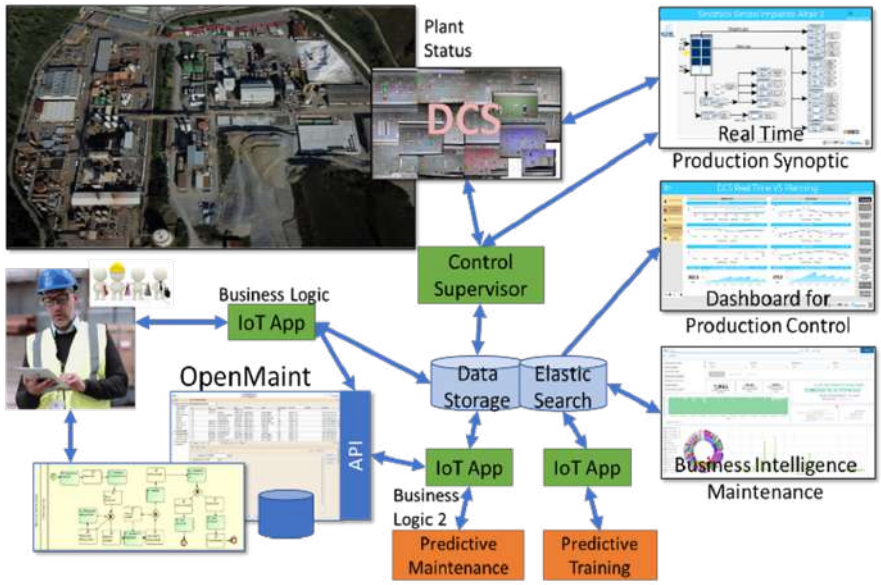


Figure 3.2: Functional architecture.

which are based on machine learning. As a first step, we started with the study of the relationships among the data collected from DCS and the events on the Ticketing system. To this end, a Business Intelligence Maintenance tool based on Elastic Search and Kibana has been designed and set up. It allows to perform visual queries on the plant status and on the history of the maintenance interventions. For example, segmenting the plant and filtering the interventions in a given area and/or production line, analysing causes and effects of certain faults in the plant, perform direct statistics on the kind of faults occurred per category, per period, per product, etc. (see Figure 3.2). All the data collected by DCS and the Control Supervisor are also visible on a set of Dashboards realized by using Snap4City tool [36]. The Dashboards allow to keep trace of the production status with respect to the historical data, and also to monitor the planned production with respect to the actual production for each product of the whole chemical plant. The Control Supervisor collects in real time the DCS data and also data coming from the administrative database, energy costs, etc., to compute the best plan for the next hours and days [22]. In addition, it produces, in real time,

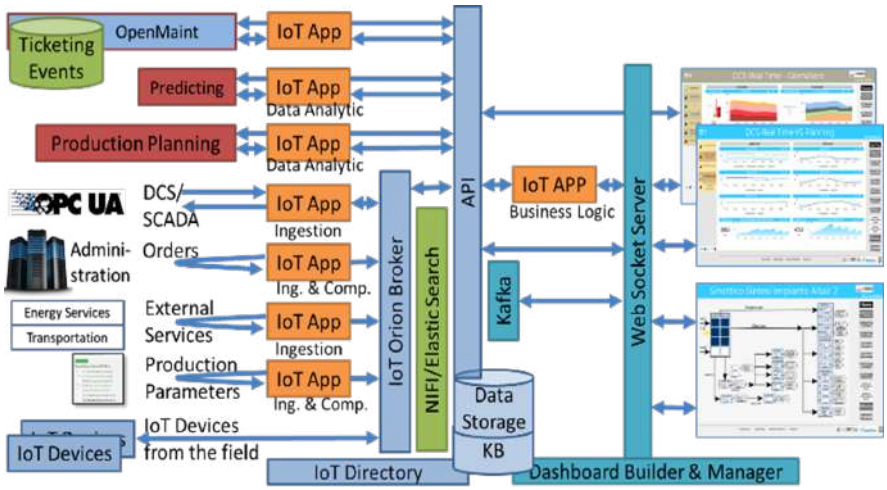


Figure 3.3: Technical architecture of the data ingestion and data flows: from data sources, to IoT app processes, storage and dashboards with custom widgets and synoptics.

the data driven streams towards a set of synoptics for the control room. Eventual alarms identified by the Predictive Maintenance are registered in the store and passed (in an event-driven manner) to the Control Supervisor and thus on the Synoptics, and also on the Ticketing via the API.

From the technical point of view, the solution has been implemented by using Snap4Industry development environment which in turn is based on Snap4City technology [25]. In Figure 3.3, the technical implementation regarding data flow of the functional architecture of Figure 3.2 is presented. In the solution, IoT Apps are Node-RED processes on Docker containers deployed on private cloud as IoT Edge. The IoT App connected with the OpenMaint ticketing system is the interface to access the new events produced by the operators, and to provide new events that may be identified by the Control system. Data are received in push and/or pull, and almost every data message with several attributes is considered an IoT Device instance. To this end, the IoT Devices have been registered, and their data structure/model formalized [22].
Once registered, the received IoT messages by an IoT App can be sent into the system via the IoT Broker, which saves them automatically into the

Data Storage. The IoT Apps can collect data in Push/Pull modalities, and in some cases, a preliminary computation is performed. IoT Devices from the field (sensors and actuators) can directly send/receive data to/from the IoT Brokers, and the messages are directly saved into the Data Storage at each change, via Apache NIFI. Processes of IoT App, IoT Brokers, Web Sockets, etc., are data driven and thus they are activated/fired by the arrival of new messages/events or, in the case of IoT App working in pull, by an internal scheduler.

The arrival of a new message in the Broker may provoke the sending of a new message into the data storage as well as a set of messages towards the user interface clients (dashboards and mobile Apps). In fact, each client browser shows one or more Dashboards and each mobile App connected to the Dashboard Manager need to have established a number of permanent WebSocket connections to receive in real time any change on the data.

DCS/SCADA data messages are received in push from OPC-UA. The DCS data includes measured values from productions flows, the settings planned and reached by the plant, status of the material storages, etc., such as for AcidoEsausto, FeCl2pot, FeCl3pot, FeCl3std, HCl32, HCl35, HCl36, KOH, HCl, K2CO3aq, NaOH50, that are the most important to represent the plant status and these are the values used by the planner algorithm.

3.3.2 Business intelligence for maintenance

The goal is to have a general overview of maintenance events, both planned maintenance and breakdown events that lead to downtime. It is possible to analyze the number of maintenance events per plant and per type of intervention by filtering on a chosen time frame.
The tool of Figure 3.4 presents six sections. In the first section, it is possible to choose the time frame by selecting the date (absolute) or by selecting years, months, or days (relative). In the second section, the user can see the number of maintenance events in the selected period: the average and median of the number of hours needed to complete a maintenance intervention (intended as the difference between the start and end of intervention datetime). In the third section, it is possible to select data by specifying and/or the: Plant, Specialty, Work type. A graph shows the trend of the working plants (daily); if the time span is very wide the plants will be grouped by week or month. In the fourth section, through a bar graph, we keep track of the maintenance events carried out per day (weekly or monthly if the

time frame is wide). It is possible to have several maintenance events for the same plant; therefore, a table at the bottom of the dashboard lists all the details by maintenance event Time, Permission List, Plant, Signature, Specialty, Status, Job Type, Air Temperature, air humidity and rain. In the fifth section, we have a visual representation of the maintenance events by Plant, Specialty, and Job Type. The font size depends on the frequency of the records. By choosing an item the dashboard widgets are filtered according to the choice. Finally, in the sixth section we have a pie representation of maintenance events by Plant, Specialty, and Job Type. From this dashboard it is possible to access the general management and control plant dashboards. In another section, we keep track of MTTF (mean time to Failure), MTTR (mean time to repair, and MTBF (mean time between failure, as MTTR+MTTF), current and trend values. Other information are:

- Average temperature of the day in which failures occurred in the plant.
- Average Humidity for the day on which failures occurred in the system.

Figure 3.4: Maintenance dashboard, business intelligence for maintenance.

3.4 First predictive maintenance model

According to the previous description, the challenge was to predict the plant failure 60 minutes before it happened. This section has two subsections; the first includes some descriptive notes about the dataset, while the second describes the architecture of the predictive model exploiting the LSTM model and related validation.

3.4.1 Data Description and Engineering

The data collection has been conducted on the basis of about 300000 observations from 2020-04-28 to 2021-01-04 (nonstop for COVID-19 since the production is mainly on chemical products for food industry). Regarding production data, the collected dataset is composed by a set of data coming from the DCS (such as plant: production, storage, status, several temperatures of elements, gear plants, process/safety parameters, chemicals compounds produced etc.) measured every minute. Therefore, a multi-feature dataset composed by 1-minute observation was one of the inputs. A total of 343.183 observations for 147 features/variables were measured. Regarding the faults, we had the list all the details coming from the Business Intelligence tool for maintenance including: event datetime, Permission List, Plant, Signature, Specialty, Status, Job Type, Air Temperature, air humidity and rain. Ticket and stop classification as "GENERAL PLANT STOP", "ORDINARY", "PLANT STOP" and "EMERGENCY". The label "ORDINARY" concerns all planned maintenance operations, while the labels "PLANT STOP" and "GENERAL PLANT STOP" concern programmed machine stops and finally "EMERGENCY" concerns machine stops due to an unexpected fault in the plant. In this way, a multi-feature annotated dataset has been created, considering data with the label "EMERGENCY" as faults, while all the other above-mentioned labeled data as regular working conditions, in order to implement a predictive binary classification model (as described in details in 3.4.2). In total, in the selected period, there were a number of breakdowns in emergency, and they produced a wider number of time intervals in which the production has been stopped/reduced for dysfunction and repaired (see Figure 3.5).

From the analysis, 28 variables have been identified removing those that replicated the same information in the production flow, as reported in Table 3.2.

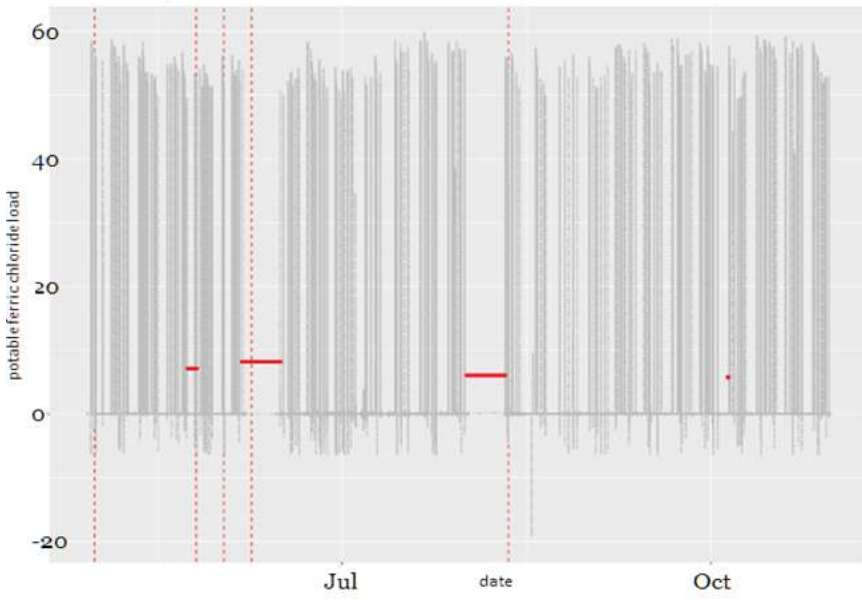


Figure 3.5: Example of a variable affected by a failure in terms of percentage of plant production capability.

Feature	Plant	Description	Unit of measure
TempreactoreR4001	chlorine paraffins (CPS)	Reactor temperature indication R 4001	°C
TempreactoreR4002	chlorine paraffins (CPS)	reactor temperature indication R 4002	°C
TempreactorR4003	chlorine paraffins (CPS)	reactor temperature indication R 4003	°C
S4304	chlorine paraffins (CPS)	level indication	%
standardFerric	Chloride Potable Ferric std	flow rate measurement and totalization	m3
S904C	Potable Ferric std	level indication	%
S904B	Potable Ferric std	level indication	%
S904A	Potable Ferric std	level indication	%
potFerricChloride	Potable Ferric Chloride	flow rate measurement and totalization	m3
S904E	Potable Ferric Chloride	level indication	%
S904D	Potable Ferric	Chloride level indication	%
QuantNaOHBatchNaClO_2	NaOH KOH	flow rate measure and totalization	lt
QuantNaOHperBatchNaClO	NaOH KOH	flow rate measure and totalization	m3
ConversionNaOH	NaOH KOH	electrolysis load adjustment (production)	kA

ConversionKOHlinea1	NaOH KOH	electrolysis load adjustment (production)	kA
KOH_1_charge	NaOH KOH	flow rate measure and totalization	m3
KOH_2_charge	NaOH KOH	flow rate measure and totalization	m3
S487	NaOH KOH	level indication	%
S484	NaOH KOH	level indication	%
S5104	NaOH KOH	level indication	%
hypo sodium	sodium hypochlorite	quantity of material produced	m3
S857	sodium hypochlorite	level indication	%
S856	sodium hypochlorite	level indication	%
S851	sodium hypochlorite	level indication	%
S852	sodium hypochlorite	level indication	%
S854	sodium hypochlorite	level indication	%
S871	HCl	level indication	%
RedoxFeCl3Pot	Ferric Chloride std	potential measure redox Ferric Chloride	mV

Table 3.2: Overview of feature measured at a given time.

Metrics S857, S856, S851, S852, S854, S871, S487, S484, S5104, S904E, S904D, S4304, S904C, S904B, S904A represent the level of the storages containing the chemical product. On this regard, we calculated the difference with the previous minute to highlight the total daily production of a given substance. These derived metrics were added to the above-described set of variables, thus obtaining a total of 43 features for 343183 minutes and few events of failure but a large number of minutes in which the plant have been

stopped/reduced in the operative level. Then we have, on a total of 343183 1-minute observation data, 37286 minutes of failure leading to downtime

3.4.2 Classification model LSTM

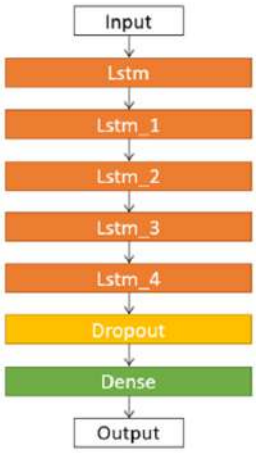
The aim of this first model is to predict the status of the plant (i.e., if the plant is properly working, under some failure for anomalies, and thus failures which may lead to downtimes/stops). This is equivalent to a multi-variate binary classification problem (considering two classes labelled as "Normality" and "Fault" for normal working conditions and failures, respectively), taking into account also temporal dependency. The prediction should be 1 hour in the future, considering the data measured in the previous 20 minutes with respect to the current observation.

Given the requirements, a deep learning model based on LSTM was adopted. The fact that we take only 20 minutes of data with respect to the current is marginally relevant, since LSTM model per se keep tracks the temporal evolution for much larger number of time instants. The model architecture is composed by LSTM layers with a Rectified Linear Unit (ReLU) activation function. During the training and hyperparameter optimization phases, we noticed that adding more LSTM layers improved the quality of prediction results. To this end, 5 LSTM layers have been placed. Since our goal was a binary classification, we used a Dense output layer with a single neuron and a sigmoid activation function. The model is compiled to minimize the log loss (in our case, the binary_crossentropy metric) with an Adam optimizer.

In order to properly represent the dataset for the classification task, and to suitably prepare the input data for the LSTM layers, data have been reshaped into sequences, considering the previous 20 time-steps (i.e., 20 minutes) for each observation, and aiming at predicting the plant status 60 minutes in the future (please note that the 20 minutes are just the time windows at the last time interval while the network has the capability to keep memory of a much longer time interval in its hidden layers). Therefore, the input dataset has been organized as the following time series:

$(X_1, X_2, ..., X_{20})(Y_{80})$

$(X_2, X_3, ..., X_{21})(Y_{81})$



Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 20, 200)	195200
lstm_1 (LSTM)	(None, 20, 200)	320800
lstm_2 (LSTM)	(None, 20, 200)	320800
lstm_3 (LSTM)	(None, 20, 200)	320800
lstm_4 (LSTM)	(None, 100)	120400
dropout (Dropout)	(None, 100)	0
dense (Dense)	(None, 1)	101

Total params: 1,278,101
Trainable params: 1,278,101
Non-trainable params: 0

Figure 3.6: LSTM architecture model.

.....

$(X_n, X_{n+1}, ..., X_{n+19})(Y_{n+79})$

where $X_1, X_2, ..., X_n$ are multi-feature data, and Y_n the plant status measured at temporal instants (representing minutes) $t = 1, t = 2, ..., t = n$. Then, we split our dataset into 70% for train data, 15% for validation data and 15% for test data.
To avoid overfitting, a dropout layer was inserted and an early stopping procedure was defined, monitoring the validation loss. This means that, if after a certain number of training iterations (set to 10 epochs in our case) the validation loss does not decrease, then the training is stopped. In this way, it is possible to prevent overfitting due to over training epochs when the validation loss is no longer improved. Finally, an automated hyperparameters optimization was performed through a Randomized Search Cross-Validation. The best model resulting from the whole process of parameters optimization and cross-validation is represented in Figure 3.6.

The above-described model has been adopted for predictions of the plant working status on unseen test data, with the capability of being executed in real-time on real production data.

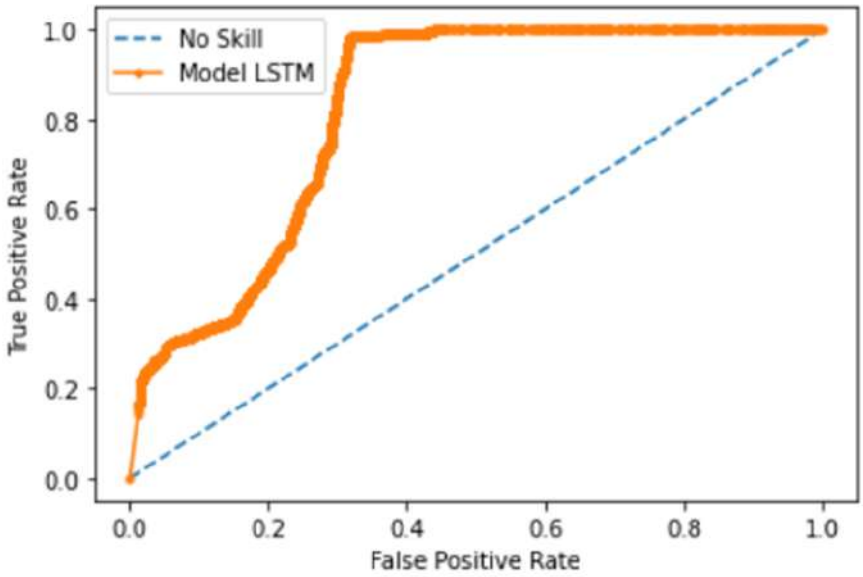


Figure 3.7: ROC curve for the LSTM model.

3.4.3 Validation of the LSTM MODEL

The dataset used for validation is composed of 15% of the total data. Overall accuracy was calculated as the number of 1-minute observations during normal minutes working conditions, plus the number of correctly classified 1-minute observations of failures minutes as a fraction of the total number of minutes 1-minute observations. The resulting accuracy was 87.40%. The ROC curve has been computed and plotted in Figure 3.7. It represents the classification performances, with True Positive Rate (TPR) on y-axis against the False Positive Rate (FPR) on x-axis. Moreover, the associated Area Under Curve (AUC) was calculated. The resulting AUC value is 0.822.

The confusion matrix is reported in Table 3.3.

Predicted Class		

Actual Class	Normality	Fault
Normality	43485	3229
Fault	3246	1436

Table 3.3: Confusion matrix for the LSTM model.

If we consider as class "Normality" and "Fault", recalling the definitions of True Positive (TP) as an outcome where the model correctly predicts the positive class, False Positive (FP) as an outcome where the model incorrectly predicts the positive class, and False Negative (FN) as an outcome where the model incorrectly predicts the negative class, we can compute the following classification metrics:

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$
$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Finally, the model predictive classification performance is reported in Table 3.4, where $weighted\_avg = Support\_Class1 * Score\_Class1 + Support\_Class2 * Score\_Class2$, and $Support$ represents the number of cases inside the test dataset.

	Precision %	Recall %	F1 score %
weighted_avg	0.87	0.87	0.87

Table 3.4: Predictive classification model evaluation results for the LSTM model.

3.5 Advanced Predictive model with CNN-LSTM

With the aim to get more accurate data we tried to reduce the effect of noise by adding a CNN layer to our model. The CNN-LSTM approach provides the advantages of combining CNN powerful feature extraction with the capability of LSTM in capturing temporal dependencies. CNN are actually useful for learning spatial local features from input time series [161], since they have the capability of performing an optimized smoothing of the signal (through the 1D convolutional and pooling layers), while maintaining the underlying data trend. Therefore, they can extract local features of time-series data (including multi-variate time-series) more accurately, thus improving the performances of subsequent LSTM layers in learning temporal dependencies [164].

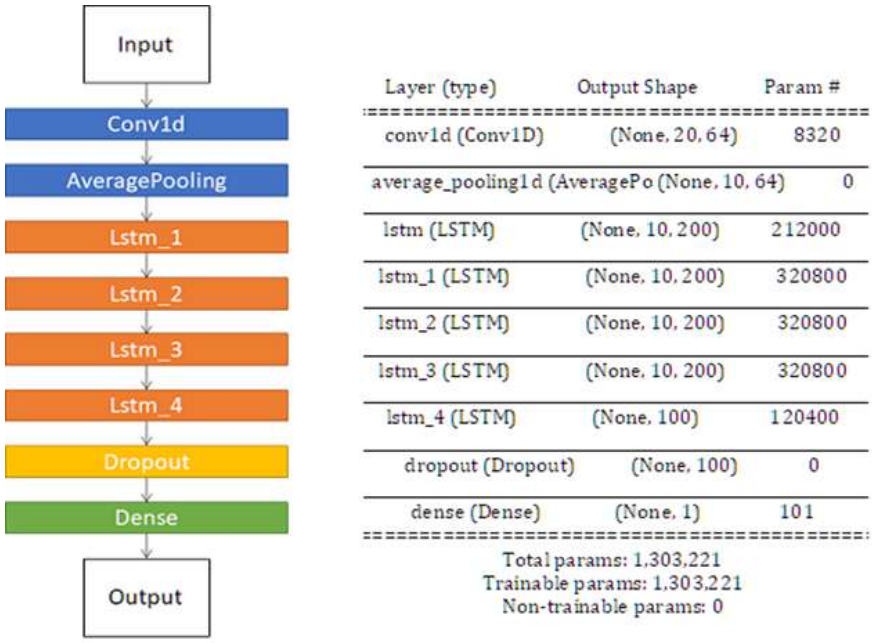


Figure 3.8: CNN-LSTM architecture model.

3.5.1 Classification model CNN-LSTM

In this case, the model architecture is composed by a one dimensional convolutional layer, followed by an Average Pooling layer that computes the average on the output of the previous convolutional layer across all time steps. Then we added LSTM layers with a Rectified Linear Unit (ReLU) activation function. During the training and hyperparameter optimization phases, we noticed that adding more LSTM layers improved the quality of prediction results. To this end, 5 LSTM layers have been placed. Since our goal was a binary classification, we used a Dense output layer with a single neuron and a sigmoid activation function. The model is compiled to minimize the log loss (in our case, the binary_crossentropy metric) with an Adam optimizer. Finally, an automated hyperparameters optimization was performed through a Randomized Search Cross-Validation. The best model resulting from the whole process of parameters optimization and cross-validation is represented in Figure 3.8.

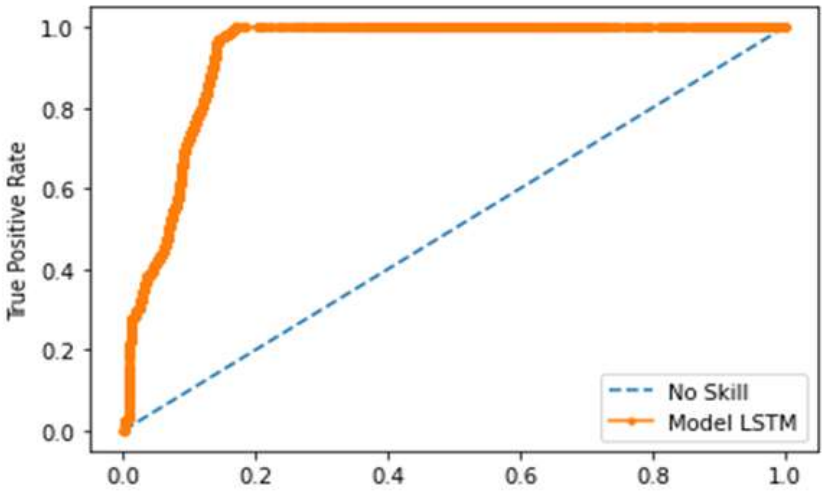


Figure 3.9: ROC curve for the CNN-LSTM model.

3.5.2 Validation of CNN-LSTM model

As before for the LSTM, the dataset used for validation has been composed of 15% of the total data. Overall accuracy was calculated as the number of 1-minute observations during normal minutes working conditions, plus the number of correctly classified 1-minute observations of failures minutes as a fraction of the total number of minutes1-minute observations. The resulting accuracy was 91.81%. The ROC curve is reported in Figure 3.9, and the resulting AUC value is 0.934. Showing in this case better performance of CNN-LSTM with respect to the LSTM.

Predicted Class / **Actual Class**	Normality	Fault
Normality	45811	903
Fault	3306	1376

Table 3.5: Confusion matrix of CNN-LSTM approach.

	Precision %	Recall %	F1 score %
weighted_avg	0.90	0.92	0.90

Table 3.6: Predictive classification model evaluation results for the LSTM model.

The confusion matrix is reported in Table 3.5. And the classification performance are reported in Table 3.6.

3.5.3 Explainable CNN-LSTM to exploit the results

In order to interpret the results, the Shapley additive explanation (SHAP) has been used. Through this analysis, it has been possible to understand how much each feature contributes (positively and negatively) to the prediction of a failure, and therefore it is possible to have both an overview on how to intervene at maintenance level on future failures and ideas to improve our model. The SHAP explanation method computes Shapley values from coalitional game theory. The feature values of a data instance act as a player in a coalition. Explanations obtained by the Deep SHAP method are represented graphically. In Figure 3.10, we have an explanation regarding the failure prediction (see Figure 3.10 a) and the normal operation prediction (see Figure 3.10 b) using our test data set. The SHAP results at each prediction time instant are recorded together with the Boolean output of

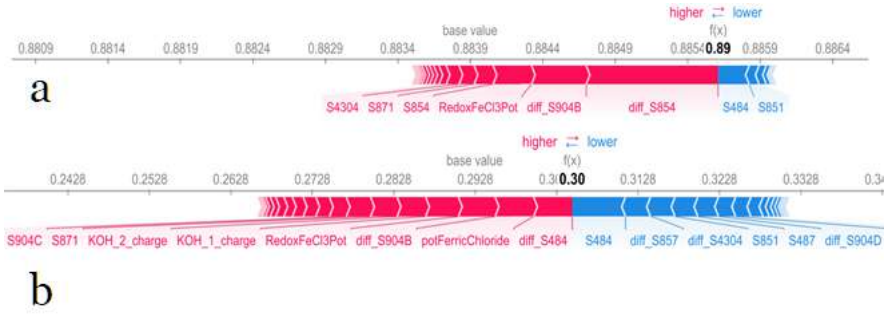


Figure 3.10: Explanation of prediction generated by model for fault (a) and normality (b). Please note that this is an example of a specific event.

the predictor expressing the probability of fault. The trends of the SHAP for each of the variables describe the activation of the network and thus are used to identify which are the causes of the predicted fault when it occurs. Therefore, with tune thresholds with respect to typical trends, a list of critical variables are produced, and these variables correspond to DCS/IoT devices into the plant and thus to sections of the production plant. Thus a maintenance team and the personnel in the control room are informed providing this list and area as a fundamental information at the support of their decision.

Through the image, we can see how different features contribute positively to the failure output (shown to the left of the base value in red) and those that contribute negatively to the failure output (shown to the right of the base value in blue). As we can see in Figure 3.10 a, the most relevant features in the fault determination are the production derived features diff_S854, diff_S904B of two different production lines (i.e., sodium hypochlorite and Potable Ferric std). Other relevant features are the RedoxFeCl3Pot and the charging features. Production anomalies are a warning symptom of a fault. Even for the plant normality situation (see Figure 3.10 b), the charge features and the derived production features are the main contributors to the prediction of our CNN-LSTM model.

With the aim of a better understanding of the obtained model, we compared the above results with a Principal Component Analysis (PCA). Thirteen principal components were extracted from the PCA, explaining a cumulative variance of 77%. We considered those principal components with

eigenvalue greater than 1. Table 3.7 shows the main components obtained with their description.

The characteristics of the derived production variables of the sodium hypochlorite and NAOH KAOH lines and the charge features of the production line are relevant in PCA. We can observe that PCA and SHAP provide quite different outcomes, showing that the deep learning model might be able to find hidden correlations among the data features that can enable and improve fault prediction.

PC	Feature	Description	% variance
PC1	diff_S856, diff_S851, diff_S871, diff_S857, diff_S484, diff_S487	Sodium hypochlorite production correlated with HCl and NAOH KAOH	19.14
PC2	S852, S854, S871, S856, S857, KOH_1_charge, KOH_2_charge, potFerricChloride	Sodium hypochlorite fill level of the tank correlated with HCl and NaOH KOH tank	12.36
PC3	TempreactorR4003, TempreactorR4001, TempreactorR4002, standardFerric Chloride, hyposodium	Reactor temperature indication	8.88
PC4	S851	Sodium hypochlorite fill level	7.91

PC5	RedoxFeCl3Pot ConversionNaOH	Redox potential measurement Ferric chloride correlated with electrolysis load regulation (production)	5.53
PC6	ConversionKOH1, S4304, S487	Electrolysis load	4.04
PC7	diff_S904A, diff_S852	Ferric Chloride std production	3.33
PC8	S904D, S484	Potable Ferric Chloride fill level of the tank correlated with NaOH KOH fill level of the tank	3.29
PC9	QuantNaOHperBatchNaClO, S904C	Flow rate measurement and totalization NaOH KOH	2.81
PC10	diff_S5104, S5104, S904B	NaOH KOH production	2.65
PC11	diff_S904B, diff_S854	Ferric Chloride standard production	2.38
PC12	QuantaNaOHperBatchNaClO_2, S904E	Flow rate measurement and totalization NaOH KOH	2.33

PC13	diff_S904D, S904A	Potable Ferric Chloride production	2.30

Table 3.7: Principal components.

3.6 Final Considerations

In this chapter, a predictive maintenance model for classification of failures in a real industrial process has been presented. The proposed solution is based on a deep learning CNN-LSTM architecture, predicting the working status of the productive process in the Altair chemical plant. The proposed model CNN-LSTM provides a one-hour prediction of the plant status and indications on the areas in which the intervention should be performed by using explainable LSTM technique. Assessing the proposed method with real production data, experimental results show an average Accuracy of 91.8% and an average F1-score of 90%, which are very good results considering that the proposed model provides predictions of the plant working status one hour in the future, and it is capable of running in real time (thus aiming at resolving some lacks found in other state of the art solutions). The explanation of the predictions provides suggestions for the maintenance teams. The chapter also introduced business intelligence tools on maintenance data and the architectural infrastructure for the integration of predictive maintenance approach into the whole control and management system of ALTAIR industry 4.0 plant in the context of SODA and large renovation of the production infrastructure.

Chapter 4

Predicting and Understanding Landslide Events with Explainable AI

Rainfall induced landslide is one of the main geological hazard in Italy. Each year it causes fatalities, casualties and economic and social losses on large populated areas. Accurate short-term and long-term predictions of landslides can be extremely important and useful, in order to both provide local authorities with efficient early warning and increase the resilience to manage emergencies. There is an extensive literature addressing the problem of landslides forecasting, while the most recent solutions are based on machine learning and deep learning approaches. However, some state of the art models are still empirical. They are all based on a wide range of features. These systems typically do not use explainable artificial intelligence techniques to allow understanding better their outcomes and feature relevance. In this chapter, the state of the art on landslide prediction for early warning has been carefully reviewed. In order to find a better solution, a number of machine learning solutions has been implemented and assessed (e.g., random forest (RF), extreme gradient boosting (XGBoost), convolutional neural networks (CNN) and autoencoders (AE)). These models have been trained, validated and compared one another and with the SIGMA approach from the literature. The

validation has been performed in the context of the Metropolitan City of Florence, data from 2013 to 2019. The method based on XGBoost achieved better results, demonstrating that it is the most reliable and robust against false alarms. Finally, we applied explainable artificial intelligence techniques to the XGBoost model (locally and globally), so as to provide a deeper understanding of the predictive model's outputs and feature relevance. Solutions have been implemented on `https://www.snap4city.org/` *infrastructure* [1] [2].

4.1 Introduction

Landslides are increasingly frequent geologic events which may involve rural areas, as well as cities and impact on largely populated areas. Typically, "wake-up call" and early warning systems are setup to inform the population about the occurrence of landslides in quasi real time. On the other hand, even a short term prediction of few hours could save a relevant number of people. Longer term predictions of landslide events, for example 1 day, could be a very powerful tool in the hands of authorities to organize evacuations and manage an emergency since its inception, thus preventing human injuries due to such catastrophic events.

Many studies and researchers too, have proposed solutions on this regard. The most common approaches, as to forecasting landslides over large areas, rely on statistical or empirical approaches, since the use of deterministic approaches need a high number of parameters which are rarely available for areas larger than a single slope or a small river basin.

In particular, as to rainfall induced landslides, in [114] and [101] authors highlighted the correlation of the amount of rain falling in the days preceding the landslide event (from 3 to 245 days), by means of statistical analysis [114], [101], while other scholars used the empirical method of rainfall thresholds to identify rain conditions associated with such landslide triggering [78],

[1] Part of the work presented in this chapter has been submitted and is currently under review as "Predicting and Understanding Landslide Events with Explainable Artificial Intelligence" for *IEEE Access*.

