Big Data Architectures
03/12/2024

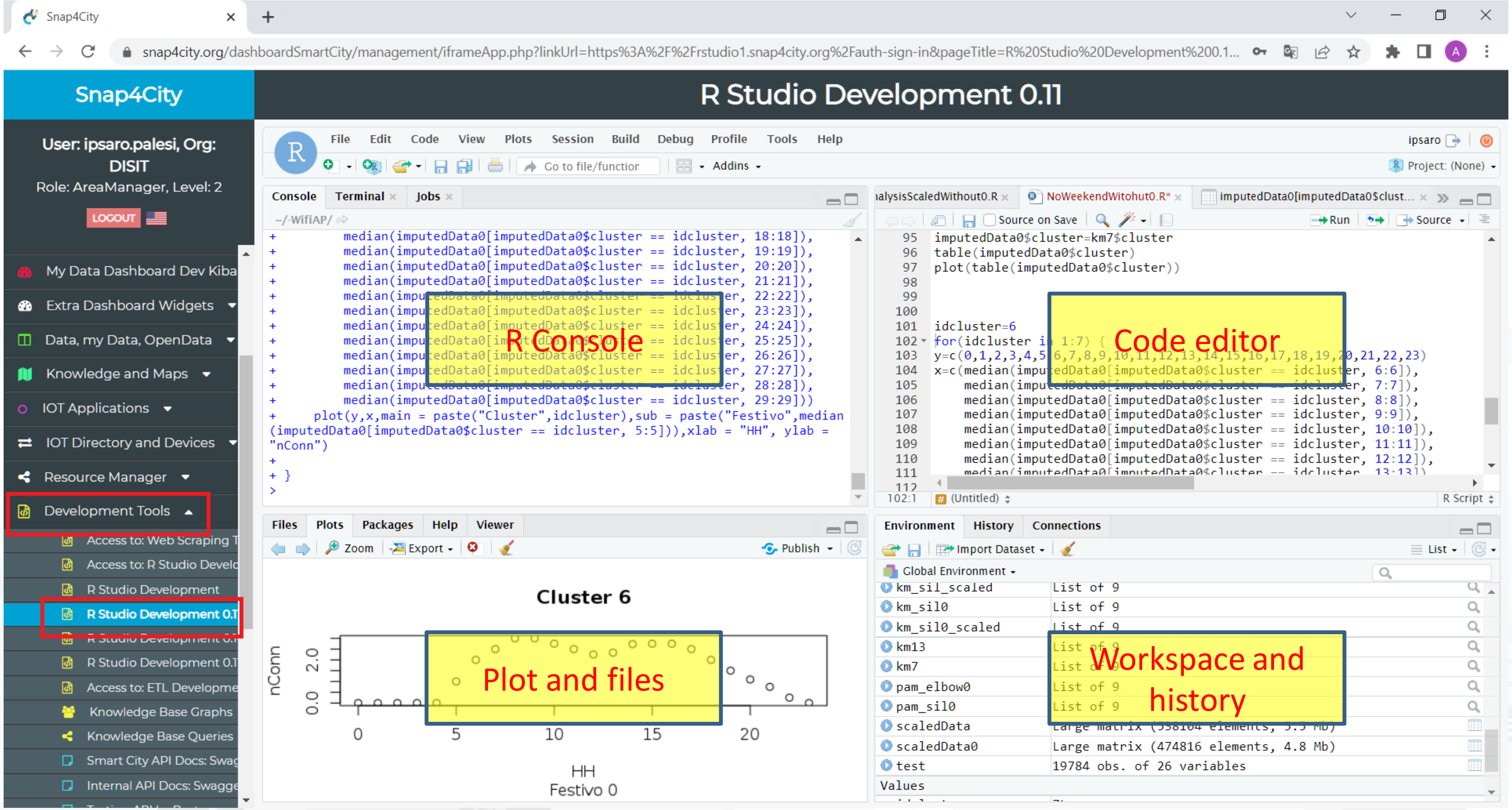
- **Clustering:** *Fashion Retail Recommendations*
- **XAI**: *Explainable artificial intelligence (Predictive Maintenance - Predicting Land sliding - Predicting free parking slots)*

Fondo europeo di sviluppo regionale (Fesr)
Le ali alle tue idee

Fashion Retail Recommendations

Feedback Pilot

Rstudio


R Studio Development 0.11
Snap4City
User: ipsaro.palesi, Org: DISIT
Role: AreaManager, Level: 2
R Console
Code editor
Plot and files
Workspace and history
Cluster 6
Development Tools
R Studio Development 0.1

R code

- Installing and loading R packages

install.packages("cluster")

From GitHub

install.packages("devtools")

devtools::install_github("kassambara/factoextra")

- Getting help with functions in R

?kmeans

- Importing your data into R

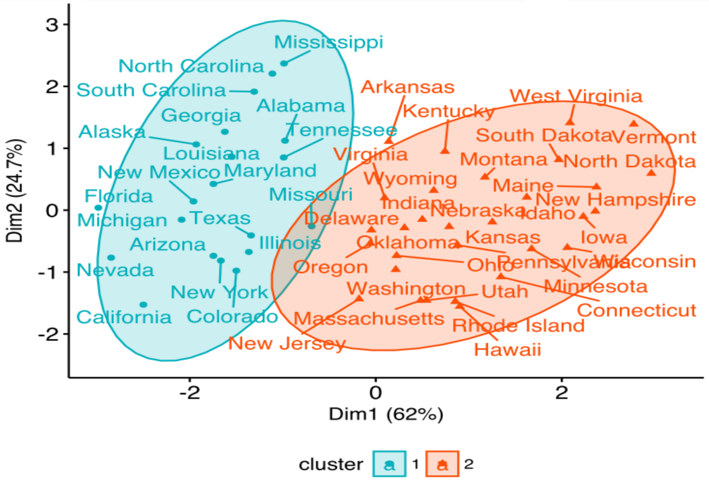
 # .csv file: Read comma (",") separated values

my_data <- read.csv(file.choose())

Clustering

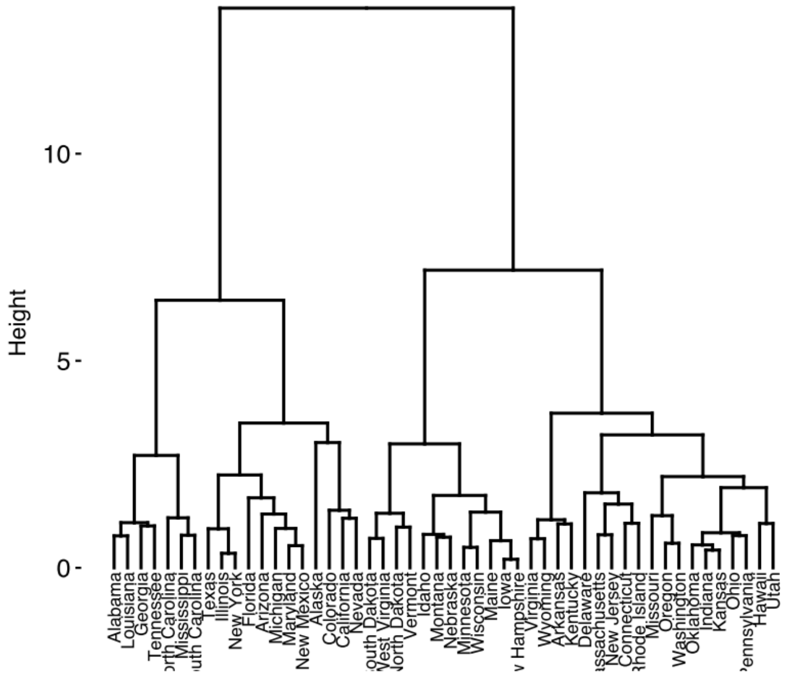
Partitioning

- K-Means Clustering
- K-Medoids
- CLARA - Clustering Large Applications



Hierarchical

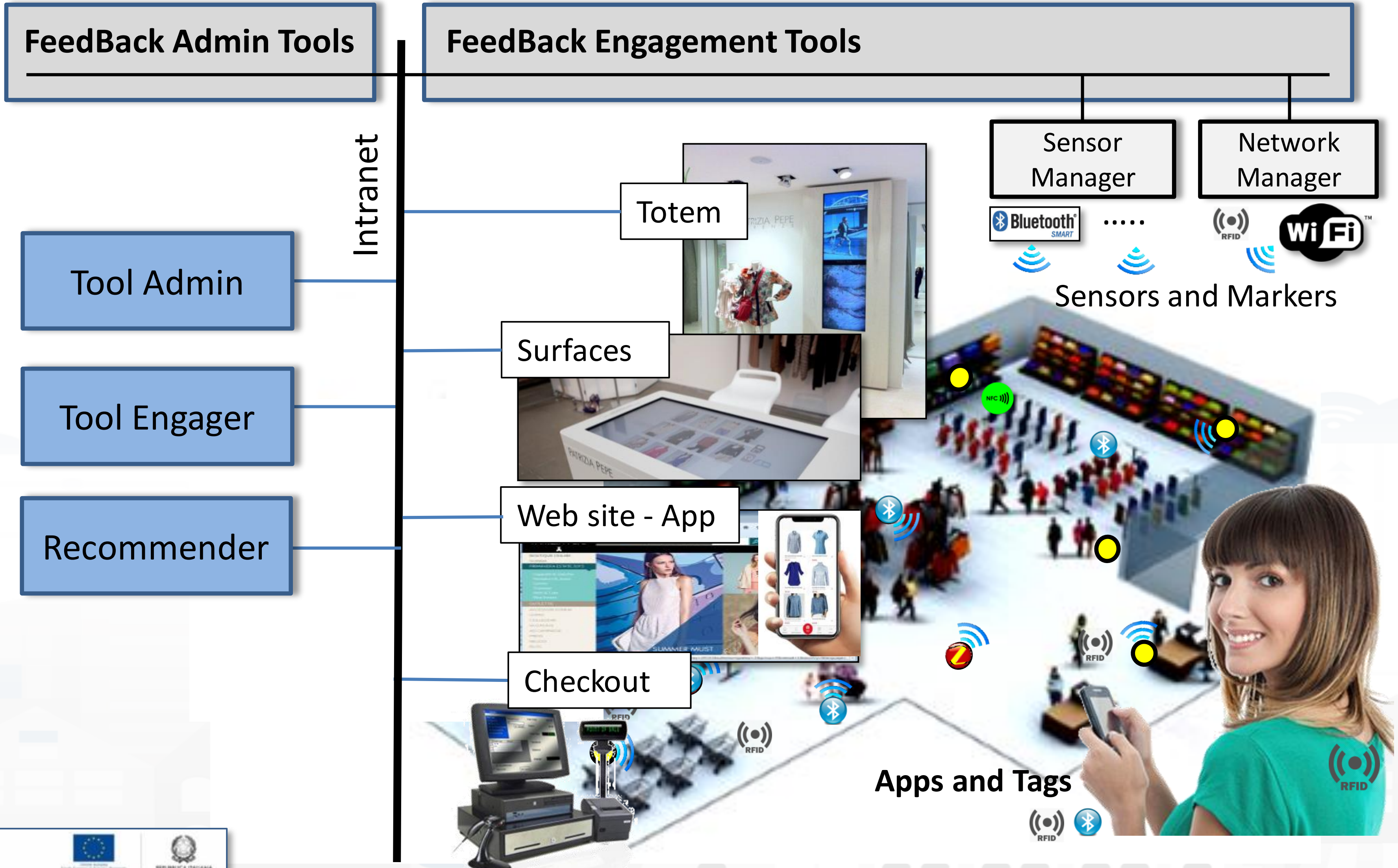
- Agglomerative Clustering



Smart Retail

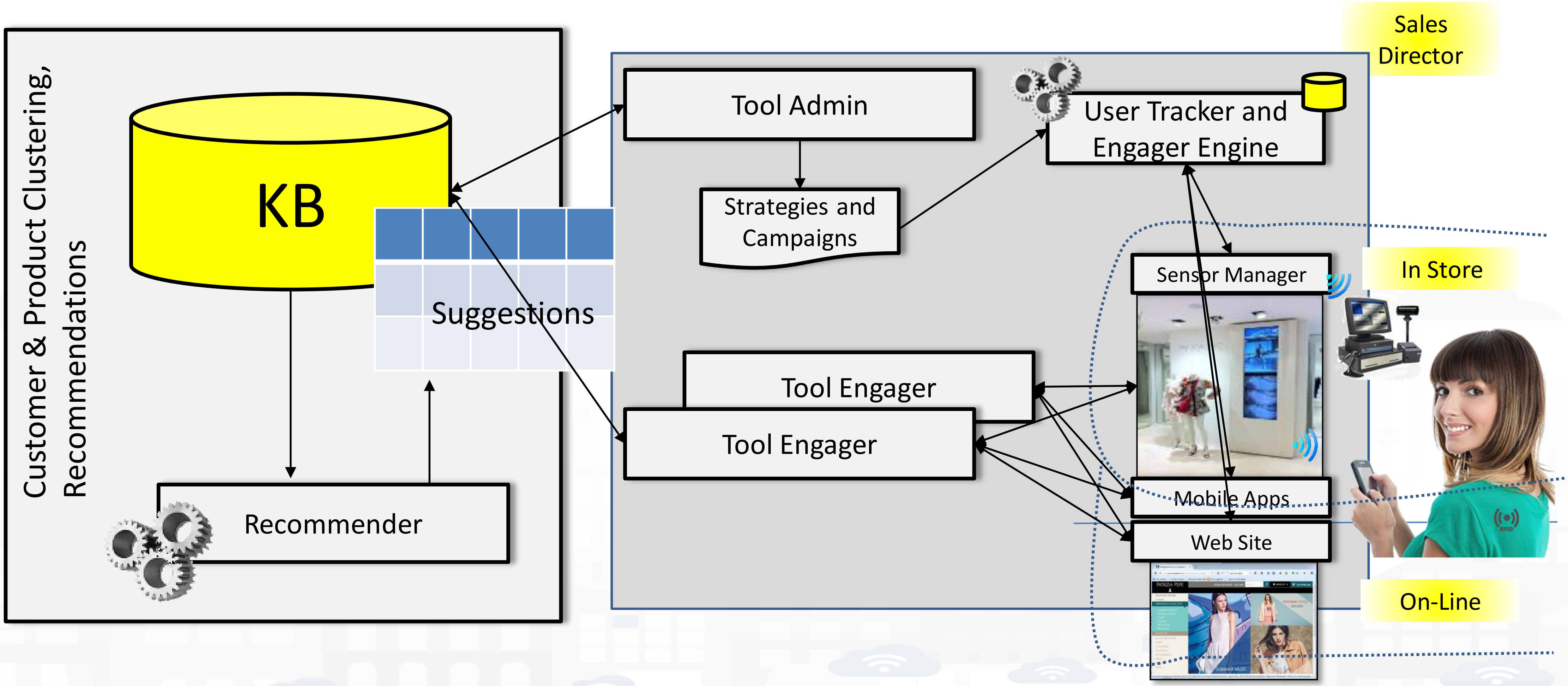
- **Feedback Project:**
  - Flexible Advanced Engagement Exploiting User Profiles and Product/Production Knowledge
  - VAR, PatriziaPepe (Tessilform), DISIT, Effective Knowledge, SICE
  - Keywords: retail, GDO, …
- **Goals and drivers:**
  - adaptive user engagement, customer experience
  - Advanced user profiling, user behaviour analysis
  - IOT and instrumentation
  - Predictive models for engagement
  - Integrated in city customer experience
- **Aiming to solve current State of the Art issues:**
  - Cold start problems in generating recommendations for new users, also addressing seasonality of products and items
  - GDPR compliance

Scenario

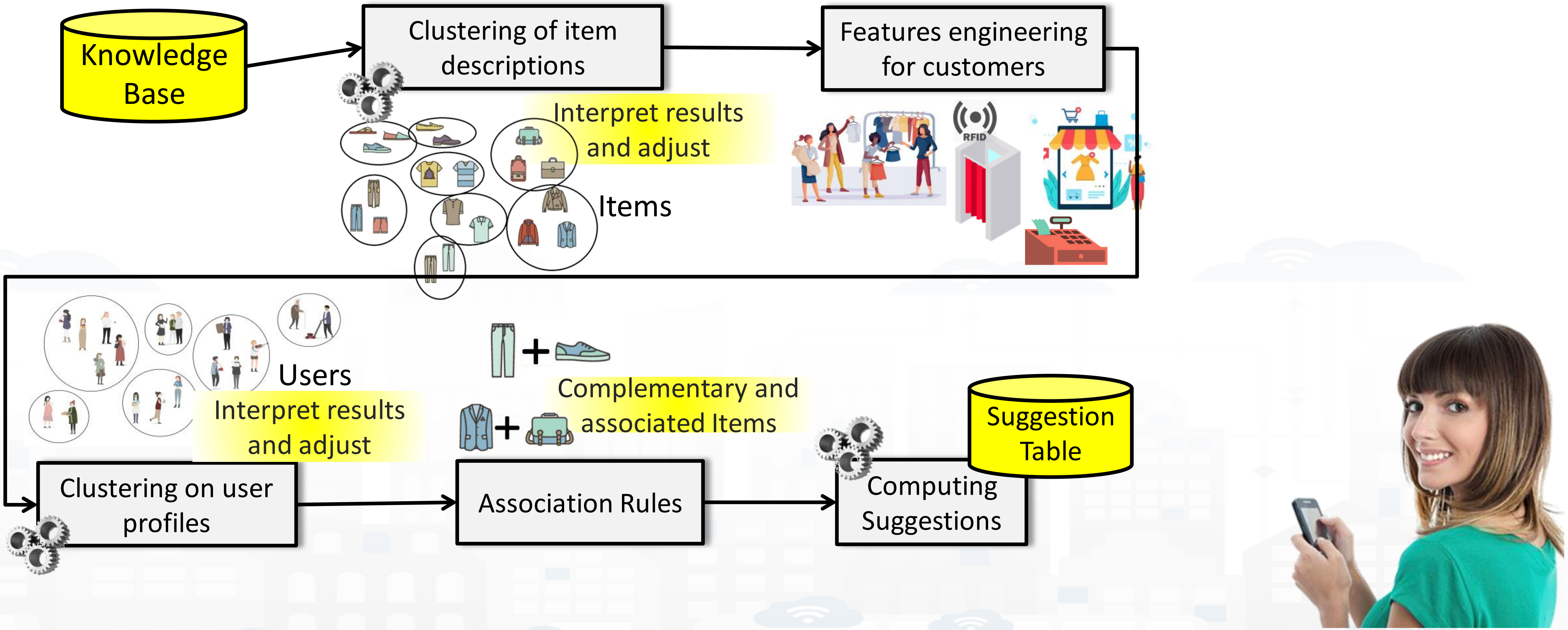


2014 2020 POR CreO PROGRAMMA OPERATIVO REGIONALE CRESCITA e OCCUPAZIONE
Fondo europeo di sviluppo regionale (Fesr)
Le ali alle tue idee

feedback


Sales Director
Customer & Product Clustering, Recommendations
KB
Suggestions
Recommender
Tool Admin
Strategies and Campaigns
User Tracker and Engager Engine
Tool Engager
Tool Engager
Sensor Manager
In Store
Mobile Apps
Web Site
On-Line