[2] *Acknowledgments:* Our thanks goes to Ente Cassa di Risparmio di Firenze and University of Florence, DISIT Lab of DINFO department and Earth Science Department which funded the research reported in this chapter. Snap4City and Km4City are technologies and infrastructures of DISIT Lab of UNIFI.

[12]. Machine learning approaches are widely used in landslide forecasting [72], but almost only for spatial analyses, while as to temporal forecasting, very few examples exist [61]. The triggering of landslides is caused by the loss of cohesion in the soil, due to its saturation from rainwater or from the raising of groundwater level. This reduction of cohesion leads to the reduction of the shear stress of the slope, therefore restraining the factor of safety. On such reasons the groundwater level (which is, in turn, influenced by rainfalls) is an important factor in the occurrence of landsides [49].

Other relevant factors in slope stability are the type of vegetation, and soil, the slope of the topographic surface, profile curvature, distance from rivers, altitude, and soil landslide critic level (as assessed by experts). They are factors that may influence the stagnation level of rainwater in the soil [95]. Therefore, they influence somehow the consistency of the soil, and thus the groundwater level is an important factor correlated with the land instability and the occurrence of landslide events [12]. In many research works, field data have been the starting point for computing predictions, taking into account databases of registered geological and natural events (e.g., earthquakes, landslides, floods, river or lakes overflows) as reference event values. In most cases, the events have been catalogued by experts according to their severity, depth, size, and persistence over time, and they are typically collected from blogs (RSS, etc.) or web pages [28], [27], recommendation systems of alert (e.g., like the ones from Civil Protection, national institutes of geophysics, etc.), sensor networks, statistical data and annual reports, etc. [46]. Moreover, current studies on landslide identification are based on optical images using pixel-based or object-oriented methods, and the digital terrain model (DTM) is combined with optical images and digital elevation model (DEM) derivatives to identify translational landslide scars using object-oriented methods [159], [142].

The creation of accurate forecasting models useful for early warning activities may be grounded on a wide range of data provided by different sources. This implies to manage a variety of: data licenses, protocols, tools and formats for data retrieval, as well as several standards for data collection and distribution. In addition, a multitude of historical and real-time data must be analyzed, so the data size and their processing speed are considerable. When it comes to the combination of these aspects, we can consider to be in the context of Big Data, for volume, variety, velocity, veracity, and value of data. Moreover, with the aim of producing predictions in a data driven ap-

proach, many different machine learning and deep learning algorithms have been applied in a variety of use cases: Logistic regression (LR), Support vector machine (SVM), Random forest (RF), Boosting, Convolutional neural network (CNN), as stated in [49], [159], [102], [128]. The SIGMA algorithm, which was firstly developed in Emilia Romagna Region [114] and then tested in India [13] is a landslide early warning model based on the analysis of the probability related to exceedance of defined rainfall amounts. The latter has been also used and calibrated in our study area, which is the Province of Florence and then compared with some machine learning algorithms.

In this chapter, the problem of landslide prediction has been addressed, and the state of the art carefully analyzed to draw a comparison on the basis of the same metrics, as much as possible and according to accessible data and information. In addition, we have tested and tuned a number of features and a set of possible machine learning solutions. The identified features have been weather, rain, slope, vegetation, temperature, humidity, wind, soil kind, altitude, etc. The addressed machine learning techniques are: random forest, extreme gradient boosting, convolutional neural network, autoencoder. They have been compared one another, as well as with SIGMA model. Solutions have been trained and validated by using data in the Metropolitan City of Florence from 2013 to 2019. The area is quite prone to landslide events. Finally, techniques of explainable artificial intelligence have been adopted to identify the global relevance of features in predicting landslides, and local relevance to understand relationships among the most relevant features. Thus, producing results to explain the approach and the phenomena. The research activity (named PC4City, civil protection for the city) has been partially funded by Foundation Cassa di Risparmio di Firenze and has been developed in collaboration with the Department of Earth Science of the University of Florence. The solution has been developed exploiting the data available in the area and the smart city infrastructure and living lab named Snap4City: `https://www.snap4city.org`

This chapter is structured as follows. Section 4.3 describes the architecture of PC4City while stressing its relationships with the Snap4City framework adopted in the area. Section 4.4 describes the exploited data and the identified and computed features. In Section 4.5, both adopted machine learning techniques and SIGMA model have been presented along with their running parameters and metrics for result assessment. Section 4.5.2 presents

the results of the validation phase after training. Section 4.6 is focused on the local and global explanation of any best results obtained with the XGBoost method. Consideration are drawn in Section 4.7.

4.2 Related Work

The problem of landslide prediction has been addressed through different approaches and this section presents the state of the art of Artificial Intelligence solutions, as summarized in Table 4.1.

Nam and Wang in [122] used Stacked Autoencoders combined with Random Forest, (RF), for the landslide susceptibility assessment. The areas of study were in Oda and Gotsu Cities in the Shimane Prefecture, in Japan, where 90 landslides occurred due to extreme precipitation from May to October 2013. The data used refer to the Digital Elevation Model (DEM), remote sensing and geological factors. Researchers compared Support Vector Machines (SVM), Stacked Autoencoders (St-AE), Sparse Autoencoders (Sp-AE), and RF classifiers. As a result, they identified the best solution combining St-AE with RF, obtaining a True Positive Rate (TPR) of 0.93.

The Autoencoders have been also used by Huang et al., to predict the landslide susceptibility in the Sinan Country of Guizhou Province in China [84]. In that case, 306 landslide events were registered from the 1980s to 2010s. The data sources for the landslide predictions regarded 27 environmental factors considering: topographic, geological, hydrological, and land covers features. The tested solutions were: SVN, Backpropagation Neural Network (BPNN), and Fully Connected Sparse Autoencoder (FC-SAE). The reported validation metrics have been the True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN), Positive Predictive Rate (PPR), Negative Predictive Rate (NPR), Accuracy. The FC-SAE architecture achieved its best results with an Accuracy of 85.2%, compared to 81.56% for the SVN and 80.86% for the BPNN.

Pham et al., in [133] used a Machine Learning (ML) technique for landslide susceptibility analysis, the Convolutional Neural Network (CNN) with a specific optimization algorithm for parameters selection. The study area of Lai Chau is a mountainous province of Vietnam, the dataset consisted in 2374 points of landslides and randomly selected non landslides with 12 features (Elevation, Aspect, Slope, Stream Power Index (SPI), Compound Topographic Index (CTI), Curvature, NDWI, NDVI, Normalized dif-

ference build-up index NDBI, Distance to river, River Density, Precipitation). The exploited assessment metrics have been the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Area under Receiver Operating Characteristics (AUC), Overall Accuracy (OA). The proposed CNN architecture achieved better results compared to Random Subspace, Random Forest and CNN using conventional Adagrad optimizer, with OA of 80.105%.

The CNN architecture has been also used by Pei et al., in [131]. Their study focused on the influence between time-varying trigger factors and periodic landslide displacement. The specific area of this study is in Zigui, Hubei Province, China. In order to find the best solution for the landslide displacement, researchers compared the 1-D CNN with the SVR. They stated that the 1-D CNN yields to more precise predictions, due to its feature extraction ability, and indeed the results in terms of RMSE/mm and MAE/mm are 9.97 and 8.29, respectively, compared to the 15.35 and 11.14 obtained by the SVR.

In the case study of Karunanayake et al., [96], for the prediction of the landslides riskiness, the implemented ensemble learning techniques (RF) achieved better results compared to deep learning techniques. The work is based on the Badulla and the Nuwara Eliya districts in Sri Lanka. The dataset is made up of 81 landslides registered in each district, including measurements of the current weather and most significant and dynamic geographical conditions of that particular area. As evaluation metric researchers chose the TPR. The RF technique achieved better results compared to the DNN (Deep neural network) with a TPR on the test set of Badulla district of 96.29%, compared to the 92.59% obtained by the DNN, and a TPR of 100% for the Nuwara Eliya district, whereas the DNN correctly classified 26 out of 27 landslides. According to the above summarized TPR percentages, decision tree models outperformed the neural network models.

The ensemble learning techniques have been also used in the work of Chen et al., [52] for the landslide prediction in the area of Tsengwen River Watershed, Central Taiwan. Using optimal hydrological, geological, and topographical variables the RF technique achieved an overall Accuracy of 99,7%.

Researchers stated that, despite different resolutions between ground reference and predicted maps that could determine an exaggeration in the landslide mapping accuracy, the used methods could provide reliable spatial and

quantitative information on landslides.

Wang et al., in [157] compared the SVM Classifier, RF, and the Extreme Gradient Boosting Machines (XGBoost) for the classification of landslide stability. Researchers used the topographic features extracted by the DEM elevation, slope, aspect, curvature and shape. The best classification technique turned out to be the XGBoost, providing an Accuracy of 89% and a Recall of 94%, outperforming the RF (which obtained an Accuracy of 88% and Recall of 91%) and the SVM (which achieved an Accuracy of 76% and Recall of 86%).

The SIGMA model has been used by Abraham et al., [13] in order to forecast landslides in the area of study of the Idukki district in India. Researchers used rainfall data and divided the district of study into 4 reference areas according to the topographic variability and location of rain gauges. The dataset used covers the years from 2009 to 2018 and the last one has been used to validate the SIGMA model. The model obtained a 79.31% mean Accuracy over the four areas.

Authors	Target	Features	Dataset	Model	Results
Nam and Wang, 2020 [122]	Landslide susceptibility	Landslide, Altitude, Slope, Plan curvature, Distance to stream, SPI (Stream Power Index), TWI (Topographic Wetness Index), NDVI (Normalized Difference Vegetation Index), NDWI (Normalized Difference Water Index), Rainfall, Distance to road, Geological age, Lithology	Oda City and Gotsu City in Shimane Prefecture Japan	Stacked Autoencoder combined with Random Forest	St-AE + RF TPR 93.2%
Huang et al., 2020 [84]	Landslide susceptibility	Elevation, Aspect, Plan Curvature, Surface roughness, Surface cutting depth, Slope Form, Geomorphic map, Total surface radiation, Surface temperature, Average annual rainfall, Topographic wetness index, Distance to river, MNDWI (modified normalized difference water index), Population density, Land use types, Distance to road, BSI (bare land soil index), NDBI (normalized difference building index)	Sinan Country of Guizhou Province in China	Fully connected sparse autoencoder neural network	FC-SAE True pos 6177 True neg 5677 False pos 1279 False neg 780 PPR 82.85% NPR 87.92% Accuracy 85.20%

Pham et al., 2020 [133]	Landslide susceptibility	DEM, Aspect, Slope, CTI , SPI , Curvature, NDVI, NDWI, NDBI, Distance to river, River density, Rain, Historical landslides occurrences, Water level, Velocity of the water level, Precipitation, Periodic displacement	Lai Chau province of Vietnam	Convolutional Neural Network with Optimized Moth Flame Algorithm	CNNFMO RMSE 0.3685 MAE 0.2888 AUC 0.889 OA 80.11%
Pei, Meng and Zhu, 2021 [131]	Landslide displacement	Water Level, Velocity of the water, Precipitation, Periodic Displacement	Three Gorges Reservoir area	CNN	CNN RMSE/mm 9.97 MAE/mm 8.29
Karunanayake et al., 2019 [96]	landslide riskiness	Overburden Land use Slope Rainfall	Badulla and Nuwara Eliya districts, Sri Lanka	RF	RF TPR 98.15%
Cheng et al., 2021 [52]	Landslide prediction	LULC (land-use/land-cover) types, Recharge of ground water, Distance to the bank of rivers, Distance to old landslides, Distance to dip slope, Geological line density, Distance to roads, River density, Geological line density, Aspect, Slope, NDVI, Wetness	Tsengwen River Watershed, Central Taiwan	RF	RF OA 99.7% Kappa Coefficient 0.99

Wang et al., 2021 [157]	Landslide stability	Slope, Elevation, Curvature, Aspect	Santai County	XGBoost	XGBoost Accuracy 0.89 Recall 0.94
Ngo et al., 2021 [125]	Landslide susceptibility	Altitude, slope degree, profile curvature, distance to river, aspect, plan curvature, distance to road, distance to fault, rainfall, geology and land-sue	Iran	RNN, CNN	RNN AUC 0.88 MSE 0.007 RMSE 0.083
Abraham et al., 2021 [13]	Landslide prediction	Rainfall data	Idukki, India	SIGMA	SIGMA Accuracy 79.31% Sensitivity 0.88 Specificity 0.79 Likelihood Ratio 5.62%

Table 4.1: Related Works Table.

There is a large literature on landslide prediction, and most recent solutions are adopting machine learning and deep learning approaches. On the other hand, a deep analysis with explainable artificial intelligence techniques has not been performed yet, with the needed attention. Moreover, most techniques available in the state of the art have been assessed on the basis of heterogeneous metrics, so as to understand their precision, accuracy, errors, sensitivity, etc.

4.3 PC4City Architecture

According to the above reported state of the art, some solutions aiming at computing some early warnings have been proposed. Early warning systems can be regarded as 24 hours predictors or early pattern detectors. The complexity, in this case, is mainly due to the heterogeneity of data and the amount of data to be processed in short time.

The solution presented in this chapter is called PC4City, and it has been set up by exploiting the Snap4City architecture and service, which is in place in the Florence/Tuscany area as well as on other regions in Europe [21], [25]. The Snap4City framework (briefly exploited in Figure 4.1, with its application within PC4City project) allows to collect data of any kind, to save them into a big data store where they can be queried for recovering specific historical data segments, and by filtering. The same storage can be used to collect data in real time and to save data analytic results.

The general workflow included activities of:

- **Data ingestion**, historical and real time data to be updated, for example, rainfall, weather, data coming from satellites regarding vegetation, etc.
- **Data set construction** for predictive model training and validation. This activity is preparing the data set for the next step where the predictive model is produced and validated.
- **Predictive Model training and validation**. This activity is focused on producing the Predictive Model (Model Fit) (for example, based on machine learning or other solutions). The produced model is validated in other areas to assess its reliability, sensitivity and robustness.

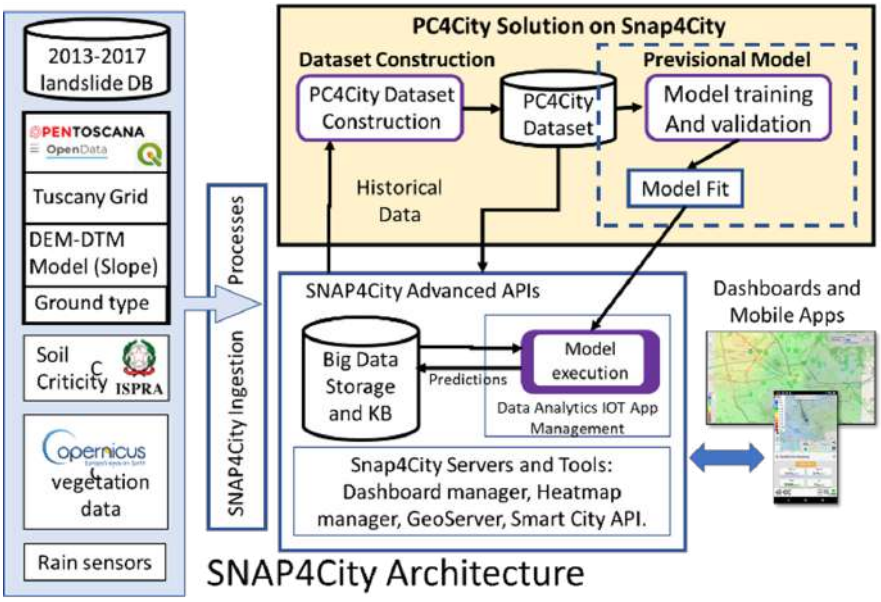


Figure 4.1: PC4City Datasets and solution in the context of Snap4City architecture.

- **Model execution**, takes in input both real time data and the Model Fit to produce predictions which could be estimated 24 hours in advance and may be used to inform civil protection authorities, municipality, etc. The resulting model assesses in real time the probability of landslide events as early warning/prediction.
- **Publication of results** on specific Dashboards, Mobile Apps, etc.

In PC4City, data ingestion processes, as well as activation of data analytics, are performed by using Node-RED processes on docker containers. Node-RED flows can exploit the platform MicroServices with a specific library of node.js [25]. In addition, Data Analytics processes have been developed by using Python and/or Rstudio. In the case of PC4City, some Node-RED IoT Applications have been developed for data ingestion and specific Python processes have been developed for implementing the Predictive Model Training and validation, and for the Model Execution. The IoT App in Node-RED governing the Python for Model Fit Execution may also decide to send alerts via Telegram, SMS, email. Finally, resulting data, as well as previous data, are visually presented by using a Dashboard exploiting the Dashboard Builder.

4.4 Feature and data preparation

In order to test and validate our approach, we have collected a large dataset in Tuscany, in the Florence province (also called Metropolitan City Area) from 2013 to 2019, with the aim of developing and validating a solution for the early warning and 1-day ahead prediction of landslide events. In the observation and analysis area, historical data regarding landslide events have registered 341 landslides from 2013 to 2019 [114]. To each and every landslide event we assigned an ID, the date when that landslide occurred, latitude and longitude expressed in EPSG:4326. Those points are located in their actual coordinates, and for each of them a given number of parameters is accessible such as: wideness, severity, duration, etc.

4.4.1 Grid definition

With the aim of computing a prediction/early warning in each point of the area, a dense grid of points was defined where the prediction could be estimated. The size of the grid is a critical aspect, since the prediction should

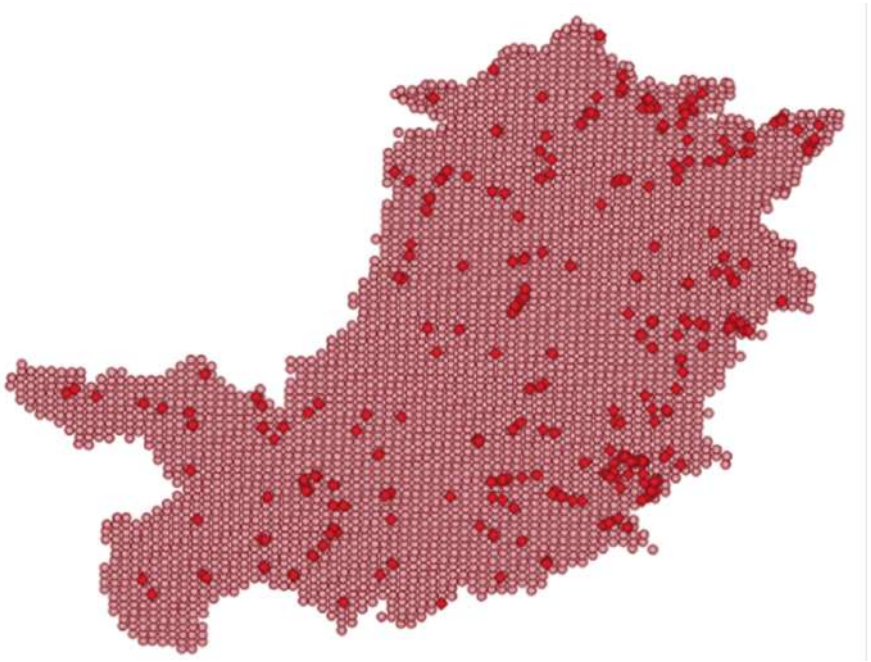


Figure 4.2: Grid and landslide events in the Florence Metro Area (Tuscany, Italy) from 2013 to 2019. An area in which live 1.5 M inhabitants.

be as much precise as possible, while the data would not be accessible with high precision and number of points would be prohibitive for computation.

So that a compromise is needed, the grid size has been defined according to the size of the landslide events, at least $\frac{1}{2}$ to be sure to sample the event. For these reasons, the grid has been defined as a compromise (points distance of 1000 mt in both directions, obtaining 3582 areas, covering the whole Florence Metro area of 3514 Km^2, and a little more at the borders) as depicted in Figure 4.2, in which the RED dots are the events of landslide registered in 2013-2019.

The area presented a large number of landslide events having a relevant range of different features in terms of: criticism, altitude, slope, vegetation, cumulated rain, type of soil, etc. As a result, the set of points in the grid may have a set of associated data that would be taken from: sensors (for example: rain, temperature, humidity, etc.), geographical information systems of the

territory, satellite services, and from the landslide occurred dataset, too.

4.4.2 Feature selection

The features in each area segment of the grid have been selected by analyzing the state of the art in studying landslides and from specific authoritative providers in the area. This allowed us to identify a number of possible features that may influence (and/or may be used as predictors of) landslide events.

One of the most relevant features that influences the landslides is the **soil water content**. These aspects can be directly measured with sensors in the soil, which is unfeasible for large areas and usually rain sensors on ground are not adopted. The same information could be indirectly measured based on the rainfall received in past days. The value of rain in each area of the grid cannot be estimated due to the lack of dense sensors, whereas data coming from satellite are very heavy to be processed and not precise, since also clouds contain water while covering the view of the terrain. On these reasons we decided to indirectly measure the amount of rainfall which reaches the ground from a number of sensors (so called SIR Sensors in Tuscany). The values of sensors have been interpolated by using IDW (Inverse Distance Weighting) algorithm [88], which is also used in Snap4City to create Heatmaps. On the basis of such scattered data, we have estimated 4 derived features: *Day1, Day3, Day5, Day30*, which compute for each day the amount of rain in mm arriving on ground within a specific area in the last day, 3 days, 5 and 30 days, respectively, as performed in SIGMA model [13].

A second parameter which may be related to the landslide proneness may be the **geological nature** and the **terrain slope**. Geology is known to be a controlling factor when it comes to large and deep landslides, while small and shallow landslides ($depth < 2$ m) are somehow independent from the bedrock's geological nature, since they are usually located in more surficial layers of soil. On the other hand, while the terrain slope, which is known as one of the main controlling factors of shallow landslides, may radically change in different parts of the same area. A Digital Terrain Model has been created by processing the Lidar survey carried out in 2017 and available among the Open Data of the Metropolitan City of Florence. The **Slope** feature has been associated to each area of the grid (as a percentage). Please note that these values change sporadically over time. Therefore, an update performed every month/year would be more than enough.

An additional aspect to consider is the **land usage** of the area. For this purpose, land use and land cover datasets of regional government, and in particular Tuscany Region geoserver, provided the data. This allowed to associate a value describing the type of **Ground** to each grid area in terms of identifiers referred to the CORINE Land Cover, CLC technical guidelines [1]. This work has been performed on a QGIS tool. Please note that these values change very slowly in time, and thus they have to be updated once a month or year.

A similar view but for a different purpose has been the identification of the vegetation which may also influence landslide events. **Vegetation** may keep the land connected to the ground. To this end, Copernicus satellite data have been collected exploiting the services of Snap4City Platform (`https://www.snap4city.org/671` [33]) which automatically harvests, downloads and processes several different kinds of Copernicus data. The vegetation level may change over time, and thus the satellite data can give the precise, and almost real time information of the vegetation level. On the other hand, some processing has been made, since the satellite data may be influenced by clouds coverage, and they need also to be remapped from large to small grid areas.

Features have been enriched with some conditioning factors coming from the historical archives of the Regional Hydrological and Geological Sector (SIR). Tuscany region has a network in telemetry consisting of over 700 sensors for meteo-climatic data monitoring; such sensors are located in a homogeneous manner throughout the regional territory. 7 conditioning factors were obtained from these sensors involving wind speed, temperature, precipitation, daily hydrometric level and data providing information related to groundwater resources (water table data). Another feature enrichment was made with data regarding temperature, moisture and average wind speed from the historical archive "ilmeteo.it". Compared to SIR data, ilmeteo.it could provide information associated with larger areas, such as cities (in our case the municipality of Florence).

Regarding the insertion of landslide data, 341 registered landslide events have been mapped over time to the grid, based on their positions and date of occurrence, and they have been labeled with the following criteria: value of 1 has been assigned to all grid cells included in an area of 1.5 km radius, centered on the coordinates of each landslide, in the previous day of its occurrence (for a total of 2342 areas impacted by landslide events); the value

of 0 has been assigned to all other cells. The haversine formula has been used for distance evaluation. Please note that 7 years, multiplied by 365 daily values on 3582 areas compose a dataset of 9.153010 million elements, among which 2342 represent areas affected by landslide events.

At the end of the process, for each grid point, features composing the dataset have been the ones reported in Table 4.2.

Feature	Description	Unit	Example
Date	Observation date, in the format YYYY-MM-DD	Day	2013-01-14
Latitude	Latitude of the area, EPSG:4326 format	Deg	43.86239
Longitude	Longitude of the are in the EPSG:4326 format	Deg	11.51586
Altitude	Altitude of the area	m	467.204
Slope	Acclivity of the area	%	45.942
Vegetation	Vegetation of the area	%	0.262
Ground	Soil type at the event site (class UCS)		223-Oliveti
Day1	Rainfall in the day before the observation	mm	12.453
Day3	Rainfall in the 3 days preceding the observation	mm	15.072
Day15	Rainfall in the 15 days preceding the observation	mm	16.160
Day30	Rainfall in the 30 days preceding the observation	mm	51.515
Temperature	Mean Temperature on the observation day (ilmeteo.it)	°C	6.965
MinTemperature	Minimum temperature on the observation day (ilmeteo.it)	°C	2.99
MaxTemperature	Maximum temperature on the observation day (ilmeteo.it)	°C	9.942

Humidity	Humidity (average) on the observation day (ilmeteo.it)	%	92.96
WindSpeed	Average wind speed on the observation day (ilmeteo.it)	Km/h	5.991
VelMedSIR	Average wind speed on the observation day (SIR)	m/s	0.9
VelMaxSIR	Maximum wind speed on the day of observation (SIR)	m/s	1.8
LevelSIRFre	Phreatimetric data on the observation day (SIR)	m	-4.34
LevelSIRIdr	Water level recorded on the observation day (SIR)	m	0.8
PrecipSIR	Precipitation on the observation day (SIR)	mm	0
MinTempSIR	Minimum temperature on the observation day (SIR)	°C	0.5
MaxTempSIR	Maximum temperature on the observation day (SIR)	°C	3.5
Value	1 if there will be a landslides in the day after the observation, 0 otherwise		1

Table 4.2: Features Details.

4.5 Data analytic solutions

On the basis of the above-described dataset, a number of techniques to predict landslide events has been tested. Aiming at creating an early warning

can be traced back to the estimation of areas presenting a high probability of landslide event occurrence in the next day, as in this case. Therefore, the dataset included several items representing non-slide events (referred hereafter as negative events) and items representing landslide cases (referred hereafter as positive events). As described in the previous section, the considered dataset is composed of about 9 million estimations, among which 2342 positive events (labeled with Value = 1). The input dataset was composed by the following variables:

- X = independent variables = Latitude, Longitude, Altitude, Slope, Vegetation, Day1, Day3, Day15, Day30, Ground, Temperature, MinTemperature, MaxTemperature, Humidity, WindSpeed, VelMedSIR, VelMaxSIR, LevelSIRFre, LevelSIRIdr, PrecipSIR, MinTempSIR, MaxTempSIR
- Y= dependent variable = Value

In order to build the model, we have divided the dataset into two groups: training set (80%) and test set (20%). The selection belonging to the data of the two sets has been performed randomly but considering the same ratio of distributions for both positive and negative cases in training and test sets.

4.5.1 Adopted Machine Learning Models

Most state of the art works addressing the problem of landslide prediction are formulated as a classification problem. As a further development we investigated the possibility of predicting the occurrence of landslides 1-day in advance for this case study in the Florence Metropolitan Area. In this section, the considered machine learning techniques are compared with the aim of predicting landslide events.

Therefore, each model is presented with a short overview and related information about how it has been used in this context.

Random Forest, RF, is a learning algorithm based on a set that includes n collections of uncorrelated decision trees. In our case, the model has been realized exploiting the RandomForestClassifier of the sklearn library. In order to classify the dataset, a high number of trees in the forest has been used (n_estimators = 100), each reaching a maximum depth given by: max_depth = 30. The criterion used to estimate the quality of each division is entropy. Since the input Dataset is unbalanced (in terms of negative and

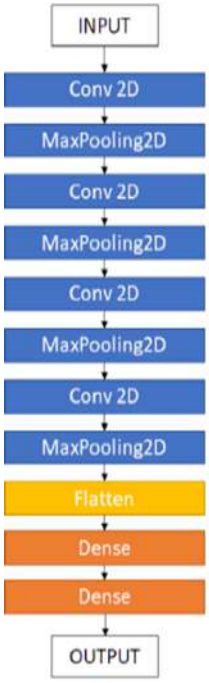
positive events), a weight to the classes in the dataset has been assigned, to give the right meaning to each value (through class_weight).

eXtreme Gradient Boosting, XGBoost, is a specific implementation of the Gradient Boosting method using more accurate approximations to find the best tree model. A high number of trees in the forest has been used for classification (n_estimators=180), each reaching a maximum depth denoted by max_depth=40.

Convolutional Neural Network, CNN, is useful for learning spatial local features from input. It is a feedforward neural network using convolution instead of general matrix multiplication in at least one of its layers. It can capture global and local features with the aim of improving efficiency and accuracy. The model architecture is composed of four pairs of 2-dimensional convolutional layer, Conv2D, followed by a MaxPooling2D layer that downsamples the input along its spatial dimensions (height and width) by taking the maximum value over an input window for each input channel. Then, we added a flatten layer and finally we added 2 Dense layers, the former with 64 neurons and Relu activation function and the latter with a single neuron and a sigmoid activation function. An automated hyperparameters optimization was performed through a Randomized Search Cross-Validation. The best model resulting from the whole parameter optimization process and its related cross-validation is represented in Figure 4.3. The model is compiled to minimize the log loss (in our case, the binary_crossentropy metric) with an Adam optimizer.

Autoencoders represents an unsupervised model generating an output by compressing the input in a space of latent variables. The model architecture is composed of five Dense layers, the first 4 with Relu activation function and the last one with linear activation function. The best model resulting from the whole parameter optimization process and its related cross-validation is represented in Figure 4.4. The model is compiled to minimize the log loss (in our case, the mean_squared_error metric) with an Adam optimizer. The training process of the Autoencoder has been made only on non-landslide data, as it occurs in anomaly detection the typical process is learnt. Then, whenever a landslide event is given in input to the trained model, the reconstructed output is likely not to follow the pattern of a typical process, and therefore it should be classified as an anomaly.

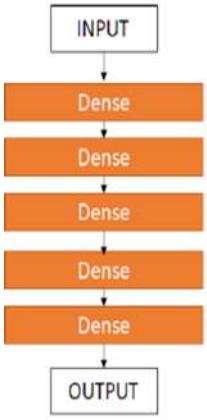
The used Autoencoder reconstruction error has been the MSE and the threshold has been evaluated at 0.4 on the test set, based on the precision



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 1, 22, 16)	32
max_pooling2d (MaxPooling2D)	(None, 1, 22, 16)	0
conv2d_1 (Conv2D)	(None, 1, 22, 32)	544
max_pooling2d_1 (MaxPooling2D)	(None, 1, 22, 32)	0
conv2d_2 (Conv2D)	(None, 1, 22, 64)	2112
max_pooling2d_2 (MaxPooling2D)	(None, 1, 22, 64)	0
conv2d_3 (Conv2D)	(None, 1, 22, 64)	4160
max_pooling2d_3 (MaxPooling2D)	(None, 1, 22, 64)	0
flatten (Flatten)	(None, 1408)	0
dense (Dense)	(None, 64)	90176
dense_1 (Dense)	(None, 1)	65

Total params: 97,089
Trainable params: 97,089
Non-trainable params: 0

Figure 4.3: The adopted CNN model Architecture.



Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	736
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 16)	272
dense_3 (Dense)	(None, 32)	544
dense_4 (Dense)	(None, 22)	726

Total params: 2,806
Trainable params: 2,806
Non-trainable params: 0

Figure 4.4: Autoencoder Architecture Model.

and recall curves reported in Figure 4.5 a. If the reconstruction error is higher than the chosen threshold it will be classified as landslide; this is visible in the reconstruction error for the validation set on Figure 4.5 b.

The decisional algorithm **SIGMA** has been taken into account, too (see Figure 4.6). The Sigma model has been calibrated for the city of Florence area according to the procedure described in [114], [13]. Since it is based on statistical analysis of rainfall data, rain gauges with at least 20 years of rainfall recordings have to be used and 9 rain gauges with the proper data series have been identified in the study area. For each rain station, the cumulative rainfall from 1 to n days is analyzed and the mean rain values and several standard deviation values (from 1 to 3, with steps of 0.5 standard deviation) are calculated. Then several Sigma curves, i.e. curves with the same standard deviation value for several time intervals, are defined (Figure 4.6) a).

Figure 4.6) b reports the flow chart of the Sigma algorithm for early warning. Such scheme compares the cumulative rainfall in the days leading up to the event with a sigma coefficient. In order to make this sigma value more accurate, it was interpolated through the IDW algorithm (same methodology used previously to estimate Day_i cumulative rainfall and described in Section 4.4), at each point in the dataset. In the schema reported in Figure 4.6) b, values of C_1, C_3, C_{15} and C_{30} correspond to the *Day1, Day3, Day15* and *Day30* values in the dataset, respectively, while the sigma symbols stand for standard deviation multiples (expressed in mm of rainfall) that must be exceeded to assign a level of criticality.

One of the possible methodologies to compare the PC4City algorithms described in this section and SIGMA algorithm, consists therefore in examining the same dataset with both algorithms and making some assumptions as reported in Table 4.3.