Workflow


Knowledge Base
Clustering of item descriptions
Features engineering for customers
Interpret results and adjust
Items
RFID
Users
Interpret results and adjust
Complementary and associated Items
Suggestion Table
Clustering on user profiles
Association Rules
Computing Suggestions

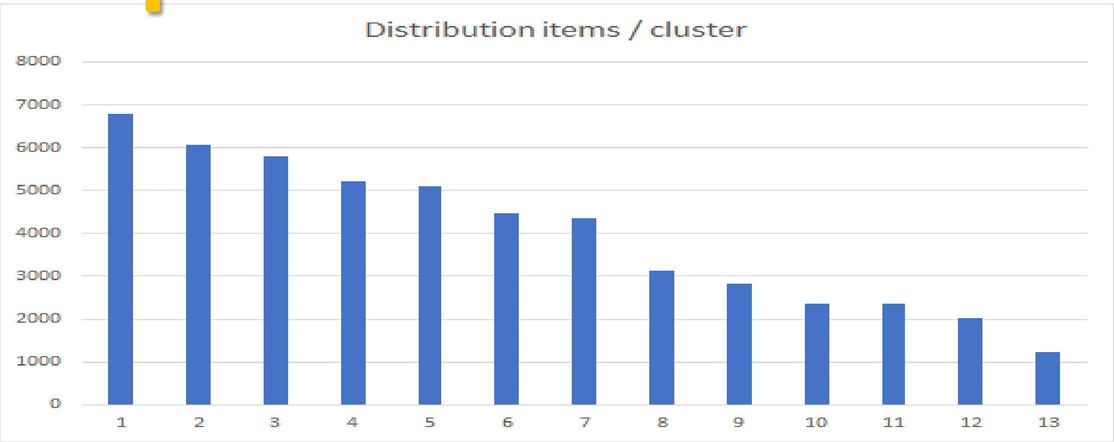
Clustering of Item Descriptions: Features

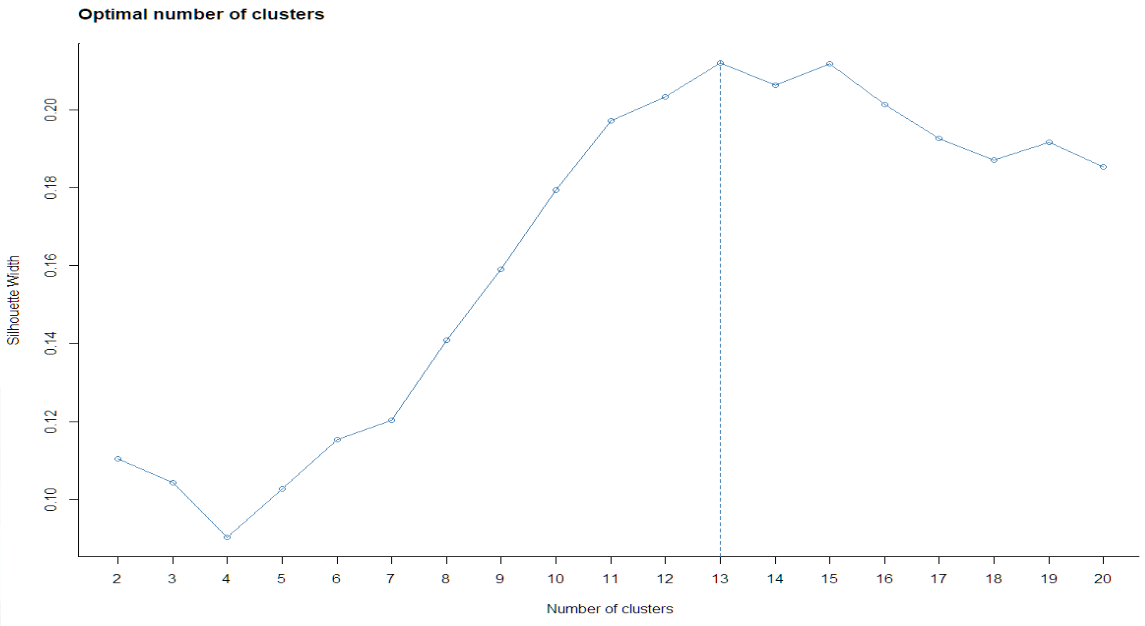
Field ID	Item Description	Example
TYPE	Type	“1A0145”, “1A0333”,…
CONFIGURATION	Configuration	“DRESS” ,“JACKET”,…
PATTERN	Color	“White”, “Red”, “Navy blue”, …
MODEL	Alphanumeric code model	“1A0145”, “1A0333”, …
PACKAGING_TYPE	Type packaging	"Packaging Basic PE", "Packaging Basic-Contin, …
PRODUCTION_CATEGORY	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	“Jewelry”, “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)
5 X NRM_CAT_LVL	Code normalized business classification level 1….5	“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)

Clustering of Item Descriptions: Results

Method: **K-medoids**

Calculate optimal number of clusters: **Silhouette analysis** (The location of the maximum is considered as the appropriate number of clusters)





Cluster	*Derived descriptions of the item clusters*	*# items sales*
1	BAG	969
2	DRESS	1171
3	TROUSERS	794
4	KNIT	678
5	T-SHIRT	674
6	ACCESSORIES (HAT - FOULARD - SCARF - NECKLACE)	596
7	SHIRT	838
8	COAT	388
9	SHOES	341
10	SKIRT	530
11	JACKET	292
12	BELT	237
13	CHILDREN'S CLOTHING	126

Features engineering for customers

Interest Cluster1	Interest Cluster2	………… …	Interest Cluster13	# buyed cluster1	# buyed cluster2	………….	# buyed cluster13

Max interest for item in the cluster

Number purchased item in the cluster

0: No interest

1: Observed (Totem, Online, etc.)

2: Tried

3: Purchased item

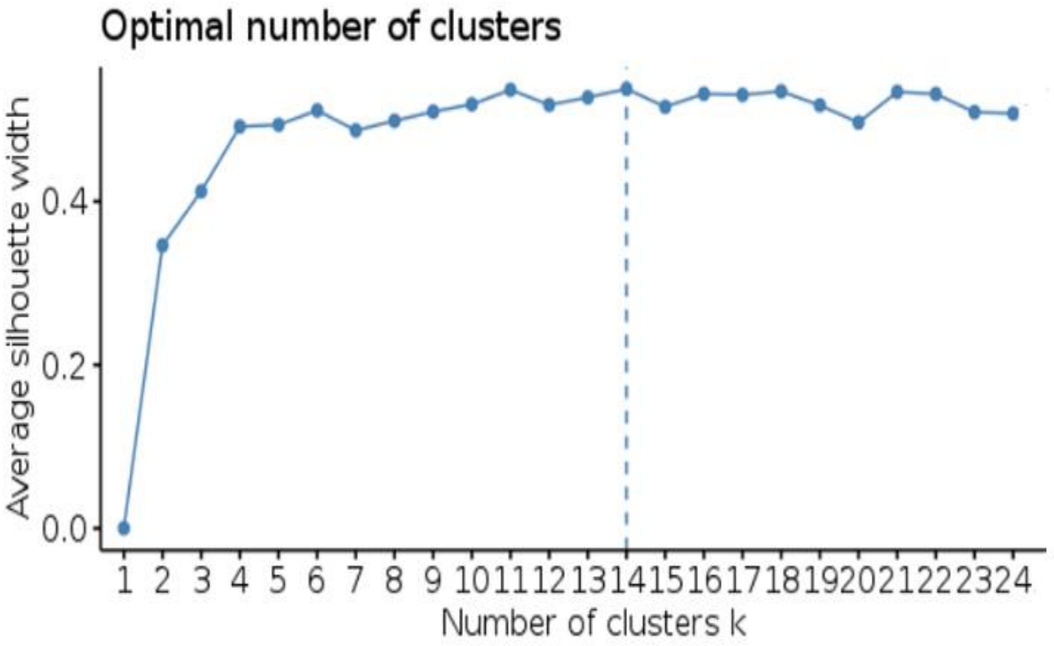
- ***Recency*** *is defined as the number of days passed since the last visit or access in a store or online;*
- ***Frequency*** *represents the frequency of purchase in number of days;*
- ***Average spending*** *is the average value of single ticket for the customer (estimated on the basis of the admin track record)*

Clustering on user profiling

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

Method: **K-means**

Calculate optimal number of clusters: **Silhouette analysis** (The location of the maximum is considered as the appropriate number of clusters)



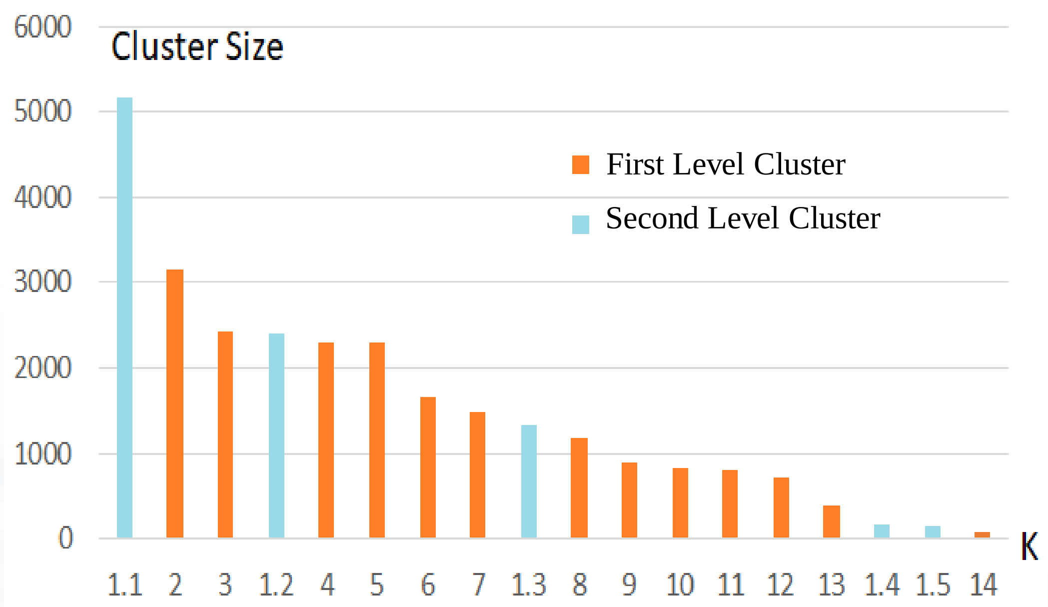
Clustering on user profiling

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 frequent online, but also buying in store with 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 mid term 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 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	9

Clustering on user profiling

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

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	5167
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



Suggestions

customer similarity for each customer cluster the most representative items are suggested;
item similarity: considering the last items purchased by the customer according to the information contained into its profile, and randomly selecting items in the same item clusters;
item complementary: considering items that may complement the last items that have been bought by the customer according to a table of complementary items;
item associated: in order to improve a customer's purchase frequency, we generated suggestions for customers who purchased an item in the last three months;
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;

Item Cluster	Complementary Item 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	3	0.34351004	0.6259701	1.355196	16776
	7	0.32391425	0.5902612	1.298495	15819
	8	0.31392182	0.5720522	1.310504	15331
	4	0.29840080	0.5437687	1.364367	214573
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

Item selection
1. Item previously not purchased
2. Confidence recommended item. Confidence established with Market Basket Analysis

2014 2020 POR CreO PROGRAMMA OPERATIVO REGIONALE CRESCITA e OCCUPAZIONE
Fondo europeo di sviluppo regionale (Fesr)
Le ali alle tue idee
feedback

Validation

- Where: store located in Florence

How

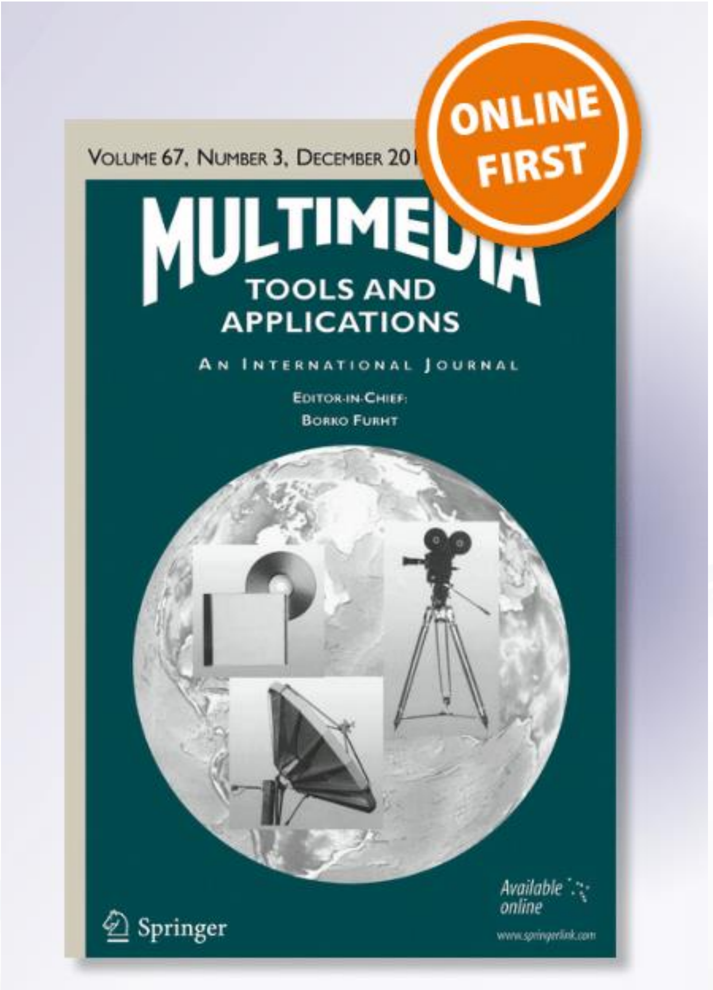
- **data collected until December 2019** to test and tune the solution, verifying if the suggestions produced were also provided by the Assistant in shops and finally acquired by the customers.
- **January - June 2020**, through transactions and verifying the shop assistants (which are the reference experts), if there was a match between suggestions and items purchased by customers. This analysis showed that on about 400 customers who bought, about 10000 suggestions were generated. On suggestions generated, the 6.36% items were purchased or tested.
- **July 2020 until December 2020**, the recommendation system was tuned on operative 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 from 67 selected customers in the trial, 3050 suggestions have been generated, while only about the 20% has been actually sent to the customers (on shops and/or email). On the items suggested, the 9.84% of them were actually acquired or tested.