Sigma *Criticity*	PC4City *Value*
Very High	1
High	0
Moderate	0
Ordinary	0

Table 4.3: Mapping criticality of sigma model vs PC4City

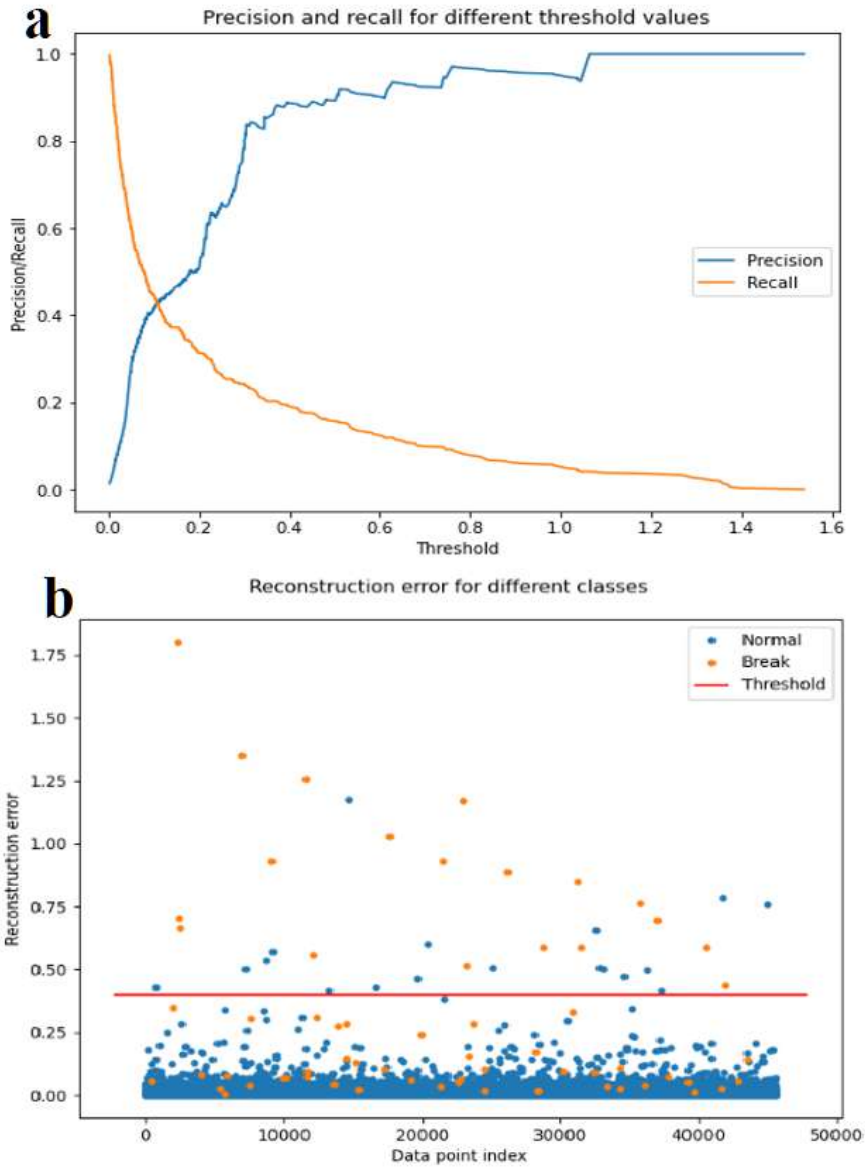


Figure 4.5: Autoencoder Model identified: (a) Precision and Recall Plot - (b) Reconstruction error plot for the validation set.

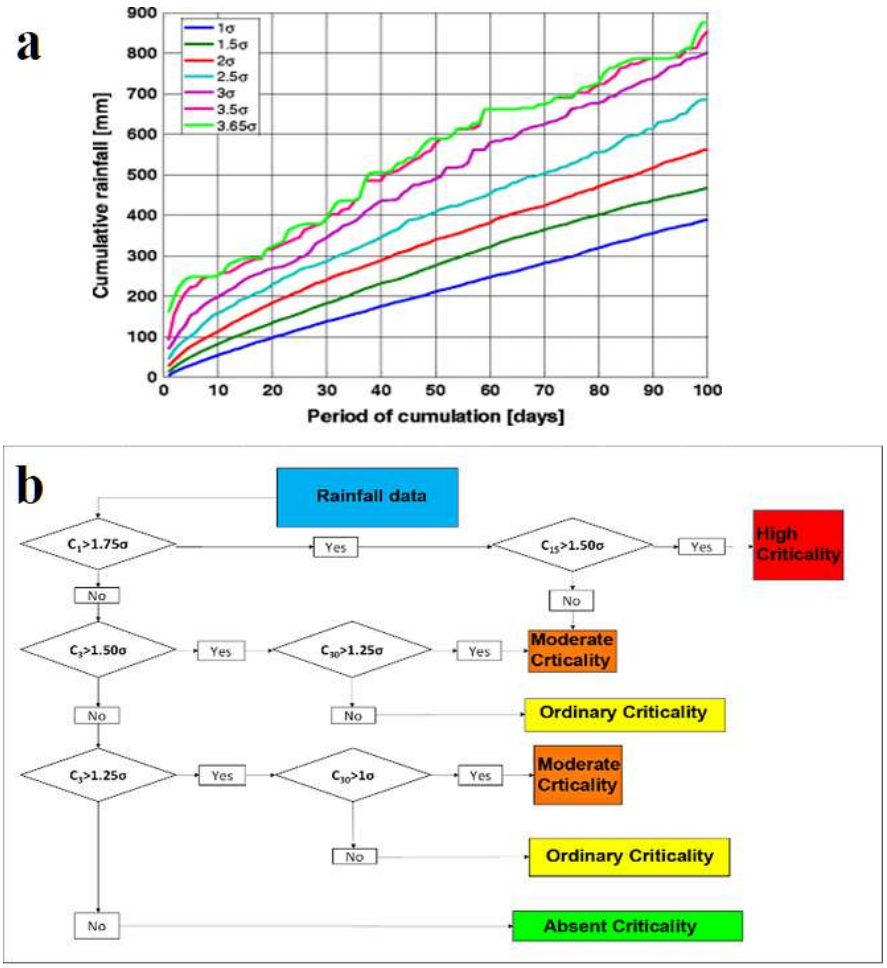


Figure 4.6: Sigma: (a) example of Sigma curves for duration from 1 to 100 days; from [114]; (b) flow chart of the algorithm: C_x represents the cumulative rainfall in x days, while the sigma symbol represents the standard deviation.

4.5.2 Assessment of Results and Best Model Selection

Due to the unbalanced data set, we have balanced the number of landslide cases in training dataset and test dataset in order to improve the RF, CNN and XGBoost classifiers' performance. As to Autoencoder, all points located within a radius of less than 5 km of any landslide have been removed from dataset to prevent a non-landslide point, located in the vicinity of a landslide, from presenting values of conditioning factors extremely similar to those associated with an actual landslide event.

Table 4.4 shows the obtained results for the models RF, XGBoost, CNN and Autoencoders for landslide event predictions, whereas evaluation metrics, as reported in the Table 4.4 are defined as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y_i}|$$

$$\text{Mean Squared Error MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2$$

$$\text{RMSE} = \sqrt{MSE}$$

$$\text{Accuracy} = \frac{TP + TN}{TN + FP + FN + TP}$$

$$\text{Sensitivity} = \frac{TN}{TN + FP}$$

$$\text{probability of false alarm, P.f.a} = P\left(positive|\, negative\right)$$

$$\text{F1} - \text{score} = 2 * \frac{Recall * Precision}{Recall + Precision}$$

$$\text{Matthews correlation coefficient (MCC)} =$$

$$= \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

$$\text{Overall Accuracy(OA)} = \text{Accuracy} + \text{F1} + \text{MCC}$$

$$\text{Kappa Index} = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

AUC : Area Under the Receiver Operating Characteristics (ROC) Curve.

Model	XGBoost	RF	CNN	Autoencoder	Sigma
MAE	**0.000173**	0.000334	0.000600	0.009218	0.004169
MSE	**0.000173**	0.000334	0.000259	0.009218	0.004169
RMSE	**0.0131**	0.0182	0.0160	0.0960	0.064572
Accuracy	0.99	0.99	0.99	0.99	0.99
Sensitivity	**0.79**	0.36	0.24	0.19	0.06
Specificity	0.99	0.99	0.99	0.99	0.99
PfA	**0.01%**	0.02%	**0.01%**	0.11 %	0.39%
Precision	0.63	0.35	0.33	**0.64**	0.003
F1 score	**0.70**	0.36	0.27	0.29	0.007
MCC	**0.70**	0.36	0.28	0.35	0.01
OA	**2.40**	1.72	1.55	1.64	1.02
Kappa	**0.70**	0.36	0.27	0.29	0.01
AUC	0.89	0.68	**0.99**	0.92	0.53

Table 4.4: Prediction Results.

Let's evaluate the MAE metric by comparison. The best model, in this case, is the XGBoost with a MAE of 0.000173, compared to 0.000334 of the RF, 0.0006 of the CNN and 0.009218 of the Autoencoder. To provide a complete representation of results, ROC curves are reported in Figure 4.7, which gives evidence on CNN being better.

In this work, we have compared the architectures used in the state of the art as to landslide metrics evaluation for 1-day ahead landslide prediction in the area of Tuscany, Italy. The XGBoost model achieved better results compared to the Autoencoders, CNN, RF, and SIGMA models. The comparison with the STOA related state of the art works reported in Section 4.2 is difficult to evaluate, because different types of landslide evaluation metrics were

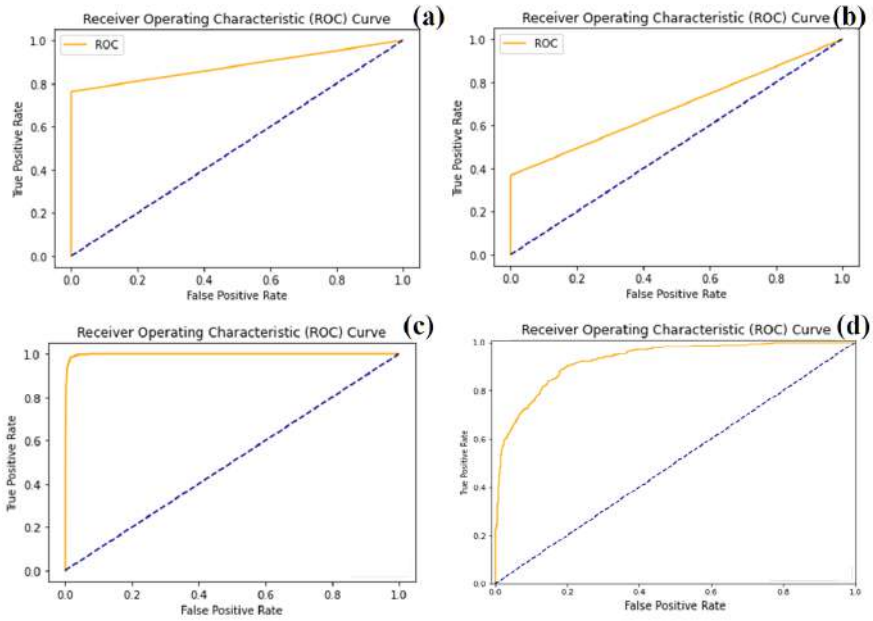


Figure 4.7: ROC Curves for the 1-day ahead landslide prediction, for (a) XGBoost, (b) Random Forest, (c) CNN, and (d) Autoencoder.

used and for the heterogeneous range of case studies such works were based on. If considering the differences with respect to the specific target, we can make a comparison only considering the evaluation metrics independently on the numbers of landslides events.

4.6 Explanation of the predictive model

In order to understand better the relevance of feature and their dependencies and correlation, we have interpreted the values predicted by the XGBoost model via SHAP, both globally and locally. SHAP (SHapley Additive exPlanation) allows to improve of the understanding the predictive model outputs and to explore relationships among individual features for that predicted case [109]. Theoretically, it is an approach from game theory explaining the output of machine learning models. The feature values of a data instance act as players in a coalition. Through SHAP analysis, it is possible to understand which factors are the most influential ones in classifying a landslide or no landslide. This is very useful also for prediction purposes, as in this case. We trained the SHAP explainer with the entire training dataset.

4.6.1 Global XGBoost Model Interpretation

In Figure 4.8, the graph describes the overall impact of features on predictions. The importance of such features is calculated as the average of the absolute Shapley values of the entire dataset. For example, features contributing most to the prediction of a landslide event or its absence are Day3, MaxTempSIR, and LevelSIRdr. Therefore precipitation, temperature and water table data are the main aspects in the prediction of a landslide event. Regarding temperature, localized temperatures (SIR) are more influential than temperatures in Florence (ilmeteo.it). Thus, meteorological phenomena play an important role in the prediction rather than other location-related features. Among variables concerning location, the most influential one is Latitude.

Figure 4.9 shows the distribution of SHAP values for each feature, sorted by relevance. The x-axis represents the specific SHAP value while the y-axis represents features.

Each point represents the samples of our dataset, the color of the point stands for the value of a specific feature, with blue indicating a small value,

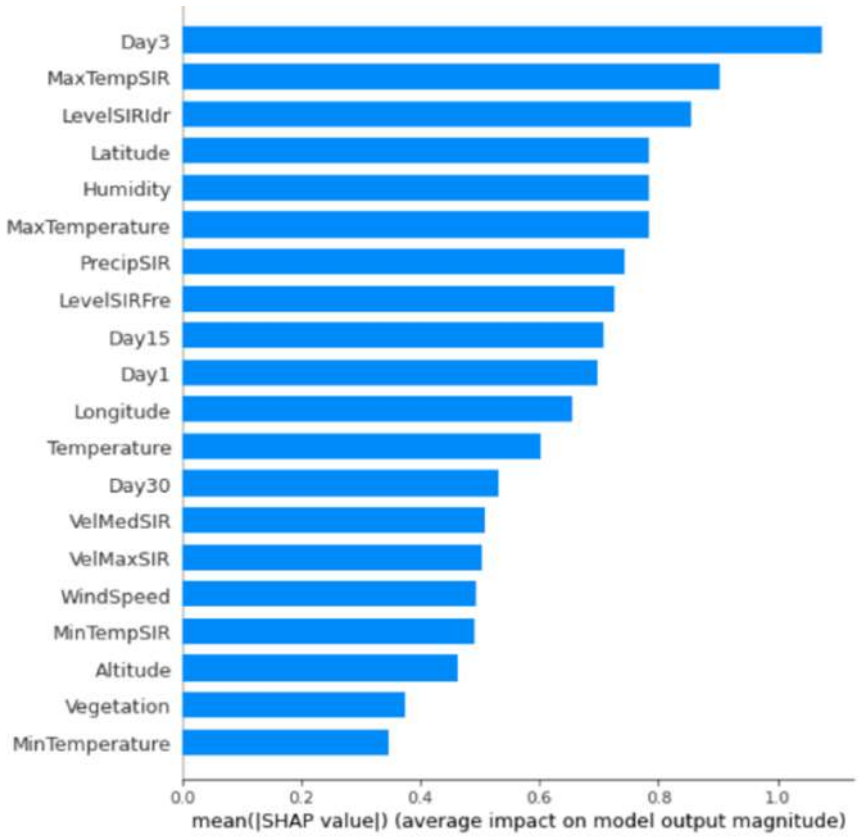


Figure 4.8: Global feature relevance as mean of the absolute SHAP global features importance for XGBoost.

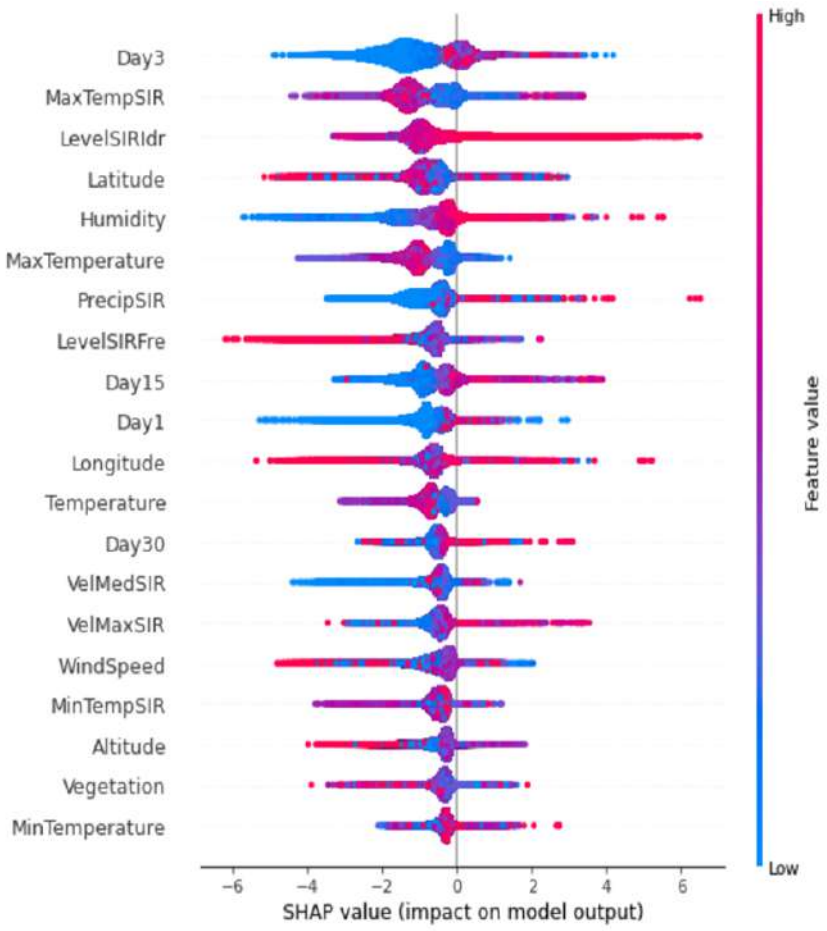


Figure 4.9: SHAP summary plot for XGBoost. x-axis reports the SHAP value of the feature, while on y-axis the features. The color codes the magnitude of the value, and the size the density of values.

while red largest values for that feature. The horizontal position of the point denotes whether the feature value leads to a positive or negative prediction. For example, as to feature LevelSIRdr or Humidity or rain values (Day1, Day3, Day15, Day 30), high values (red dots) contribute positively to the classification of a landslide. We can get a confirmation from the graph that high rainfall values associated with high temperatures and high levels of water within the soil have their main correlation with the prediction of landslide events.

4.6.2 Local XGBoost Model Interpretation

In addition to the global interpretation of the entire data set, each single point and thus prediction can be interpreted locally using SHAP. Figure 4.10 illustrates 3 examples of the local classification of landslide events (a) and (b) and a non-landslide (c). This SHAP plot decomposes final classification into the sum of contributions for input variables highlighting their contributions. The base value, in our case 0.4311, represents the value that would be predicted by the model if there were no knowledge of the features for current output. SHAP values are calculated in log odds. Features which increased prediction value towards a positive classification as landslide events are shown in red on the left, while features which lowered prediction value towards a negative classification are shown in blue. In our case in Figure 4.10 (a) the value of VelMaxSIR, MaxTempSIR, Day3 and Humidity contributed significantly to the classification of the observation as a landslide event. In Figure 4.10 (b), values related to rainfall in the last days, LevelSIRIdr and Humidity gave a relevant contribution to the landslide event prediction. While, in Figure 4.11 (c), values of features: Day3, MaxTempSIR, MaxTemperature, Temperature and LevelSIRdr have been determinant for the classification of the observation into a no landslide event.

4.6.3 Features Dependency

In this section, some features that associate high SHAP values are furtherly analyzed. In order to understand the effect that a single feature has on the output of the model, the SHAP value of the features has been plot against the feature value for all instances of the dataset under consideration. The analysis reported in Figure 4.11 presents the graphs for the most relevant features with respect to feature that has major influence or dynamic with

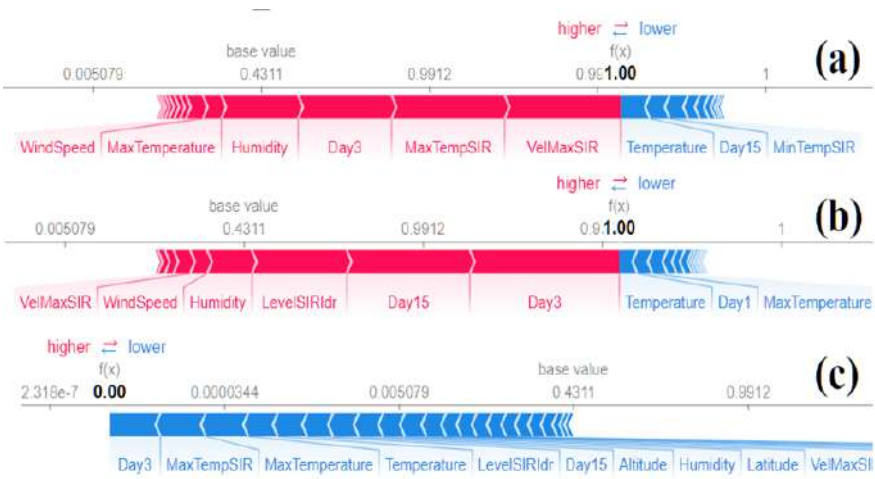


Figure 4.10: Local feature relevance via SHAP, as interpretation of events in terms of feature values: (a) and (b) are events with predictions of landslide, (c) a no landslide event.

them. Each point of the graphs in Figure 4.11 represents an instance of the dataset. On the horizontal line we have the actual value of the selected feature, while the left y-axis presents the SHAP value associated with the feature. When a value along Y is positive the feature contributes positively to the occurrence of landslide event, if negative it favors the classification of the instance as a non-landslide event. The fact that the slope is upward, as in Figure 4.11 (a,b) (where we have high values of variable with high value of SHAP), means that a higher value of the feature leads to a landslide event classification. Thus, high Humidity values or high water levels (LevelSIRIdr) are associated with high SHAP values in predicting landslide events. Regarding the colored bar on the right, this is a reference scale for the values of a correlated second feature, the MaxTemperature. In Figure 4.11 (b), we can see that high temperatures are typically associated with low SHAP values, thus no landslide. While in Figure 4.11 (a), it can be seen that high temperature with high level of humidity may lead to landslide. These graphs lead to immediate interpretation of the model. For example, similar values for a feature, as shown in Figure 4.11 (c), can lead to both positive and negative SHAP values to predict a landslide value. This means

that the mean value of Day30 associated with high temperatures leads to higher SHAP values.

In Figure 4.11 (e), the high values of SHAP correspond to almost any kind of value for Day3. This means that having rain in the previous day is not enough to determine a landslide. While from Figure 4.11 (d) we see high levels of SHAP with low levels of PrecipiSIR, which indicates the amount of rain of the day after. This may lead to confirm that the sliding may occur when the water had the time to penetrate and become stagnant.

4.7 Final Considerations

In this chapter, the problem of landslide event prediction has been addressed, for early warning. A careful review of related works and solutions proposed in literature has been performed, making a comparative analysis of their results, where possible. Traditional approaches are based on empirical algorithm as SIGMA, while most recent state of the art solutions are based on machine learning and deep learning approaches. Their main limitations are represented by the fact that these systems have a low reliability and they do not often provide interpretability of results. They do not apply a specific analysis of predictive outputs and features relevance, based for instance on explainable artificial intelligence techniques. To this purpose, this chapter reports the implementation, tuning and testing of four machine learning methods, based on Random Forest (RF), Extreme Gradient Boosting (XGBoost), Convolutional Neural Networks (CNN) and Autoencoders (AE). These systems have been trained and validated by exploiting data collected in the context of the Metropolitan City of Florence since 2013 up to 2019; they have been compared with SIGMA decisional model, which is currently adopted in both Emilia Romagna and India. Comparative results showed that the method based on XGBoost achieved better results in terms of Sensitivity, MAE, MSE and RMSE. Moreover, a further analysis based on Shapley additive explanation (SHAP) has been carried out, globally and locally, for the XGBoost model which obtained best results. In this way, a deeper understanding of the predictive model outputs, as well as the relevance of features and their interdependency, has been provided.

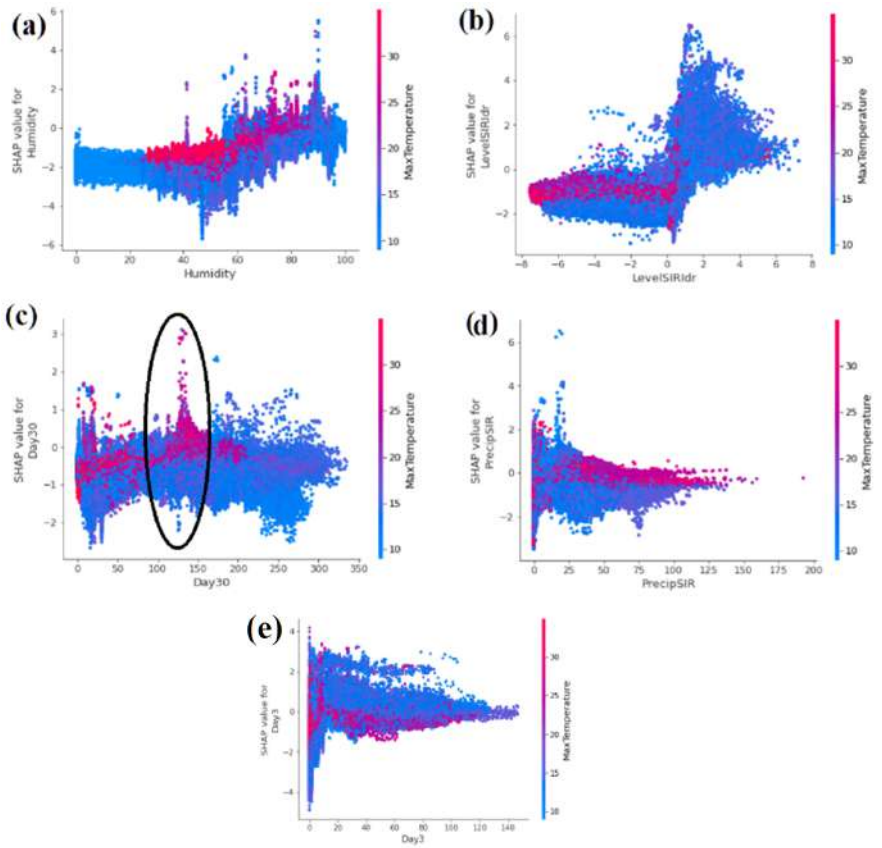


Figure 4.11: Plots of SHAP value wrt to the feature value. Color of dots depends on MaxTemperature feature: color legenda in (e) is valid for all the plots.

Chapter 5

What-If Analysis for Traffic Flow in Smart City Context

What-if analysis solutions have to cope with high complex situations of city scenarios addressing unexpected events, and plans. In this chapter, a solution for what-if analysis has been proposed and validated with the major focus on traffic flow which has a strong impact since most of the simulations in the context of the cities are based traffic flow, including: parking, pollutant, people flow, accidents, commercial sites, tourism, etc. The contributions of the chapter are: (i) the definition and formalization of the what-if analysis framework, including formalization of what-if scenarios with multiple connected areas, (ii) the definition and implementation of large traffic flow reconstruction and simulation against what-if scenarios, (iii) the validation of the traffic flow reconstruction against complex scenarios, (iv) the high performance obtained in the what-if analysis providing traffic flow predictions on large changes on city road traffic. Point (ii) extended the solution for traffic flow reconstruction at the state of the art by (a) dynamically reshaping the road graph network on the basis of the scenarios with multiply connected critical areas, (b) computing multiple reconstructions in consecutive time slots taking into account the evolution of road graph, junction redistribution and of traffic flow data, (c) computing and comparing traffic flow KPI at the support of the what-if analysis, consider-

ing the complexity of their comparison since sensors and roads may be involved into the blocked areas as well. The architecture of the what-if analysis tools has been based on Snap4City framework, in which the business logic is defined by using Node-RED and Snap4City MicroService Libraries. The solution is presently under usage of the Snap4City framework in several cities [1] [2].

5.1 Introduction

Recently, in the context of smart cities, tools for What-If analysis are more often demanded to provide support to decision makers about strategic or tactical plans and/or to provide data evidence support with respect to changes needed to adapt/react with respect to unexpected events in the spirit of city resilience with the aim of maintaining a high quality of service [19]. What-If analysis tools are adopted to perform simulations/plans of complex city subsystems on the basis of specific scenarios/hypotheses (in control rooms and remotely). For example, to identify which are the best solutions when one or more city areas should be closed/modified to (i) study/plan city changes for implementing civil works, or events (sport, markets, manifestations, etc.), as well as to (ii) react to unplanned changes forced by some natural and non-natural disaster (e.g., human factors, terrorists, broken pipe, a cloudburst, a broken metro, unplanned manifestation, etc.). When the closures/changes are planned (case (i)), there is time to identify the best location and how to cope with services. For example, choosing among a limited number of solutions for setting up the open market for the city patron, or for defining the path for the visit of the Pope, or for a large manifestation. On the contrary, in case (ii), the critical area(s) are mainly imposed by the event, thus the rearrangement of the services inside/around the area has to be performed as fast as possible to recover functionalities and to reduce the risk/damage/discomfort to the rest of city users according to the resilience strategies [60],

[1] Part of the work presented in this chapter has been submitted and is currently under review as "What-Iif Analysis for Traffic Flow in Smart City Context" for *FGCS, Future generation of computer systems.*

[2] *Acknowledgments:* Our thanks goes to the MIUR, the University of Florence and the companies involved for co-founding Sii-Mobility national project on smart city mobility and transport. Km4City and Snap4City `https://www.snap4city.org` are open technologies and research of DISIT Lab. Sii-Mobility is grounded and has contributed to the Km4City open solution.

[30]. Probably rescue teams of civil protection, and fire brigades have been already informed. At the same time, the city needs to immediately react by analyzing the context, taking decision for blocking some services, and rerouting others such as: changing the path of public transport, inform private mobility with multimodal information (panels, mobile apps, connected drive messages, etc.), activate maintenance teams that have to access the location in a safe manner, taking into account the main services as drinkable water, gas, and energy, and thus understanding how they can be reactivated and for which part of the area and city. Thus, a set of integrated analyses, simulations and predictions should be performed in quasi real-time before taking decisions.

The closure of a part of the city is an extreme case, while less critical situations such as changing road direction, closing a single road, may be more frequent. In case of unplanned events, the number of possible cases may be very high, since they may be vary in terms of locations and entities, and impact on the city services (the first actions could be the evacuation of people involved by rescue teams, to secure the area, as well as to inform the neighboring population to let them avoid moving in the direction of the critical area and let them be informed of the actual situation for their parents, etc.). This large range of changes made almost impossible to train Artificial Intelligence, AI, systems, and thus is very complex to make predictions. Ideally, experts could foresee the occurrence of a certain event, while other details are hard to be estimated in most cases and they may be somehow unexpected, such as precise date and time, the relevance, the location, the context, the reactions of city users, the cascade effects on multiple services, etc. They are the so called unexpected unknowns [127], when the model is changing at any event occurrence the long-terms prediction is almost infeasible. On these cases, a relevant number of simulations should be performed to select the best solution among the possible identified on the basis of KPIs, Key Performance Indicators, for each objective. In most cases, the possible scenarios to be compared are not infinite, while the number of aspects and the variables to be taken into account may be huge and not easy to be manually inspected.

Therefore, What-If analysis can be defined as a data intensive simulation with the aim to analyze the behavior of a complex cyberphysical system under some given scenarios and hypothesis [73], [29]. What-If analysis should not be confused with sensitivity analysis which aims at evaluating how sensi-

tive is the behavior of the system/subsystem with respect to small changes of one or more parameters. Moreover, there is an important difference between What-If analysis and predictions. Predictions are produced by extrapolating short/long-term trends from the historical time series and may take into account a number of features [148]. The approaches for predictions may be based on AI, machine learning provided that enough data are accessible for training the model, and also the simulation of data for machine learning training may be an option since the context is unknown. Then, What-If analysis requires to have an input context which describes the scenario in the future (and in this sense it may exploit predictions), thus the What-If analysis should simulate the complex outcomes and phenomena. For example, one may predict the temperature 6 months in advance with a certain approximation, but to be precise on predicting the eventual occurrence and damages of a large storm would need to pose a set of additional hypotheses, thus creating a larger space of possible solutions and scenarios. Then, this may be solved by performing a simulation with a certain margin of error, and it is impossible to be automatically computed only by AI, due to the lack of data for their training. Thus, the direct prediction of the outcome could not be possible since the detailed scenarios may be typically unknown. Then, only a simulation of the subsystem, or a mixed solution (AI + simulation), may reproduce with satisfactory approximation reliable results, when the subsystems to be simulated is not trivial, for example when certain phenomena are at our best modeled by complex differential equations (such as in the case of traffic, diffusion processes, etc.), and in any case the data for training are not accessible by monitoring the physical system in the unknown conditions. Therefore, the design of an effective What-If application is more complex than making predictions by AI or any other techniques. In addition, when the space of input parameters can be automatically varied around reference values for each simulated/predicted scenario, the sensitive analysis could be also automatically computed.

Therefore, What-If analysis involves (i) a scenario with a set of input parameters on space (e.g., a number of city areas in which the traffic has to be locked or changed) and time which in some cases may be chosen by the user; (ii) one or more simulations of different subsystems to take into account scenario and parameters to produce an outcome for each time slot of the scenario; and (iii) a set of goals according to which the outcomes could be assessed with the aim of making a decision. For example, what happen if

a city is closing a number of areas in city downtown, for about 5000m2, for 4 days next month for a fair? The fact may impact on private traffic, on public transportation, on parking slots on the roads and on silos, on sharing economy, on pollutant production, on pedestrian movements, and on commercial services, etc. Which is the best combination of closed areas to avoid creating critical traffic conditions, minimizing the public transportation changes, still preserving the capabilities of rescue teams to guarantee the security, and creating a good experience for the attendees?
The identification of the possible solutions for each service can be performed on the basis of one or more objective KPIs. Thus, a decision support system based on What-If analysis could help to ground a data driven decision, thus reducing the risk. The best solution would probably be a compromise to minimize the dysfunction for the population, minimize the risk (which is the damage costs for recovering), reducing the damage to commercial activities, etc. The best criteria to be applied may be different for planned/unplanned events and thus they may also depend on the location in which the event occurred and on the services which are involved in the closures.