Discussion

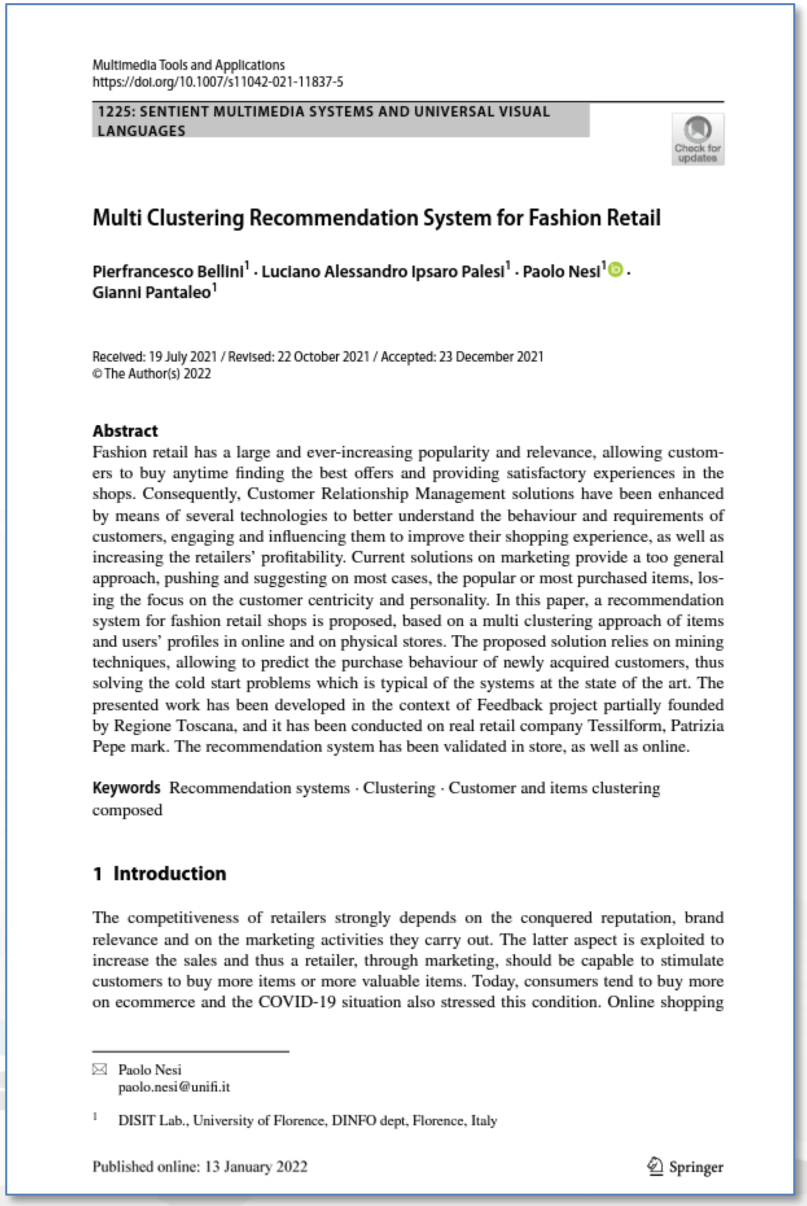
- Using the stimulus of the recommendation system, we have increased the customers’ attention of the 3.48%
- The solution is also functional in presence of a low number of customers and items
- The solution solved the cold start problems
- GDPR compliant

- P. Bellini, L. A. Ipsaro Palesi, P. Nesi, G. Pantaleo, "Multi Clustering Recommendation System for Fashion Retail", Multimedia Tools and Applications, Springer, 2022.

- https://link.springer.com/article/10.1007/s11042-021-11837-5



Springer



Multimedia Tools and Applications
https://doi.org/10.1007/s11042-021-11837-5

1225: SENTIENT MULTIMEDIA SYSTEMS AND UNIVERSAL VISUAL LANGUAGES

Check for updates

Multi Clustering Recommendation System for Fashion Retail

Pierfrancesco Bellini[1] · Luciano Alessandro Ipsaro Palesi[1] · Paolo Nesi[1] · Gianni Pantaleo[1]

Received: 19 July 2021 / Revised: 22 October 2021 / Accepted: 23 December 2021

Abstract
Fashion retail has a large and ever-increasing popularity and relevance, allowing customers to buy anytime 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 paper, a recommendation system for fashion retail shops is proposed, based on a multi clustering approach of items and users' profiles in online and on 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 Feedback project partially founded 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.

Keywords Recommendation systems · Clustering · Customer and items clustering composed

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 the 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

✉ Paolo Nesi
paolo.nesi@unifi.it

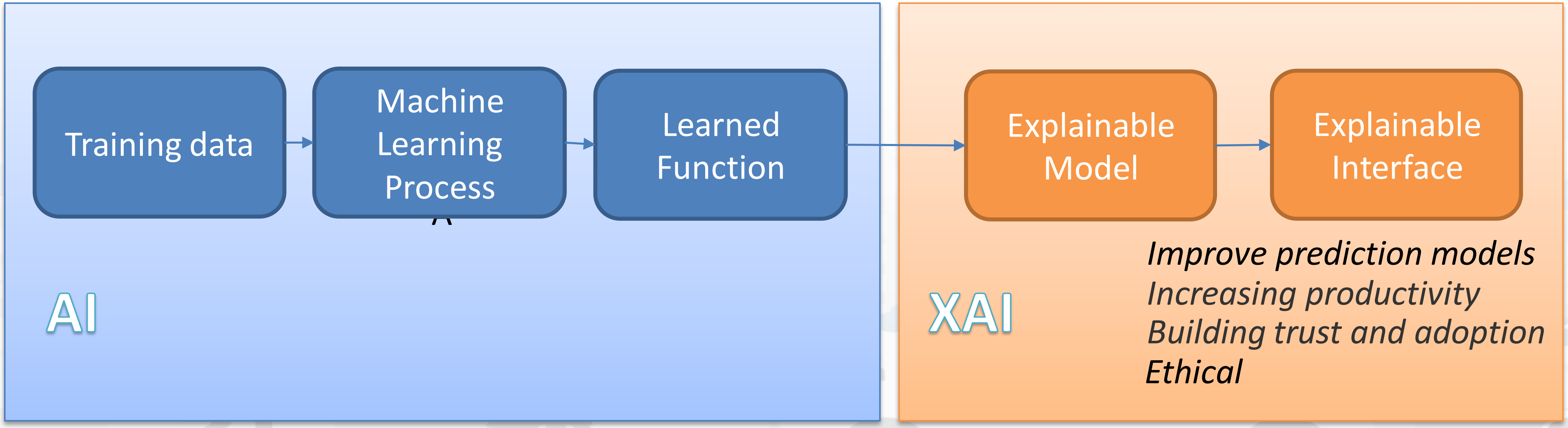
1 DISIT Lab., University of Florence, DINFO dept, Florence, Italy

Published online: 13 January 2022

Springer

XAI: Explainable artificial intelligence

Explainable artificial intelligence (XAI) is a set of processes and methods that allows human users to comprehend and trust the results and output created by machine learning algorithms.



White Box vs. Black Box Models

A **white-box** model is explainable by design. Therefore, it does not require additional capabilities to be explainable:

- Linear regression,
- Logistic regression,
- Decision Tree,
- Naive Bayes,
- KNNs
- ....

A **black-box model** is not explainable by itself. Therefore, to make a black-box model explainable, we have to adopt several techniques to extract explanations from the inner logic or the outputs of the model.

- CNN
- LSTM
- ....

Challenges in XAI Design

Model Developers: To improve or debug the model.

Business Owners/Administrators: To evaluate AI capabilities, ensure regulatory compliance, and guide adoption decisions.

Decision-Makers: To build trust in AI and make informed decisions based on its outputs.

Impacted Groups: To seek recourse or contest decisions that affect their lives.

Regulatory Bodies: To audit for compliance with legal, ethical, and safety standards.

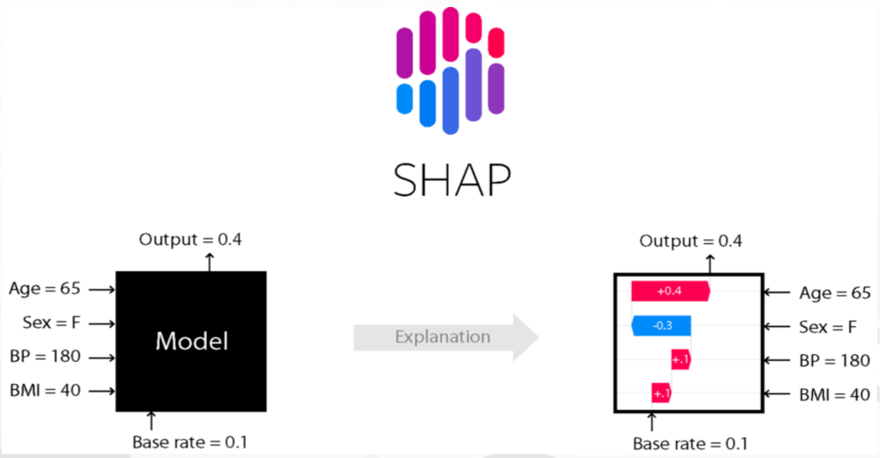
XAI Question/Methods

Question	Ways to explain	Example XAI methods
How (global model-wide)	Asking about the general logic or process the AI follows to have a global view	ProfWeight, Global feature importance, Global feature inspection plots, Tree surrogates
Why (a given prediction)	Asking about the reason behind a specific prediction.	LIME, SHAP, LOCO, Anchors, ProtoDash
Why Not (a different prediction)	Asking why the prediction is different from an expected or desired outcome.	CEM, Counterfactuals, ProtoDash
How to Be That (a different prediction)	Asking about ways to change the instance to get a different prediction.	CEM, Counterfactuals, Counterfactual instances, DiCE
How to Still Be This (the current prediction)	Asking what change is allowed for the instance to still get the same prediction.	CEM, Anchors
What if	Asking how the prediction changes if the input changes.	PDP, ALE, ICE

SHAP

SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions

https://github.com/slundberg/shap



SHAP

Advantages and Limitations

Advantages:

Fairness: Based on Shapley values from game theory.

Model-Agnostic: Works with any ML model.

Additivity: Attributions sum to the model's prediction.

Visualization: Intuitive plots for interpretation.

Captures Interactions: Handles individual and feature interaction effects.

Limitations:

High Computational Cost: Expensive for large/complex models.

Approximation Errors: KernelSHAP may introduce inaccuracies.

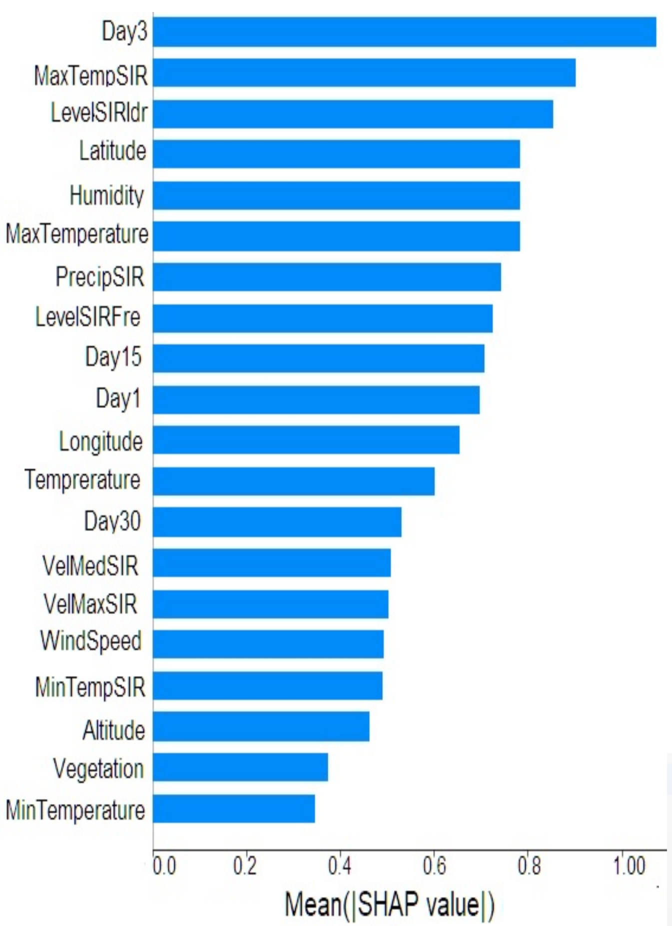
Baseline Sensitivity: Results depend on the chosen baseline.

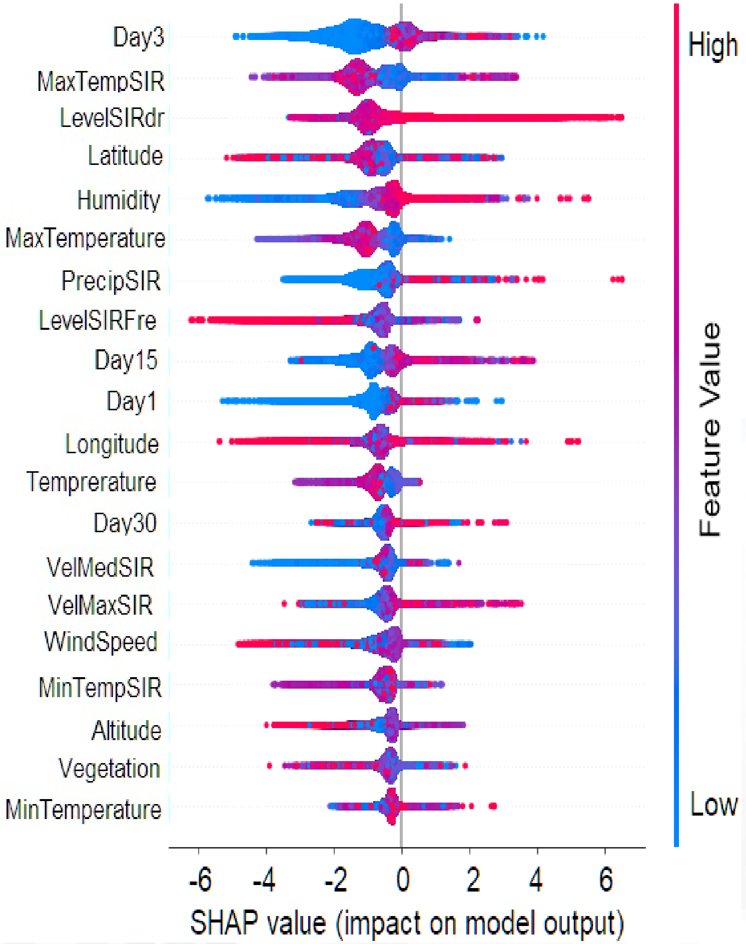
Scalability Issues: Struggles with high-dimensional data.

Expertise Required: Can be challenging to interpret.

Global interpretability

```
with tf.device('/device:GPU:0'):
    explainer = shap.TreeExplainer(MODEL)
    shap_values = explainer.shap_values(X_train)
```





shap.summary_plot(shap_values, features_names, plot_type="bar")

shap.summary_plot(shap_values, X_train,features_names)

•***Feature importance***: Variables are ranked in descending order.

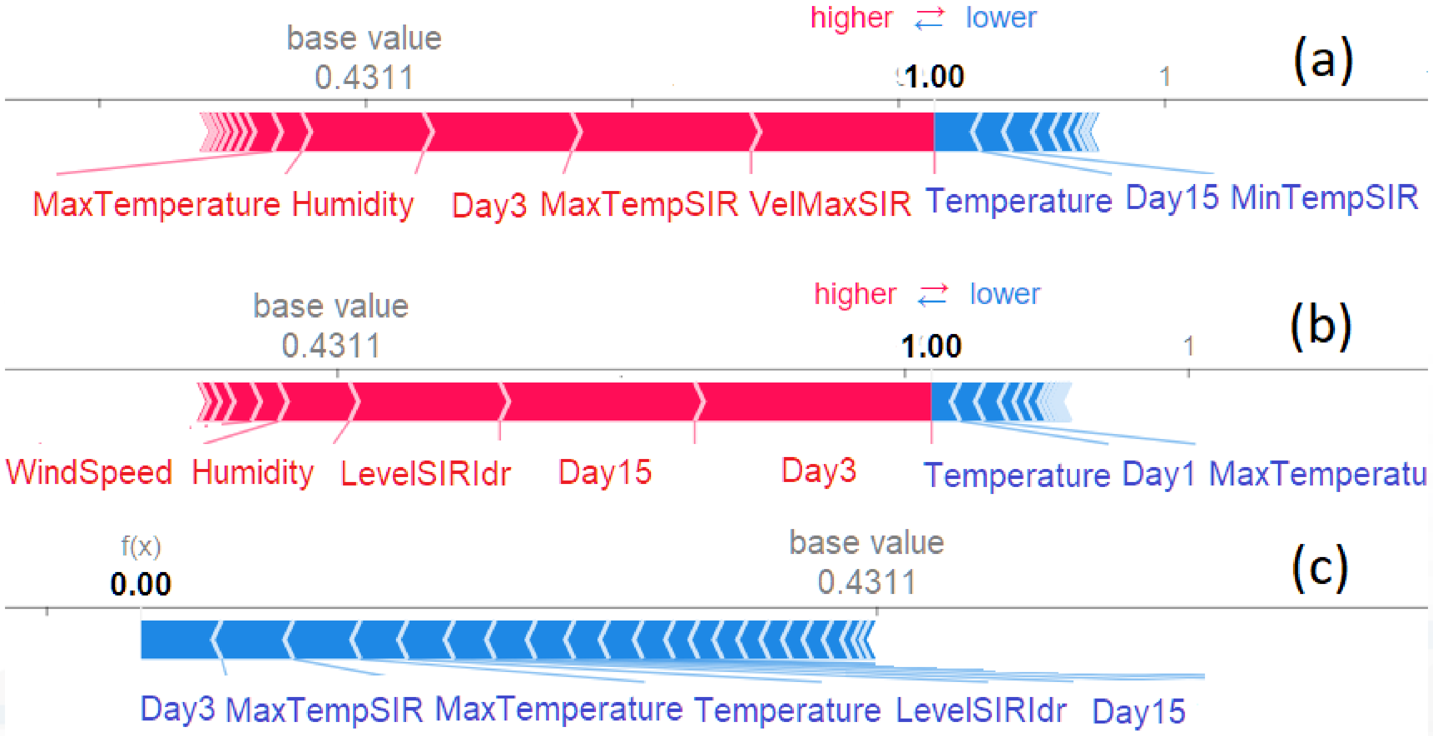
•***Impact***: The horizontal location shows whether the effect of that value *is associated with a higher or lower prediction*.

•***Original value***: Color shows whether that variable is high (in red) or low (in blue) for that observation.

•***Correlation***: A *high* level of "Day3" or "PrecipiSIR" content has a high and *positive* impact on the classification. The "high" comes from the red color, and the "positive" impact is shown on the X-axis.

Local interpretability

```
with tf.device('/device:GPU:0'):
    explainer = shap.TreeExplainer(MODEL)
    shap_values = explainer.shap_values(X_train)
```




```
shap.force_plot(explainer.expected_value,
shap_values[7,:],fields)
```

The ability to explain each prediction, is a very important promise in an explainable AI.

(a) value of VelMaxSIR, MaxTempSIR, Day3 and Humidity contributed significantly to the classification of the observation as a landslide event.

(b) values related to rainfall in the last days, LevelSIRIdr and Humidity given a relevant contribution to the landslide event prediction.

(c) the value of features: Day3, MaxTempSIR, MaxTemperature, Temperature and LevelSIRdr have been determinant for the classification of the observation into a no landslide event.

Local interpretability – CNN-LSTM

Explanation of prediction generated by model for fault

Explanation of prediction generated by model for normality


```
DE_mis = shap.DeepExplainer(classifierLoad,X_test_df)
shap_values_mis = DE_mis.shap_values(X_test_df[(minutes-21):(minutes-20)])
```

TOP

Predictive Maintenance

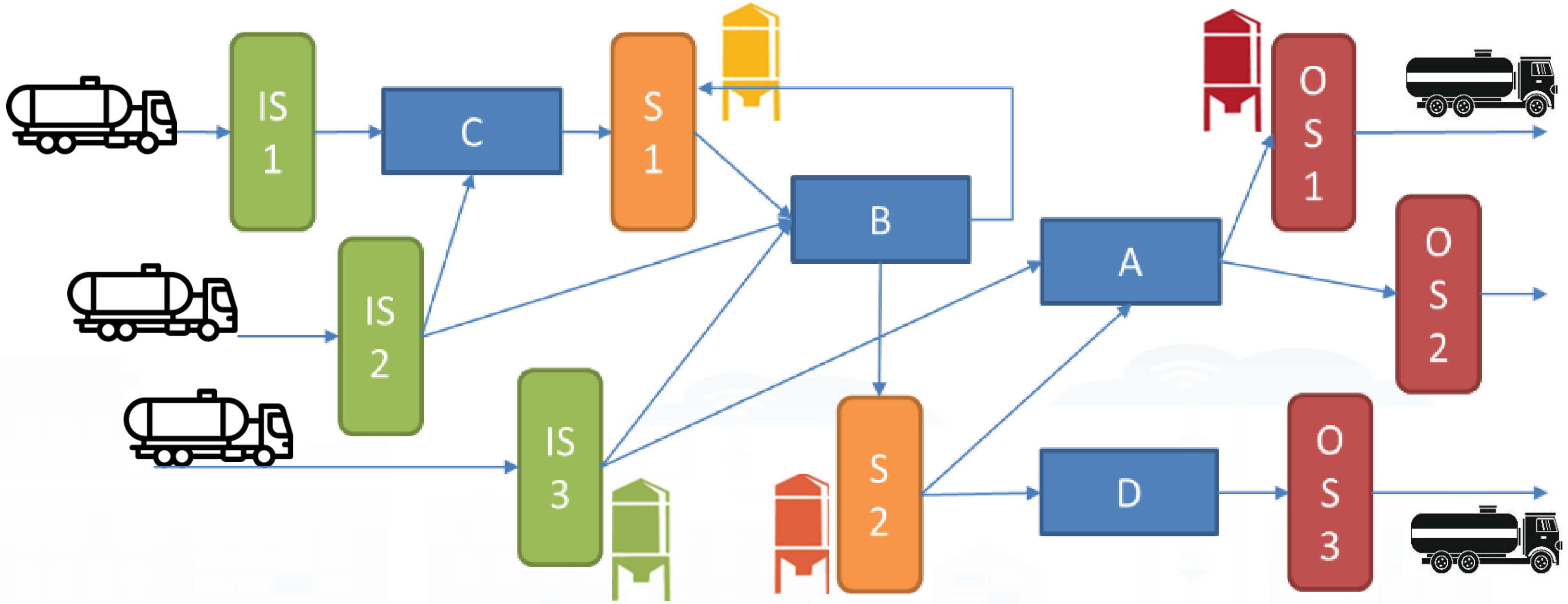
Data Analytic

Industry 4.0

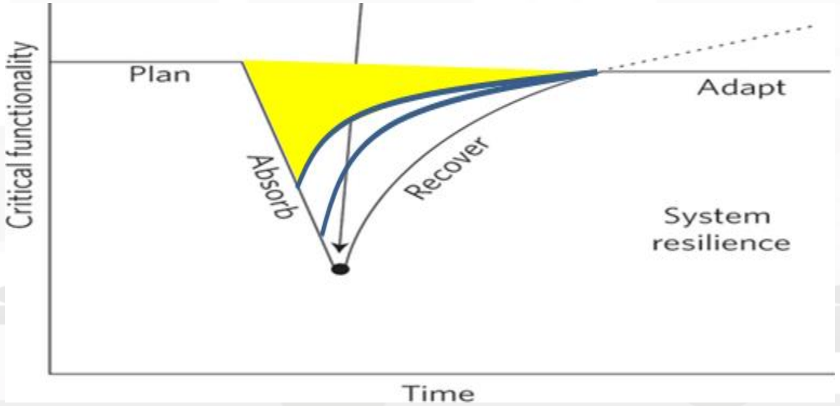
- **ALTAIR SODA-4.0 project**
  – maximize the efficiency and productivity of plants, reducing downtime
  – in order to improve competitiveness in the market

- **Goals and drivers:**
  – Business intelligence tools on maintenance data
  – predictive maintenance approach into the whole control and management systems Predictive models for engagement
  – predict plant failures 60 minutes before it happens
  – Provide indications on the area of failure via XAI

Complex cause-effect realtionships



- **Elements:**
  - Machines: A…C
  - Storage: silos…
  - Flows:…
- **Dependencies**
  - Cascade effects
- **Early warning**
  - Reduction of costs
  - Recovering from failure is more expensive than correcting in advance
  - Possible advanced replan and reschedule: secondary solutions



Workflow

Data acquisition
Production - Maintenance

Feature selection,
Feature engineering

CNN-LSTM Training

Validation AI
Model

Classification
Model

CNN-LSTM
Engine

F/NF

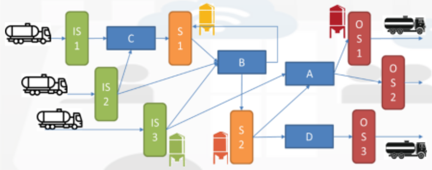
Explainable/XAI
(Shap)

{features (t)}

Business Intelligence for
Maintenance

integration of workflow management system for maintenance with general control systems and data flow

Structure of the plant

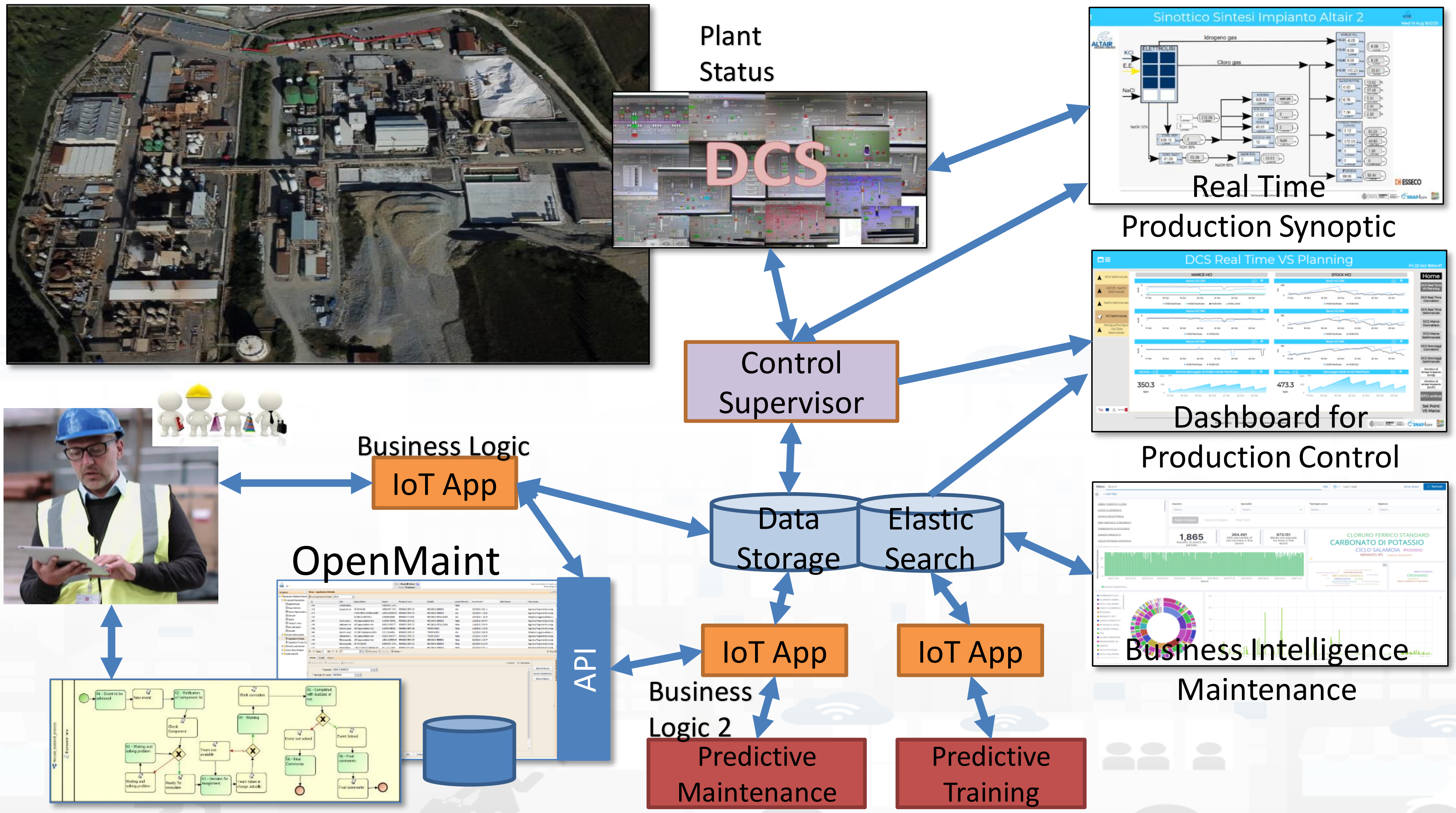


Decision making
on more cases

KPI

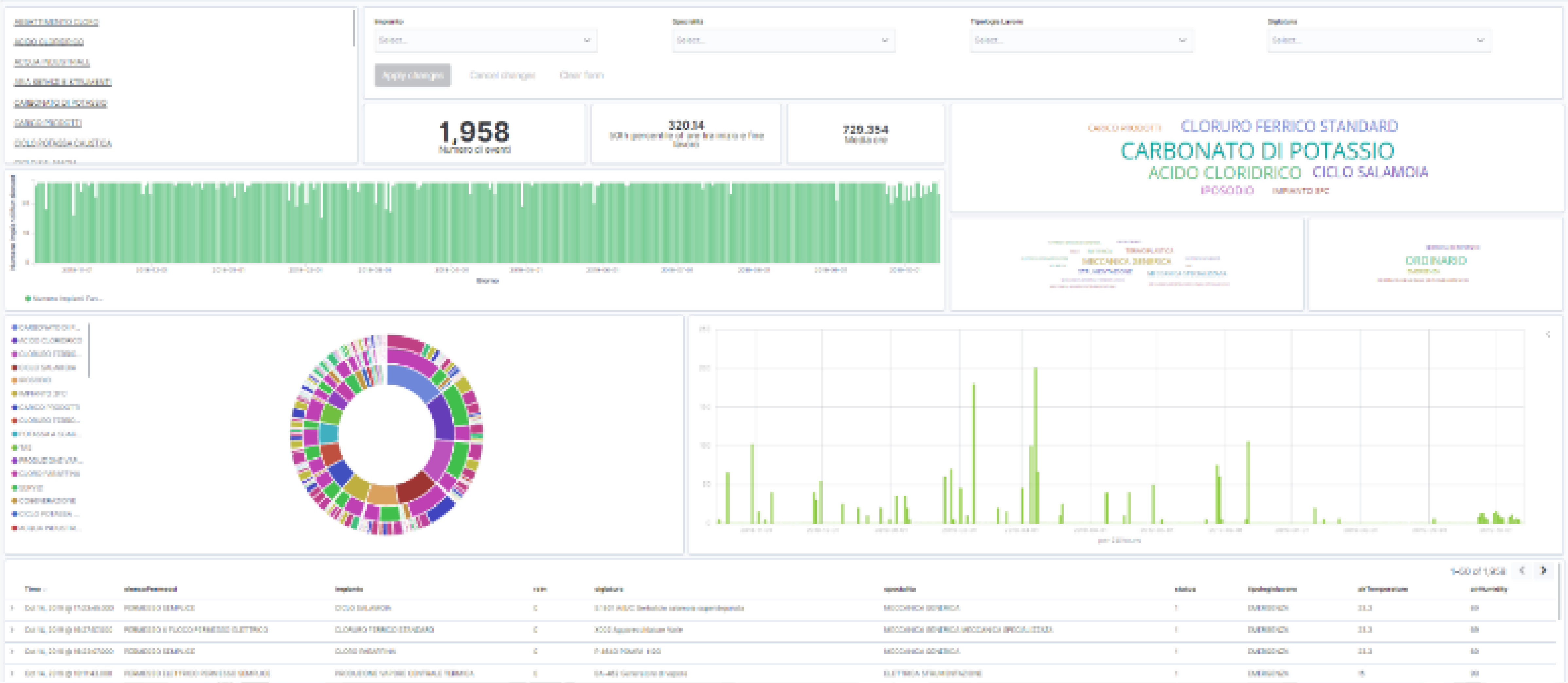
CNN: Convolutionary NN
LSTM: Long Short-Term Memory
(deep learning model)

Solution



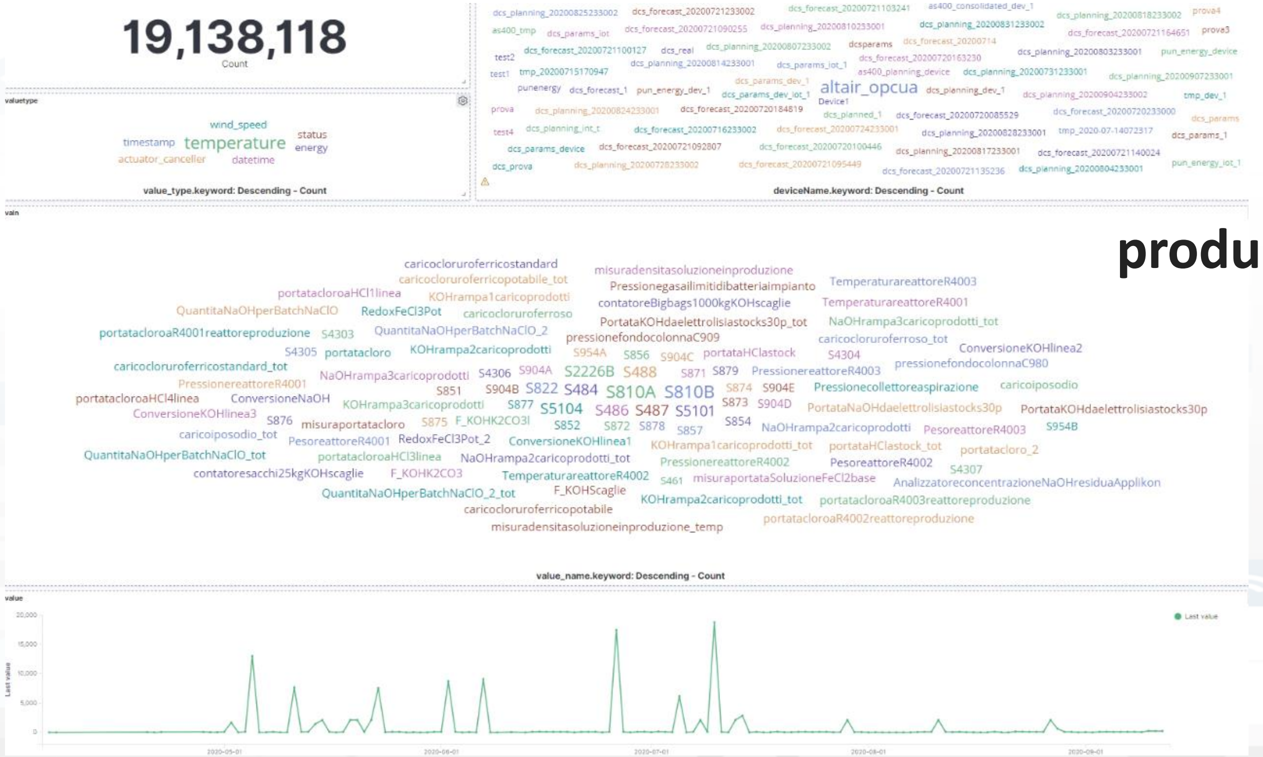
Plant
Status
DCS
Sinottico Sintesi Impianto Altair 2
Real Time
Production Synoptic
DCS Real Time VS Planning
Dashboard for
Production Control
Control
Supervisor
Business Logic
IoT App
OpenMaint
API
Data
Storage
Elastic
Search
IoT App
IoT App
Business
Logic 2
Predictive
Maintenance
Predictive
Training
Business Intelligence
Maintenance