One of the most complex city services to be simulated is the traffic [63], from which almost all physical activities depend: commerce, tourism, education, etc. Therefore, in the case of a What-If analysis tool to cope with traffic, the decision should be focused on identifying the solutions which minimize the traffic dysfunction, reducing/limiting the crowding conditions as much as possible. When a city area is closed the simulation of traffic can be performed on the basis of historical data of traffic flow, traffic predictions, origin destination matrices, typical time trends of traffic flow on sensors, and/or by taking into account the typical paths of the city users in the city. On the other hand, small scale simulations or predictions (covering few roads or single sensors) may fail in providing a solution, since typical cascade effect may occur provoking consequences in many other areas. For example, blocking a part of the city may put in crisis other parts of the city, apparently unconnected to the former area. In general sense, the whole amount of vehicles entering and exiting the city over time would not change [40], while the internal traffic could lead to a radical change of behavior. Moreover, in case of unplanned events, a certain inertia of the traffic could cause large problems, and also city users that could be far from the event and could be involved in short time in some dysfunctions should be informed as soon as possible (or in advance for planned events) to give them the time to resched-

ule their trip, thus avoiding to travel towards the blocked areas. Therefore, when the problems are solved and the normal viability reestablished, the traffic would take a while for returning to regular conditions.

In this chapter, the What-If analysis paradigm and architecture are formalized to cope with the complexity of city scenarios addressing unexpected and planned events. The validation of the approach has been performed with the major focus on traffic flow What-If analysis, while other simulations can be easily added. As described above the What-If analysis implies the formal definition of scenarios, the estimation of predictions to be used in the simulations and the production of integrated simulations and KPI which can allow to provide support for decision makers. As put in evidence in the sequel, the reconstruction of the traffic flow is at the basis of most of the What-If analyses for parking, traffic, pollutant, people flow, etc. In this context, the contributions of this chapter are: (i) the definition and formalization of the What-If analysis framework, including formalization of What-If scenarios; (ii) the definition and implementation of large traffic flow reconstruction and simulation against What-If scenarios (addressing the changes into the road graphs as well as on the number of the operating sensors, and automatically estimating the redistribution of traffic on modified crossroads); (iii) the validation of the traffic flow reconstruction against complex What-If scenarios obtaining high precision; (iv) the high performance obtained in the What-If analysis providing traffic flow predictions on large changes on city road traffic. Point (ii) extended the solution for traffic flow reconstruction proposed in [40] by (a) dynamically reshaping the road graph network on the basis of the scenarios with multiply connected critical areas, (b) computing multiple reconstructions in consecutive time slots taking into account the evolution of road graph, junction redistribution and traffic flow data, (c) computing and comparing traffic flow KPI at the support of the What-If analysis. The architecture of the What-If analysis tools has been based on Snap4City framework, in which the business logic is defined by using Node-RED and Snap4City MicroService Libraries [25]. The solution is presently under usage of the Snap4City framework in several cities. The chapter is structured as follows. In Section 5.3, the formalization of the What-If analysis solution is provided, together with a summary of techniques for computing very long-term predictions, and road graph network modeling and restructuring. Section 5.4 presents the algorithm and solution for What-If analysis of traffic flow, providing details on traffic flow KPI on which the decision

can be made. In Section 5.5, the validation of the solution with respect to simple cases and actual data is reported. Section 5.6 provides the validation of the What-If analysis tool when multiply connected area are blocked. The validations have been based on KPI which took into account the traffic flow distribution according to the fundamental curve of traffic, of the road traffic features as betweenness, centrality, eccentricity, and of road saturation. In Section 5.7, the performance analysis is reported providing the evidence that the approach taken permits to perform quasi real-time What-If analysis with respect to the state of the art. Consideration are drawn in Section 5.8.

5.2 Related Work

According to the above discussion, the related work in this area also involves the following aspects:

- **definition of scenarios and contextual data**. For unplanned events the short terms predictions can be sufficient (1 hour), while for planned events, a very long-term prediction of traffic flow (6/12 months for example) could be needed;
- **conditional routing** (routing taking into account scenarios, limitations, traffic, etc.), which could be an element of the computation;
- computing/simulation of **traffic flow reconstruction** in the whole city taking into account different constraints and limitations;
- **computing KPI** to assess the city and/or traffic conditions in objective manner, and thus its evolution in the time span of the scenario;
- **architectural aspects of a What-If analysis** tool which should be capable to take into account multiple simulations in addition to traffic flow reconstruction, such as to assess/simulate the impact of changes of traffic and routing on: public transport, parking, people flow, pollutant, etc.

In order to provide a formal definition of scenarios to which we will refer in the following, a What-If scenario is a simply or multiply connected area on a city map, representing spatial blocking constraints which interdict specific kinds of traffic (private, public transportation, cycling, pedestrian, etc.) according to the road graph in the areas. A scenario typically presents

data/time range in which the constraints are active. In the context of ICT solutions and frameworks addressing the aspects of What-If analysis for traffic simulation and management, a crucial role is played by the capability of the system in performing simulations dynamically, in real-time or near real-time, in order to cope with unexpected or planned events which may change the regular traffic viability (e.g., vehicle accidents, natural, weather related and non-natural disasters, other events such as public manifestations, evacuation plans etc.). These aspects can be suitably modelled by the above defined What-If scenarios, which can well describe and represent the complexity of multiple geographical and temporal constraints. However, current state of the art solutions for traffic What-If analysis typically seems to perform simulations only at level of single elements of the road graph, in cases of general road blocking conditions [67], road bottlenecks and congestion, as well as disaster management [80], [132]. Traffic conditional routing in vehicular networks has been addressed in literature, as a measure for mitigating as well as preventing traffic congestions. Several strategies have been proposed, such as Dynamic Shortest Path (DSP), Random k Shortest Paths (RkSP) [38] and Entropy Balanced k Shortest Paths (EBkSP) [132]. Some works address also the traffic routing problem due to natural disasters, such as in [17], where a traffic rerouting system is presented exploiting Bayesan Networks Analysis to provide possible reroute paths to avoid flooded areas. However, to our knowledge, proposed methods for traffic routing are typically based on traffic congestion, without taking into account the possibility to manage whole blocked areas in which the vehicular traffic is interdicted.

The Traffic Flow Reconstruction is the process to produce a value of traffic density (flow) - e.g., vehicle per meter (vehicles per minute) - for each road (or road segment, or a large number of road segments) by starting from a limited number of traffic sensors measuring traffic density (flow) in the road. The measures of traffic density are typically obtained by stationary sensors on strategic positions. The current literature is quite poor about What-If analysis related to traffic flow reconstruction algorithms in large scale which may provide responses in real-time and for long terms analysis, for example months in advance. In the most of cases, the studies concerning traffic flow algorithms are focused on the mathematical aspects concerning the solution at the junctions in the theory of LWR (Lighthill-Whitham-Richards) model [43], [152] and the analysis of the error in the related traffic estimation. For example, at the intersections and/or on the roads far from the inter-

section when vehicles are moving with a more stable velocity. The problem of the traffic flow reconstruction could be regarded as the solution of the LWR model [107], [138], which is modeling the traffic density in terms of Partial Differential Equation (PDE). The solution of the LWR model is not a trivial matter for large networks due to its computational complexity and constraints [43], [71], [91]. In alternative, the traffic flow reconstruction can be performed by using agent-based solutions which are typically more problematic to scale since they need a specific process for each agent/vehicle. In literature, many studies concerning What-If analysis' concepts in Intelligent Transportation Systems, ITS, have focused on routing [118] and [77], signal control systems [103] and [45], and autonomous driving [140]. In [112], the strategies for dynamic adaptive transport policies are reviewed and formalized, putting in evidence the needs of continuous improvement and simulation. The literature also presents a set of traffic or city simulators which at the first glance could be used for implementing What-If analysis solutions. CitySymXML allows to simulate the energy condition of buildings taking into account weather, geographic information, and building details [149]. DEUS [134], a Discrete-Event Universal Simulator used to simulate a Vehicular Ad-Hoc Network (VANET) [59]. VANET has been also used with SUMO (Simulation of Urban Mobility, `http://sumo.dlr.de`) which can create microsimulations of traffic (for example, few crossroads) providing all the information regarding crossroad distribution of vehicles. MATSim (Multi-Agent Transport Simulation, `https://www.matsim.org/`) is an agent-based simulator which can cope with a limited number of agents/vehicles [155]. Most of these simulators are not suitable for large scale analysis of changes to understand how they impact in the whole city. A large part of them can import the structure of the city by starting from Open Street Map (OSM), while at the end they are capable to process a limited number of agents/vehicles, and the model can be limited to small size of the whole road graph. InterSCSimulator [141] is also an agent-based simulator which may scale up to large roads and cases at the expense of memory and computational time, for instance 51 Gbyte of memory for 50.000 nodes, 22 minutes of computation. Also, commercial solution such as OPTIMA proposed by PTV has a limited capability in forecasting traffic behaviour at 60 minutes (`https://www.ptvgroup.com/en/solutions/products/`).

According to the What-If scenarios, the road network can be strongly modified. A set of KPIs has to be computed for: (i) the same date and

time without scenarios changes (that would be simple forecast), (ii) the case in which the scenarios changes are applied (that would be a forecast in the new conditions). The difference among the values in computed KPI would be used to make the decision. The traffic flow is at the basis of many predictions and simulations (as described in the sequel), so that the network analysis can help to describe the topological features of each scenario, since the related road network is modelled as a directed graph. And thus, a new graph implies a different redistribution of traffic. For example, in [40] is showed that the highest value for betweenness [154] is located in proximity of one of the typical areas where traffic congestion often occurs. In [66] it is proven that the nodes characterized by higher values of betweenness potentially represent critical regions (intersections) of the network, since they correspond to the nodes where traffic can be expected to most-likely concentrate in the near future by computing betweenness over a time-varying weighted graph. On the other hand, nodes having high eccentricity [154] are located in the decentralized zones of the urban graph admitting more distance from the other side of the network. However, such studies assume that the suitable data structure for the related urban network is unchanging, that is, a given road graph (or a given data sub-structure) is also considered in the computation when the access is not permitted in the next future, for example in the case of roads maintenance, local events, etc.
In [64] some integrated performance indicators in urban road infrastructure are also developed for evaluating network functionality and the impact of transport system interventions. To determine a value of traffic congestion in the network, the most informative metric seems to be the so called average degree of saturation. At the present, there is no unified and standard evaluation measure for traffic level conditions. Traditional indicators used to measure traffic congestion include various delays, saturation flow rate, average travel time in specific cases [111]. In many studies, the urban traffic state is explored in different ways. For example, in [145] travel speed and travel time are directly obtained through the loop detector, GPS, video, etc. In [82] the traffic volume and occupancy are integrated to form a new value for network-wide traffic states observation and analysis via pseudo-color maps. In [160], a comprehensive traffic state estimator derived from traffic flow variables (flows, mean speeds, and densities) is presented.

The architectural aspects of a What-If analysis tool should be also addressed [130]. It has to be capable to take into account multiple simulations,

and probably most of them are going to exploit traffic flow reconstruction in the modified context to assess/simulate the impact of changes of traffic on: public transport, parking, people flow, commercial areas, etc. Some developers of custom What-If analysis complain about the lack of formalism to facilitate the setup of the solutions and compute the several simulations on the basis of the same scenario [73]. The solution has to be capable to address, static, historical, real-time/dynamic, and forecasting information, in a functional model, on which the processes (simulations, predictions, data transformations) can be integrated with business logic of the visualization and user interaction. Despite the large literature of What-If analysis and its complexity for managing multiple simulations on the basis of progressively computed results, there is no a common methodology and open tool to be applied for merging different simulations according to the scenario. In fact, the simulation for people flow, pollutant propagation and parking may depend on the estimation of traffic flow reconstruction in a completely different context with respect to the actual. So that the classic prediction models cannot be used. Most of the solutions have limited performance to use the What-If analysis to cope with unplanned events that have to be managed in short time. For these reasons the following aspects become relevant: the formalization of the scenarios, the definition of a flexible and scalable infrastructure for What-If analysis, supporting KPI computation, and the definition of custom business logic for harmonizing the different simulations with computing tools (in container and parallel processing), and permitting the scripting of the user interface business logic.

In Table 5.1, a comparative overview of the reviewed methods for What-If analysis in literature is reported.

Paper	Aspects covered	Case	Techniques
[67]	simulations only at level of single elements of the road graph	general road blocking conditions	
[80]	simulations only at level of single elements of the road graph	road bottlenecks and congestion, disaster management	

[132]	simulations only at level of single elements of the road graph	road bottlenecks and congestion, disaster management	
[38]	Traffic conditional routing in vehicular networks	measure for mitigating as well as preventing traffic congestions	Dynamic Shortest Path (DSP), Random k Shortest Paths (RkSP)
[132]	Traffic conditional routing in vehicular networks	measure for mitigating as well as preventing traffic congestions	Entropy Balanced k Shortest Paths (EBkSP)
[17]	Traffic conditional routing in vehicular networks	measure for mitigating as well as preventing traffic congestions	Bayesan Networks Analysis
[107], [138]	Traffic Flow Reconstruction	modeling the traffic density in terms of Partial Differential Equation (PDE)	LWR (Lighthill-Whitham-Richards)
[118], [77]	What-If analysis' concepts	focused on routing signal control systems	
[140]	What-If analysis' concepts	autonomous driving	
[112]	What-If analysis' concepts	strategies for dynamic adaptive transport policies	
[149]	traffic or city simulators	simulate the energy condition of buildings taking into account weather, geographic information, and building details	DEUS [134], a Discrete-Event Universal Simulator used to simulate a Vehicular Ad-Hoc Network (VANET) [59].

[155]	Traffic Flow Reconstruction Multi-Agent Transport Simulation	agent-based simulator	
[64]	performance indicators in urban road infrastructure	evaluating network functionality and the impact of transport system interventions.	
[111]	Traffic Flow Reconstruction	Traditional indicators used to measure traffic congestion include various delays, saturation flow rate, average travel time in specific cases	
[160]	Traffic Flow Reconstruction	estimator derived from traffic flow variables (flows, mean speeds, and densities)	
[130]	architectural aspects	multiple simulations: public transport, parking, people flow, commercial areas, etc.	

Table 5.1: Comparative overview of the main related works for What-If analysis.

5.3 Design of What-If Solutions

According to the above described What-If analysis concepts, Table 5.2 summarizes what can be obtained according to a list of incrementally capable approaches and data. Cases 1 and 2 describe which kind of answers can be provided by using historical and real-time data, respectively. Case 3 refers to

short terms predictions which can be obtained exploiting the historical and real-time data with some predictive technique (STP, short term prediction). They may be used to produce answers to questions such as "What is going to happen?". The usage of some Analytical Model and Scenarios Modeling can allow to perform some short range simulations, answering to "what is going to happen if a scenario occurs in the future?" (Case 4). To perform long-term What-If analysis on complex scenarios, more powerful predictions and simulations are needed as in Case 5. On the other hand, the actual usage of the technique may lead to assess the impact of the cases using KPI, Case 6.

Available data and techniques	What happened	What is going on now	What is going to happen	What-If: what is going to happen if a scenario occurs in the future	Which impact will be provoked
Historical Data, HD	Yes	No	No	No	No
Real Time Data, RTD	No	Yes	No	No	No
HD + RT + Short term Predictions, STP	Yes	Yes	Yes	No	No
HD + RT + Analytical Model, AM + Scenario Model, SM	Yes	Yes	Yes	(Yes)	No
HD + RT + Short and Very Long Term Predictions, SLTP + AM + SM + Simulation, S	Yes	Yes	Yes	Yes	No
HD + RT + SVLTP + AM + SM + S + KPI based Decision	Yes	Yes	Yes	Yes	Yes

Table 5.2: What can be obtained according to the available data and techniques, as a list of incrementally capable approaches.

According to Table 5.2 for What-If analysis, predictions and simulations are needed, and, from the literature, it is evident that historical and real-time data are needed for their computation. Thus, a complex network of dependencies among data, data analytics and simulation tools is created to implement the above cases. For example, if we consider to perform What-If analysis in a city scenario blocking a couple of areas for 3 days 6 months ahead, in order to answer to what is going to happen to traffic, people, parking, pollutant, and public means, the following aspects should be addressed:

- Formal **Definition of the scenario** that may cover areas of parking, traffic and public means, etc., on specific days, and at specific time slots;
- **Predicting the traffic flow** 6 months ahead in regular conditions, in all the city and thus also at the border of the scenario areas, which may include also traffic flow sensors, unfortunately;
- **Reconstructing/simulating the traffic flow** 6 months ahead in the whole city to understand what is going to happen as impact of changes due to the scenario, according to the modified road graph as defined in the scenario in the considered time slots;
- **Simulating/computing conditions of the public transport** in the changed conditions of traffic and road graph. It may lead to move bus stops, change rides, add/change lines, etc. This can be addressed, for example, by using a simulation of public transportation [20];
- **Simulating/computing conditions of the parking** lots 6 months in advance taking into account the new condition of traffic. This can be achieved, for example, by using a simulation of parking on the basis of traffic, day, weather, etc. [26];
- **Simulating/computing pollutant** changes on the basis of the traffic flow (for NO2, CO2, for example), etc. [135];
- **computing KPI** addressing all the above aspects and their impact on the rest of city services, with the aim of providing objective indicators to choose the best compromise to reduce city user dysfunction, traffic conditions, costs, preserving security, quality of service, still enabling a routing for emergency vehicles, etc.

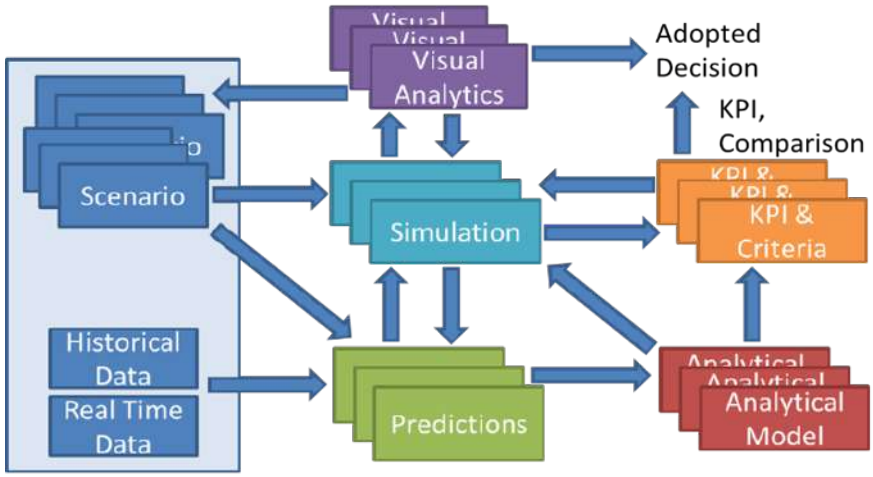


Figure 5.1: Functional architecture of the What-If analysis solutions, actually implemented as an instance of the Snap4City framework.

5.3.1 Architecture for What-If Analysis

In order to cope with the above described complexity, a flexible What-If analysis architecture has been defined and implemented as reported in Figure 5.1. It exploits: Historical Data, Real Time Data, to compute predictions and the Analytical Models to perform Simulations and computing KPIs. The Simulations may be performed for each Scenario which is referring to date and time, and specific conditions and data. The results are assessed by the decision makers by using (i) some Visual Analytics tool (namely the Snap4City Dashboard Builder [36], [24]) according to the simulated data outcome and (ii) KPI, also adopting some selection criteria which are formalized in terms of KPIs for each Simulation outcome. The Analytical Models are used for both: performing the Simulation and Computation of the KPIs. The Computation of KPIs and thus the comparison of the outcomes of multiple simulated scenarios can be performed only if the results of the simulations are formalized and saved to be reused for the KPI computation. Moreover, according to the obtained results on Visual Analytic and KPI, the decision maker may also decide to perform some other simulation as well as to make changes on former Scenarios or create others. The actual implementation of the solution has been performed by using Snap4City framework, which

allows to develop the processes as MicroServices for implementing: Simulations, Analytical Models, Predictions of any kind, and KPI computations exploiting R-Studio or Python processes running as containers in the back office. In addition, the business logic determining the data flows has been implemented in Node-RED, as well as the manipulation of the scenarios, the search of data, and the Business Logic behind the Visual Analytics in the Snap4City Dashboard Builder and visual tools [36]. This approach resulted in a flexible framework in which any kind of simulation, KPI, prediction, and criteria can be added to the tools according to the needs.

In this chapter, only the aspects regarding the traffic flow reconstruction and related KPI are discussed since the computing of the traffic flow is a fundamental aspect of the What-If analysis in city cyberphysical systems, due to the fact that most of the other predictions/simulations strongly depend on the distribution of traffic flow density in the scenarios under study of the What-If analysis.

5.3.2 Modeling Scenarios

In the context of What-If analysis, each scenario has been denoted as SC_{ID}, and it is identified by a unique ID, a description, the involved areas, a set of time intervals and additional constraints, such as the category of users or vehicles which are going to be restricted/enabled. Therefore, SC_{ID} can be formalized by the following tuple:
$SC_{ID} = ID, D, A, T, C, R,$
where:

- ID is the unique scenario Identifier;
- D is a textual description of the scenario;
- A is a simply or multiply connected blocked Area, that is a set composed by one or multiple blocked areas: $A = A\_1, A\_2, ..., A\_N$, where N is the number of blocked Areas;
- T represents a set of time slots or intervals: $T = T_S, T_E$ where T_S and T_E represents, respectively, the starting and ending dates and times of the period in which the blocking constraints are active;
- C is a set of blocking constraints representing which transportation mean is being blocked/limited, for instance:

$C = C_{ped}, C_{priv}, C_{pub}, C_{spec}, C_{cyc}, ...,$
where C_{ped} represents a pedestrian constraint, C_{priv} a private vehicles constraint, C_{pub} a Public transport constraint, C_{spec} a special vehicles constraint, C_{cyc} bike cycling paths constraint, etc. Please note that, changes in the specific constraints of a road segment (e.g., velocity, direction) can be defined as a set of constraints;

- R is the reference road graph network.

The Scenarios can be produced for: planned or unplanned events, with the aim of solving/limiting problems, as well as managing the events or the changes. In the case in which an area has to be closed to perform some recovery maintenance work, multiple Scenarios ($SC_i = \{i,\ D_i, \boldsymbol{A}_i, \boldsymbol{T}_i, \boldsymbol{C}_i,\ R\}$, where $i = 1, ..., N$ represents different scenario IDs) could be defined on the basis of changes in the intervention. Thus, different changes on the road graph, for example in the viability, direction of roads, etc. can be applied. The impact of the application of a certain scenario $SC_{i,\widehat{\boldsymbol{T}}} = \left\{i,\ D_i, \boldsymbol{A}_i, \widehat{\boldsymbol{T}}, \boldsymbol{C}_i,\ R^*\right\}$ should be compared with other $SC_{j,\widehat{\boldsymbol{T}}} = \left\{j,\ D_j, \boldsymbol{A}_j, \widehat{\boldsymbol{T}}, \boldsymbol{C}_j,\ R^*\right\}$, as well as against the original scenario in the same time slots $\widehat{\boldsymbol{T}}$ without any restrictions, which is called the *UnChanged Scenario*, $SC\_UC_{i,\widehat{\boldsymbol{T}}} = \left\{i,\ D_i, \emptyset, \widehat{\boldsymbol{T}}, \emptyset,\ R\ \right\}$. Blocking areas of the scenario are drawn via the user interface (see Figure 5.2) providing geometric shapes (rectangles, circles and polygons) on the map, supplying some the above described metadata, among which those mandatory are identifier, blocking areas, starting and ending times and constraints. Saved scenarios can be loaded and used by the What-If routing engine to provide dynamic alternative routings for different types of traffic means: private cars, public transportation busses, bikes and pedestrians. The routing engine exploits the open source GraphHopper library [2], which has been extended to manage simply or multiply connected blocking constraints. A formal definition of the Scenario is saved into a storage and it is provided to all algorithms and modules in JSON format as a Node-RED message.

5.3.3 Computing Predictions

The short and long terms predictions are computed on the basis of the historical data plus some contextual data. For example, short terms predictions in the range of minutes and hours of traffic flow may be predicted on the

Figure 5.2: Example of a What-If scenario with 3 areas and visual verification of possible routing around the blocked areas. Please note that start and end points can be moved to see in real time the routing. Also intermediate points can be imposed. `https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjE5MA==`.

basis of historical data [148], [26]; while midterms predictions within a day may be computed also taking into account weather conditions and forecast, since the city user decide for short range traveling on the basis of weather conditions. A similar consideration and approach may be applied for: parking predictions which exploit historical data, traffic and weather, for bike sharing which in most cases exploit historical data and weather, pollutants which also exploit traffic flow and weather condition which may influence the movements of particles with humidity and wind. On the contrary, very long-term predictions, for example 3 months in advance, are very hard to be obtained in reliable manner, weather conditions in the close range are not accessible, and neither in the form of long range weather forecast. Then, for very long-term predictions, most of the predictive algorithms are not capable to produce satisfactory results with errors smaller than the 15%. In those cases, the computation of the typical time trends for the traffic flow (similar for parking, pollutant, temperature, people flow, etc.) of the period can be computed taking into account the historical data of the same day of the week, of the month and of the year. For example, for predicting the traffic flow in the first Monday of July 2022, the typical trends of the traffic flow can be

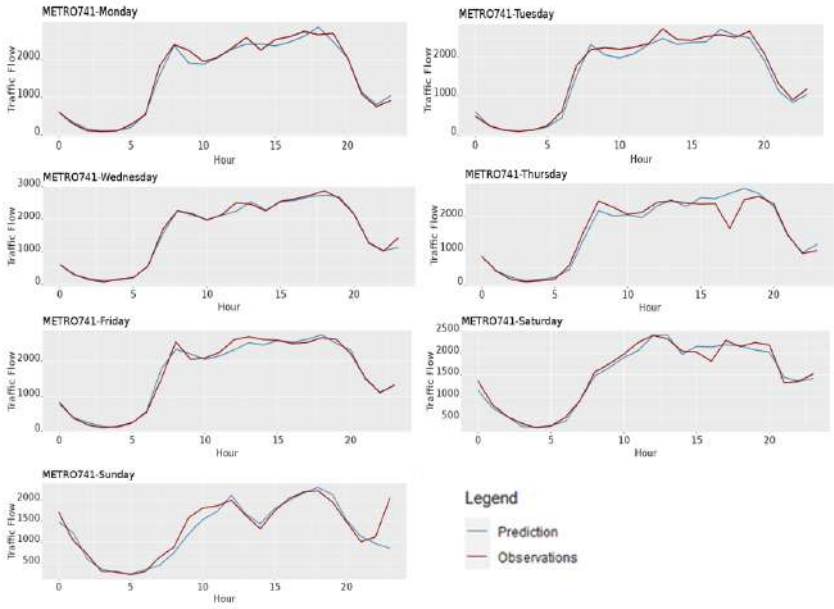


Figure 5.3: Example of predicted traffic flow density (in this case vehicles per hour) with respect to the actual values in a week of the period in which the What-If analysis is performed.

computed taking into account the values of the Mondays in June/July 2021 (a part for COVID-19 considerations).

In Figure 5.3, the typical time trends of traffic flow are compared with the corresponding data of an actual traffic measure collected from the traffic flow sensors (e.g., METRO 741 in Florence area). The typical time trend has been computed as the median on the basis of data of October 2018, while the actual values are those of the same day of the week of October 2019. The computed averaged MAPE (Mean Absolute Percentage Error) over the whole set of city sensors in that period resulted to be of 11.2%. The MAPE has been calculated as follows:

$$MAPE = \frac{\sum_{i=1}^{n} \left| \frac{obs_i - pred_i}{obs_i} \right|}{n} * 100$$

5.3.4 Computing Road Graph KPI

According to the above presented formal model, the KPIs have to be calculated on the basis of a certain scenario $SC_{i,\widehat{\boldsymbol{T}}}$, and they have to be compared against those obtained in some other scenario, $SC_{j,\widehat{\boldsymbol{T}}}$, as well as against the original scenario in the same time slots $\widehat{\boldsymbol{T}}$ without any restrictions, which is called the *UnChanged Scenario*, $SC\_UC_{i,\widehat{\boldsymbol{T}}}$. In the context of What-If analysis, which implies reconstructing the traffic flow density in scenarios under analysis, some of the KPIs can be directly computed on the basis of the changed road graph before computing the traffic flow in the What-If analysis conditions. As it will be clarified in the following, one of the first steps of the analysis consists in creating a modified road graph and thus a new traffic network. In fact, betweenness, centrality and eccentricity metrics are computed on the basis of the topological structure of the traffic network of the city, without taking into account the actual flow. On the other hand, [40], [66] showed that the highest value for *betweenness* is located in proximity of areas where traffic congestions often occurs. The vertex betweenness (also known as betweenness centrality) of a node v of the road graph network R is the number of shortest paths which pass through vv in R, formally we have

$$b_R(v) = \sum_{i \neq j, \quad i \neq v, \quad j \neq v} g_{ivj}/g_{ij}$$

where g_{ij} is the total number of the shortest paths from node i to node j in R, and g_{ivj} is the number of those paths passing-through v in R. The vertex betweenness represents the degree to which nodes stand between each other and it measures the extent to which a vertex lies on paths between other vertices. Nodes having high betweenness may have considerable influence within a road network by virtue of their control over traffic data passing between others. Such nodes are also the ones whose removal from the network will most disrupt communications between other vertices, because they lie on the largest number of paths inside the network. The correctness of a given information which goes through the nodes of a network is depends also related to on other issues related to node properties. An example is given by the *(eigenvector) centrality* [154] of a vertex v in R, labelled with $c_R(v)$, which is a measure of the influence of the vertex v in the road graph network R. In general, vertices with high (eigenvector) centralities are those which are connected to many other vertices which are, in turn, connected to many others (and so on). Another topological metric is given by the *eccen-*

tricity [154] of a vertex v in R, labelled with $e_R(v)$, which is defined as the shortest path distance that a given vertex v has from the farthest other node in the road graph network R. The nodes having high *eccentricity* are located in the decentralized zones of the urban graph admitting more distance from the other side of the network. Then, such metrics can be considered as structural features of the road network describing a given scenario, and they can help us to understand how the related and restricted road network changes in terms of connectivity. Formally, the connectivity KPI (denoted by $\boldsymbol{KPI}_C$)according to a certain scenario $SC_{i,\widehat{\boldsymbol{T}}} = \left\{i,\, D_i, \boldsymbol{A}_i, \widehat{\boldsymbol{T}}, \boldsymbol{C}_i,\, R^*\right\}$ can be defined as

$$\boldsymbol{KPI}_C(SC_{i,\widehat{\boldsymbol{T}}}) = \{B_{R^*}, C_{R^*}, E_{R^*}\}$$

where

- $B_{R^*} = (\max\{b_{R^*}(v) : v \in R^*\}, b)$ with b representing the corresponding node assuming the maximum value,
- $C_{R^*} = (\max\{c_{R^*}(v) : v \in R^*\}, c)$ with c representing the corresponding node assuming the maximum value,
- $E_{R^*} = (\max\{e_{R^*}(v) : v \in R^*\}, e)$ with e representing the corresponding node assuming the maximum value.

So that, computing $\boldsymbol{KPI}_C(SC_{i,\widehat{\boldsymbol{T}}})$ and $\boldsymbol{KPI}_C(SC\_UC_{i,\widehat{\boldsymbol{T}}})$, it may produce different values, and distinct representative nodes can be observed in the map in order to understand the changes in terms of road graph network connectivity.

5.4 Constrained Traffic Flow Reconstruction and Simulation

In this section, the traffic flow reconstruction approach is described in the form to be used as a constrained simulation tool for computing the traffic flow density in all segments of the city roads. This process involves the following steps: (i) the road graph definition, (ii) the definition of the What-If analysis scenarios, and (iii) the computation of traffic flow predictions on the same points in which the regular traffic flow sensors are located. As

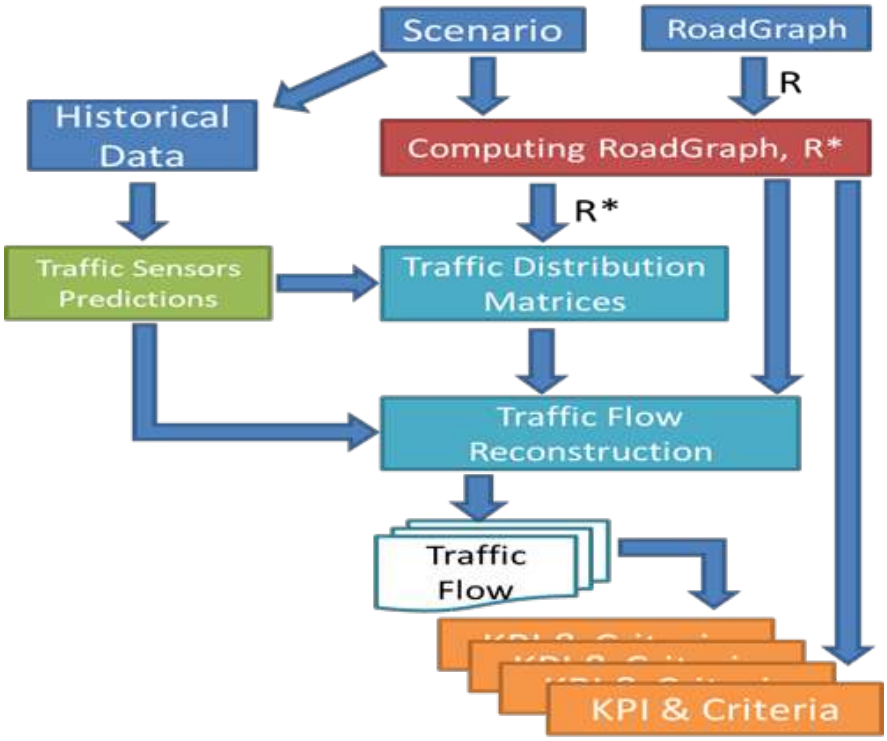


Figure 5.4: Data flow for the simulation process of traffic flow reconstruction on the basis of What-If scenario and predicted traffic flow sensor data. The color of each phase is in accordance with the functional architecture of the What-If Analysis solution shown in Figure 5.1.

above described, the most effective solutions for large scale traffic flow reconstructions are those based on differential equations. Thus, in agreement with traffic flow reconstruction approach of [40], taking into account the reference scenario of What-If analysis, a number of steps are needed to compute the traffic flow in each time slot and day of the scenario. Figure 5.4 is presenting the process data flow. To this end, the following phases are needed.

The process can be summarized as follow:

1. **computation of KPI in the UnChanged Scenario**, $SC\_UC_{i,\widehat{\boldsymbol{T}}}$,

which can be performed before the simulation and the computation of the modified road graph, R^*. The KPI for the unchanged scenario has to be computed for each time slot of the What-If analysis scenario.