Business Intelligence



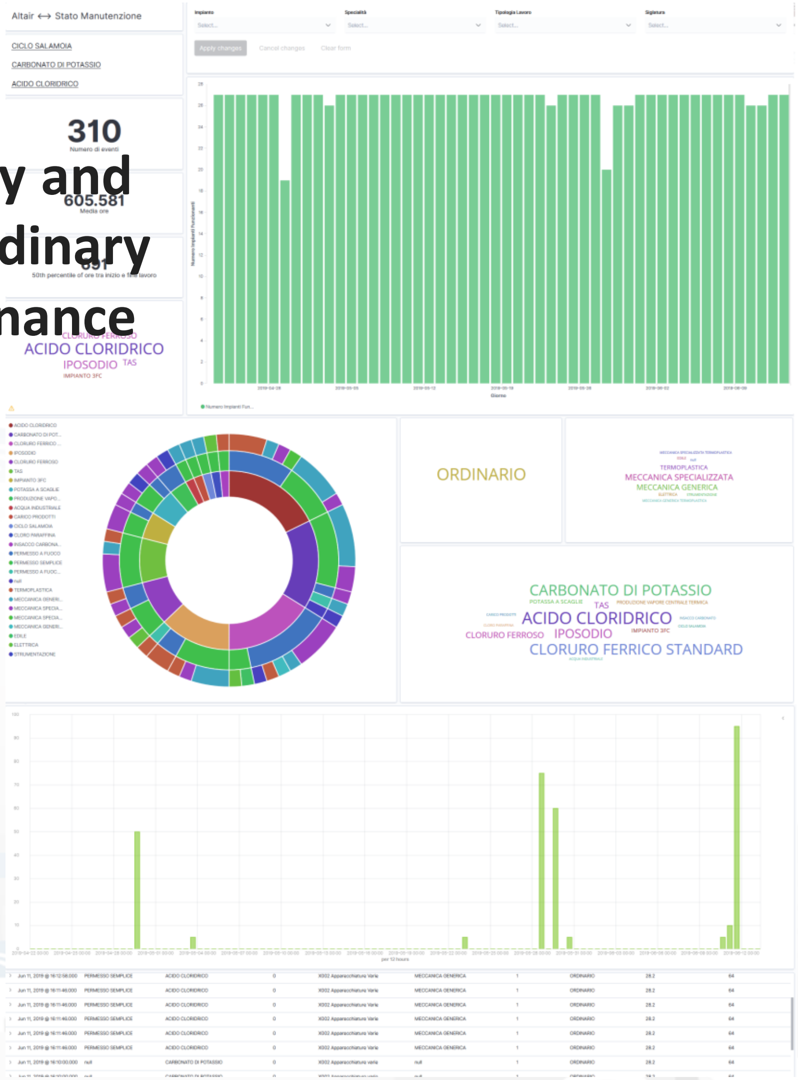
Dashboard for monitoring:

- Tool: ElasticSearch – Kibana
- Realtime



production

ordinary and extraordinary maintenance



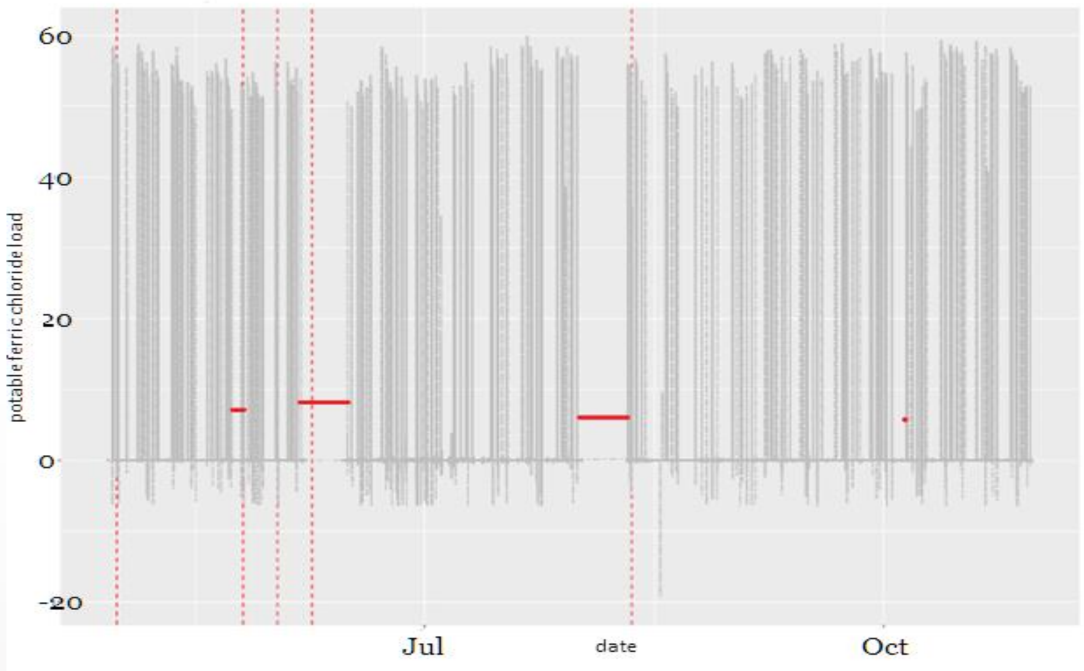
Production:

- 1-minute observation from 2020-04-28 to 2021-01-04
- 343.183 observations for 147 features/variables
- production, storage, status, several temperatures of elements, gear plants, process/safety parameters, chemicals compounds produced

Fault:

- List all the details: 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 "

Example of a failure



Overview Features

Feature	Plant	Description	Unit of measure
TempreactoreR4001 - TempreactoreR4002 - TempreactorR4003	chlorine paraffins (CPS)	reactor temperature indication	°C
S904A - S904B - S904C	Potable Ferric std	Storage level indication	%
S4304	chlorine paraffins (CPS)	Storage level indication	%
standardFerric Chloride	Potable Ferric std	flow rate measurement and totalization	m3
potFerricChloride	Potable Ferric Chloride	flow rate measurement and totalization	m3
S904E - S904D	Potable Ferric Chloride	Storage level indication	%
QuantNaOHperBatchNaClO - QuantNaOHBatchNaClO_2	NaOH KOH	flow rate measure and totalization	lt – m3
ConversionNaOH - ConversionKOHlinea1	NaOH KOH	electrolysis load adjustment (production)	kA
KOH_1_charge - KOH_2_charge	NaOH KOH	flow rate measure and totalization	m3
S487 - S484 - S5104	NaOH KOH	Storage level indication	%
hypo sodium	sodium hypochlorite	quantity of material produced	m3
S851 - S852 - S854 - S856 - S857	sodium hypochlorite	Storage level indication	%
S871	HCl	Storage level indication	%
RedoxFeCl3Pot	Ferric Chloride std	potential measure redox Ferric Chloride	mV

Features engineering

For S857, S856, S851, S852, S854, S871, S487, S484, S5104, S904E, S904D, S4304, S904C, S904B, S904A (level of the storages)

43 features for 343183 minutes

difference with the previous minute to highlight the total daily production of a given substance.

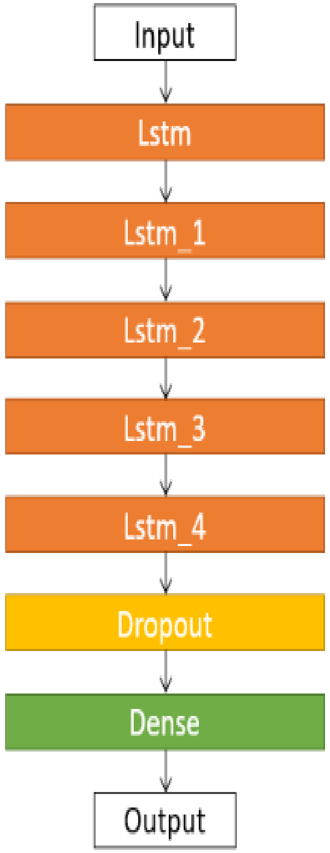
37286 minutes of failure leading to downtime.

Classification model

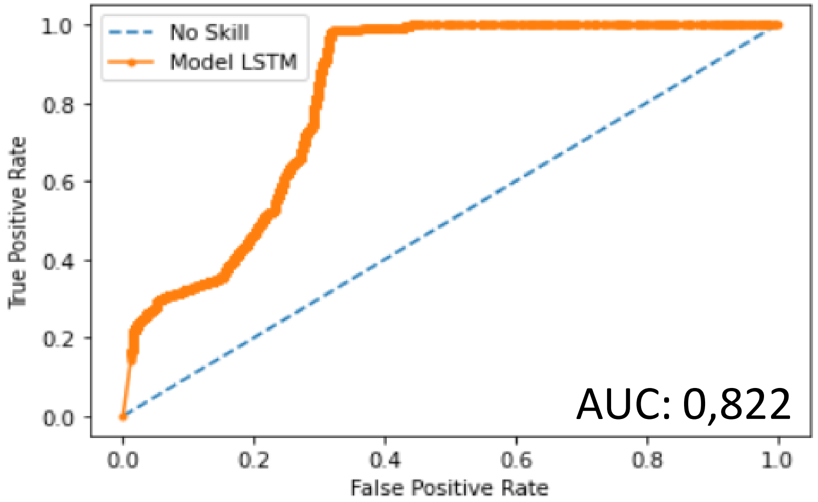
Input: Time series 20 minutes
Prediction 1 hour in the future

$$(X_1, X_2, \dots \dots, X_{20})\ (Y_{80})$$
$$(X_2, X_3, \dots \dots, X_{21})\ (Y_{81})$$
$$\dots \dots \dots \dots$$
$$(X_n, X_{n+1}, \dots \dots, X_{n+19})\ (Y_{n+79})$$

Classification model LSTM




```
_________________________________________________________________
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
```

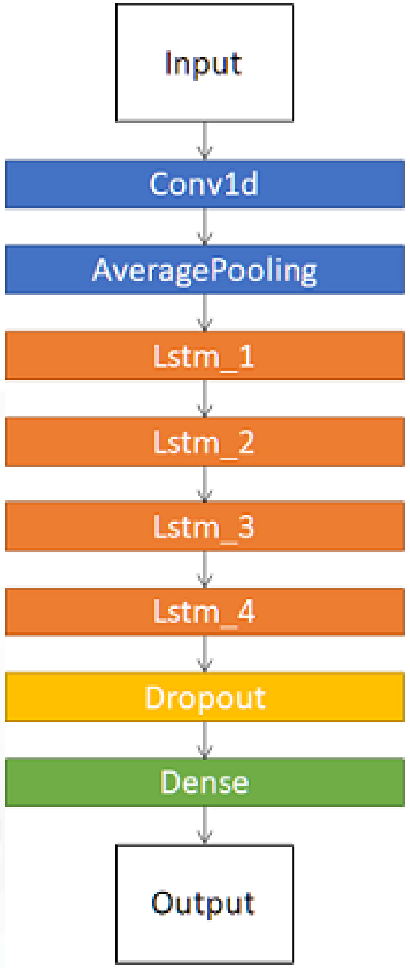


Predicted Class / Actual Class	Normality	Fault
Normality	43485	3229
Fault	3246	1436

Accuracy %
0,874

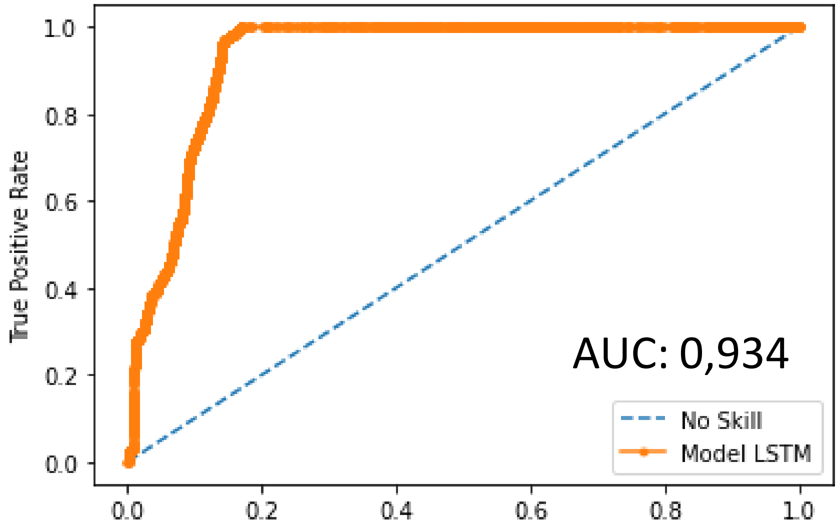
	Precision %	Recall %	F_1 score %
weighted avg	0.87	0.87	0.87

Classification model CNN-LSTM



Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 20, 64)	8320
average_pooling1d (AveragePo	(None, 10, 64)	0
lstm (LSTM)	(None, 10, 200)	212000
lstm_1 (LSTM)	(None, 10, 200)	320800
lstm_2 (LSTM)	(None, 10, 200)	320800
lstm_3 (LSTM)	(None, 10, 200)	320800
lstm_4 (LSTM)	(None, 100)	120400
dropout (Dropout)	(None, 100)	0
dense (Dense)	(None, 1)	101

Total params: 1,303,221
Trainable params: 1,303,221
Non-trainable params: 0



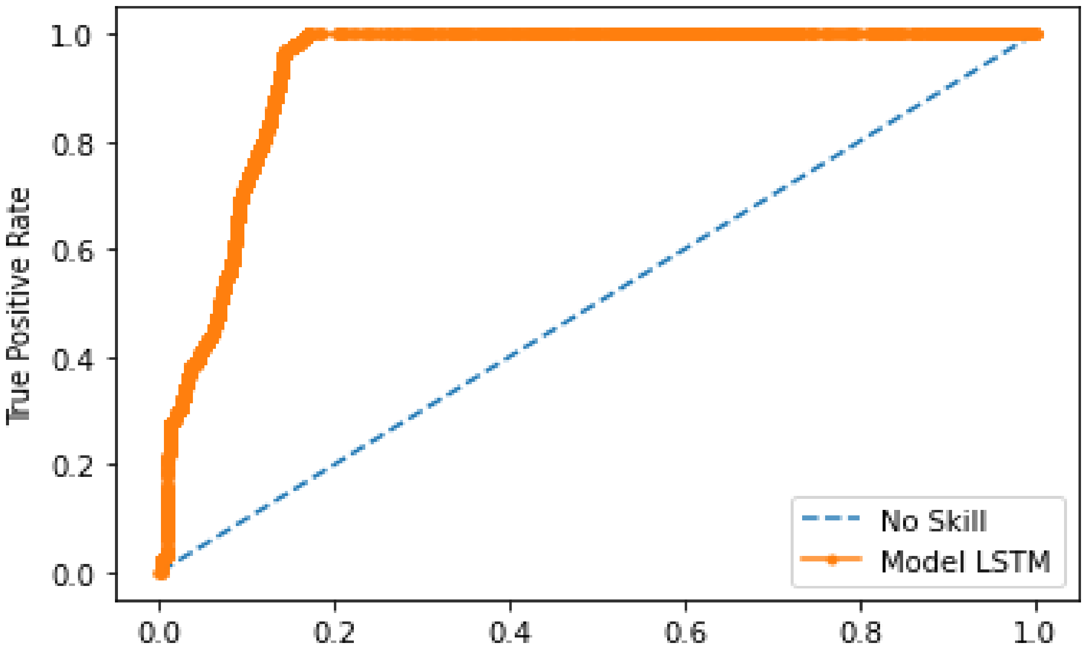
Predicted Class / Actual Class	Normality	Fault
Normality	45811	903
Fault	3306	1376

Accuracy %
0,918

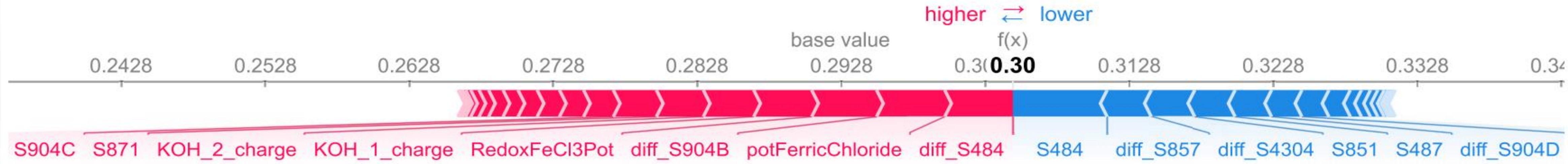
	Precision %	Recall %	F_1 score %
weighted avg	0.90	0.92	0.90

Preditive capabilities

- Deep Learning: LSTM, CNN-LSTM approached
- Explainable AI: Identification of possible causes of fault



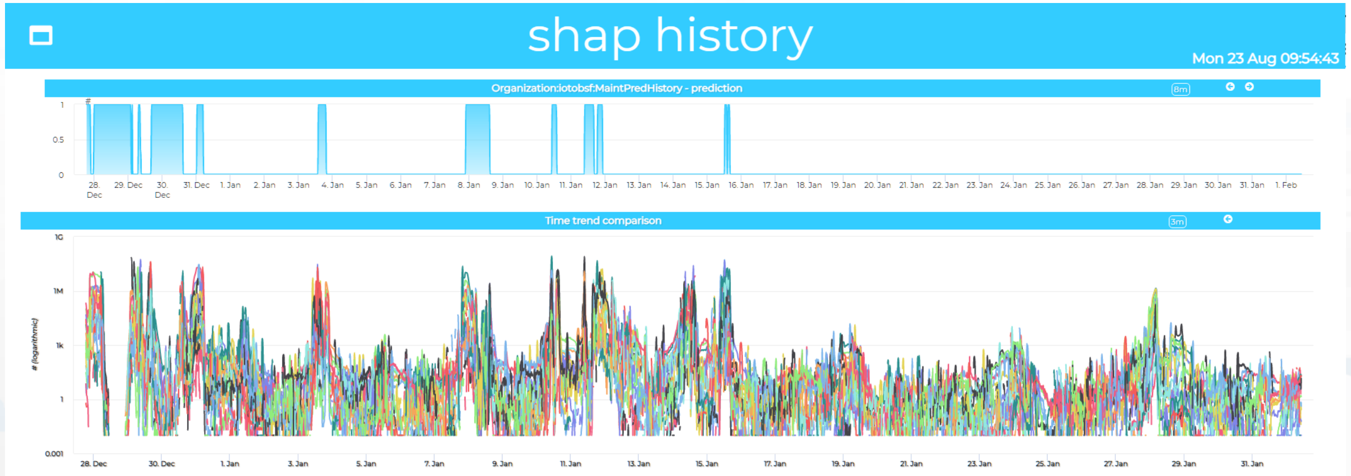
	Precision %	Recall %	F_1 score %
weighted avg	0.90	0.92	0.90



Explainable/XAI - CNN-LSTM (SHAP)

Explanation of prediction generated by model for fault

Explanation of prediction generated by model for normality

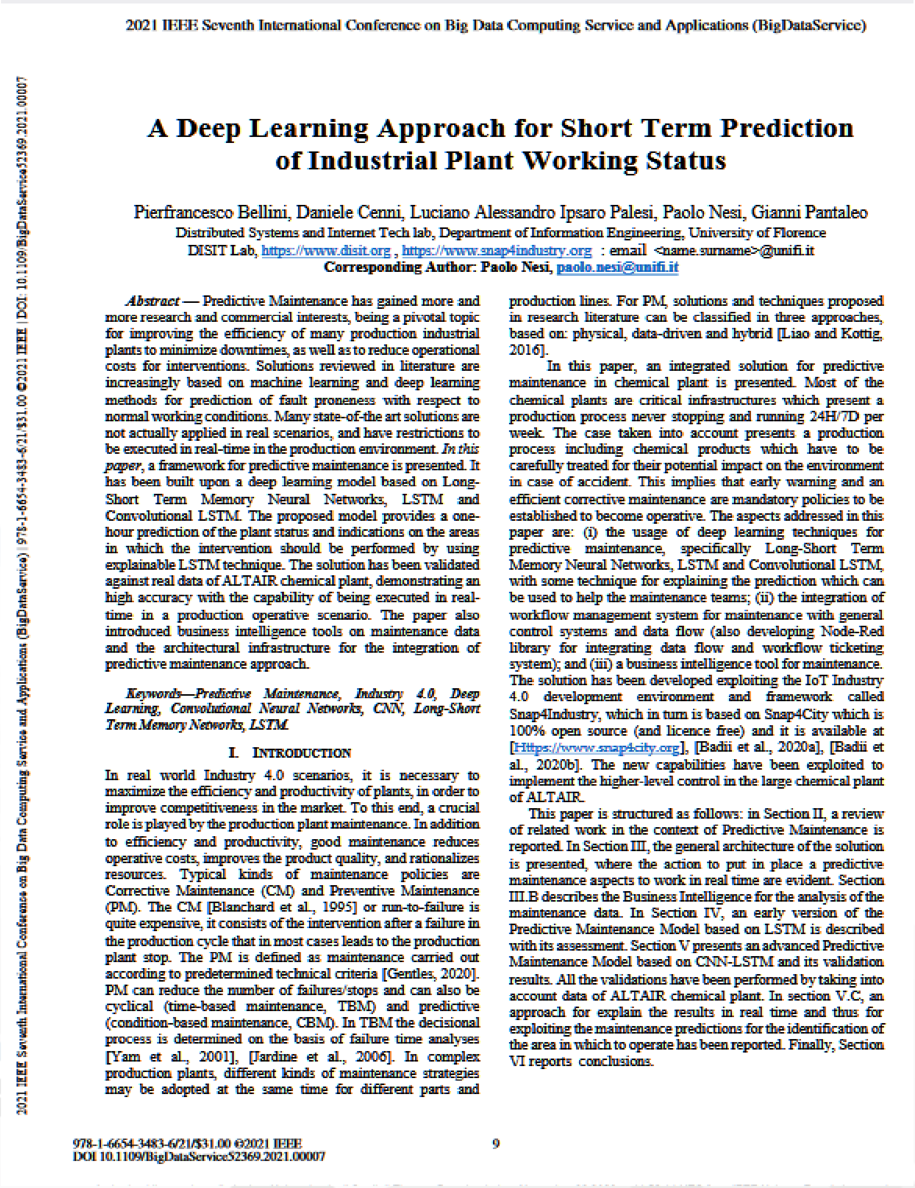


Considerations

- Experimental results shown an average Accuracy of 91.8% and an average F1-score of 90%, which are very satisfactory results
- Explanation of the predictions provides suggestions for the maintenance teams in terms of areas of intervention.
- Large renovation of the production infrastructure.

7th IEEE International Conference on Big Data Service and Machine Learning

- 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," doi: 10.1109/ACCESS.2022.3158328.
- https://ieeexplore.ieee.org/abstract/document/9564391



2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)

2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService) | 978-1-6654-3483-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigDataService52369.2021.00007

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

Pierfrancesco Bellini, Daniele Cenni, Luciano Alessandro Ipsaro Palesi, Paolo Nesi, Gianni Pantaleo
Distributed Systems and Internet Tech lab, Department of Information Engineering, University of Florence
DISIT Lab, https://www.disit.org , https://www.snap4industry.org : email <name.surname>@unifi.it
Corresponding Author: Paolo Nesi, paolo.nesi@unifi.it

Abstract — 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 literature are increasingly based on machine learning and deep learning methods for prediction of fault proneness with respect to normal working conditions. 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 paper*, 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 high accuracy with the capability of being executed in real-time in a production operative scenario. The paper also introduced business intelligence tools on maintenance data and the architectural infrastructure for the integration of predictive maintenance approach.

Keywords—Predictive Maintenance, Industry 4.0, Deep Learning, Convolutional Neural Networks, CNN, Long-Short Term Memory Networks, LSTM.

I. 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 the product quality, and rationalizes resources. Typical kinds of maintenance policies are Corrective Maintenance (CM) and Preventive Maintenance (PM). The CM [Blanchard et al., 1995] 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 [Gentles, 2020]. 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 [Yam et al., 2001], [Jardine et al., 2006]. 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 and hybrid [Liao and Kottig, 2016].

In this paper, an integrated solution for predictive maintenance in chemical plant is presented. Most of the chemical plants are critical infrastructures which present a production process never stopping and running 24H/7D per week. 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 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 paper 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], [Badii et al., 2020a], [Badii et al., 2020b]. The new capabilities have been exploited to implement the higher-level control in the large chemical plant of ALTAIR.

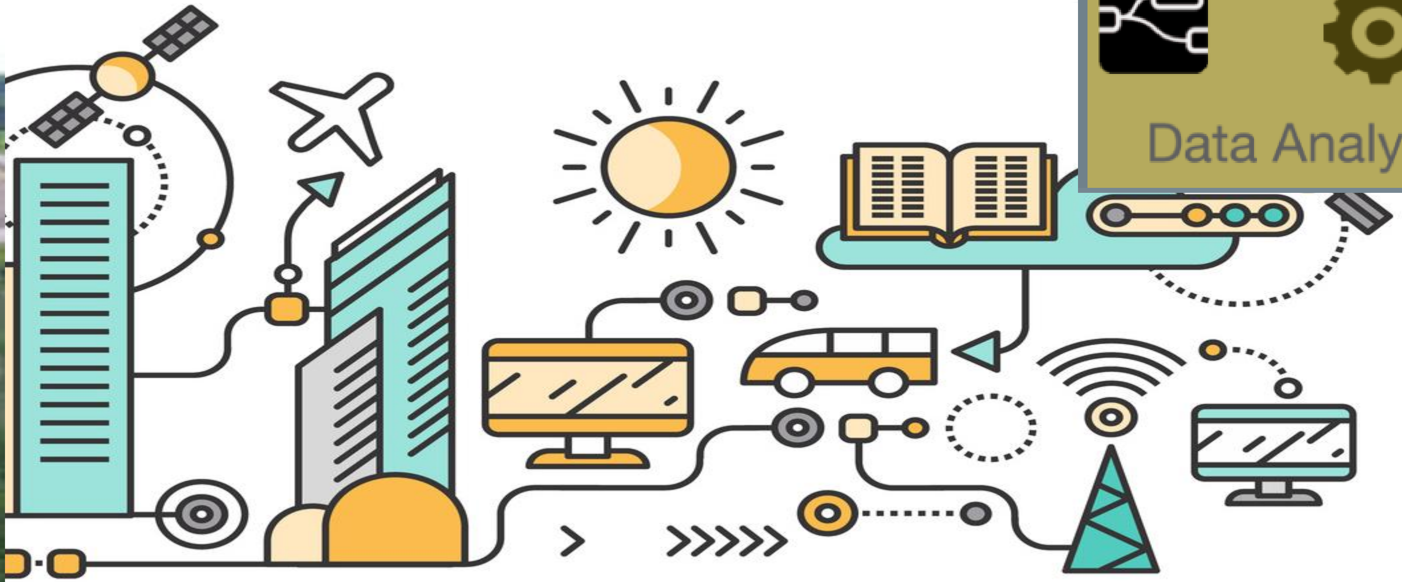
This paper is structured as follows: in Section II, a review of related work in the context of Predictive Maintenance is reported. In Section III, the general architecture of the solution is presented, where the action to put in place a predictive maintenance aspects to work in real time are evident. Section III.B describes the Business Intelligence for the analysis of the maintenance data. In Section IV, an early version of the Predictive Maintenance Model based on LSTM is described with its assessment. Section V 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 V.C, an approach for explain 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 VI reports conclusions.

978-1-6654-3483-6/21/$31.00 ©2021 IEEE
DOI 10.1109/BigDataService52369.2021.00007

9

Predicting Land sliding

TOP



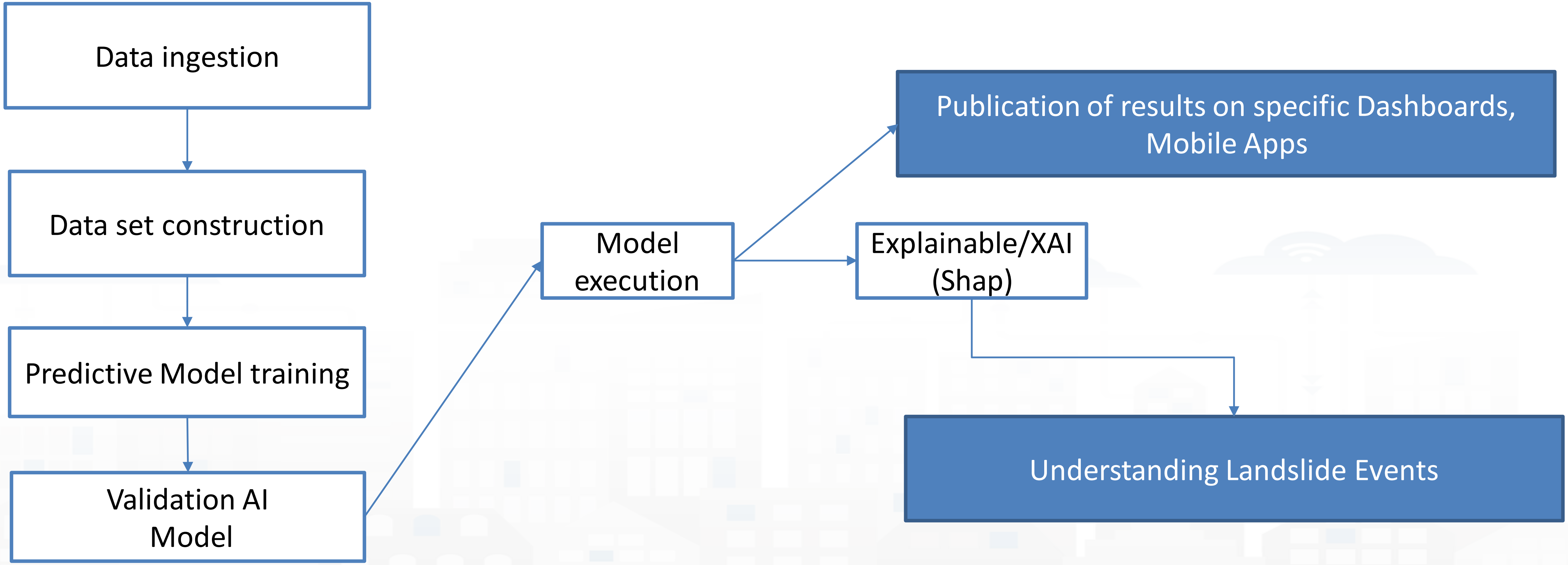
Data Analytic

PC4City

- **PC4City project**
  - Development of a Civil Protection platform through Kn4City
- **Goals:**
  - Predicting Landslide Events 24 h
  - Understanding Landslide Events with Explainable AI

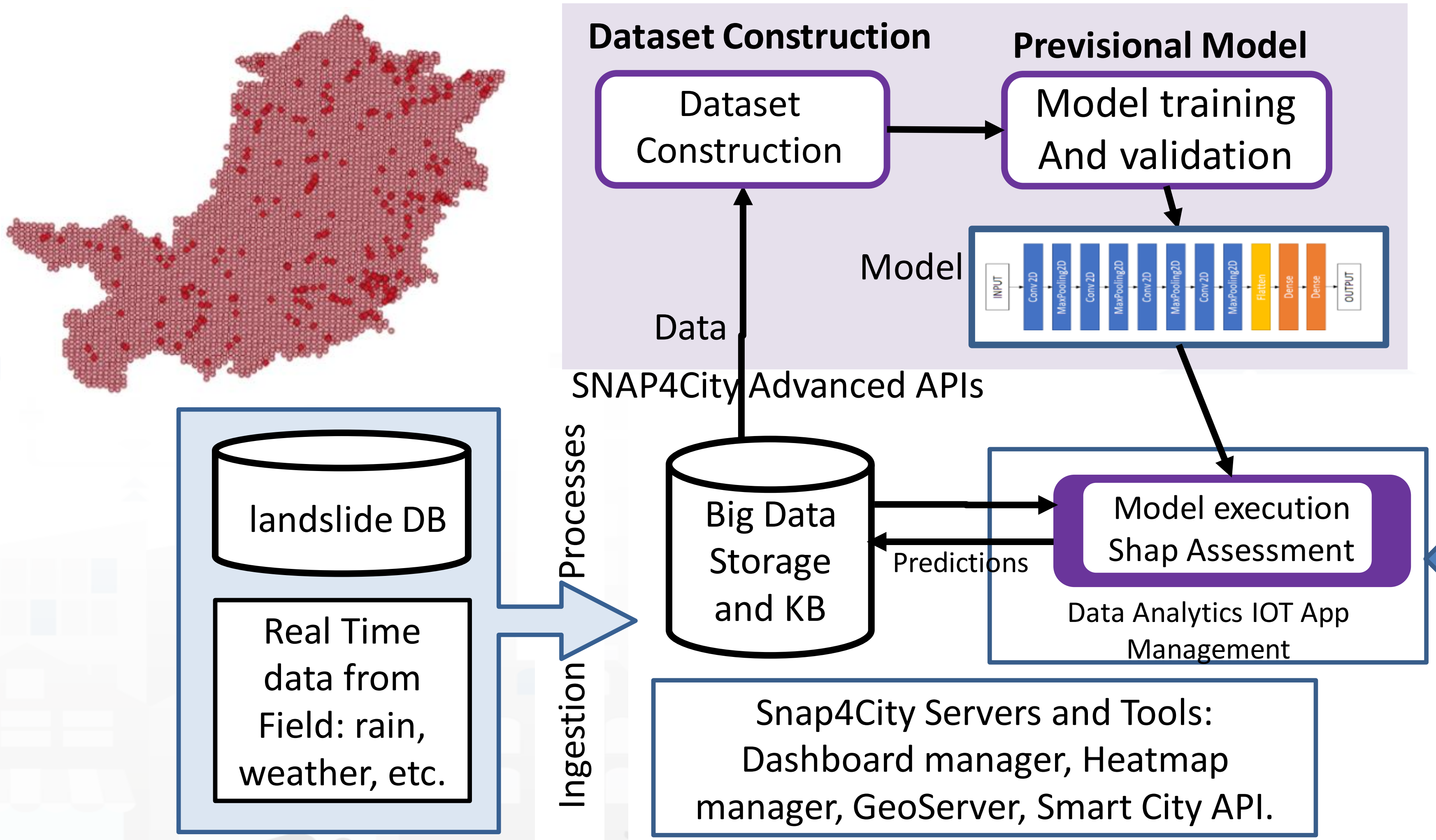
Regione Toscana

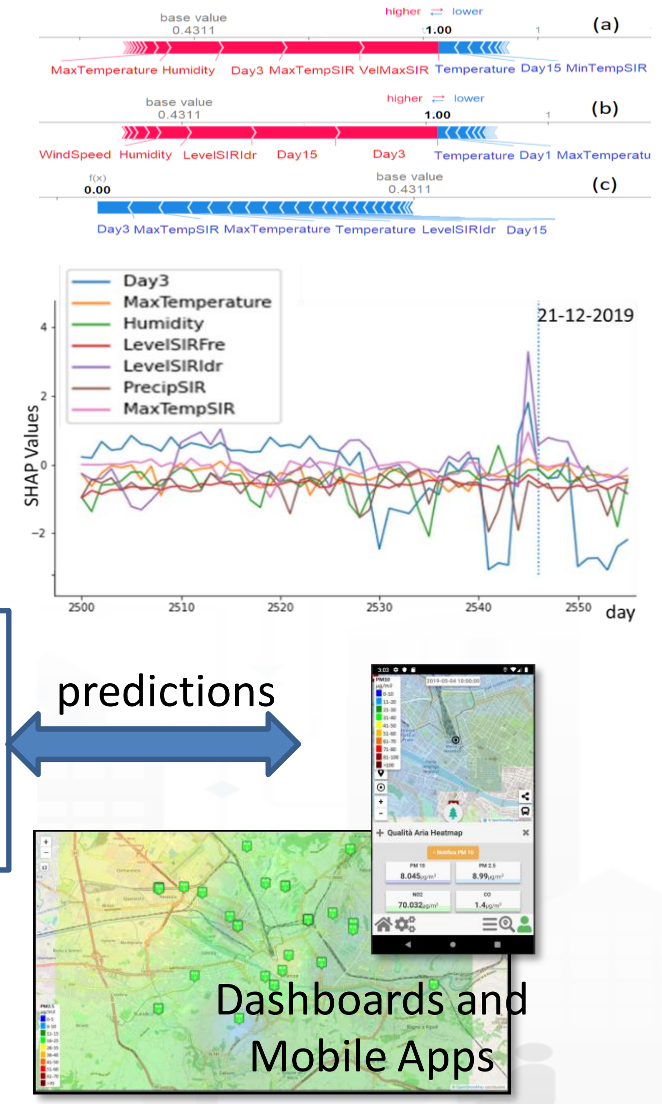
Workflow



Data ingestion
Data set construction
Predictive Model training
Validation AI Model
Model execution
Publication of results on specific Dashboards, Mobile Apps
Explainable/XAI (Shap)
Understanding Landslide Events