2. **Review of the road graph network** on the basis of the scenario to compute the modified road graph: R^*. According to the What-If analysis scenario, some no-go areas are delimited by means of polygonal definitions on the map in the selected time slots of the days, then the road network structure modification is needed through the following steps:

   - exclusion of the (road graph) arcs that admit at least one point (described by the segmentation of the arcs every 20m) inside the no-go areas,
   - exclusion of the junctions having the related node inside the no-go areas, also modifying all the intersections in which a given arc is no longer considered in the new setting and modifying in an appropriate way the distribution of the incoming and outgoing traffic according to the characteristics of the remaining arcs,
   - exclusion of the sensors that admit relative node inside the no-go areas.

3. **computing some road graph metrics KPI**, such as betweenness, centrality and eccentricity, performed on the original road graph R, and immediately after to have obtained the graph R^*. This means that that $\Delta\overline{KPIc}$ (.) and some considerations can be performed before proceeding to the complete simulation also observing the changes on the map and the absolute values of those KPIs.

4. **Computing Traffic Distribution Matrices**, TDM, which is the traffic flow distribution at junctions. According to [40], when the road graph is changed the TDM is changed as well since the distribution of traffic flow in the junctions which present close roads are obviously changed, but also those that could be influenced by the changes on road graph. In order to model the traffic distribution at junctions, a distribution matrix can be used to describe the percentage of vehicles getting out each outcoming road with respect to those getting in each incoming road. Thus, the traffic distribution matrix is defined as $TDM=\{w_{ji}\}_{j=n+1,\ldots,n+m,i=1,\ldots,n}$ so that $0 < w_{ji} < 1$ and

$\int_{j=n+1}^{n+m} w_{ji}= 1$, for $i= 1,\ldots,n$ and $j=n+1,\ldots,n+m$, where w_{ji} is the percentage of vehicles arriving from the i-th incoming road and taking the j-th outcoming road (assuming that, on each junction, the incoming flux coincides with the outcoming flux). The real values of w_{ji} may depend on the time of the day, on the road size, cross light settings, etc., and thus, it is unknown a priori. In the following, w_{ji} coefficients are called *weights*. When one (or more) roads are closed, then the traffic is redistributed in other directions. In particular, we have a reassessment of the traffic distribution in the junctions such that $\int_{j=n'+1}^{n'+m'} v_{ji} = 1$, for $i = 1,\ldots,n'$ and $j = n'+1,\ldots,n'+m'$ where $n' \leq n$, $m' \leq m$ and

$$v_{ji} = w_{ji} \frac{\int_{j=n+1}^{n+m} x_{ji}}{\int_{j=n'+1}^{n'+m'} x_{ji}}$$

where x_{ji} are the values in the weight assignment giving the lower mean error by means of the stochastic relaxation technique as described in [40].

5. The **traffic flow reconstruction** is performed by solving a nonlinear model based on the conservation of vehicles described by the following scalar hyperbolic conservation law, in a single road,

$$\frac{\partial \rho\,(t,x)}{\partial t} + \frac{\partial f\,(\rho\,(t,x))}{\partial x} = 0,$$

where: $\rho\,(t,x)$ is the traffic density of vehicles, which admits values from 0 to ρ_{max}, where $\rho_{max} > 0$ is the maximal traffic density; $f\,(\rho(t,x))$ function is the vehicular flux which is defined by means of the product $\rho\,(t,x)\ \ v\,(t,x)$, where $v\,(t,x)$ is the vehicle speed; and boundary conditions $\rho\,(t,a) = \rho_a(t)$, $\rho\,(t,b) = \rho_b(t)$, initial values $\rho\,(0,x) = \rho_0(x)$, with $x \in (a,b)$. In the case of first order approximation, we assume that $v\,(t,x)$ is a decreasing function, depending on the density, then the corresponding flux is a concave function. Thus, we consider the local speed of the vehicles as $v\,(\rho) = v_{max}(1 - \frac{\rho}{\rho_{max}})$ and then $f\,(\rho) = v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right)\rho$, where v_{max} is the limit speed on a given road segment (these assumptions are known in the literature as the Greenshield's Model.) The solution is obtained by an iterative process, at finite differences on the basis of the traffic flow data in the sensors points or, as in this case, by exploiting the predictions for each time

slot of the What-If analysis scenario. For the whole sequence, and for each time slot, the traffic flow reconstruction is performed producing a value of traffic density in each city road segment of the graph, which are typically of 20mt. Traffic flow reconstruction algorithm has to be computed progressively and on a parallel architecture, since the estimation of traffic flow density for the city (in Florence there are about 30.000 segments) at time instant t would depend on traffic flow at time t-1 and on the new measures coming from sensors/predictions.

6. Once the traffic flow reconstruction is complete for each time slot of the What-If analysis scenario, a set of **traffic flow KPIs** are computed. In addition, it is also possible to compute the average values of **KPI** and the differences which may provide support to make some decisions among multiple scenarios and solutions.

7. The final step consists in **making the decision** among different scenarios to take the most acceptable compromise according to the identified KPI.

5.4.1 Computing Traffic Flow KPIs

Actually, each scenario is defined via temporal features according to the selected hours of the days describing the related scenario application. In this case, the dynamic system representing the reconstruction traffic flow model admits a current state which influences a future state. So that, some KPIs are needed in order to compare the traffic viability at each state (or time slot) of the model. In order to do that, the common metrics used in the area of vehicular traffic flow theory can be considered, which are the *traffic density* and the *traffic flow* values, that is, the number of vehicles in terms of road occupancy and the number of vehicles crossing the supervised location during a given period of time (which is usually equal to one hour), respectively. More precisely, the traffic density of a road graph network R at a given time slot t, denoted by $D_R(t)$, is defined as the array $D_R(t) = \{\rho_R(i,t) : i = 1, \ldots, S\}$ where S is the total number of the road segments in R having length of about 20 meters. Analogously, the corresponding traffic flow is defined as $F_R(t) = \{f_R(i,t) : i = 1, \ldots, S\}$.

Fixing the i-th road segment in R (with $i = 1, \ldots, S$) such that $\rho = \rho_R(i,t)$ in $D_R(t)$ and $f = f_R(i,t)$ in $F_R(t)$, if $\rho = 0$ then $f = 0$.

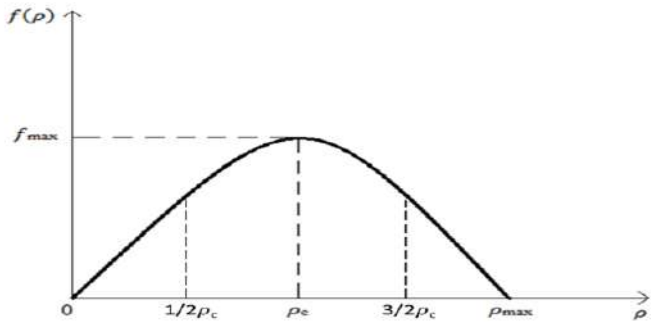


Figure 5.5: A simple classification of the fundamental diagram in order to define "FREE", "FLUID", "HEAVY" or "VERY HEAVY" traffic states for the context of decision support results.

Otherwise, when ρ grows, also f grows up to the maximum f_{max} for which the vehicular density assumes its critical value $\rho_c = \rho_{max}\ /2$. In the case where $\rho > \rho_c$, other increases of the traffic density conduct to a congested scenario of the traffic flow up to its maximum value ρ_{max} where the speed of the cars is 0 and then $f = 0$. Of course, the critical density associated to the i-th road segment in R depends on its road capability according, for example, to its number of lanes. So that, some thresholds on the road segments of R can be defined in terms of traffic density. Critical values of the traffic density (according to the number of lanes) can be observed when some congestion situations are occurring. Such trends are related with the data coming from the observed sensors' behaviors in different locations and their well-known *fundamental diagrams* where congestions are examined. This approach allows us to obtain a set of thresholds $\{\rho_c\,(i) : i = 1, \ldots,\ S\ \}$ in terms of traffic density determining the state of traffic of the i-th segment road in R, for each $i = 1, \ldots,\ S$. It can establish "*FREE*", "*FLUID*", "*HEAVY*" or "*VERY HEAVY*" traffic states by assigning numerical intervals on the value $\rho = \rho_R\,(i, t)$ according to the corresponding threshold $\rho_c = \rho_c\,(i)$, for each $i = 1, \ldots,\ S$, considering for example $0 < \rho < \rho_c/2$, $\rho_c/2 \leq \rho < \rho_c$, $\rho_c \leq \rho < 3/2\rho_c$ and $3/2\rho_c \leq \rho < 2\rho_c$, respectively, as a regular partition of the fundamental diagram depicted in Figure 5.5.

Actually, a more accurate classification could be carried out in such a context, but the presented one is easily used as an immediate comparative

tool in the decision support field. In particular, the percentages of road segments that admit the above traffic states, with respect to the total number of road segments in R, can be considered in order to determine KPIs via temporal features of a given scenario, making it easy to understand the comparative results.

Moreover, in [64] some integrated performance indicators in urban road infrastructure are also developed for evaluating network functionality and the impact of transport system interventions. To determine a value of traffic congestion in the network, the most informative metric seems to be the so called **average degree of saturation**. More precisely, the average degree of saturation of a road graph network R at a given time slot t, denoted by $S_R(t)$, is defined as

$$S_R(t) = \frac{\sum_{i=1}^{S} \frac{\rho_R(i,t) \quad l(i)}{\rho_c(i)}}{\sum_{i=1}^{S} l(i)} ,$$

where $l(i)$ is the length of the i-the road segment in R. Please note that, in the considered road graphs, $l(i)$ is almost constant equal to 20m. Thus

$$S_R(t) \cong \sum_{i=1}^{S} \frac{\rho_R(i,t)}{\rho_c(i)}.$$

Please note that $\rho_c(i)$ is not constant, since it depends on the structure of the road in terms of lanes. Formally, the *traffic flow KPI* (denoted by $\boldsymbol{KPI}_F$) according to a certain scenario $SC_{i,\widehat{\boldsymbol{T}}} = \left\{ i,\ D_i, \boldsymbol{A}_i, \widehat{\boldsymbol{T}}, \boldsymbol{C}_i,\ R^* \right\}$, where

$$S_{R^*}(t) \cong \sum_{i=1}^{S*} \frac{\rho_{R*}(i,t)}{\rho_c(i)},$$

can be defined as

$$\boldsymbol{KPI}_F(SC_{i,\widehat{\boldsymbol{T}}}) = \{\boldsymbol{FR}_{R^*}, \boldsymbol{FL}_{R^*}, \boldsymbol{HE}_{R^*}, \boldsymbol{VH}_{R^*}, \boldsymbol{S}_{R^*}\}$$

where

- $\boldsymbol{FR}_{R^*} = \{ \frac{|FR_{R^*}(t)|}{S^*} 100 \ : t \in \widehat{\boldsymbol{T}} \}$ is the FREE traffic state percentage for each time t in $\widehat{\boldsymbol{T}}$ and
  $|FR_{R^*}(t)| = | \left\{ i \in [1, S^*] : \rho_{R^*}(i,t) < \frac{\rho_c(i)}{2} \right\} |$
  is the number of road segments in R^* admitting FREE traffic state at time t in $\widehat{\boldsymbol{T}}$, where S^* is the total number of road segments in R^*.

- $\boldsymbol{FL}_{R^*}=\{\frac{|FL_{R^*}(t)|}{S^*}100 \ : t\in \widehat{\boldsymbol{T}}\}$ is the FLUID traffic state percentage for each time t in $\widehat{\boldsymbol{T}}$ and
  $|FL_{R^*}(t)| = |\left\{i \in [1,S^*] : \frac{\rho_c(i)}{2} \leq \rho_{R^*}(i,t) < \rho_c(i)\right\}|$
  is the number of road segments in R^* admitting FLUID traffic state at time t in $\widehat{\boldsymbol{T}}$.
- $\boldsymbol{HE}_{R^*}=\{\frac{|HE_{R^*}(t)|}{S^*}100 \ : t\in \widehat{\boldsymbol{T}}\}$ is the HEAVY traffic state percentage for each time t in $\widehat{\boldsymbol{T}}$ and
  $|HE_{R^*}(t)| = |\left\{i \in [1,S^*] : \rho_c(i) \leq \rho_{R^*}(i,t) < \frac{3\rho_c(i)}{2}\right\}|$
  is the number of road segments in R^* admitting HEAVY traffic state at time t in $\widehat{\boldsymbol{T}}$.
- $\boldsymbol{VE}_{R^*}=\{\frac{|VH_{R^*}(t)|}{S^*}100 \ : t\in \widehat{\boldsymbol{T}}\}$ is the VERY HEAVY traffic state percentage for each time t in $\widehat{\boldsymbol{T}}$ and
  $|VH_{R^*}(t)| = |\left\{i \in [1,S^*] : \frac{3\rho_c(i)}{2} \leq \rho_{R^*}(i,t) < 2\rho_c(i)\right\}|$
  is the number of road segments in R^* admitting VERY HEAVY traffic state at time t in $\widehat{\boldsymbol{T}}$.
- $\boldsymbol{S}_{R^*} = \{S_{R^*}(t) : t \in \widehat{\boldsymbol{T}}\}$ is the collection of the average degree saturation values for each t in $\widehat{\boldsymbol{T}}$.

Finally, the scenario KPI (denoted by $\boldsymbol{KPI}$) according to a certain scenario $SC_{i,\widehat{\boldsymbol{T}}} = \left\{i,\ D_i, \boldsymbol{A}_i, \widehat{\boldsymbol{T}}, \boldsymbol{C}_i,\ R^*\right\}$ can be defined as the union of the related traffic flow KPI and the related connectivity KPI (see **Section 5.3.4**), so that

$$\boldsymbol{KPI}(SC_{i,\widehat{\boldsymbol{T}}}) = \boldsymbol{KPI}_F(SC_{i,\widehat{\boldsymbol{T}}}) \cup \boldsymbol{KPI}_C(SC_{i,\widehat{\boldsymbol{T}}}).$$

The KPIs produce different values for each time slot in the simulation/scenario of the What-If analysis. Thus, the comparison among the effects of different choices and scenarios have to be performed on the basis of MIN, MAX, average, median or the values obtained in each time slot. Thus computing: $\boldsymbol{KPI}(SC_{i,\widehat{\boldsymbol{T}}})$, $\boldsymbol{KPI}(SC\_UC_{i,\widehat{\boldsymbol{T}}})$, for the averages $\overline{KPI}(SC_{i,\widehat{\boldsymbol{T}}})$, $\overline{KPI}(SC\_UC_{i,\widehat{\boldsymbol{T}}})$, and the derived differences of KPIs in different scenarios we may have: $\Delta\overline{KPI}$ (.).

5.5 Validation in a Simple Case

In order to validate the tool, a real case in Florence city has been considered. The case consisted in performing the What-If analysis of the traffic flow in

Figure 5.6: Graphical representation of the blocking area scenario defined in the city of Florence. The scenario is named "ScenarioAnov2019" and it represents the real blocking area caused by the road works started at the beginning of December 2019 around the square named "Piazza della Libertá" for the future rails location in Florence. Red dot represents the location of the road graph network highest betweenness before the works in the city, Green dot the position when the graph road changes have been applied.

the case of changes in the viability due to the creation of large restructuring area for creating a new tram line in Florence. The blocking area reported in Figure 5.6 describes the shape of the scenario of the road works started at the beginning of December 2019 around the square named "Piazza della Libertá". Such a modification in the road network removed a number of road segments within the square to allow the construction of the tramway rails line #3 in Florence.

The selected presents a high level of traffic flow, and it is one of the key junctions in the Florence city. This fact is also shown by means of the location of the road graph node having the highest betweenness as depicted in Figure 5.6 (in the case of the blocking area is not yet considered) and such a node is very closed to the selected area. The presence of blocking area in the road network slightly modifies the related suitable directed graph and, for instance, the highest value of the betweenness, which determines the typical junctions where the traffic congestion often occurs changed location. In particular, in the case of blocking area scenario, the junction assuming

the highest value is the Green node depicted in Figure 5.6 .

By setting $SC_{i,\widehat{\boldsymbol{T}}}$ as the described scenario named "ScenarioANov2019", we have

$\boldsymbol{KPI}_C(SC_{i,\widehat{\boldsymbol{T}}}) =$

$= \{(469640.5,\ node_{253179120}),(1,node_{3262140609}),(88,node_{298511990})\}$

and

$\boldsymbol{KPI}_C(SC\_UC_{i,\widehat{\boldsymbol{T}}}) =$

$= \{(464605.71,\ node_{246843224}),(1,node_{3262140609}),(88,node_{298511990})\}$

where the nodes' indexes correspond to the OSM (Open Street Map) indexing. In this case, the highest value of the betweenness in $SC_{i,\widehat{\boldsymbol{T}}}$ admits an increment of about 1% with respect to the one in $SC\_UC_{i,\widehat{\boldsymbol{T}}}$ and a North-West shift appears in the corresponding node position. Note that, the highest values of eccentricity and centrality are assumed by the same nodes in both cases of the road network, since their modification only happens when significant changes in the related suitable directed graph are operated and this is not the case since a small blocking area is under review (see later for a case with more relevance changes).

From December 2019, the presence of road works in the area has contributed to some inconveniences in vehicular traffic by causing an increase in vehicular density during daylight hours. Since the number of the road segments is reduced in the area, then the drivers have forced choices for their travel directions by converging on the same option. So that, a more congested situation has been observed in the relevant and primary roads surrounding the square site during the days (and months).

In order to validate the What-If approach and simulation tools, we have run the simulation of the traffic flow reconstruction imposing the modifications in the road graph network. The aim has been two folds: (i) to study the effect of changes (see Section 5.5.1), and (ii) to predict the actual traffic behavior months in advance (see Section 5.5.2), also verifying that the predicted values have been precise enough with respect to the actually produced effects on traffic flow by the works on the roads.

5.5.1 Assessing the effects of changes

By setting $\widehat{\boldsymbol{T}}$ ={2019-11-20T07:00, 2019-11-20T20:00}, the results of the simulation of $SC_{i,\widehat{\boldsymbol{T}}}$ (representing the described scenario named "ScenarioANov2019") are related to a working day period when the traffic is usually heavy with respect to the weekend days. The simulation has been based on

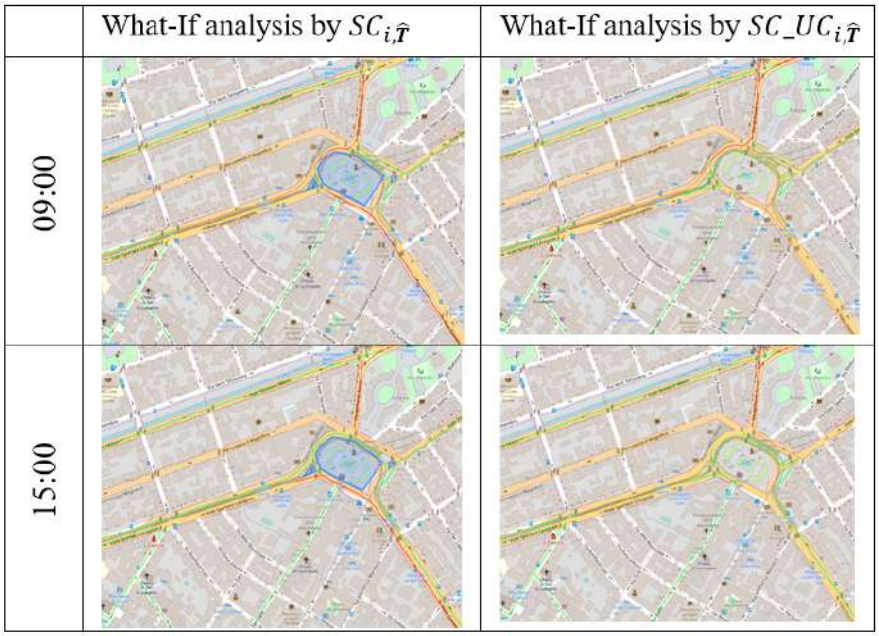


Figure 5.7: Graphical comparison between the What-If analysis for $SC_{i,\widehat{T}}\boldsymbol{SC}_{\boldsymbol{i},\widehat{\boldsymbol{T}}}$ (representing the described scenario named "ScenarioANov2019") with respect to $SC\_UC_{i,\widehat{T}}\boldsymbol{SC\_UC}_{\boldsymbol{i},\widehat{\boldsymbol{T}}}$. Only two selected frames are considered, at time 09:00 and 15:00, respectively.

the above described traffic predictions for Traffic Flow on sensors computed during the six weeks from 2019-10-14 to 2019-11-24. Thus, computing the What-If analysis regarding traffic flow related to $SC_{i,\widehat{T}}$ and $SC\_UC_{i,\widehat{T}}$, we can observe different vehicular traffic situations. Figure 5.7 shows a graphical comparison of the related traffic states at time $09:00$ and $15:00$ for $SC_{i,\widehat{T}}$ and $SC\_UC_{i,\widehat{T}}$ respectively, where the interested area is under review.

The What-If analysis including the traffic flow reconstruction by simulation of $SC_{i,\widehat{T}}\boldsymbol{SC}_{\boldsymbol{i},\widehat{\boldsymbol{T}}}$ produced a more congested situations with respect to $SC\_UC_{i,\widehat{T}}\boldsymbol{SC\_UC}_{\boldsymbol{i},\widehat{\boldsymbol{T}}}$ and the related computation of $\boldsymbol{KPI}_F(SC_{i,\widehat{T}})$ and $\boldsymbol{KPI}_F(SC\_UC_{i,\widehat{T}})$ are graphically compared in Figure 5.8, where the suitable road graph network is restricted to the area under review, that is, a circle area having its center at the center of "Piazza della Libertá" and ra-

dius 1 km, so that, it is a small part of the city (about the 10% of the whole city). The bar-series provides a comparative distribution, hour by hour, of the distribution of free, fluid, heavy and very heavy traffic roads in terms of density in the central hours of the day. From the second part of Figure 5.8,, it is evident that the new configuration is going to produce a certain increment of *traffic saturation* in the area for the last hours of the day (from 13 : 00 to 20 : 00), which are the most critical. The mean increment of the saturation is equal to 5.1% while the maximum saturation increment has been equal to 12.4% at time 18 : 00 (for the city part considered).

5.5.2 Validating the Effect of changes with respect to actually measured effects of changes

As a validation approach, the computation of the What-If analysis for $SC_{i,\widehat{\boldsymbol{T}}}$, modeling the effect of the road words for tramlines, have been compared with respect to actual measured traffic flow when the works have been actually performed. By setting $\widehat{\boldsymbol{T1}} = \{2020\text{-}02\text{-}05T07{:}00,\ 2020\text{-}02\text{-}05T20{:}00\}$, $R_{\widehat{\boldsymbol{T1}}}$ denotes the real traffic situation modelled when the road works were actually taking place. Both the simulations related to $SC_{i,\widehat{\boldsymbol{T}}}$ and $R_{\widehat{\boldsymbol{T1}}}$ admit a similar vehicular traffic behavior. Figure 5.9 shows a graphical comparison of the related traffic states at time 09 : 00 and 15 : 00 for $SC_{i,\widehat{\boldsymbol{T}}}$ and $R_{\widehat{\boldsymbol{T1}}}$ respectively, where the interested area is under review.

Moreover, the related computation of $\boldsymbol{KPI}_F(SC_{i,\widehat{\boldsymbol{T}}})$ and $\boldsymbol{KPI}_F(R_{\widehat{\boldsymbol{T1}}})$ can be graphically compared in Figure 5.10, where the suitable road graph network is restricted to the area under review, that is, a circle area having its center at the center of "Piazza della Libertá" and radius 1 km.

In order to estimate the error between the results obtained with the What-If analysis on $SC_{i,\widehat{\boldsymbol{T}}}$ with respect to those coming from actual traffic flow reconstruction on $R_{\widehat{\boldsymbol{T1}}}$, the traffic densities in the corresponding road segments has been compared. To this end, *mean absolute error* (MAE) is considered. Since $R_{\widehat{\boldsymbol{T1}}}$ and $SC_{i,\widehat{\boldsymbol{T}}}$ admit the same road graph network, named R^*, then $MAE(t) = \frac{\sum_{i=1}^{S*} |\rho_{R*}(i,t) - \rho'_{R*}(i,t)|}{S*}$, where $\rho_{R*}(i,t)$ and $\rho'_{R*}(i,t)$ are the traffic densities in the i-th road segment of R^*(with $i = 1, \ldots, S^*$) according to $R_{\widehat{\boldsymbol{T1}}}$ and $SC_{i,\widehat{\boldsymbol{T}}}$ respectively, at the corresponding time slot t. In order to estimate such an error in the roads which are located around the blocking constraints where discrepancies are expected, the analysis is conducted in the road segments which are inside a circle having its

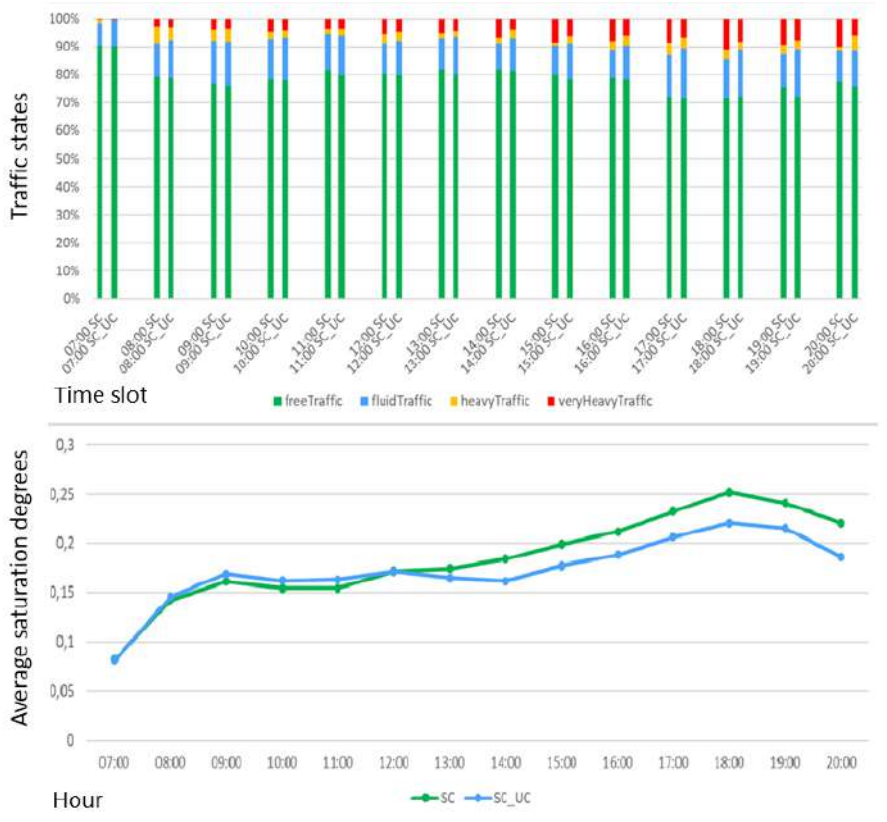


Figure 5.8: Graphical comparison between $\boldsymbol{KPI_F}(SC_{\boldsymbol{i},\widehat{\boldsymbol{T}}})$ and $\boldsymbol{KPI_F}(SC\_UC_{\boldsymbol{i},\widehat{\boldsymbol{T}}})$. In the top, the traffic states are compared for each time slot hour by hour the distribution of free, fluid, heavy and very heavy traffic segments. In the bottom, the related average saturation degrees over time values are depicted. The green line is related to $SC_{\boldsymbol{i},\widehat{\boldsymbol{T}}}$ and it represents a more congested situation with respect to the blue line related to $SC\_UC_{\boldsymbol{i},\widehat{\boldsymbol{T}}}$.

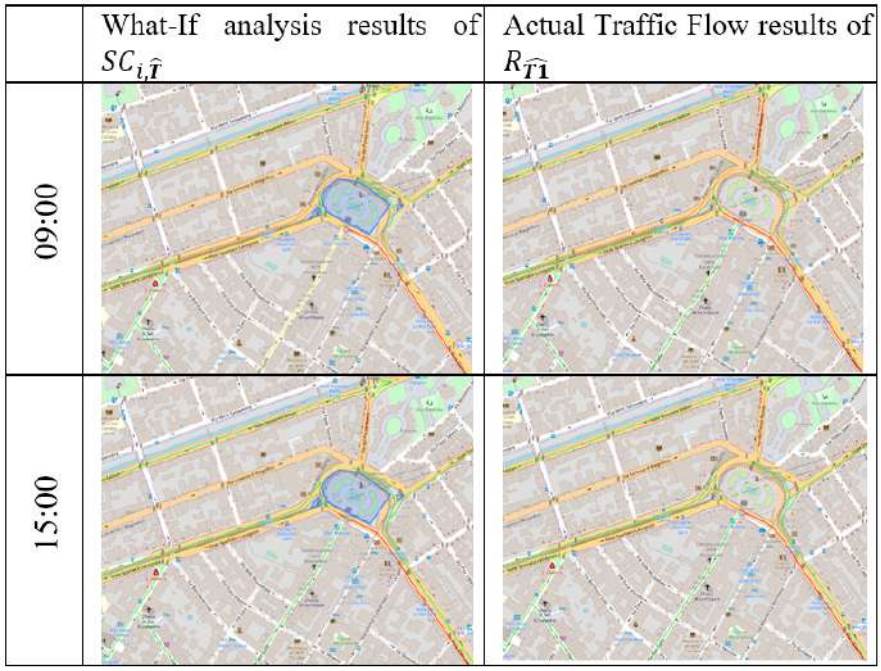


Figure 5.9: Graphical comparison between the simulation $SC_{i,\widehat{T}}$ with respect to $R_{\widehat{T1}}$ where two selected frames are considered, at time 09:00 and 15:00, respectively.

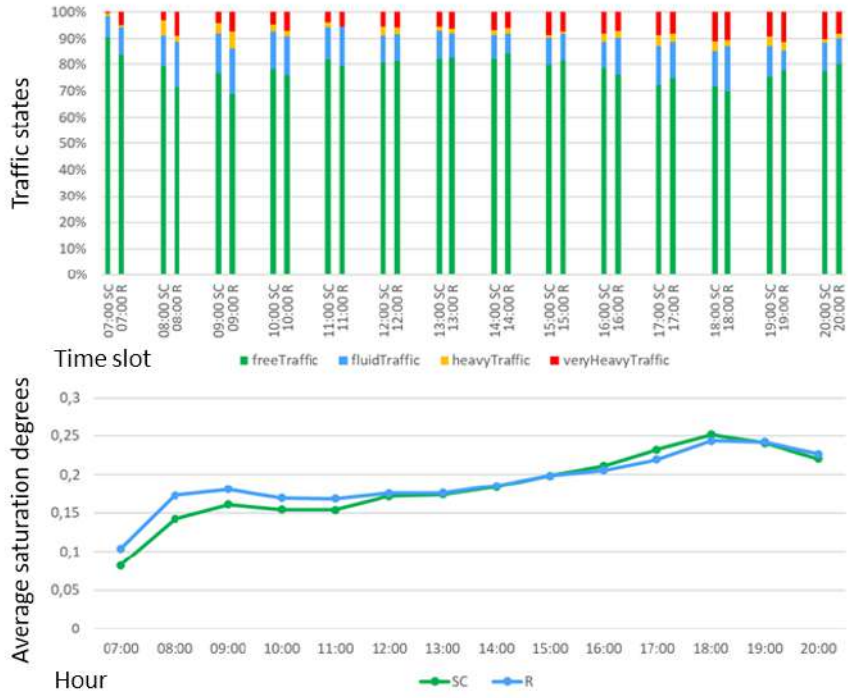


Figure 5.10: Comparison between $\boldsymbol{KPI_F}(SC_{\boldsymbol{i},\widehat{\boldsymbol{T}}})$ and $\boldsymbol{KPI_F}(R_{\widehat{\boldsymbol{T1}}})$. In the top, the traffic state is compared for each time slot of the day. In the bottom, the related average saturation degree values are depicted. The green line is related to $SC_{\boldsymbol{i},\widehat{\boldsymbol{T}}}$ and it represents a similar behavior with respect to the blue line related to $R_{\widehat{\boldsymbol{T1}}}$.

center at the center of "Piazza della Libertá" and radius 1 km, which is equal to 1/10 of the whole city, for a total of about 3000 road segments. Then, $MAE(t)$ is estimated in such a circular area and the results are shown in Table 5.3, together with the related percentage error (with respect to the average traffic density) $MAE_p\,(t) = \frac{MAE(t)}{d(t)}100$, where $d(t)$ is the average traffic density in the considered circular area at time t. The values of $MAE_p(t)$ seems to be in accordance with the comparison between $\boldsymbol{KPI}_F(SC_{i,\widehat{\boldsymbol{T}}})$ and $\boldsymbol{KPI}_F(R_{\widehat{\boldsymbol{T1}}})$ presented in Figure 5.10.

Hour of the day	MAE (vehicles/20 m)	MAE_p (%)
07:00	0.0591	22.16
08:00	0.0966	20.89
09:00	0.0833	16.83
10:00	0.0791	17.67
11:00	0.0768	18.30
12:00	0.0531	12.09
13:00	0.0617	13.93
14:00	0.0694	15.09
15:00	0.0617	11.97
16:00	0.0852	16.06
17:00	0.1080	18.41
18:00	0.1111	17.29
19:00	0.1225	19.43
20:00	0.1055	18.67

Table 5.3: Error estimation between traffic flow density coming from What-If analysis $SC_{i,\widehat{\boldsymbol{T}}}\mathbf{SC_{i,\widehat{T}}}$ and traffic flow reconstruction from direct sensors in $R_{\widehat{\boldsymbol{T1}}}\mathbf{R_{\widehat{T1}}}$ in terms of $\boldsymbol{MAE}$ and $\boldsymbol{MAE_p}$. The estimation is conducted in the road segments contained in a circle with radius of 1 Km and centered in "Piazza della Libertá", for each hour of the simulations.