Predicting Land slides

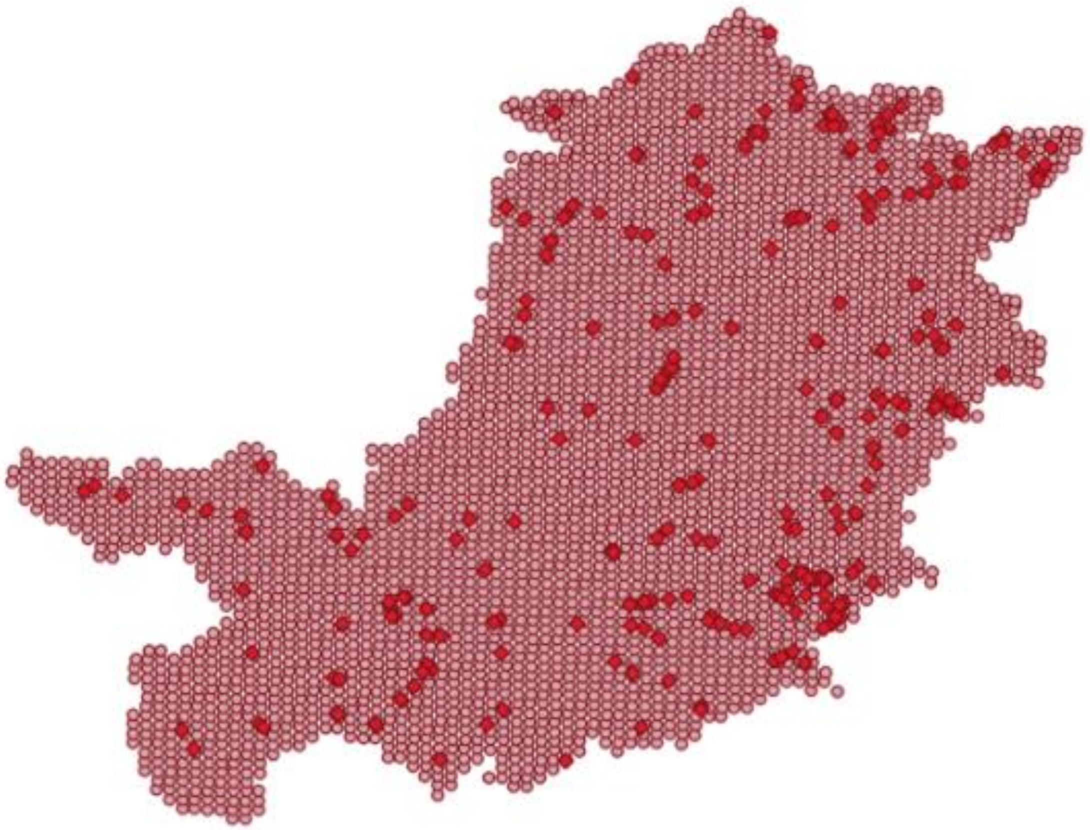




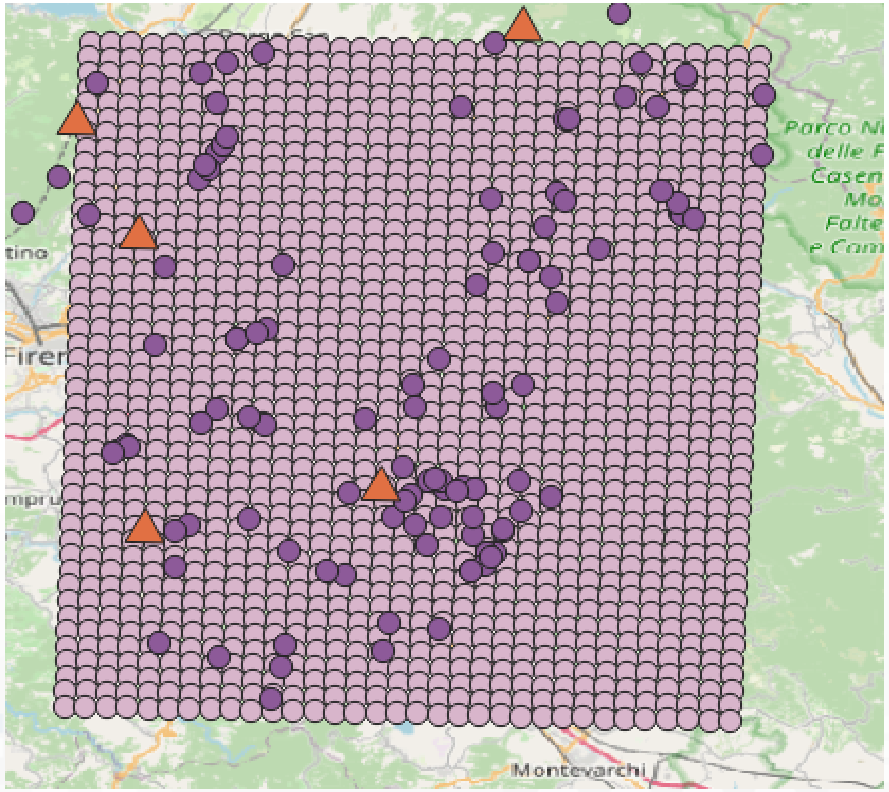
E. Collini, L. A. I. Palesi, P. Nesi, G. Pantaleo, N. Nocentini and A. Rosi, "Predicting and Understanding Landslide Events with Explainable AI," in *IEEE Access*, doi: 10.1109/ACCESS.2022.3158328.
https://ieeexplore.ieee.org/abstract/document/9732490

Grid definition

- 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
- RED dots are the events of landslide registered in 2013-2019



Features as Predictors: static + dynamic data

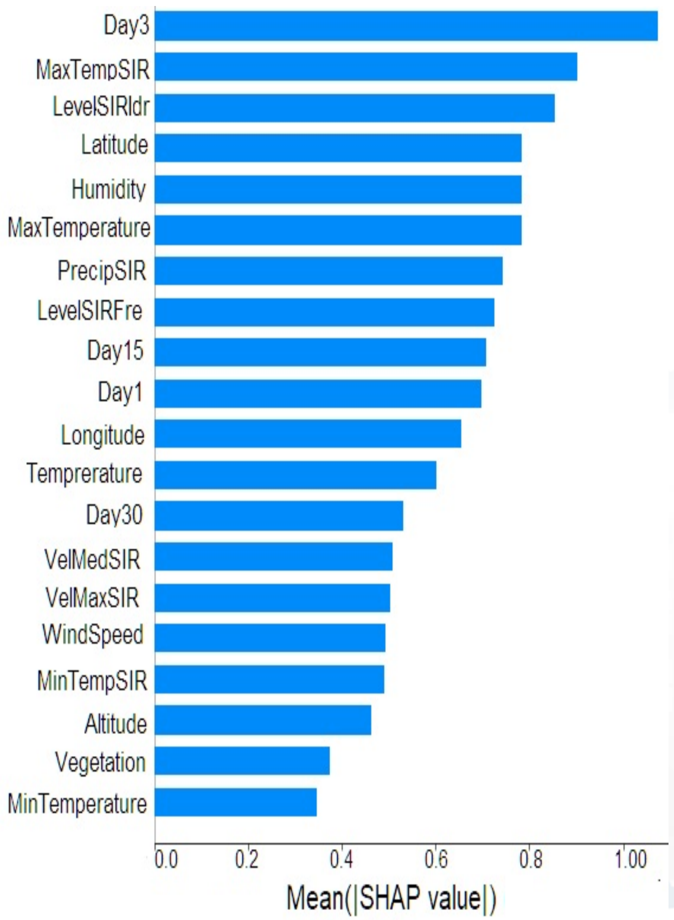


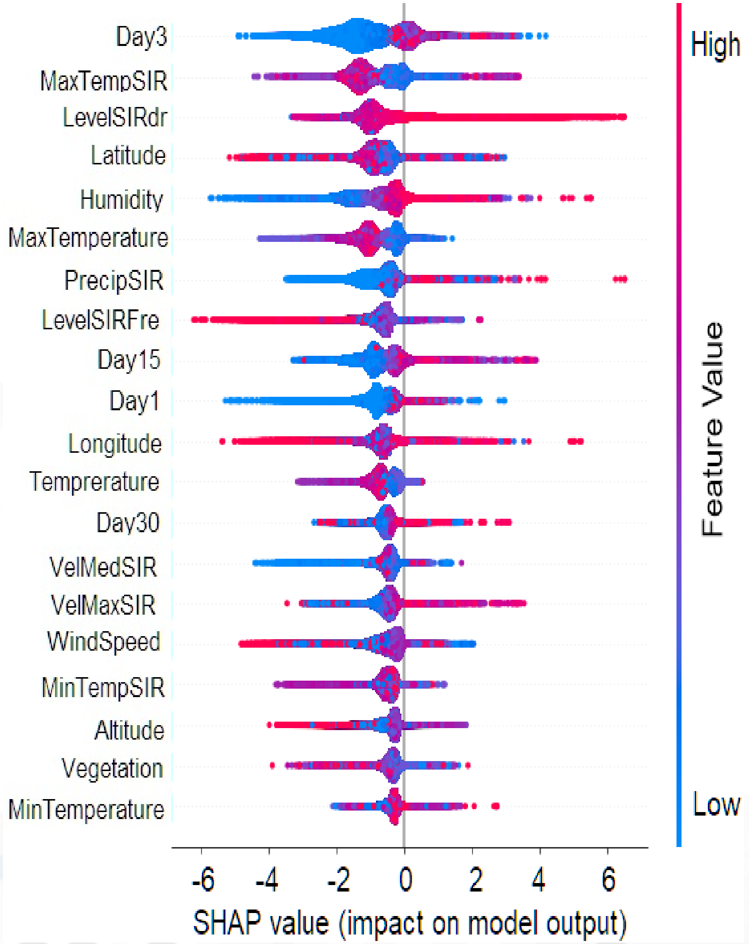
landslide events   rain gauges   grid

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 area 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 on the day before the observation	mm	12.453
Day3	Rainfall on the 3 days preceding the observation	mm	15.072
Day15	Rainfall on the 15 days preceding the observation	mm	16.160
Day30	Rainfall on 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 (river) 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

Comparing Predictive Model/architectures

Model	XGBoost	RF	CNN	Auto encoder	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
TSS	0.78	0.35	0.23	0.18	0.05
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

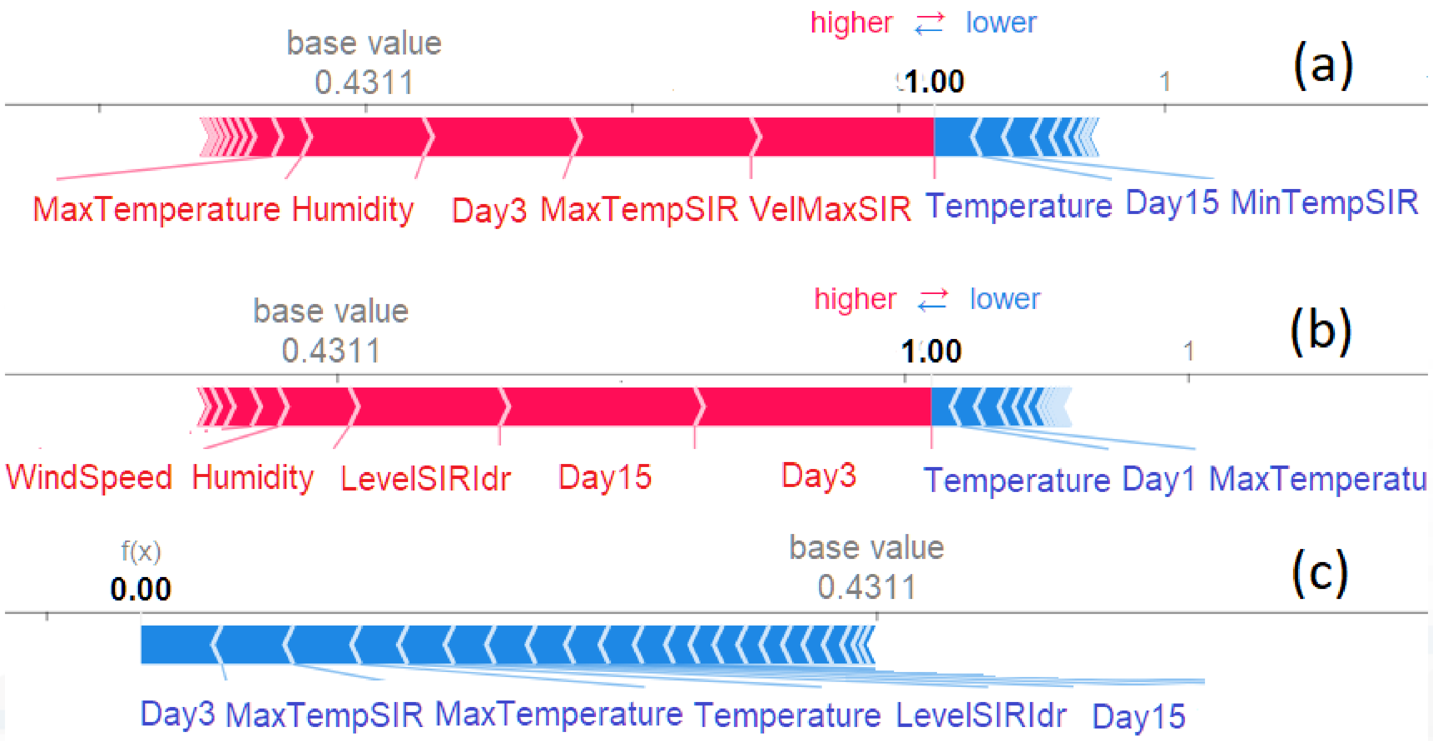


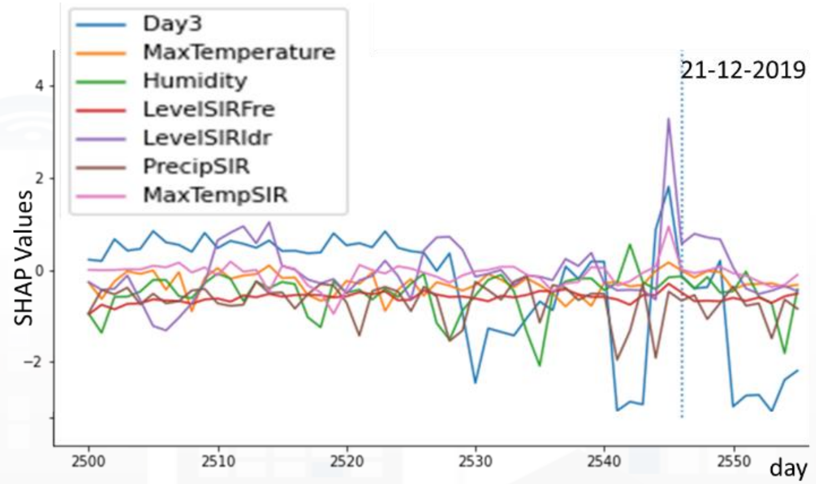


Global Explainable AI
- Feature relevance

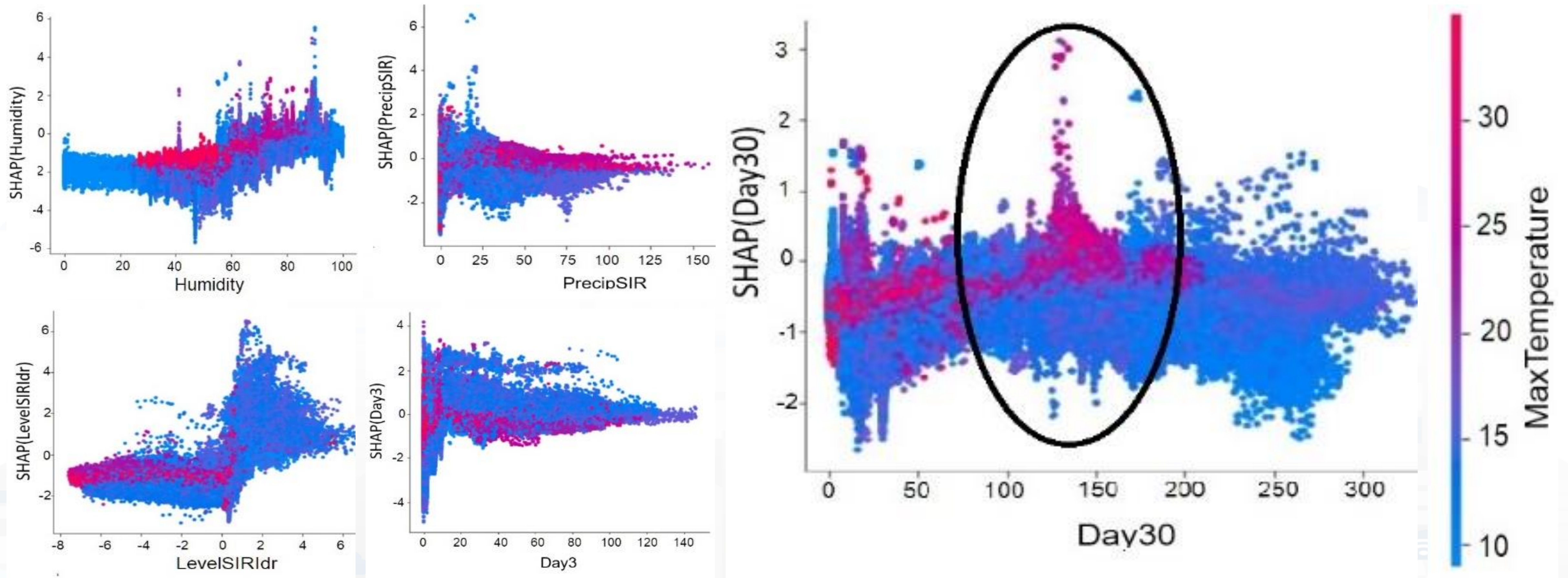
- Red: positive, blue: negeative;
- vs intensity and impact

Local Explainable AI: understanding the single events





Impact of Features on corresp. SHAP Values vs MaxTemp



Considerations

- Comparative results showed that the method based on XGBoost achieved better results in terms of Sensitivity
- A deeper understanding of the predictive model outputs, as well as the relevance of features and their interdependency, has been provided.