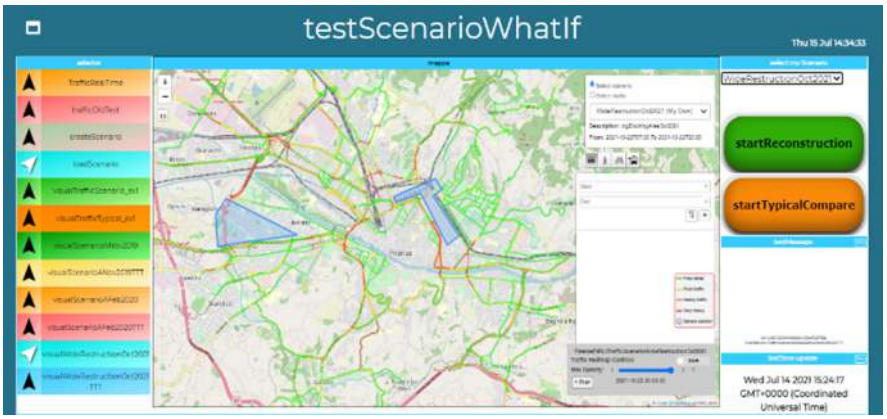


Figure 5.11: What-If analysis tool with complex scenario presenting multiply connected blocking areas in the city.

5.6 Multiply Connected Scenarios

The proposed model and tools allow to perform the What-If analysis taking large scenarios of changes in the original road network by defining large blocking areas by means of multiply connected blocking constraints. In this manner, the road network structure can be strongly modified and complex scenarios can be analyzed.

An example of scenario having large multiply connected areas, named "WideRestructionOct2021", is shown in Figure 5.11, where two polygons covering a considerable part of the city road network graph are defined according to the description in Section 5.3.2. This complex scenario has been defined for stressing the computational complexity of the solution and assessing the performance. Please note that the logic of the What-If analysis, implying to integrate one or more simulations and KPI assessment, is performed by scripting data flow business logic in Node-RED, according to the so called IoT Apps of Snap4City [25], [23].

By setting $SC_{j,\widehat{\boldsymbol{T}'}}$ as the described scenario named "WideRestructionOct2021" and considering $\widehat{\boldsymbol{T}'} = \{$2021-10-22T07:00, 2021-10-22T20:00$\}$, then the scenario represents a situation of the network graph occurring in a determined period in the future, from 2021-10-22T07:00 to 2021-10-22T20:00, where two multiply connected areas (depicted in blue in

the Figure 5.11) are prohibited to the vehicular traffic. When a scenario is defined, then the reconstruction model can be performed according with the selected constraints (selecting the button named "startReconstruction" showed in the right side of Figure 5.11), by applying the above described computational steps according to the reconstruction model in presence of blocking components. In particular, the scenario "WideRestructionOct2021" admits 32299 units of road network segmentation of length 20m and 1463 junctions, reducing the original road network of 1529 units and 77 junctions. That is, a 4.7% of reduction on traffic road segments, and a 5% of reduction in terms of junctions.

In the given analysis, the reconstruction model allows to calculate the traffic state at each hour of the selected period, in the case from 2021-10-22T07:00 to 2021-10-22T20:00. In particular, the traffic state at a given hour influences the traffic state of the successive hours of the scenario in terms of traffic flow propagation. By running the simulation on such a model, the user can understand how the system would evolve over time as a consequence of the given hypothetical change (i.e., a specific What-If condition). Moreover, the resulting calculation of the reconstruction model can be graphically visualized by means of the corresponding 14 traffic flow maps (one for each hour, from $07:00$ to $20:00$) that can be sequentially selected via a control panel and widgets, and the related animation can be presented as well. Please note that each traffic flow density map is distributed from a GeoServer via WMS protocol to any dashboard and tool requesting them.

When traffic flow reconstruction algorithm is computed with scenarios featuring, then it is possible to compare the produced results via *scenario KPI* (see Section 5.4.1), in order to evaluate the impact of such a scenario in terms of changes of traffic patterns with respect to, for example, the unchanged road graph network, denoted by $SC_{UC}{}_{j,\widehat{\boldsymbol{T}'}}$ and representing a typical traffic behavior during the same period of time (select the button named "startTypicalCompare" showed in the right side of Figure 5.11), or another selected scenario.

5.6.1 Assessment of Traffic Flow

In the sequel, $\boldsymbol{KPI}(SC_{j,\widehat{\boldsymbol{T}'}})$ and $\boldsymbol{KPI}(SC\_UC_{j,\widehat{\boldsymbol{T}'}})$ which are the collection of the *connectivity* and *traffic flow KPIs* related to $SC_{j,\widehat{\boldsymbol{T}'}}$ and $SC_{UC}{}_{j,\widehat{\boldsymbol{T}'}}$, are considered and compared. In the case, the *connectivity KPIs* are

$\boldsymbol{KPI}_C(SC_{j,\widehat{\boldsymbol{T}'}}) =$
$= \{(323488.76, node_{2471190}), (1, node_{4924703865}), (94, node_{298511990})\}$ and
$\boldsymbol{KPI}_C(SC\_UC_{j,\widehat{\boldsymbol{T}'}}) =$
$\{(464605.71, node_{246843224}), (1, node_{3262140609}), (88, node_{298511990})\}$
where the nodes' indexes correspond to the OSM indexing. In particular, the presence of blocking areas of high dimensions in the road network strongly modifies the related suitable directed graph, and the junctions assuming the highest values of betweenness and centrality are the nodes depicted in Figure 5.12. The road graph network modification according to the scenario "WideRestructionOct2021", by which big areas are removed from the original graph, causes a new orientation of the traffic viability by means of a redefined graph balancing with respect to the unchanged graph.

Moreover, the related computation of the *traffic flow KPI* can be also conducted. In particular, $\boldsymbol{KPI}_F(SC_{ij,\widehat{\boldsymbol{T}'}})$ and $\boldsymbol{KPI}_F(SC\_UC_{ij,\widehat{\boldsymbol{T}'}})$ can be graphically compared in Figure 5.13, where each traffic state presents a similar behavior. Please note that areas having typical congestion situations are included in the blocking components and the nodes assuming the highest values of betweenness and centrality in $\boldsymbol{KPI}_C(SC\_UC_{j,\widehat{\boldsymbol{T}'}})$ are included in the blocking components. Moreover, note that $S_R(t)$ and $S_{R^*}(t)$ are computed on different road graphs, while the number of vehicles involved should be the same. When large city areas are blocked, they may include traffic flow sensors which may strongly influence the simulation of traffic reconstruction. If the sensors are not included in the blocked area, or they are in the center of the city, the diffusive approach of the computational model is compensating the lacks. On the contrary, when the sensors included into the blocked areas are those at the city border, then their exclusion may reduce the number of simulated vehicles entering/exiting into/from the city. Thus, in order to preserve the balance on the boundary conditions of the traffic flow mathematical model [40], the flows measured by specific virtual sensors are added, or the total number of vehicles involved in the simulation have to compensate the missing sensors. This compensation is performed by scaling the density by hour in the computation of saturation by a corrective factor, according to the actual volume of predicted vehicles. The resulting saturation degree by hour is reported in Figure 5.13. The average change in saturation of the changed solution with respect to original unchanged is equal to 2.9 % computed for the whole city.

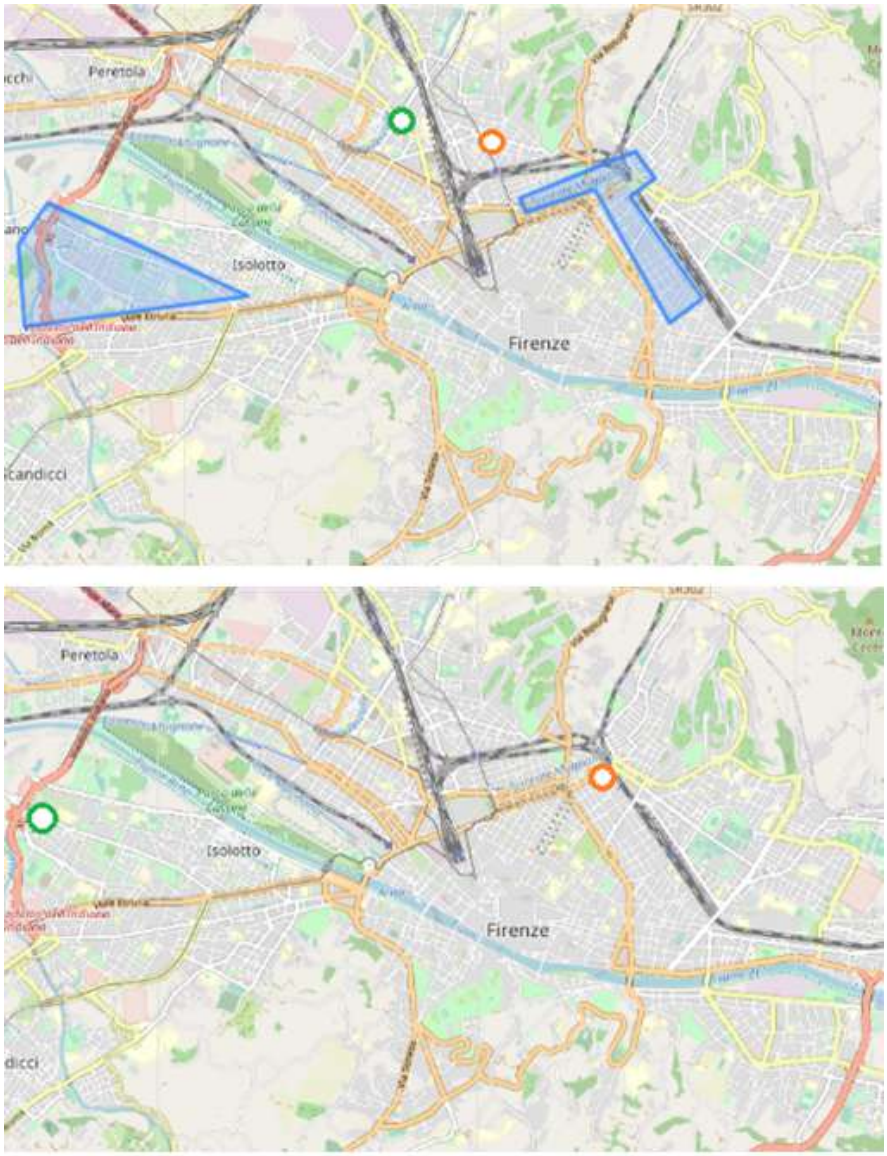


Figure 5.12: Connectivity KPI comparison: on the left the results of $\boldsymbol{KPI_C}(SC_{\boldsymbol{j},\widehat{\boldsymbol{T'}}})$, on the right the ones of $\boldsymbol{KPI_C}(SC\_UC_{\boldsymbol{j},\widehat{\boldsymbol{T'}}})$. The nodes assuming highest betweenness and centrality are depicted in orange and green, respectively.

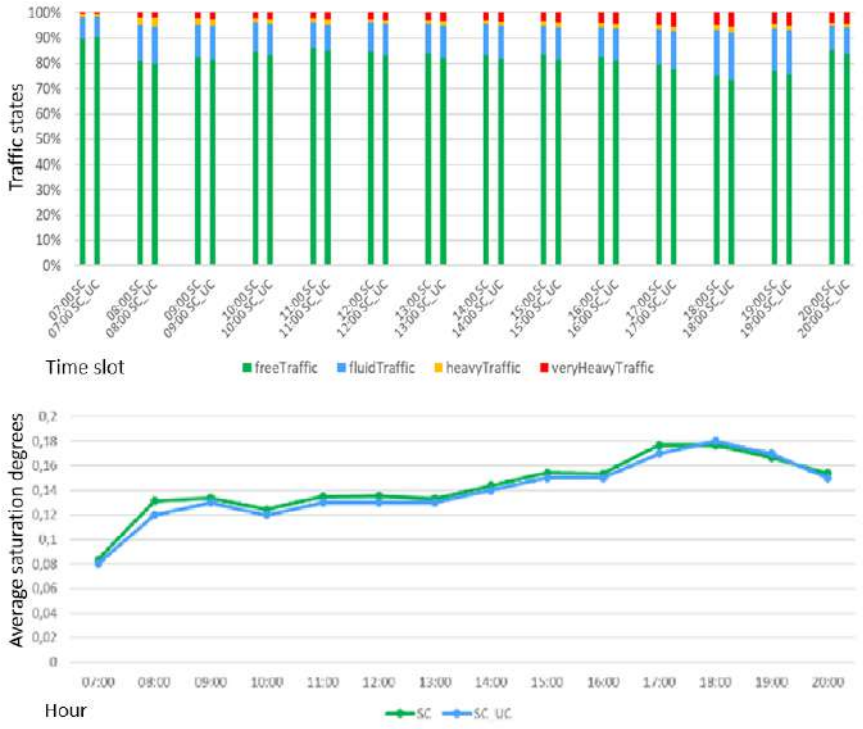


Figure 5.13: Graphical comparison between $\boldsymbol{KPI_F}(SC_{\boldsymbol{j},\widehat{\boldsymbol{T}'}})$ and $\boldsymbol{KPI_F}(SC\_UC_{\boldsymbol{ji},\widehat{\boldsymbol{T}}})$. In the top, the traffic states are compared for each time slot. In the bottom, the saturation degree values are depicted for the whole city due to the wideness of the involved area. The green line is related to $SC_{\boldsymbol{ij},\widehat{\boldsymbol{T}}}$ and it represents a slight increment with respect to the blue line related to $SC\_UC_{ij,\widehat{\boldsymbol{T}}'}$.

5.7 Performance Assessment

From the point of view of the performance assessment, the major costs are devoted to the algorithms for traffic flow reconstruction also when they are exploited as simulation. As described about, the complexity is due not only to the traffic flow reconstruction, but also to the computation of the road graph, and to the computation of the TDM. On the other hand, the present model overcomes the limitation of the solutions proposed in literature which are based on agents simulation models (as above described). Most of them present relevant limitations on the dimension of the road network graph and on the number of changes they can afford. While in most real cases of What-If analysis, the considered graph may be very large and multiply connected as above presented, thus changing large number of road segments from the original graph.

When large network graph simulations have to be addressed, then high computational costs and considerable memory usage are required, as in the What-If analysis presented in [141]. On the other hand, What-If analysis framework approach presented in this chapter is based on a fluid dynamics model of vehicular traffic via LWR PDE model and equation maintains the same computational cost presented in [40] also when large blocking area scenarios are defined.

The computational speed of the solution depends on the dimension of the considered road network. In particular, the computational complexity is an O(H(V+U)) where: V is the number of nodes, U is the number of units (consisting of a roads segmentation of about 20 mt) and H is the number of iterations. Since U is larger than V, then we definitively have an O(HU). Therefore, a small modification in the urban graph by considering modest no-go areas does not meaningfully change the computational speed.

Concerning the estimation of computational cost of our model, the scenario "WideRestructionOct2021" is considered. Such a scenario admits a suitable road graph of the metropolitan network of Florence constituted by 1463 junctions (representing the number V), and a total amount of units U=32299 occurring in the network when the related blocking constraints are considered. The given scenario consists of 14 sequential hours in which the computation has to be performed in continuous reconstructing the traffic model determining a certain traffic situation for each hour of the selected scenario, from 2021-10-22T07:00 to 2021-10-22T20:00. By using a machine having (CPU) 20 cores at 2.20 GHz and 128 GB Ram, the execution time

has been obtained as mean value of results taken on 10 distinct executions of the scenario "WideRestructionOct2021" It admits an average time execution for the complete performance of the selected scenario equals to 16 minutes and 52 seconds and each reconstructed scenario of one hour takes about 62.43 seconds for its computation. This largely overcome the state of the art solutions.

5.8 Final Considerations

The formalization of What-If analysis solutions and architectures is a complex task that presently could address multiple techniques from modeling, simulation, predictions, and KPI. The solutions have to cope with high complex situations of city scenarios addressing unexpected and planned events. A good What-If analysis starts always from the formal definition of scenarios, the estimation of predictions to be used in the simulations and the production of integrated simulations and KPI which can allow to provide support for decision makers. In this chapter, a solution for What-If analysis has been proposed and validated with the major focus on traffic flow which has a strong impact since most of the simulations in the context of the cities are based traffic flow, including: parking, pollutant, people flow, accidents, commercial sites, tourism, etc. As in this context, the contributions of this chapter are: (i) the definition and formalization of the What-If analysis framework, including formalization of What-If scenarios with multiple connected areas; (ii) the definition and implementation of large traffic flow reconstruction and simulation against What-If scenarios (addressing the changes into the road graphs, on the number of the operating sensors; and automatically estimating the redistribution of traffic on modified crossroads); (iii) the validation of the traffic flow reconstruction against complex What-If scenarios obtaining high precision; (iv) the high performance obtained in the What-If analysis providing traffic flow predictions on large changes on city road traffic. The architecture of the What-If analysis tools has been based on Snap4City framework, in which the business logic is defined by using Node-RED and Snap4City MicroService Libraries. Point (ii) extended the solution for traffic flow reconstruction at the state of the art by (a) dynamically reshaping the road graph network on the basis of the scenarios with multiply connected critical areas, (b) computing multiple reconstructions in consecutive time slots taking into account the evolution of road graph, junction

redistribution and of traffic flow data, (c) computing and comparing traffic flow KPI at the support of the What-If analysis, considering the complexity of their comparison since sensors and roads may be involved into the blocked areas as well.

Chapter 6

SMARTBED: Linking Automation to artificial Intelligence to reveal sleep Dysfunctions

Insomnia consists of a condition of dissatisfaction related to the quantity or quality of sleep. Just as daytime experiences affect sleep, sleep also affects daytime activities; in fact, insomnia correlates with problems with concentration or performance ability or absenteeism from work. SMARTBED is a project with the purpose of creating an integrated platform for the evaluation of sleep quality in the general population. The project aims to develop smart mattresses that can identify insomnia, thereby reducing its socio-economic cost and increasing individual well-being. In this chapter, we deal with studying and developing a mobile application for the smart bed and then analysis and design, communication, mobile data collection, data management, user management with the server, statistical views and evaluation, menus and suggestions. The proposed solution is GDPR compliant and provides user interfaces with statistical data and advanced statistical data about a mattress for recorded nights for both the end-user and doctors. The architecture of the proposed solution was based on Snap4City in which the business logic is defined using NodeRED

and Snap4City MicroService Libraries [1].

6.1 Introduction

SMARTBED is a project with the purpose of creating an integrated platform for the evaluation of sleep quality in the general population. The SmartBed idea is the study and realization of smart mattresses that bring within them several sensors of various types to monitor the sleep of patients/users (movements, heart, breath, sounds, etc.). The University of Florence, Department of Information Engineering (DINFO) - DISIT Laboratory, has dealt with 1) Study and develop a mobile application for the smart bed and then analysis and design, communication, mobile data collection, data management, user management with the server, statistical views and evaluation, menus and suggestions. 2) Study and develop the central software and then analysis and design, user management, data collection, statistical views, user interface, advanced data views. 3) General activities: testing, validation assistance. The integrated system can collect and provide data according to the type of users, and provide data that may be useful to researchers. The data are, in the various contexts, shown by day, week, month and year. They are aggregated and shown in statistical form, both in time and by geographical location and user characteristics (gender, age, habits, etc.). The specific data of the sensors are shown in an aggregated way to the users and can be turned on individually and in an aggregate way for researchers and/or physicians.

This chapter is structured as follows: In Section 6.2, a review of related work hardware and software for data visualization. In Section 6.3, the general architecture of the solution is presented, in 6.4 the process of raw data transfer on the server is described, in 6.5 data ingestion on Snap4City platform, and in Section 6.6 a possible data visualization with a mockup app. Finally, Section 6.7 reports final considerations.

[1] *Acknowledgments:* Our thanks goes to LAID - SMART BED project and partners for which we have developed a part of the solutions described in this chapter, and Regione Toscana for the partial founding Por Fesr 2014-2020.

6.2 Related Work

The SMARTBED concept is not new, there are several solutions on the market, which have different types of precision, and sometimes very high costs, see for example SleepNumber. In some cases, SMARTBED can also have actuators to manage the movement of the head and legs, to produce sounds, and/or vibrations, etc. Others present summary information derived from the measurements directly on smartphones or on screens integrated into the bed. This type of solution, which sees the bed instrumented with sensors, is part of the Internet to things approach. As low-cost solutions, smartphone apps for sleep monitoring are quite popular. Some of them are based on the presence of a bracelet and/or watch that can detect the heartbeat, movements, etc; such as FitBit, Jawbone UP. Others ask the user to place the smartphone on the mattress, such as Sleep as Android, MySleepBot, SleepCycle, etc. In the latter case, the sensors of the smartphone are directly exploited, recording not only the vibrations but also its own, etc. These systems are typical of low quality and have big problems with accuracy and reliability. In some cases, some mobile applications try to integrate the evaluation of sleep with information regarding the consumption of food, water, alcohol, coffee; and also sports activity, or not during the day, and more. Other mobile applications add functions to facilitate sleep such as music, sounds, lights, etc.; or to stimulate the interruption of snoring, generate the alarm clock at better times, etc. The professional methods are mainly based on detection tools that must be worn by the patient such as chest belts, bracelets, child mats, EEG, etc. These solutions are often too intrusive to be used directly at the user's home and without affecting the measurement itself.

On technology websites, such as WIRED in the article [11] it talks about sleep trackers with the functions and hardware in Table 6.1:

Functions	Hardware
heartbeat	Smartband
motion detection	Smartwatch
sleep disorders	Helmet
breathing	Radio frequency sensor
accelerometer	
movement	
oxygen level	

temperature	
hormones	
sleep apnea	
insomnia	
breathing disorders	
brain waves	
how deeply you slept	
how many times he got nervous	

Table 6.1: Functions and hardware in the WIRED article.

On medical websites, such as paginemediche in the article [5], [4] it talks about sleep trackers with the functions and hardware in Table 6.2:

Functions	Hardware
heartbeat	sensors on the mattress
brainwaves	
oxygen level in the blood	
sleep duration	
quality of sleep	
the time you spend waking up in bed before you sleep	
night movements	
breath frequency	
sleep phases and cycles	

Table 6.2: Functions and hardware in paginemediche article.

As regards the most used and reviewed apps are (in Table 6.3 there are the details of the features, device, rating e sensor):

- **Pillow Automatic Sleep Tracker**: "Landscape mode gives you immediate access to incredibly detailed statistics all the way back to your

Figure 6.1: Screenshot app - Pillow automatic sleep tracker.

first sleep session. Connect Pillow with Apple's Health app to compare your Sleep quality with various health metrics like your weight, caffeine, blood pressure and more" [6]. Screenshot of the app shown in Figure 6.1

- **Sleep Cycle**: "Sleep Cycle tracks and analyzes your sleep, waking you up at the most perfect time, feeling rested." [10]

  Screenshot of the app shown in Figure 6.2

- **Sleep as Android: Cicli del sonno, Sveglia**: "Unlocks the Sleep as Android application - the alarm clock with sleep cycle tracker. This is not a subscription but a lifetime license, eligible to be added to Google Family library. Install this "Unlock" and enjoy all features. Sleep as Android is a smart alarm clock with sleep cycle tracking. Wakes you gently in optimal moment for pleasant mornings." [8] Screenshot of the app shown in Figure 6.3

- **Runtastic Sleep Better**: "Track your sleep cycle, monitor dreams, improve bedtime habits, sleep patterns & wake up better with the free Sleep Better sleep tracker app with smart alarm clock from Runtastic! Sleep Better sleep cycle app offers you a simple and engaging way to get better sleep using a sleep tracker, sleep timer and sleep clock." [9] Screenshot of the app shown in Figure 6.4

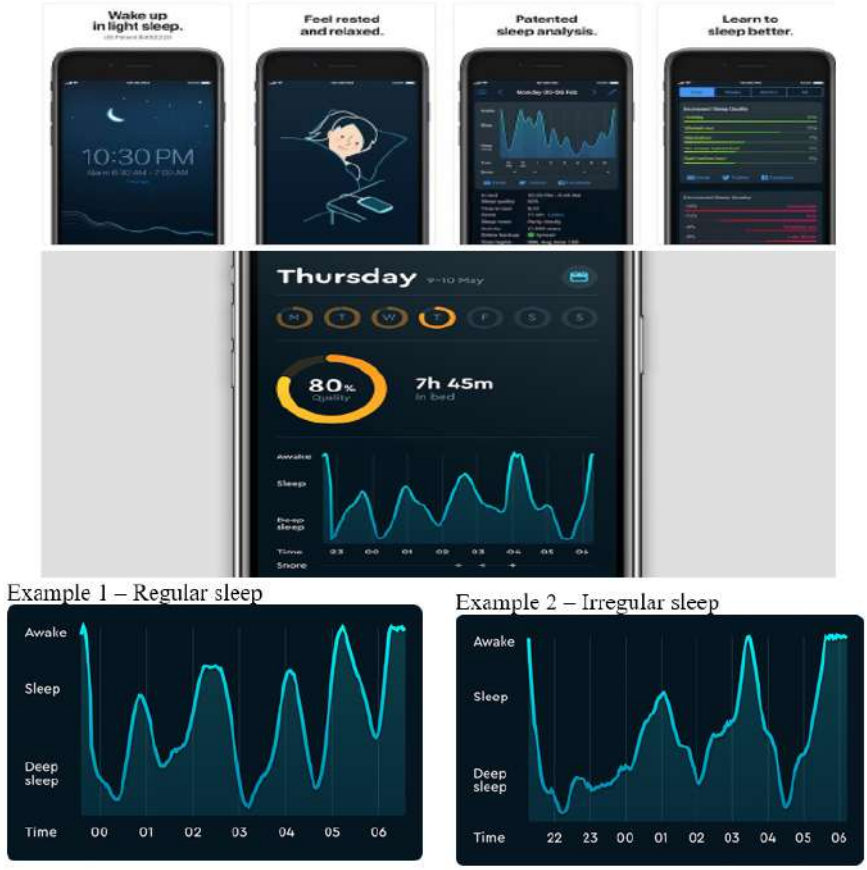


Figure 6.2: Screenshot app - Sleep cycle.

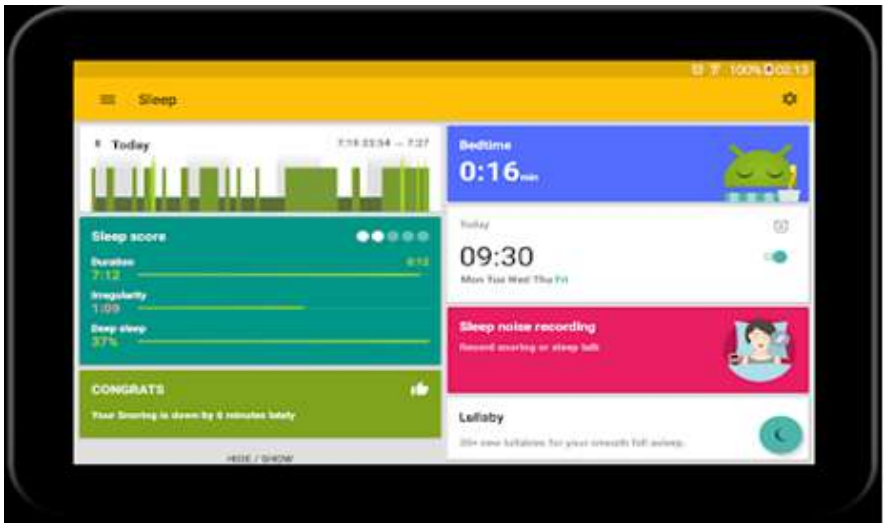


Figure 6.3: Screenshot app - Sleep as android.

Figure 6.4: Screenshot app - Runtastic Sleep Better.

App	Rating	Device	Sensor	Information per user
Pillow Automatic Sleep Tracker	#27 in Health & Fitness 4.4, 46.2K Ratings	iOS	Apple Watch	FREE FEATURES: ⋆ track your sleep automatically ⋆ sleep & heart rate analysis ⋆ compare your sleep quality with apple health ⋆ listen to important audio events during sleep ⋆ snooze lab ⋆ detailed sleep history and sleep trends ⋆ backup and synchronize ⋆ mood reporting & sleep notes ⋆ sleep aid melodies and wake up sounds PREMIUM FEATURES: ⋆ Unlimited sleep history ⋆ Sleep trends Sleep notes ⋆ Heart rate analysis ⋆ Listen to your sound recordings ⋆ Access to all wake up melodies & sleep aid sounds ⋆ Snooze lab: Personalised sleep tips ⋆ Compare your sleep quality with Apple Health metrics ⋆ Wake up melodies from your iTunes library ⋆ Runkeeper integration ⋆ Database export in CSV format
Runtastic Sleep Better	PlayStore 4,1	Android	Device by your sleep pillow	FREE FEATURES: ⋆ Sleep monitor: ⋆ Works in airplane mode ⋆ Sleep timer ⋆ Track caffeine & alcohol consumption ⋆ Monitor moon phases and find out if it impacts your sleep cycle. ⋆ Keep a dream & sleep diary to track your dreams (good, bad, or neutral) ⋆ Note your mood when you wake up directly in your sleep diary. ⋆ Sleep track with your tablet ⋆ Share sleep tracking sessions. PREMIUM FEATURES: ⋆ Use the Smart Alarm Clock to wake up at the ideal time within your personalized wake Up Window. ⋆ Enjoy a variety of smart alarm sounds & snooze functionality.⋆ View sleep history including daily stats and overviews for a longer time ⋆ Filter history taking daily variables

Sleep Cycle	APP STORE #16 in Health & Fitness 4.7, 205.3K Ratings PlayStore 4,5	iOS / Android	phone on mattress	FREE FEATURES: ⋆ Sleep analysis with Sleep Cycle patented sound technology or accelerometer ⋆ Detailed sleep statistics and daily sleep graphs ⋆ Fully integrated with Apple Health, exchanging sleep analysis and heart rate ⋆ Carefully selected alarm melodies ⋆ Snooze by shaking or double-tapping the phone lightly ⋆ Customizable wake up window. From instant (regular alarm clock), up to 90 minutes PREMIUM FEATURES: ⋆ Sleep Aid - Our library of sleep stories, relaxation guides and calm sleep sounds is specially designed to help you fall asleep easier. (Mindfulness & meditation) ⋆ Trends – gather long-term trends on your sleep patterns ⋆ Comparison Data - Compare your sleep patterns to world sleep statistics ⋆ Snore Recorder & Trends – Capture snoring behavior and view historical snore trends data ⋆ Weather & Sleep - See how different types of weather affect your sleep quality ⋆ Heart Rate Monitor - Measure your heart rate (RHR) every morning using the built-in camera in your device ⋆ Sleep Notes - See how events such as drinking coffee, stress, working out, or eating late affect your sleep quality ⋆ Wake up Mood - See how Sleep Cycle affects your wake up mood over time ⋆ Online Backup - Let's you secure your sleep data online ⋆ Data Export – Download sleep data to Excel for detailed analysis ⋆ Philips HUE Light Bulb Support - Simulate a natural sunrise to give you an even softer wake up

Sleep as Android: Cicli del sonno, Sveglia	PlayStore 4,6	Android	phone on mattress	PREMIUM FEATURES: ⋆ Sleep cycle tracking with smart wake up ⋆ Sleep graph history ⋆ Google Fit, S Health ⋆ Pebble, Android Wear, Garmin Connect IQ ⋆ Sleep deficit, deep sleep and snoring statistics ⋆ Social sharing (FaceBook, Twitter) ⋆ Gentle volume nature sound alarms (birds, sea, storm...) ⋆ Music playlists from alarm ⋆ Nature sound lullabies with binaural tones for fast fall asleep ⋆ Never oversleep again with CAPTCHA wake up verification (Math, Sheep counting, Phone shaking, Bathroom QR code or NFC tag scanning...) ⋆ Sleep talk recording, snoring detection and anti-snoring ⋆ Jet lag prevention

Table 6.3: Details of the features, device, rating e sensor in the app.

In Table 6.4 regarding some hardware solutions

Name	Price	Type	Features
Rhythm Dreem Headband	450 €	Helmet	⋆ Measuring and scoring sleep ⋆ valuation Insomnia and CBT-I
SleepScore Max	145 €	Monitor / sensor	⋆ Smart Alarm ⋆ Set Goals ⋆ Sleep History ⋆ Sleep Chart ⋆ Sleep Report For Doctor
EverSleep	179 €	Bracelet	Blood Oxygen ⋆ possible sleep apnea ⋆ Movement ⋆ restless sleep ⋆ Snoring ⋆ airway obstructions ⋆ Pulse Rate - autonomic disturbance ⋆ Insomnias

Table 6.4: Name, price, type and features for hardware solutions.

6.3 Architecture and processes

The purpose of the project is to provide a user with interfaces with basic and advanced statistical data related to a mattress for the recorded nights. There are two possible scenarios:

FINAL USER: a user through an app, log in and have views of their data for each night recorded by the mattress.

DOCTORS: a doctor through an app, log in and have available views of their patient data for each night recorded by the mattress.

User data are stored in a centralized database on a server, where they are processed. Delegations to view the data for each device are managed using the Snap4City tool.

The architecture shown in Figure 6.5 is divided into three stages:

- Raw data transfer on the server
- Data processing and ingestion on snap4city platform
- Data visualization

In the following sections, the process for each stage is described.

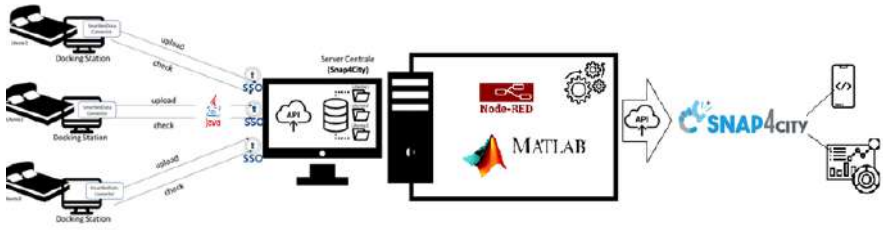


Figure 6.5: System architecture.

6.4 Raw data transfer on server

The SmartBedDataConnector application is designed to perform the following actions in a scheduled way: - It takes the files generated by the docking station and sends them to the server every 10 minutes. Every 10 minutes it checks if there are new files and sends them to the server through an authenticated access (SSO) to a server (SMARTBED_UPLOAD); - Every hour it sends the name of the files generated in the last 3 months and checks if they are all present on the server. If it finds any missing, it sends the file. Sending is done through authenticated access (SSO) to a server. (SMARTBED_CHECK); The file generated by the docking station with the sensor data must be in the form ID_SMARTBED_S_.

The following items describe the installation and configuration procedure of the application.

1. Unzip the SmartBedDataConnector.rar file into C:

2. In C:
   SmartBedDataConnector open the file userId.properties with a text editor and fill in as follows:

   smartBedId=YOUR_ID_SMARTBED

   userSnap= YOUR_USERNAME_SNAP4CITY

   passwordSnap= YOUR _PASSWORD_SNAP4CITY

3. In C:
   SmartBedDataConnector open the file parameter.properties with a text editor and fill in as follows:

client_id=java-smartbed

url = `https://fileserver.snap4city.org/smartbed`

patchDir = C://Users/ipsaro/Desktop

keyclock.endpoint=`https://www.snap4city.org/auth/realms/master/protocol/openid-connect`

patchDirLog = C://SmartBedDataConnector

4. Open the Windows Scheduler:
   a. Press the WIN+R keys to open the form to execute the commands
   b. Write taskschd.msc and press OK
   c. The Planning Utility will open
5. Schedule the UPLOAD activity of the files generated by the docking station by following these steps:
   a. Open the Action Menu and select Create Basic Activity
   b. Enter a Name, for example, SMARTBED_UPLOAD. Click Next
   c. In the Activation field, select When starting the computer. Click Next
   d. In the next step specify Start program and click Next;
   e. In the Program or script field: indicate the path of the program
   f. lunchUPLOAD.vbs, for example, C://SmartBedDataConnector/lunchUPLOAD.vbs. Click Next
   g. On the next screen, select "Open the Properties dialog box when Finish is selected".
   h. In the Activation Tab click on Edit and activate Repeat activity every: 10 minutes for the duration of Unlimited.
6. Open the Windows Scheduler:
   a. Press the WIN+R keys to open the form to execute the commands
   b. Write taskschd.msc and press OK
   c. The Planning Utility will open
7. Schedule the CHECK activity of the files generated by the docking station by following these steps:

a. Open the Action Menu and select Create Basic Activity
b. Enter a Name, for example, SMARTBED_ CHECK. Click Next
c. In the Activation field, select When starting the computer. Click Next
d. In the next step specify Start program and click Next;
e. In the Program or script field: indicate the path of the program
f. lunchCHECK.vbs, for example, C:// SmartBedDataConnector/ lunchCHECK.vbs. Click Next
g. On the next screen, select "Open the Properties dialog box when Finish is selected".
h. In the Activation Tab click on Edit and activate Repeat activity every: 1 hour for the duration of Unlimited.

6.5 Data processing and ingestion on Snap4City platform

The server receives the files with name
[Id_SmartBed]_S_[Date-Format_YYYY_M_DD_HH_MM_SS]
in the folder Id_SmartBed/Date-Format_YYYY-MMM-DD.
Example in Table 6.5:

Name	Dimension
N003_S_17-Jan-2019_2019_1_17_12_10_35.mat	196 Kb
N003_S_17-Jan-2019_2019_1_17_12_11_35.mat	196 Kb
N003_S_17-Jan-2019_2019_1_17_12_12_35.mat	196 Kb
N003_S_17-Jan-2019_2019_1_17_12_13_35.mat	196 Kb
N003_S_17-Jan-2019_2019_1_17_12_14_35.mat	196 Kb
N003_S_17-Jan-2019_2019_1_17_12_15_35.mat	196 Kb

Table 6.5: Example files in the server.

FREQUENCY: Data is recorded every minute since the start of recording. (The frequency in seconds is given in the blocksize structure and can be set

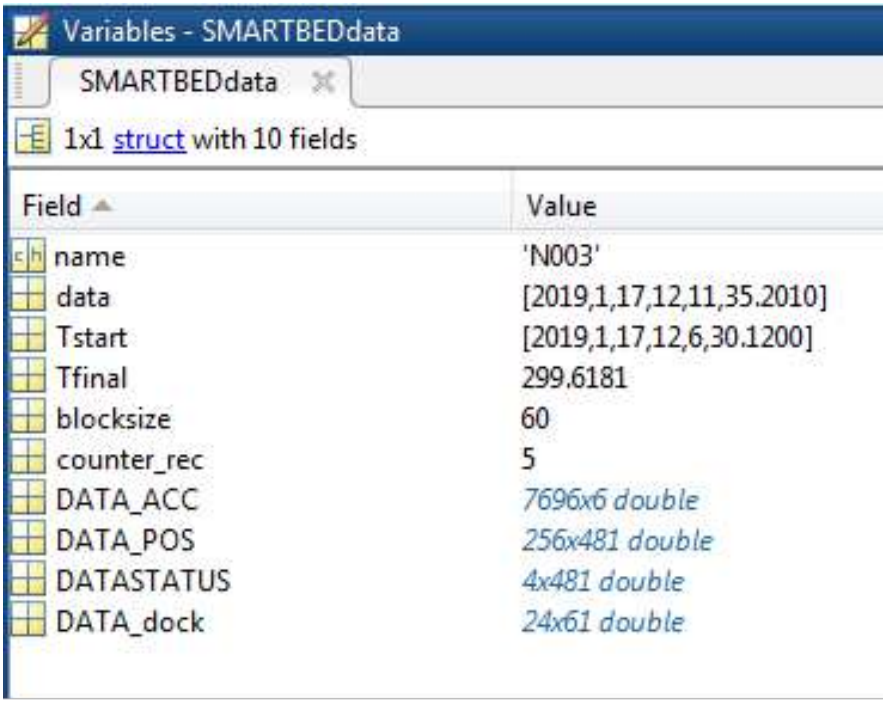


Figure 6.6: Structure of the MAT file.

in the docking station.) Figure 6.6 presents the structure of the MAT file and Table 6.6 provide the explanation.

Field	Type	Description	Example
name	char	username SMARTBED	N003
data	1X6 double	Datetime of when the block was recorded, in the format [YYYY,MM,DD, HH,MM,SS.MMMM]	[2019,1,17,12,11, 35.201000000000000]
Tstart	1X6 double	Recording start datetime	[2019,1,17,12,6, 30.120000000000000]

Tfinal	double	Actual time in seconds from the start of recording to saving the data block. Difference in seconds between date and Tstart (should be multiple blocksize but sometimes unpredictable latencies are present)	299.6181 sec
blocksize	double	Duration in seconds of the data storage interval (set in the docking station)	60 sec
counter_rec	double	Progressive index indicating the data block number (the first data block has counter_rec=1)	5
DATA_ACC	7696X6 double	The size of the rows is variable depending on the seconds acquired Accelerometric data, (acceleration according to 3 axes X, Y and Z) timeXchannel matrix (sampling freq =128Hz) the matrix has size: time x (N*3) Where N is the number of accelerometers enabled in acquisition. Frequency: 128 measurements per second	

DATA_POS	256X481 double	The size of the columns is variable depending on the seconds acquired Position data, 256Xtime matrix (sampling freq =8Hz) A vector of length 256 is saved at each instant of position sampling (fs=8Hz). The length 256 is due to the fact that the position matrix is organized in 16X16 Frequency: 8 measurements per second	
DATASTATUS	4X481 Double	The size of the columns is variable depending on the seconds acquired Data coming from the serial of the acquisition card with the flags related to the single packets received The 4 saved flags related to each packet sent from the serial (packets are sent with fs 8 Hz) are in order: 1. COUNTER 2. SIZE 3. SERVICE 4. RESPONSE Frequency: 8 measurements per second	

DATA_dock	24X61Double	The size of the columns is variable depending on the seconds acquired Environmental data, 24Xtime matrix. Frequency: 1 measurement per second	intensity vector size 24 20 rows: loudness. Ex 0.99200 1 row: temperature Ex 24.6200000000000 1 row: atmospheric pressure Ex 1.01627 1 row: humidity Ex: 35.7100000000000 1 row: brightness: Ex 0

Table 6.6: Explanation MAT Structure.

On the size of the matrices N rows of DATA_ACC / Frequency = N columns of DATA_POS / Frequency = N columns of DATASTATUS/ Frequency.

The relevant units of measurement are as follows:

- Position matrix: resistive value of the sensorized tissue

- Accelerometers matrix: value in mV. However, these measurements above are blind to the user so no units are needed.

6.5.1 Data Storage

Size in Byte For 1 day

Time/User		**1,00**	**10,00**	**100,00**	**1.000**	**10.000**	**100.000**	**1.000.000**
	1 min	1,960E+05	1,960E+06	1,960E+07	1,960E+08	1,960E+09	1,960E+10	1,960E+11
60	**1h**	1,176E+07	1,176E+08	1,176E+09	1,176E+10	1,176E+11	1,176E+12	1,176E+13
24	**1 d**	2,822E+08	2,822E+09	2,822E+10	2,822E+11	2,822E+12	2,822E+13	2,822E+14
30	**30 d**	8,467E+09	8,467E+10	8,467E+11	8,467E+12	8,467E+13	8,467E+14	8,467E+15
12	**1 year**	1,016E+11	1,016E+12	1,016E+13	1,016E+14	1,016E+15	1,016E+16	1,016E+17

For 8 h every day

Time/User		**1,00**	**10,00**	**100,00**	**1.000**	**10.000**	**100.000**	**1.000.000**
	1 min	1,960E+05	1,960E+06	1,960E+07	1,960E+08	1,960E+09	1,960E+10	1,960E+11
60	**1h**	1,176E+07	1,176E+08	1,176E+09	1,176E+10	1,176E+11	1,176E+12	1,176E+13
8	**1 d**	9,408E+07	9,408E+08	9,408E+09	9,408E+10	9,408E+11	9,408E+12	9,408E+13
30	**30 d**	2,822E+09	2,822E+10	2,822E+11	2,822E+12	2,822E+13	2,822E+14	2,822E+15
12	**1 year**	3,387E+10	3,387E+11	3,387E+12	3,387E+13	3,387E+14	3,387E+15	3,387E+16

Considering the data acquisition for 8h per night/day (every 24) for 100 users we have about 280 Gbytes per month (3000 nights).

6.5.2 Data processing

Each night is processed through a Matlab script. This script receives as input all the recorded Matlab structures for the whole night. Must configure this script with the following parameters addpath(genpath('PATH of the script Matlab')); % path of scripts data_path = 'PATH of the night records folder '; % path of data (nights) results_path = 'PATH results folder '; % saving results path ID = 'id SMARTBED'; % ID of SMARTBED EPO = 300; seconds of the analysis time window (sec)

The following files were generated for each user and each night

- DD-mmm-YYYY_results_user.dat [1 X 19]
- DD-mmm-YYYY_results_expert.csv [1 row every EPO X 16]
- DD-mmm-YYYY_date.csv [1 row every EPO X 1]

In the file DD-mmm-YYYY_results_user.dat [1 X 19] 19 values are generated, separated by a comma, the first 13 are physiological indexes and the remaining 5 environmental indexes:

1. Rate waking time vs. total bedtime (%)
2. Rate of Non-Rem sleep time compared to total time in bed (%)
3. Rate of time in sleep Rem compared to total time in bed (%)
4. Rate of Non-Rem sleep time compared to total sleep time (%)
5. Rate of sleep time Rem versus total sleep time (%)
6. Total sleep time (min)
7. Sleep efficiency (%)
8. Sleep time –> sleep latency (min)
9. Rem sleep latency (min)
10. Awakenings after falling asleep –> WASO (min)

11. Average heart rate during sleep (bpm)
12. Average respiratory rate during sleep (bpm)
13. SMARTBED sleep quality index (ordinal scale from 0 to 6 (6 levels): 0 –> very bad sleep quality,–> very good sleep quality)
14. Flag room temperature (-1: temperature too low, 0: optimum temperature, +1: temperature too high)
15. Flag room humidity (-1: humidity too low, 0: optimum humidity, +1: humidity too high)
16. Flag brightness room (0: optimum brightness, +1: too high brightness)
17. Flag room noise (0: an optimal noise level, +1: too high noise level)
18. Flag environmental quality for sleep (0: optimal environment for sleep, 1: a non-optimal environment for sleep)
19. Date night

In the file DD-mmm-YYYY_results_expert.csv [1 row every EPO X 16] is a Matlab structure and contains :

- par_smartbed: contains data related to the mattress sensors and specifically:
  - par_smartbed.BBI = average heart rate in the "EPO" window estimated by accelerometers;
  - par_smartbed.RESP_pos = average respiratory rate in the "EPO" window estimated by accelerometers;
  - par_smartbed.Ys = average accelerometric intensity in the "EPO" window estimated by accelerometers;
  - par_smartbed.pos_m = average of the respiratory matrix signal evaluated in the "EPO" window estimated by the position matrix;
  - par_smartbed.Ys_s = variance of the accelerometric intensity evaluated in the "EPO" window estimated by the accelerometers;
  - par_smartbed.pos_s = variance of the respiratory matrix signal evaluated in the "EPO" window estimated by the accelerometers;

- par_env: contains data related to environmental sensors and specifically:
  - par_env.ENVm = contains the average values in the "EPO" windows of SOUND, TEMPERATURE, ATMOSPHERIC PRESSURE, % HUMIDITY, LUMINOSITY (in order);
  - par_env.ENVs = contains the variance values in the "EPO" windows of SOUND, TEMPERATURE, ATMOSPHERIC PRESSURE, % HUMIDITY, LUMINOSITY (in order);

In the file DD-mmm-YYYY_date.csv [1 row every EPO X 1] there is the DateTime for each EPO.

6.5.3 Data Ingestion

For each SMARTBED id and therefore for each mattress, two devices are automatically created on Snap4City.

- An IoT device of model Smart Bed named SmartBed_idsmartbed_KPI receives the data for the entire night. The model has a refresh_rate of 86400 seconds and has the structure in Table 6.7:

value name	data type	value type	value unit
noiseFlag	float	noise_flag	#
environmentalQualityFlag	float	enviromental_quality_flag	#
dateObserved	timestamp	timestamp	s
wakeTimeAfterSleepOnset	float	wake_time_after_sleep_onset	min
averageHeartRate	float	average_heart_rate	bpm
averageRespiratoryRate	float	average_respiratory_rate	bpm
sleepQualityIndex	float	sleep_quality_index	#
temperatureFlag	float	temperature_flag	#
humidityFlag	float	humidity_flag	#
brightnessFlag	float	brightness_flag	#
sleepEfficiency	float	sleep_efficiency	%
asleepTime	float	asleep_time	min
remSleepLatency	float	rem_sleep_latency	min
nonremTimeWRTTotalSleepTime	float	nonrem_time_wrt_total_sleep_time	%
remTimeWRTTotalSleepTime	float	rem_time_wrt_total_sleep_time	%
totalSleepTime	float	total_sleep_time	min
wakeTimeWRTTotalBedTime	float	wake_time_wrt_total_bed_time	%
nonremTimeWRTTotalBedTime	float	nonrem_time_wrt_total_bed_time	%
remTimeWRTTotalBedTime	float	rem_time_wrt_total_bed_time	%

Table 6.7: Model device KPI in Snap4city.

- An IoT device of model SMARTBED named SmartBed_id_smartbed_night receives the data every 300 seconds for the entire night. The model has a refresh_rate of 86400 seconds and has the structure in Table 6.8:

value name	**data_type**	**value_type**	**value_unit**
varianceNoise	float	variance_noise	dB
varianceTemperature	float	variance_temperature	°C
varianceAtmosphericPressure	float	variance_atmospheric_pressure	hPa
varianceHumidity	float	variance_humidity	kg/m
varianceBrightness	float	variance_brightness	lux
dateObserved	time	timestamp	s
averageAtmosphericPressure	float	average_atmospheric_pressure	hPa
averageHumidity	float	average_humidity	kg/m
averageBrightness	float	average_brightness	lux
varianceAccelerometricIntensity	float	variance_accelerometric_intensity	mV
varianceRespiratoryMatrixSignal	float	variance_respiratory_matrix_signal	bpm
averageNoise	float	average_noise	dB
averageTemperature	float	average_temperature	°C
averageRespiratoryRate	float	average_respiratory_rate	bpm
averageAccelerometricIntensity	float	average_accelerometric_intensity	mV
averageRespiratoryMatrixSignal	float	average_respiratory_matrix_signal	bpm
averageHeartRate	float	average_heart	bpm

Table 6.8: Model device Night trend in Snap4city.

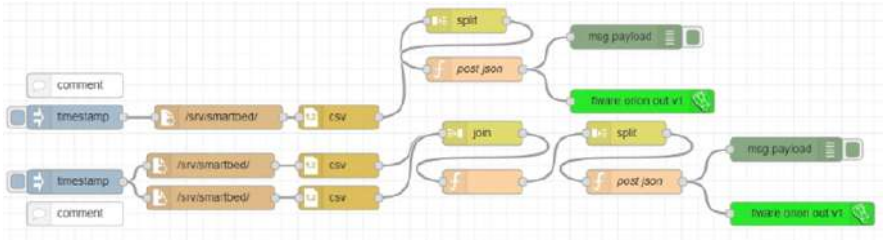


Figure 6.7: IotApp data ingestion.

Users and doctors are delegated to display their devices.

Data acquisition is performed through an IotApp (Figure 6.7) that reads the files generated by the Matlab and inserts them into the two devices

6.6 Data visualization

Once logged in with credentials (Snap4City username and password), any user can:

- View a summary of sleep quality, environmental quality and total sleep time for the night selected.
- View details and trends in heart rate, respiratory rate and intensity accelerometer (physiological variables) and sound, brightness, temperature, humidity and pressure atmospheric (environmental variables) for the selected night.
- View details regarding sleep: total duration, wakeful duration, rem phase duration and non-rem phase for the selected night.
- View the history from a selected date sleep quality, environmental quality, the total duration of the waking, rem and non-rem phases of the previous nights.

Following is the mockup for the app

LOGIN (Figure 6.8) In this screen the user can log in by entering your username and password. It is also possible (without login) to access the LAID - SMART BED project info (Figure 6.9).

Figure 6.8: Login screenshot.

Figure 6.9: Info screenshot.

Figure 6.10: Home screenshot.

PHYSIOLOGICAL DATA (Figure 6.10) In this screen, the user can select the night to be displayed. The sleep quality of the selected night is displayed (6 levels) with different colors (from very bad quality of sleep to very good quality of sleep). Below we have the environmental quality assessment green if the environment is optimal for sleep and red if the environment is not optimal for sleep. It also has the evaluation of brightness (green if optimal and red if too high), temperature (green if optimal and red if too high or too low), humidity (green if optimal and red if too high or too low) and finally noise (green if optimal and red if too high). Below it has an analysis of sleep in particular the duration in minutes and a graphic representation of the wakefulness duration of the rem phase and the non-rem phase.

Figure 6.11: Physiological data trend screenshot.

PHYSIOLOGICAL DATA TREND (Figure 6.11) In this screen, the user can select the night to be displayed. The sleep quality of the selected night is displayed (6 levels) with different colors (from very bad quality of sleep to very good quality of sleep). Below is displayed the hourly rate of heart rate, respiratory rate and accelerometric intensity.

ENVIRONMENTAL DATA TREND (Figure 6.12) In this screen, the user can select the night to be displayed. The sleep quality of the selected night is displayed (6 levels) with different colors (from very bad quality of sleep to very good quality of sleep). There is also the evaluation of brightness (green if optimal and red if too high), temperature, and finally noise.

HISTORICAL DATA (Figure 6.13)

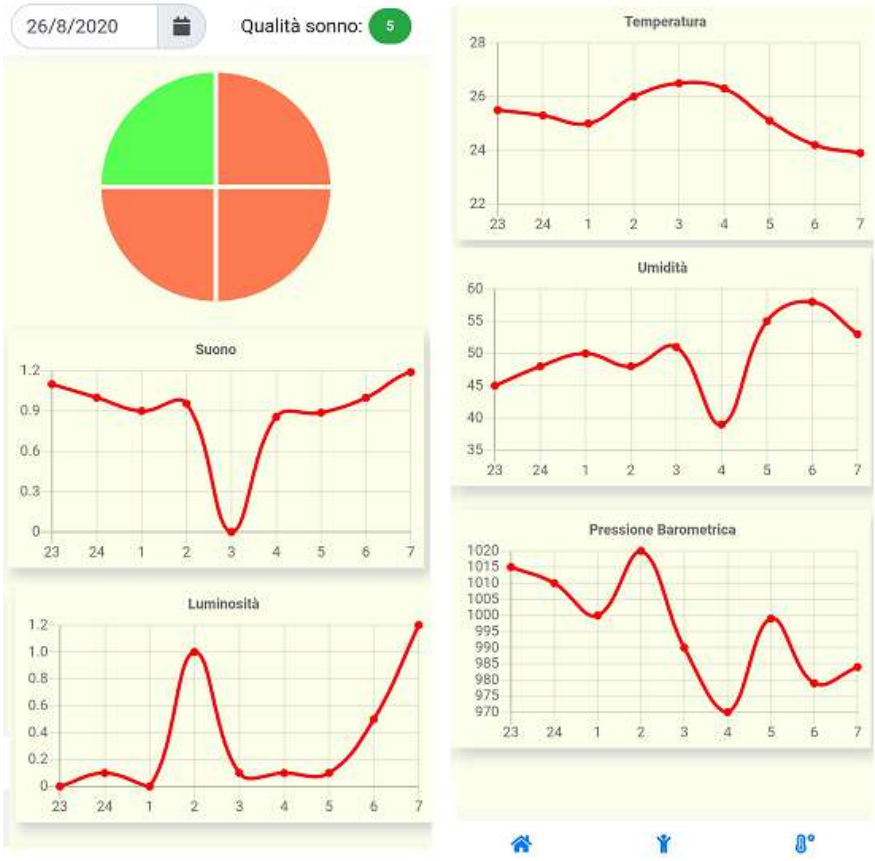


Figure 6.12: Environmental data trend screenshot.

Figure 6.13: Historical data screenshot.

In this screen, it is possible to view the sleep quality, the environmental quality and the sleep analysis of the nights before today. As far as sleep quality, the bar graph represents the 6 levels of sleep quality with different colors and lengths (from red bad sleep quality to green very good sleep quality). In the environment, there is the environmental quality assessment where green indicates an optimal situation while red indicates a non-optimal situation. In the sleep analysis there is a trend of the sleep duration and in the bar graphs the representation of the waking, non-rem sleep and the rem sleep phase.

6.7 Final Considerations

In this chapter, it has been studied and developed a mobile application for the smart bed. The development involved several stages. The first phase

involved the analysis and design of the app. Subsequently, the data collection and management phase began, and then the user management through the Snap4City platform. After, in accordance with medical guidelines, we moved on to statistical processing in order to assess the quality of sleep. Finally, data visualization. All phases focused on user privacy.

The proposed solution is GDPR compliant and provides user interfaces with advanced statistical data about a mattress for recorded nights for both the end-user and doctors. The architecture of the proposed solution was based on Snap4City in which the business logic is defined using NodeRED and Snap4City MicroService Libraries.

The research activity in this chapter has laid the foundations for future experimentation in the medical field by solving complex privacy and GDPR problems using the Snap4City tool.

Chapter 7

Conclusion

The research activity presented in this thesis has been carried out at the DISIT laboratory (Distributed Data Intelligence and Technologies) of the DINFO department (Department of Information Engineering) of the University of Florence. The activities were performed in the context of five projects:

- Feedback has the purpose of innovating CRM-retail products to evolve them to the use of advanced profiling for users and products with adaptive/predictive and personalized user engagement.

- SODA has the purpose of developing and implementing a system of integrated management and optimization of production processes and consumption of hydrogen and chlorine.

- PC4City has the purpose of developing a platform of Civil Protection through Km4City.

- Sii-Mobility's purposes are: (i) reduction of social costs of mobility, (ii) simplify the use of mobility systems, (iii) developing working solutions and application, with testing methods, (iv) contribute to standardization organs, and establishing relationships with other smart cities' management systems.

- LAID - SMART BED aims to combine automation with artificial intelligence to reveal sleep dysfunction.

In the research activity presented in this thesis, Smart Applications in various contexts developed. In chapter 2, recommendations were produced for people in the retail context, while in chapter 6, information concerning people's health status was generated. The information generated by the smart applications is covered by GDPR and therefore the data was treated with special attention in all its issues (user management, privacy management, data access, and behavior analysis). In chapter 3, in the industrial domain, privacy is not a relevant factor so the research focused on predictive aspects such as accuracy in prediction and XAI aspects on predictive maintenance problems. These techniques explain the correlation of failure with plant features to guide humans to possible maintenance solutions within the plant. The same approach was taken in chapter 4 where predictive algorithms and XAI techniques were studied to predict landslide events and to identify which features are most correlated to the event. Finally, in chapter 5 the research focused on aspects of What If analysis that combine together the previously discussed aspects. The research on What If has recently started, so the results are not yet consolidated in a published article, but we can see the reuse of predictive models, the definition of scenarios, and typical behaviors.

In the following items, the conclusions for each individual chapter are given.

- In chapter 2, a recommendation system in the context of fashion retail has been proposed and described, relying on a multi-level clustering approach of items and users' profiles in online and physical stores. The solution has been developed in the context of the Feedback project funded by Regione Toscana, and has been conducted on real retail company Tessilform, and it has been validated against real data from December 2019 to December 2020, showing that the use of the proposed recommendation tool generated stimulus to the customers which brought to an increase of buyers' attention and purchase increase of 3.48%. The solution proposed has demonstrated to be functional also in the presence of low number of customers and items (as happens in retail shops, in which the items are of high value), and when suggestions are mediated by the assistants, as happens in the fashion retail shops. Moreover, the proposed solution addresses and solved lacks and issues which are present in current state of the art tools, such as also the cold start problems in generating recommendations for newly ac-

quired customers, since it relies on rules mining techniques, allowing to predict the purchase behaviour of new users. Our solution is also GDPR compliant, addressing the current strict policies for users' data privacy, solving one of the main issues for managing users' demographic details.

- In chapter 3, a predictive maintenance model for classification of failures in a real industrial process has been presented. The proposed solution is based on a deep learning CNN-LSTM architecture, predicting the working status of the productive process in the Altair chemical plant. The proposed model CNN-LSTM provides a one-hour prediction of the plant status and indications on the areas in which the intervention should be performed by using explainable LSTM technique. Assessing the proposed method with real production data, experimental results show an average Accuracy of 91.8% and an average F1-score of 90%, which are very good results considering that the proposed model provides predictions of the plant working status one hour in the future, and it is capable of running in real time (thus aiming at resolving some lacks found in other state of the art solutions). The explanation of the predictions provides suggestions for the maintenance teams. The chapter also introduced business intelligence tools on maintenance data and the architectural infrastructure for the integration of predictive maintenance approach into the whole control and management system of ALTAIR industry 4.0 plant in the context of SODA and large renovation of the production infrastructure.

- In chapter 4, the problem of landslide event prediction has been addressed, for early warning. A careful review of related works and solutions proposed in literature has been performed, making a comparative analysis of their results, where possible. Traditional approaches are based on empirical algorithm as SIGMA, while most recent state of the art solutions are based on machine learning and deep learning approaches. Their main limitations are represented by the fact that these systems have a low reliability and they do not often provide interpretability of results. They do not apply a specific analysis of predictive outputs and features relevance, based for instance on explainable artificial intelligence techniques. To this purpose, this chapter reports the implementation, tuning and testing of four machine learning methods, based on Random Forest (RF), Extreme Gradient

Boosting (XGBoost), Convolutional Neural Networks (CNN) and Autoencoders (AE). These systems have been trained and validated by exploiting data collected in the context of the Metropolitan City of Florence since 2013 up to 2019; they have been compared with SIGMA decisional model, which is currently adopted in both Emilia Romagna and India. Comparative results showed that the method based on XGBoost achieved better results in terms of Sensitivity, MAE, MSE and RMSE. Moreover, a further analysis based on Shapley additive explanation (SHAP) has been carried out, globally and locally, for the XGBoost model which obtained best results. In this way, a deeper understanding of the predictive model outputs, as well as the relevance of features and their interdependency, has been provided.

- In chapter 5, a solution for What-If analysis has been proposed and validated with the major focus on traffic flow which has a strong impact since most of the simulations in the context of the cities are based traffic flow, including: parking, pollutant, people flow, accidents, commercial sites, tourism, etc. As in this context, the contributions of this chapter are: (i) the definition and formalization of the What-If analysis framework, including formalization of What-If scenarios with multiple connected areas; (ii) the definition and implementation of large traffic flow reconstruction and simulation against What-If scenarios (addressing the changes into the road graphs, on the number of the operating sensors; and automatically estimating the redistribution of traffic on modified crossroads); (iii) the validation of the traffic flow reconstruction against complex What-If scenarios obtaining high precision; (iv) the high performance obtained in the What-If analysis providing traffic flow predictions on large changes on city road traffic. The architecture of the What-If analysis tools has been based on Snap4City framework, in which the business logic is defined by using Node-RED and Snap4City MicroService Libraries. Point (ii) extended the solution for traffic flow reconstruction at the state of the art by (a) dynamically reshaping the road graph network on the basis of the scenarios with multiply connected critical areas, (b) computing multiple reconstructions in consecutive time slots taking into account the evolution of road graph, junction redistribution and of traffic flow data, (c) computing and comparing traffic flow KPI at the support of the What-If analysis, considering the complexity of their comparison since

sensors and roads may be involved into the blocked areas as well.

- In chapter 6, it has been studied and developed a mobile application for the smart bed. The development involved several stages. The first phase involved the analysis and design of the app. Subsequently, the data collection and management phase began, and then the user management through the Snap4City platform. After, in accordance with medical guidelines, we moved on to statistical processing in order to assess the quality of sleep. Finally, data visualization. All phases focused on user privacy. The proposed solution is GDPR compliant and provides user interfaces with advanced statistical data about a mattress for recorded nights for both the end-user and doctors. The architecture of the proposed solution was based on Snap4City in which the business logic is defined using NodeRED and Snap4City MicroService Libraries. The research activity in this chapter has laid the foundations for future experimentation in the medical field by solving complex privacy and GDPR problems using the Snap4City tool.

Appendix A

Acronyms

AE Autoencoders

AI Artificial Intelligence

AM Analytical Model

AN Adversarial Networks

API Application Programming Interface

ARIMA Auto Regressive Integrated Moving Average

AUC Area Under Curve

BPM Beats per minute

BPNN Backpropagation Neural Network

BSI Bare land soil index

CART Classification And Regression Tree

CBM Condition-based maintenance

CM Corrective Maintenance

CNN Convolutional neural networks

CNNFMO Convolutional Neural Network with Optimized Moth Flame Algorithm

COVID Corona virus disease

CRM Customer Relationship Management

CTI Compound Topographic Index Curvature

DAN Discriminative Adversarial Networks

DBN Deep Belief Network

DCS Distributed control systems

DEM Digital elevation model

DEUS Discrete-Event Universal Simulator

DINFO Department of Information Engineering of University of Florence

DISIT Distributed Systems and Internet Tech lab & Distributed Data Intelligence Lab of UNIFI

DNN Deep Neural Network

DRL Deep Reinforcement Learning

DSP Dynamic Shortest Path

DTM Digital terrain model

EBkSP Entropy Balanced k Shortest Paths

EPSG Geodetic Parameter Dataset

FCM Fuzzy c-mean

FC-SAE Fully Connected Sparse Autoencoder

FD Functional Dependencies

FN False Negative

FP False Positive

FP-Growth Frequent Pattern Growth

FPR False Positive Rate

GDPR General Data Protection Regulation

GPS Global positioning system

HD Historical Data

ICT Information and Communication Technologies

IDW Inverse Distance Weighting

IoT Internet of Things

IT Information technology

ITS Intelligent Transportation Systems

KB Knowledge Base

kNN k-nearest neighbors

KPI Key performance indicator

LR Logistic regression

LSTM Long-Short Term Memory

LTV Life-Time-Value

LULC Land-use/land-cover

LWR Lighthill-Whitham-Richards

MAE Mean absolute error

MAPE Mean Absolute Percentage Error

MAT-Sim Multi-Agent Transport Simulation

MCC Matthews correlation coefficient

MLP Multilayer Perceptron

MNDWI Modified normalized difference water index

MSE Mean Squared Error

MTBF Mean time between failure

MTTF Mean time to Failure

MTTR Mean time to repair

NDBI Normalized difference build-up index

NDVI Normalized difference Vegetation Index

NDWI Normalized difference Water Index

NPR Negative Predictive Rate

OA Overall Accuracy

OPC-UA Open Platform Communications - Unified Architecture

OSM Open Street Map

P.f.a Probability of false alarm

PAM Partitioning Around Medoids

PCA Principal Component Analysis

PDE Partial Differential Equation

PM Preventive Maintenance

PPR Positive Predictive Rate

ReLU Rectified Linear Unit

REM Rapid eye movement

RF Random Forest

RFD Relaxed Functional Dependencies

RFID Radio-Frequency IDentification

RFM Recency-Frequency-Monetary Value

RkSP Random k Shortest Paths

RMSE Root Mean Square Error

RNN Recurrent neural network

ROC Receiver Operator Characteristic

RTD Real Time Data

RUL Remaining useful life

S Simulation

SC Scenario

SCADA Supervisory Control And Data Acquisition

SETM Set-oriented Mining

SHAP Shapley additive explanation

SIGMA Sistema Integrato Gestione Monitoraggion Allerta

SIR Servizio Idrologico Toscana

SLINK Single-Linkage

SLTP Short and Very Long Term Predictions

SM Scenario Model

SOM Self-Organizing-Maps

Sp-AE Sparse Autoencoders

SPI Stream Power Index

SSFIM Single Scan for Frequent Itemsets Mining

SSO Single sign-on

St-AE Stacked Autoencoders

STP Short term prediction

SUMO Simulation of Urban Mobility

SVD Singular value decomposition

SVM Support Vector Machines

SVR Support Vector Regression

TBM Time-based maintenance

TDM Traffic Distribution Matrices

TE Tennessee Eastman

TN True Negatives

TP True Positive

TPR True Positive Rate

TWI Topographic Wetness Index

UCS Unified Control System

UNIFI University of Florence

VANET Vehicular Ad-Hoc Network

WASO Wake Time After Sleep Onset

WMS Web Map Service

XGBoost Extreme gradient boosting

Appendix B

Publications

This research activity has led to several publications in international journals and conferences. These are summarized below.[1]

International Journals

1. P. Bellini, **L.A. Ipsaro Palesi**, P. Nesi, G. Pantaleo. Multi Clustering Recommendation System for Fashion Retail. MTAP Multimedia Tool and Application, Springer 2021

2. C. Badii, P. Bellini, S. Bilotta, D. Bologna, D. Cenni, A. Difino, **A. Ipsaro Palesi**, N. Mitolo, P. Nesi, G. Pantaleo, I. Paoli, M. Paolucci, M. Soderi, "How COVID-19 Lockdown Impacted on Mobility and Environmental data", Bollettino della SocietÃ Geografica Italiana, FuPress, June 2020

Submitted

1. P. Bellini, S. Bilotta, **L. A. Ipsaro Palesi**, P. Nesi, G. Pantaleo. What-If Analysis for Traffic Flow in Smart City Context. FGCS, Future generation of computer systems 2021

2. E.Collini, **L. A. Ipsaro Palesi**, P. Nesi, G. Pantaleo, N. Nocentin, A Rosi Predicting and Understanding Landslide Events with Explainable Artificial Intelligence IEEE Access

[1]The author's bibliometric indices are the following: H-index = 2, total number of citations = 5 (source: Google Scholar on Month 01, 2021).

International Conferences and Workshops

1. P. Bellini, **A. L. Ipsaro Palesi**, P. Nesi, G. Pantaleo, "Fashion Retail Recommendation System by Multiple Clustering", proc. of th DMSVIVA conference, Pittsburgh Conf. Center, June 29-30, 2021.
2. P. Bellini, D. Cenni, **L. A. Ipsaro Palesi**, P. Nesi, G. Pantaleo, "A Deep Learning Approach for Short Term Prediction of Industrial Plant Working Status", 7th IEEE International Conference on Big Data Service and Machine Learning Applications, 23-26 August, 2021.
3. P. Bellini, **L. A. Ipsaro Palesi**, P. Nesi, G. Pantaleo, Recommendation System for Fashion Retail Pierfrancesco Bellini, i-Cities Conference 2021

Technical Reports

1. Badii, C.; Bellini, P.; Bilotta, S.; Bologna, D.; Cenni, D.; Difino, A.; **Ipsaro Palesi, A.**; Mitolo, N.; Nesi, P.; Pantaleo, G.; Paoli, I.; Paolucci, M.; Soderi, M. Impact on Mobility and Environmental Data of COVID-19 Lockdown on Florence Area. 2020, 2020050184
(doi: 10.20944/preprints202005.0184.v1).

Bibliography

[1] "Clc2006 technical guidelines. european environment agency. eea technical report, 2017." https://www.eea.europa.eu/publications/technical_report_2007_17.

[2] "Graphhopper," https://www.graphhopper.com/.

[3] "Human-centred design for interactive systems," https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-1:v1:en.

[4] "Paginemediche polisonnografia," https://www.paginemediche.it/medicina-e-prevenzione/esami/polisonnografia-il-test-per-diagnosticare-i-disturbi-del-sonno.

[5] "Paginemediche sleep tracker," https://www.paginemediche.it/benessere/salute-digitale/sleep-tracker-per-il-monitoraggio-del-sonno-come-funzionano.

[6] "Pillow sleep tracker," https://neybox.com/pillow-sleep-tracker-en.

[7] "rfd-discovery," https://github.com/dariodip/rfd-discovery.

[8] "Sleep as android," https://play.google.com/store/apps/details?id=com.urbandroid.sleep.full.key&hl=en.

[9] "Sleep better," https://play.google.com/store/apps/details?id=com.runtastic.android.sleepbetter.lite&hl=en.

[10] "Sleepcycle," https://www.sleepcycle.com/.

[11] "Wired sleep tracker," https://www.wired.it/gadget/accessori/2018/06/02/sleep-tracker-funzionano-davvero/.

[12] M. T. Abraham, N. Satyam, A. Rosi, B. Pradhan, and S. Segoni, "The selection of rain gauges and rainfall parameters in estimating intensity-duration thresholds for landslide occurrence: case study from wayanad (india)," *Water*, vol. 12, no. 4, p. 1000, 2020.

[13] M. T. Abraham, N. Satyam, N. Shreyas, B. Pradhan, S. Segoni, K. N. Abdul Maulud, and A. M. Alamri, "Forecasting landslides using sigma model: a case study from idukki, india," *Geomatics, Natural Hazards and Risk*, vol. 12, no. 1, pp. 540–559, 2021.

[14] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, 1993, pp. 207–216.

[15] R. Agrawal, R. Srikant *et al.*, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, vol. 1215. Citeseer, 1994, pp. 487–499.

[16] R. Alaa, M. Gawich, and M. Fernández-Veiga, "Personalized recommendation for online retail applications based on ontology evolution," in *Proceedings of the 2020 6th International Conference on Computer and Technology Applications*, 2020, pp. 12–16.

[17] M. I. Alipio, J. R. R. Bayanay, A. O. Casantusan, and A. A. Dequeros, "Vehicle traffic and flood monitoring with reroute system using bayesian networks analysis," in *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*. IEEE, 2017, pp. 1–5.

[18] F. Antonello, P. Baraldi, A. Shokry, E. Zio, U. Gentile, and L. Serio, "Association rules extraction for the identification of functional dependencies in complex technical infrastructures," *Reliability Engineering & System Safety*, vol. 209, p. 107305, 2021.

[19] Y. Arafah and H. Winarso, "Redefining smart city concept with resilience approach," in *IOP conference series: earth and environmental science*, vol. 70, no. 1. IOP Publishing, 2017, p. 012065.

[20] A. Arman, P. Bellini, P. Nesi, and M. Paolucci, "Analyzing public transportation offer wrt mobility demand," in *Proceedings of the 1st ACM International Workshop on Technology Enablers and Innovative Applications for Smart Cities and Communities*, 2019, pp. 30–37.

[21] C. Badii, E. G. Belay, P. Bellini, M. Marazzini, M. Mesiti, P. Nesi, G. Pantaleo, M. Paolucci, S. Valtolina, M. Soderi *et al.*, "Snap4city: A scalable iot/ioe platform for developing smart city applications," in *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 2018, pp. 2109–2116.

[22] C. Badii, P. Bellini, D. Cenni, N. Mitolo, P. Nesi, G. Pantaleo, and M. Soderi, "Industry 4.0 synoptics controlled by iot applications in node-red," in *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*. IEEE, 2020, pp. 54–61.

[23] C. Badii, P. Bellini, A. Difino, and P. Nesi, "Sii-mobility: An iot/ioe architecture to enhance smart city mobility and transportation services," *Sensors*, vol. 19, no. 1, p. 1, 2019.

[24] ——, "Smart city iot platform respecting gdpr privacy and security aspects," *IEEE Access*, vol. 8, pp. 23 601–23 623, 2020.

[25] C. Badii, P. Bellini, A. Difino, P. Nesi, G. Pantaleo, and M. Paolucci, "Microservices suite for smart city applications," *Sensors*, vol. 19, no. 21, p. 4798, 2019.

[26] C. Badii, P. Nesi, and I. Paoli, "Predicting available parking slots on critical and regular services by exploiting a range of open data," *IEEE Access*, vol. 6, pp. 44 059–44 071, 2018.

[27] A. Battistini, A. Rosi, S. Segoni, D. Lagomarsino, F. Catani, and N. Casagli, "Validation of landslide hazard models using a semantic engine on online news," *Applied geography*, vol. 82, pp. 59–65, 2017.

[28] A. Battistini, S. Segoni, G. Manzo, F. Catani, and N. Casagli, "Web data mining for automatic inventory of geohazards at national scale," *Applied Geography*, vol. 43, pp. 147–158, 2013.

[29] E. Bellini, P. Ceravolo, and P. Nesi, "Quantify resilience enhancement of uts through exploiting connected community and internet of everything emerging technologies," *ACM Transactions on Internet Technology (TOIT)*, vol. 18, no. 1, pp. 1–34, 2017.

[30] E. Bellini, L. Coconea, and P. Nesi, "A functional resonance analysis method driven resilience quantification for socio-technical systems," *IEEE Systems Journal*, vol. 14, no. 1, pp. 1234–1244, 2019.

[31] P. Bellini, I. Bruno, D. Cenni, A. Fuzier, P. Nesi, and M. Paolucci, "Mobile medicine: semantic computing management for health care applications on desktop and mobile devices," *Multimedia Tools and Applications*, vol. 58, no. 1, pp. 41–79, 2012.

[32] P. Bellini, I. Bruno, P. Nesi, and M. Paolucci, "A static and dynamic recommendations system for best practice networks," in *International Conference on Human-Computer Interaction*. Springer, 2013, pp. 259–268.

[33] P. Bellini, D. Cenni, N. Mitolo, P. Nesi, and G. Pantaleo, "Exploiting satellite data in the context of smart city applications," in *2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*. IEEE, 2021, pp. 399–406.

[34] P. Bellini, D. Cenni, and P. Nesi, "Optimization of information retrieval for cross media contents in a best practice network," *International Journal of Multimedia Information Retrieval*, vol. 3, no. 3, pp. 147–159, 2014.

[35] P. Bellini, D. Cenni, L. A. I. Palesi, P. Nesi, and G. Pantaleo, "A deep learning approach for short term prediction of industrial plant working status," in *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2021, pp. 9–16.

[36] P. Bellini, P. Nesi, M. Paolucci, and I. Zaza, "Smart city architecture for data ingestion and analytics: Processes and solutions," in *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2018, pp. 137–144.

[37] P. Bellini, L. A. I. Palesi, P. Nesi, and G. Pantaleo, "Multi clustering recommendation system for fashion retail," *Multimedia Tools and Applications*, pp. 1–28, 2022.

[38] A. Bhandari, V. Patel, and M. Patel, "A survey on traffic congestion detection and rerouting strategies," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2018, pp. 42–44.

[39] D. Billsus and M. J. Pazzani, "User modeling for adaptive news access," *User modeling and user-adapted interaction*, vol. 10, no. 2, pp. 147–180, 2000.

[40] S. Bilotta and P. Nesi, "Traffic flow reconstruction by solving indeterminacy on traffic distribution at junctions," *Future Generation Computer Systems*, vol. 114, pp. 649–660, 2021.

[41] A. Binding, N. Dykeman, and S. Pang, "Machine learning predictive maintenance on data in the wild," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019, pp. 507–512.

[42] B. S. Blanchard, D. C. Verma, and E. L. Peterson, *Maintainability: a key to effective serviceability and maintenance management*. John Wiley & Sons, 1995, vol. 13.

[43] G. Bretti, R. Natalini, and B. Piccoli, "A fluid-dynamic traffic model on road networks," *Archives of Computational Methods in Engineering*, vol. 14, no. 2, pp. 139–172, 2007.

[44] B. Breve, L. Caruccio, S. Cirillo, V. Deufemia, and G. Polese, "Visualizing dependencies during incremental discovery processes." in *EDBT/ICDT Workshops*, 2020.

[45] Z. Cakici and Y. S. Murat, "A differential evolution algorithm-based traffic control model for signalized intersections," *Advances in Civil Engineering*, vol. 2019, 2019.

[46] M. Calvello and G. Pecoraro, "Franeitalia: a catalog of recent italian landslides," *Geoenvironmental Disasters*, vol. 5, no. 1, pp. 1–16, 2018.

[47] L. Caruccio and S. Cirillo, "Incremental discovery of imprecise functional dependencies," *Journal of Data and Information Quality (JDIQ)*, vol. 12, no. 4, pp. 1–25, 2020.

[48] L. Caruccio, V. Deufemia, and G. Polese, "Relaxed functional dependencies a survey of approaches," *IEEE Transactions on knowledge and data engineering*, vol. 28, no. 1, pp. 147–165, 2015.

[49] B.-G. Chae, H.-J. Park, F. Catani, A. Simoni, and M. Berti, "Landslide prediction, monitoring and early warning: a concise review of state-of-the-art," *Geosciences Journal*, vol. 21, no. 6, pp. 1033–1070, 2017.

[50] C. C. H. Chan, "Intelligent value-based customer segmentation method for campaign management: A case study of automobile retailer," *Expert systems with applications*, vol. 34, no. 4, pp. 2754–2762, 2008.

[51] D. Chen, S. Sain, and K. Guo, "Data mining for the online retail industry: a case study of rfm model-based customer segmentation using data mining. j database mark cust strateg manag 19: 197–208," 2012.

[52] Y.-S. Cheng, T.-T. Yu, and N.-T. Son, "Random forests for landslide prediction in tsengwen river watershed, central taiwan," *Remote Sensing*, vol. 13, no. 2, p. 199, 2021.

[53] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on web usage mining and decision tree induction," *Expert systems with Applications*, vol. 23, no. 3, pp. 329–342, 2002.

[54] A. Cuzzocrea and E. Fadda, "Data-intensive object-oriented adaptive web systems: Implementing and experimenting the oo-xahm framework," in *Proceedings of the 12th International Conference on Management of Digital EcoSystems*, 2020, pp. 115–123.

[55] A. Cuzzocrea, G. Pilato, and E. Fadda, "User emotion detection via taxonomy management: An innovative system." in *SEBD*, 2020, pp. 334–342.

[56] A. Da ú and N. Salim, "Sentiment aware deep recommender system with neural attention networks," *IEEE Access*, vol. 7, pp. 45 472–45 484, 2019.

[57] M. Danaf, F. Becker, X. Song, B. Atasoy, and M. Ben-Akiva, "Online discrete choice models: Applications in personalized recommendations," *Decision Support Systems*, vol. 119, pp. 35–45, 2019.

[58] T. T. Dang, T. H. Duong, and H. S. Nguyen, "A hybrid framework for enhancing correlation to solve cold-start problem in recommender systems," in *the 2014 Seventh IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*. IEEE, 2014, pp. 1–5.

[59] M. Y. Darus and K. A. Bakar, "Congestion control algorithm in vanets," *world applied sciences journal*, vol. 21, no. 7, pp. 1057–1061, 2013.

[60] S. De Falco, M. Angelidou, and J.-P. D. Addie, "From the smart city to the smart metropolis? building resilience in the urban periphery," *European Urban and Regional Studies*, vol. 26, no. 2, pp. 205–223, 2019.

[61] P. Distefano, D. J. Peres, P. Scandura, and A. Cancelliere, "Brief communication: Rainfall thresholds based on artificial neural networks can improve landslide early warning," *Natural Hazards and Earth System Sciences Discussions*, pp. 1–9, 2021.

[62] Y. Djenouri, M. Comuzzi, and D. Djenouri, "Ss-fim: single scan for frequent itemsets mining in transactional databases," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017, pp. 644–654.

[63] P. M. Ejercito, K. G. E. Nebrija, R. P. Feria, and L. L. Lara-Figueroa, "Traffic simulation software review," in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2017, pp. 1–4.

[64] G. Fancello, M. Carta, and P. Fadda, "A modeling tool for measuring the performance of urban road networks," *Procedia-Social and Behavioral Sciences*, vol. 111, pp. 559–566, 2014.

[65] M. Fatemi and L. Tokarchuk, "A community based social recommender system for individuals & groups," in *2013 International Conference on Social Computing*. IEEE, 2013, pp. 351–356.

[66] A. Furno, N.-E. E. Faouzi, R. Sharma, and E. Zimeo, "Graph-based ahead monitoring of vulnerabilities in large dynamic transportation networks," *PloS one*, vol. 16, no. 3, p. e0248764, 2021.

[67] M. Gavrilyuk, T. Vorobyova, and E. Shalagina, "Effects of road blocking on traffic flows in moscow," *Transportation research procedia*, vol. 50, pp. 1–11, 2020.

[68] W. M. Gentles, "Chapter 42 - htm best practice guidelines and standards of practice around the world," in *Clinical Engineering Handbook (Second Edition)*, second edition ed., E. Iadanza, Ed. Academic Press, 2020, pp. 268–275.

[69] T. F. Gharib, H. Nassar, M. Taha, and A. Abraham, "An efficient algorithm for incremental mining of temporal association rules," *Data & Knowledge Engineering*, vol. 69, no. 8, pp. 800–815, 2010.

[70] M. Giering, "Retail sales prediction and item recommendations using customer demographics at store level," *ACM SIGKDD Explorations Newsletter*, vol. 10, no. 2, pp. 84–89, 2008.

[71] S. Godunov, "A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics." *Sbornik: Mathematics*, vol. 47, no. 8-9, pp. 357–393, 1959.

[72] J. Goetz, A. Brenning, H. Petschko, and P. Leopold, "Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling," *Computers & geosciences*, vol. 81, pp. 1–11, 2015.

[73] M. Golfarelli, S. Rizzi, and A. Proli, "Designing what-if analysis: towards a methodology," in *Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP*, 2006, pp. 51–58.

[74] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, pp. 857–871, 1971.

[75] G. Greco, A. Guzzo, L. Pontieri, and D. Sacca, "Mining expressive process models by clustering workflow traces," in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2004, pp. 52–62.

[76] D. Grewal, A. L. Roggeveen, and J. NordfÃ¤lt, "The future of retailing," *Journal of Retailing*, vol. 93, no. 1, pp. 1–6, 2017, the Future of Retailing.

[77] D. L. Guidoni, G. Maia, F. S. Souza, L. A. Villas, and A. A. Loureiro, "Vehicular traffic management based on traffic engineering for vehicular ad hoc networks," *IEEE Access*, vol. 8, pp. 45 167–45 183, 2020.

[78] F. Guzzetti, S. Peruccacci, M. Rossi, and C. P. Stark, "Rainfall thresholds for the initiation of landslides in central and southern europe," *Meteorology and atmospheric physics*, vol. 98, no. 3, pp. 239–267, 2007.

[79] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *ACM sigmod record*, vol. 29, no. 2, pp. 1–12, 2000.

[80] D. Hartama, H. Mawengkang, M. Zarlis, R. W. Sembiring, M. Furqan, D. Abdullah, and R. Rahim, "A research framework of disaster traffic management to smart city," in *2017 Second International Conference on Informatics and Computing (ICIC)*. IEEE, 2017, pp. 1–5.

[81] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[82] D. Houli, L. Zhiheng, L. Li, Y. ZHANG, and Y. Shengchao, "Network-wide traffic state observation and analysis method using pseudo-color map," *Journal of Transportation Systems Engineering and Information Technology*, vol. 9, no. 4, pp. 46–52, 2009.

[83] M. Houtsma and A. Swami, "Set-oriented mining for association rules in relational databases," in *Proceedings of the eleventh international conference on data engineering*. IEEE, 1995, pp. 25–33.

[84] F. Huang, J. Zhang, C. Zhou, Y. Wang, J. Huang, and L. Zhu, "A deep learning algorithm using a fully connected sparse autoencoder neural network for landslide susceptibility prediction," *Landslides*, vol. 17, no. 1, pp. 217–229, 2020.

[85] F. Huseynov, S. Y. Huseynov, and S. Özkan, "The influence of knowledge-based e-commerce product recommender agents on online consumer decision-making," *Information Development*, vol. 32, no. 1, pp. 81–90, 2016.

[86] T. Huuhtanen and A. Jung, "Predictive maintenance of photovoltaic panels via deep learning," in *2018 IEEE Data Science Workshop (DSW)*. IEEE, 2018, pp. 66–70.

[87] H. Hwang, T. Jung, and E. Suh, "An ltv model and customer segmentation based on customer value: a case study on the wireless telecommunication industry," *Expert systems with applications*, vol. 26, no. 2, pp. 181–188, 2004.

[88] Y. Hwang, M. Clark, B. Rajagopalan, and G. Leavesley, "Spatial interpolation schemes of daily precipitation for hydrologic modeling," *Stochastic environmental research and risk assessment*, vol. 26, no. 2, pp. 295–320, 2012.

[89] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[90] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[91] P. Kachroo and S. Sastry, "Travel time dynamics for intelligent transportation systems: Theory and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 385–394, 2015.

[92] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using iot sensor data," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2017, pp. 87–90.

[93] R. Kashef, "Enhancing the role of large-scale recommendation systems in the iot context," *IEEE Access*, vol. 8, pp. 178 248–178 257, 2020.

[94] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[95] T. Kavzoglu, E. K. Sahin, and I. Colkesen, "Selecting optimal conditioning factors in shallow translational landslide susceptibility mapping using genetic algorithm," *Engineering Geology*, vol. 192, pp. 101–112, 2015.

[96] K. KBAAM and W. Wijayanayake, "Application of data mining technique to predict landslides in sri lanka."

[97] S.-Y. Kim, T.-S. Jung, E.-H. Suh, and H.-S. Hwang, "Customer segmentation and strategy development based on customer lifetime value: A case study," *Expert systems with applications*, vol. 31, no. 1, pp. 101–107, 2006.

[98] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data," *PloS one*, vol. 14, no. 2, p. e0212320, 2019.

[99] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[100] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[101] D. Lagomarsino, S. Segoni, A. Rosi, G. Rossi, A. Battistini, F. Catani, and N. Casagli, "Quantitative comparison between two different methodologies to define rainfall thresholds for landslide forecasting," *Natural Hazards and Earth System Sciences*, vol. 15, no. 10, pp. 2413–2423, 2015.

[102] C.-Y. Lee, J.-Q. Huang, W.-P. Ma, Y.-L. Weng, Y.-C. Lee, and Z.-J. Lee, "Analyze the rainfall of landslide on apache spark," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*. IEEE, 2018, pp. 348–351.

[103] W.-H. Lee and C.-Y. Chiu, "Design and implementation of a smart traffic signal control system for smart city applications," *Sensors*, vol. 20, no. 2, p. 508, 2020.

[104] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to rul prediction," *Mechanical systems and signal processing*, vol. 104, pp. 799–834, 2018.

[105] C. Li, R.-V. Sánchez, G. Zurita, M. Cerrada, and D. Cabrera, "Fault diagnosis for rotating machinery using vibration measurement deep statistical feature learning," *Sensors*, vol. 16, no. 6, p. 895, 2016.

[106] L. Liao and F. Köttig, "A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction," *Applied Soft Computing*, vol. 44, pp. 191–199, 2016.

[107] M. J. Lighthill and G. B. Whitham, "On kinematic waves ii. a theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.

[108] G. S. Linoff and M. J. Berry, *Data mining techniques: for marketing, sales, and customer relationship management.* John Wiley & Sons, 2011.

[109] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems*, 2017, pp. 4768–4777.

[110] T. Mahmood and F. Ricci, "Improving recommender systems with adaptive conversational strategies," in *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, 2009, pp. 73–82.

[111] H. C. Manual, "Hcm2010," *Transportation Research Board, National Research Council, Washington, DC*, vol. 1207, 2010.

[112] V. A. Marchau, W. E. Walker, and G. Van Wee, "Dynamic adaptive transport policies for handling deep uncertainty," *Technological forecasting and social change*, vol. 77, no. 6, pp. 940–950, 2010.

[113] M. B. Mariappan, M. Suk, and B. Prabhakaran, "Facefetch: A user emotion driven multimedia content recommendation system based on facial expression recognition," in *2012 IEEE International Symposium on Multimedia.* IEEE, 2012, pp. 84–87.

[114] G. Martelloni, S. Segoni, R. Fanti, and F. Catani, "Rainfall thresholds for the forecasting of landslide occurrence at regional scale," *Landslides*, vol. 9, no. 4, pp. 485–495, 2012.

[115] J. Mathew, M. Luo, and C. K. Pang, "Regression kernel for prognostics with support vector machines," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA).* IEEE, 2017, pp. 1–5.

[116] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 12, pp. 1650–1654, 2002.

[117] J. A. McCarty and M. Hastak, "Segmentation approaches in data-mining: A comparison of rfm, chaid, and logistic regression," *Journal of business research*, vol. 60, no. 6, pp. 656–662, 2007.

[118] S. Melo, J. Macedo, and P. Baptista, "Guiding cities to pursue a smart mobility paradigm: An example from vehicle routing guidance and its traffic and operational effects," *Research in transportation economics*, vol. 65, pp. 24–33, 2017.

[119] S.-H. Min and I. Han, "Recommender systems using support vector machines," in *International Conference on Web Engineering.* Springer, 2005, pp. 387–393.

[120] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple bayesian classifier," in *Pacific Rim International conference on artificial intelligence*. Springer, 2000, pp. 679–689.

[121] R. Mu, "A survey of recommender systems based on deep learning," *Ieee Access*, vol. 6, pp. 69 009–69 022, 2018.

[122] K. Nam and F. Wang, "An extreme rainfall-induced landslide susceptibility assessment using autoencoder combined with random forest in shimane prefecture, japan," *Geoenvironmental Disasters*, vol. 7, no. 1, pp. 1–16, 2020.

[123] M. Namvar, M. R. Gholamian, and S. KhakAbi, "A two phase clustering method for intelligent customer segmentation," in *2010 International Conference on Intelligent Systems, Modelling and Simulation*. IEEE, 2010, pp. 215–219.

[124] E. W. Ngai, L. Xiu, and D. C. Chau, "Application of data mining techniques in customer relationship management: A literature review and classification," *Expert systems with applications*, vol. 36, no. 2, pp. 2592–2602, 2009.

[125] P. T. T. Ngo, M. Panahi, K. Khosravi, O. Ghorbanzadeh, N. Kariminejad, A. Cerda, and S. Lee, "Evaluation of deep learning algorithms for national scale landslide susceptibility mapping of iran," *Geoscience Frontiers*, vol. 12, no. 2, pp. 505–519, 2021.

[126] S. T. T. Nguyen and B. D. Tran, "Long short-term memory based movie recommendation," *Science & Technology Development Journal-Engineering and Technology*, vol. 3, no. SI1, pp. SI1–SI9, 2020.

[127] A. Nikitas, K. Michalakopoulou, E. T. Njoya, and D. Karampatzakis, "Artificial intelligence, transport and the smart city: Definitions and dimensions of a new mobility era," *Sustainability*, vol. 12, no. 7, p. 2789, 2020.

[128] S. Palliyaguru, L. Liyanage, O. Weerakoon, and G. Wimalaratne, "Random forest as a novel machine learning approach to predict landslide susceptibility in kalutara district, sri lanka," in *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, 2020, pp. 262–267.

[129] M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Machine learning*, vol. 27, no. 3, pp. 313–331, 1997.

[130] J. Pearl, "The seven tools of causal inference, with reflections on machine learning," *Communications of the ACM*, vol. 62, no. 3, pp. 54–60, 2019.

[131] H. Pei, F. Meng, and H. Zhu, "Landslide displacement prediction based on a novel hybrid model and convolutional neural network considering time-varying factors," *Bulletin of Engineering Geology and the Environment*, vol. 80, no. 10, pp. 7403–7422, 2021.

[132] P. Perez-Murueta, A. Gómez-Espinosa, C. Cardenas, and M. Gonzalez-Mendoza, "Deep learning system for vehicular re-routing and congestion avoidance," *Applied Sciences*, vol. 9, no. 13, p. 2717, 2019.

[133] V. D. Pham, Q.-H. Nguyen, H.-D. Nguyen, V.-M. Pham, Q.-T. Bui *et al.*, "Convolutional neural networkoptimized moth flame algorithm for shallow landslide susceptible analysis," *IEEE Access*, vol. 8, pp. 32 727–32 736, 2020.

[134] M. Picone, M. Amoretti, and F. Zanichelli, "Simulating smart cities with deus." in *SimuTools*. Citeseer, 2012, pp. 172–177.

[135] L. Po, F. Rollo, J. R. R. Viqueira, R. T. Lado, A. Bigi, J. C. Lopez, M. Paolucci, and P. Nesi, "Trafair: Understanding traffic flow to improve air quality," in *2019 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2019, pp. 36–43.

[136] K. N. Rao, "Application domain and functional classification of recommender systems–a survey," *DESIDOC Journal of Library & Information Technology*, vol. 28, no. 3, p. 17, 2008.

[137] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*. Springer, 2011, pp. 1–35.

[138] P. I. Richards, "Shock waves on the highway," *Operations research*, vol. 4, no. 1, pp. 42–51, 1956.

[139] F. Rodrigues and B. Ferreira, "Product recommendation based on shared customer's behaviour," *Procedia Computer Science*, vol. 100, pp. 136–146, 2016.

[140] C. Ross and S. Guhathakurta, "Autonomous vehicles and energy impacts: a scenario analysis," *Energy Procedia*, vol. 143, pp. 47–52, 2017.

[141] E. F. Z. Santana, N. Lago, F. Kon, and D. S. Milojicic, "Interscsimulator: Large-scale traffic simulation in smart cities using erlang," in *International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Springer, 2017, pp. 211–227.

[142] A. S. Santos, A. C. Corsi, I. C. Teixeira, V. L. Gava, F. A. Falcetta, E. S. de Macedo, C. d. S. Azevedo, K. T. de Lima, and K. R. Braghetto, "Brazilian natural disasters integrated into cyber-physical systems: computational challenges for landslides and floods in urban ecosystems," in *2020 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2020, pp. 1–8.

[143] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111–112.

[144] B. Shao, X. Hu, G. Bian, and Y. Zhao, "A multichannel lstm-cnn method for fault diagnosis of chemical process," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[145] J. Shunping, P. Hongqin, and L. Shuang, "Urban traffic state estimation considering resident travel characteristics and road network capacity," *Journal of Transportation Systems Engineering and Information Technology*, vol. 11, no. 5, pp. 81–85, 2011.

[146] R. S. Swift, *Accelerating customer relationships: Using CRM and relationship technologies*. Prentice Hall Professional, 2001.

[147] S. Tao, T. Zhang, J. Yang, X. Wang, and W. Lu, "Bearing fault diagnosis method based on stacked autoencoder and softmax regression," in *2015 34th Chinese Control Conference (CCC)*. IEEE, 2015, pp. 6331–6335.

[148] J. W. Taylor, P. E. McSharry, and R. Buizza, "Wind power density forecasting using ensemble predictions and time series models," *IEEE Transactions on Energy Conversion*, vol. 24, no. 3, pp. 775–782, 2009.

[149] D. Thomas, C. Miller, J. Kämpf, and A. Schlueter, "Multiscale co-simulation of energyplus and citysim models derived from a building information model," in *Bausim 2014: Fifth German-Austrian IBPSA Conference*, 2014, pp. 469–476.

[150] M. Tovar, M. Robles, and F. Rashid, "Pv power prediction, using cnn-lstm hybrid neural network model. case of study: Temixco-morelos, méxico," *Energies*, vol. 13, no. 24, p. 6512, 2020.

[151] H. Tuinhof, C. Pirker, and M. Haltmeier, "Image-based fashion product recommendation with deep learning," in *International Conference on Machine Learning, Optimization, and Data Science*. Springer, 2018, pp. 472–481.

[152] A. Tveito and A. M. Bruaset, *Numerical solution of partial differential equations on parallel computers*. Springer, 2006.

[153] E. Uhlmann, R. P. Pontes, C. Geisert, and E. Hohwieler, "Cluster identification of sensor data for predictive maintenance in a selective laser melting machine tool," *Procedia manufacturing*, vol. 24, pp. 60–65, 2018.

[154] M. Van Steen, "Graph theory and complex networks," *An introduction*, vol. 144, 2010.

[155] K. W Axhausen, A. Horni, and K. Nagel, *The multi-agent transport simulation MATSim*. Ubiquity Press, 2016.

[156] B. Walek and P. Spackova, "Content-based recommender system for online stores using expert system," in *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. IEEE, 2018, pp. 164–165.

[157] C. Wang, M.-C. Zhu, Z.-G. Ma, Z.-Y. He, H. Jiang, P.-S. Li, X.-B. Zhang, J.-B. Shi, K. Chen, T. Weng *et al.*, "Classification of landslide stability based

on fine topographic features," in *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 2020, pp. 54–57.

[158] C. Wang, M. Niepert, and H. Li, "Recsys-dan: discriminative adversarial networks for cross-domain recommender systems," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 8, pp. 2731–2740, 2019.

[159] H. Wang, L. Zhang, K. Yin, H. Luo, and J. Li, "Landslide identification using machine learning," *Geoscience Frontiers*, vol. 12, no. 1, pp. 351–364, 2021.

[160] Y. Wang, M. Papageorgiou, A. Messmer, P. Coppola, A. Tzimitsi, and A. Nuzzolo, "An adaptive freeway traffic state estimator," *Automatica*, vol. 45, no. 1, pp. 10–24, 2009.

[161] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1578–1585.

[162] S. Wei, N. Ye, S. Zhang, X. Huang, and J. Zhu, "Collaborative filtering recommendation algorithm based on item clustering and global similarity," in *2012 Fifth International Conference on Business Intelligence and Financial Engineering*. IEEE, 2012, pp. 69–72.

[163] S. Wu, W. Ren, C. Yu, G. Chen, D. Zhang, and J. Zhu, "Personal recommendation using deep recurrent neural networks in netease," in *2016 IEEE 32nd international conference on data engineering (ICDE)*. IEEE, 2016, pp. 1218–1229.

[164] H. Xie, L. Zhang, and C. P. Lim, "Evolving cnn-lstm models for time series prediction using enhanced grey wolf optimizer," *IEEE Access*, vol. 8, pp. 161 519–161 541, 2020.

[165] R. Yam, P. Tse, L. Li, and P. Tu, "Intelligent predictive decision support system for condition-based maintenance," *The International Journal of Advanced Manufacturing Technology*, vol. 17, no. 5, pp. 383–391, 2001.

[166] J. Zhang, P. Wang, R. Yan, and R. X. Gao, "Long short-term memory for machine remaining life prediction," *Journal of manufacturing systems*, vol. 48, pp. 78–86, 2018.

[167] Z. Zhang and J. Zhao, "A deep belief network based fault diagnosis model for complex chemical processes," *Computers & chemical engineering*, vol. 107, pp. 395–407, 2017.

[168] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional lstm networks," *Sensors*, vol. 17, no. 2, p. 273, 2017.

[169] X. Zhao, L. Zhang, L. Xia, Z. Ding, D. Yin, and J. Tang, "Deep reinforcement learning for list-wise recommendations," *arXiv preprint arXiv:1801.00209*, 2017.

[170] Y. Zhou, Y. Gao, Y. Huang, M. Hefenbrock, T. Riedel, and M. Beigl, "Automatic remaining useful life estimation framework with embedded convolutional lstm as the backbone," *arXiv preprint arXiv:2008.03961*, 2020.