IEEE Access

- E. Collini, L. A. I. Palesi, P. Nesi, G. Pantaleo, N. Nocentini and A. Rosi, "Predicting and Understanding Landslide Events with Explainable AI," in *IEEE Access*, doi: 10.1109/ACCESS.2022.3158328.
- https://ieeexplore.ieee.org/abstract/document/9732490

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3158328, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Predicting and Understanding Landslide Events with Explainable AI

E.Collini[1], L. A. Ipsaro Palesi[1], P. Nesi[1], G. Pantaleo[1], N. Nocentini[2], A. Rosi[2]

1) DISIT lab, Dept. Information Engineering, , https://www.disit.org, https://www.snap4city.org,
2) DST, Dept. of Earth Science, University of Florence, https://www.dst.unifi.it/
University of Florence, *ref email:* paolo.nesi@unifi.it

ABSTRACT Rainfall induced landslide is one of the main geological hazard in Italy and in the world. Each year it causes fatalities, casualties and economic and social losses on large populated areas. Accurate short-term predictions of landslides can be extremely important and useful, in order to both provide local authorities with efficient prediction/early warning and increase the resilience to manage emergencies. There is an extensive literature addressing the problem of computing landslide susceptibility maps (which is a classification problem exploiting a large range of static features) and only few on actual short terms predictions (spatial and temporal). The short-term prediction models are still empirical and obtain unsatisfactory results, also in the identification of the predictors. The new aspects addressed in this paper are: (i) a short-term prediction model (1 day in advance) of landslide based on machine learning, (ii) real time features as good predictors. The introduction of explainable artificial intelligence techniques allowed to understand global and local feature relevance. In order to find the best prediction model, a number of machine learning solutions have been implemented and assessed. The models obtained overcome those of 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 best results, demonstrating that it is the most reliable and robust against false alarms. Finally, we applied explainable artificial intelligence techniques locally and globally to derive a deep understand of the predictive model's outputs and features' relevance, and relationships. The analysis allowed us to identify the best feature for short term predictions and their impact in the local cases and global prediction model. Solutions have been implemented on Snap4City.org infrastructure.

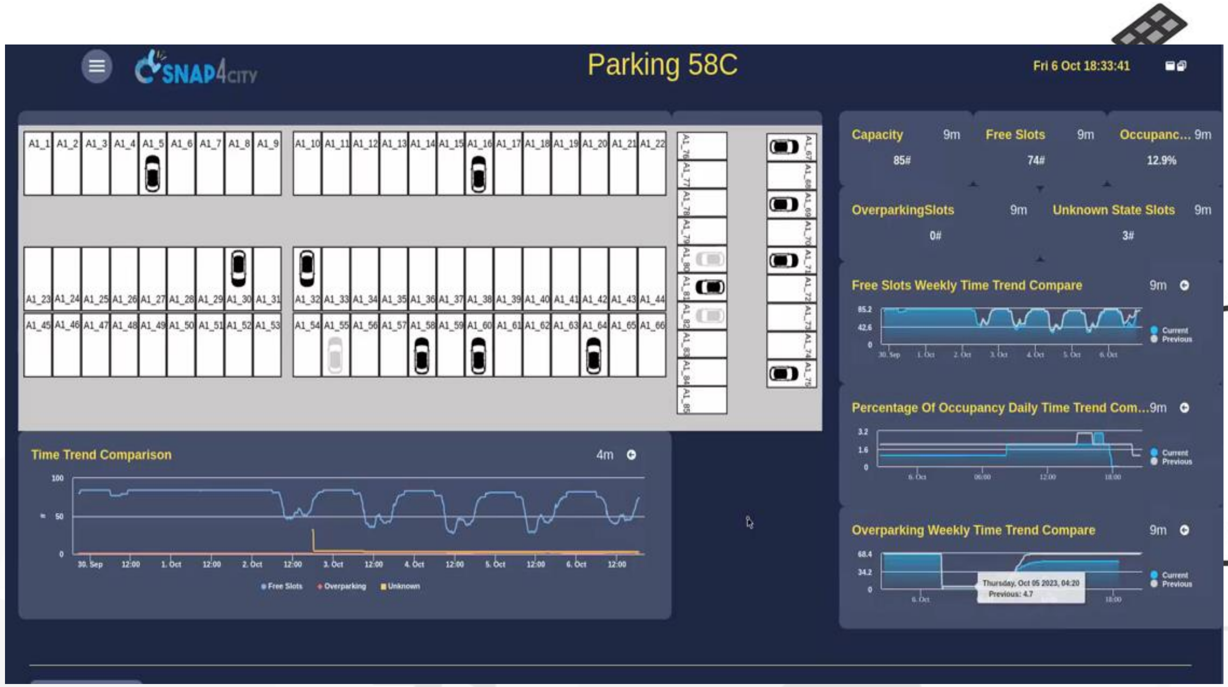
INDEX TERMS landslide prediction, machine-learning, explainable artificial intelligence, snap4city

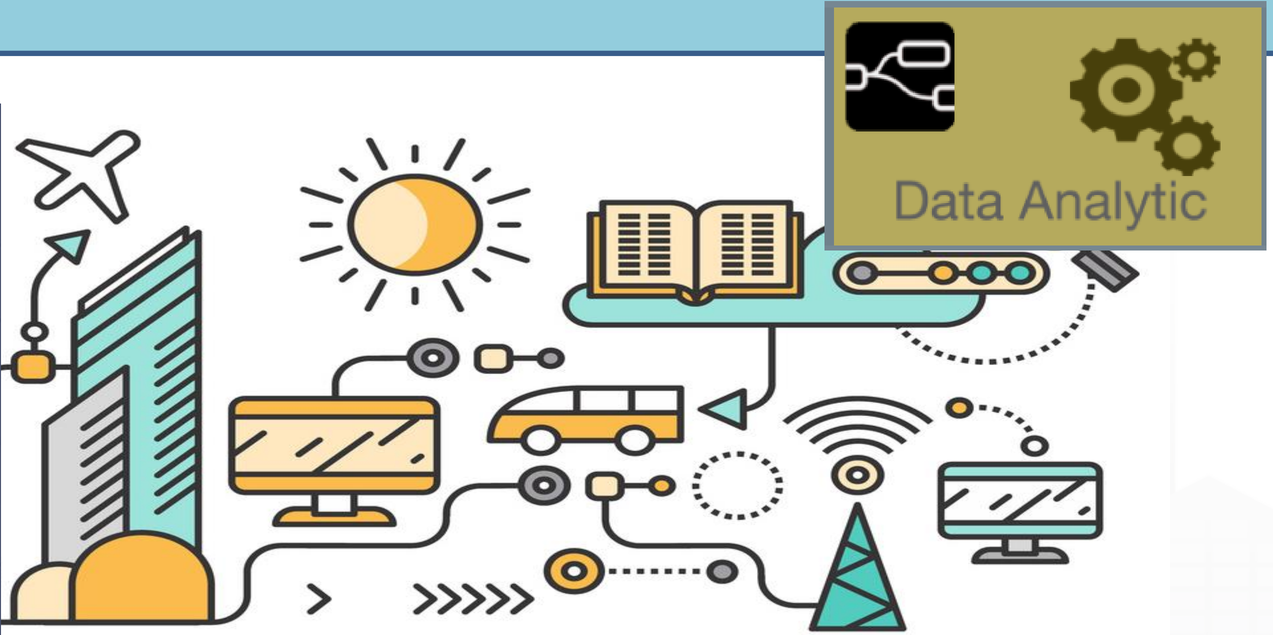
I. INTRODUCTION

Landslides are increasingly frequent geologic events which may involve rural areas, as well as cities and impact on largely populated areas. These phenomena are responsible each year of several losses and casualties; according to [1], from 2004 to 2016, 55997 people were killed in 4862 non seismic landslide events worldwide, with a major incidence in Central America, Caribbean islands, South America, along the Andes mountain chain, Asia, East Africa, Turkey and the Alps in Europe. The same authors identified rainfall as the main the triggering factor of 79% of non-seismic landslides.
Italy is the European country most affected by landslides, with about 2/3 of know landslide in Europe [2]; in fact, over 620'000 known landslides, covering almost 24'000 km^2 (7.9% of the whole national territory), are present, according to the Italian landslide inventory [3]. From 1971 to 2020, 1079 fatalities have been caused by landslides in Italy, along with 1416 casualties and over 146'000 evacuated and homeless [4]. Tuscany is an Italian region highly affected by landslides, since about 91700 landslides are present [5], covering 2107 km^2 (9% of the territory). The province of Florence, due to its geological setting, mainly made of clay-sandy deposits and its morphology, made of alternating valley and hills, is quite susceptible to landslide. These phenomena pose a real risk for the population and one of the possible solutions for its reduction is the setting up of early warning systems. Typically, "wake-up call" and early warning systems are setup to inform the population about the occurrence of landslides in quasi real time. Short term predictions, ranging from a few hours to one/two days, could save a relevant number of people. Thus, the short-term prediction of landslide events 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.

The most common approaches rely on statistical or empirical approaches mixing static information describing the terrain with real time data computed on the basis of recent days. In particular, as to rainfall induced landslides, in [6] and [7] authors highlighted the correlation of the amount of rainfall in the days preceding the landslide event (from 3 to 245 days), by means of statistical analysis [6], [7], while other scholars used the empirical method of rainfall thresholds to identify rain conditions associated with such landslide triggering [8], [9]. Machine learning approaches are widely used in landslide hazard mapping [10] which can be regarded

Predicting free parking slots
Explaining temporal impact of features





Integrated Gradients

Integrated Gradients is an explainability technique that attributes the importance of each input feature to a model's prediction. Measure the contribution of each input by integrating the gradient of the predictive function along a path between a baseline input and the actual input.

Applications: Effective for deep learning models, image classification, natural language processing, and more.

Integrated Gradients

$$IG_i = (x_i - x_i') \int_{\alpha=0}^{1} \frac{\partial F\left(x' + \alpha(x - x')\right)}{\partial x_i}\, d\alpha$$

- $F(x)$ predictive function
- x actual input
- x' baseline input
- α interpolation parameter between the baseline and the input

- **Baseline**:
  - Represents a “neutral” input, e.g., a black image or an empty sentence.
- **Path Integral**:
  - Computes the integral of the model’s gradients along a linear path from the baseline to the actual input.
- **Desirable Properties**:
  - **Sensitivity**: Features with no impact on the prediction receive zero attribution.
  - **Linearity**: Attributions are additive for linear models.

Integrated Gradients

Advantages and Limitations

Advantages:

Works with complex models.

Guarantees sensitivity and completeness properties.

Requires only access to the model's gradients.

Limitations:

Choice of baseline can affect results.

Computationally intensive for high-dimensional data.

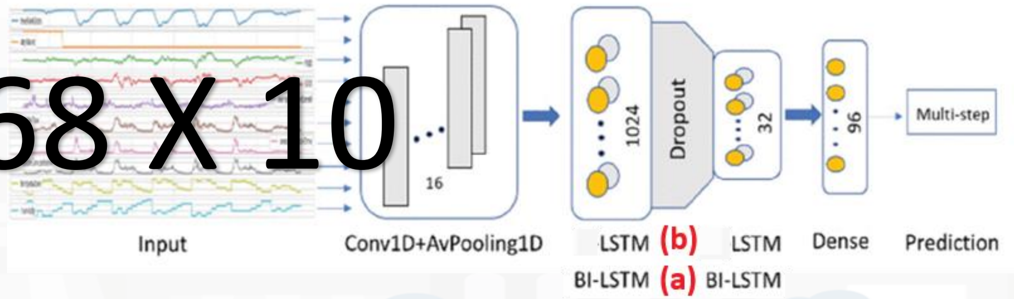
Interpretability

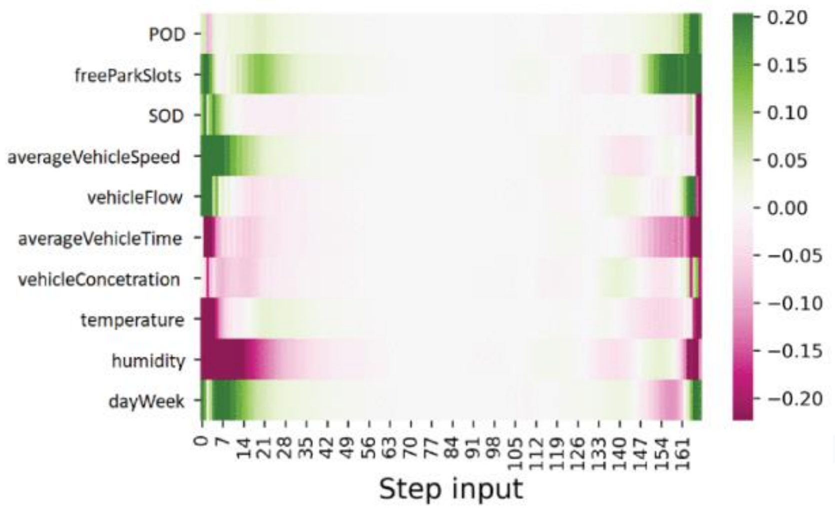
Layer (type)	Output Shape	Param #
conv1d (Conv1D)		
average_pooling1d (AveragePooling1D		
bidirectional (Bidirectional)		
dropout (Dropout)		
bidirectional (Bidirectional)		
dense (Dense)	(None, 96)	6240

Total params: 9,067,344
Trainable params: 9,067,344
Non-trainable params: 0

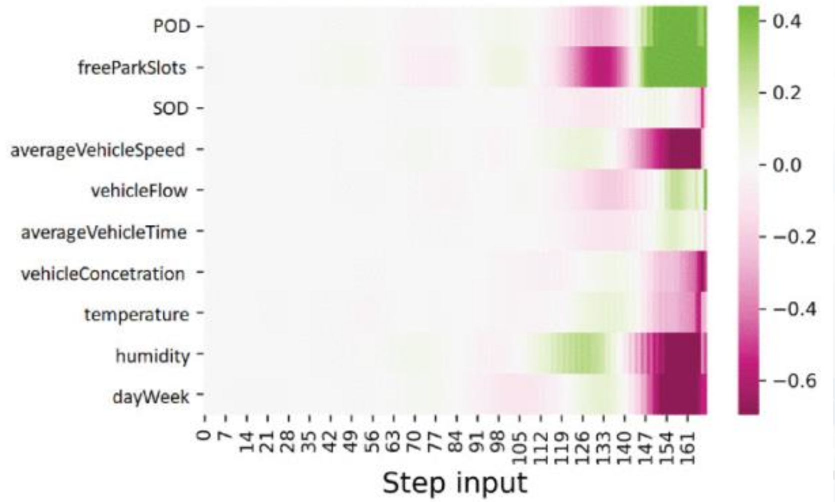
Error Comparison (1H)	Forecasting Techniques				
	BRANN	RNN	CNN-GRU	CNN-LSTM	CNN-BI-LSTM
	Careggi car park				
MASEf (1H)	10.6	2.4	1.8	2.2	1.1
MAEf (1H)	78.6	32.5	24.8	29.5	14.8
RMSEf (1H)	109.5	43.7	31.5	36.5	17.7
	Beccaria car park				
MASEf (1H)	8.2	3.3	2.7	3.3	2.0
MAEf (1H)	30.2	16.4	13.6	16.7	10.3
RMSEf (1H)	38.3	21.0	17.1	25.0	12.3
	S. Lorenzo car park				
MASEf (1H)	6.9	3.4	2.8	2.2	2.0
MAEf (1H)	22.3	16.6	13.7	10.8	9.6
RMSEf (1H)	29.1	19.3	16.5	15.1	11.7

168 X 10



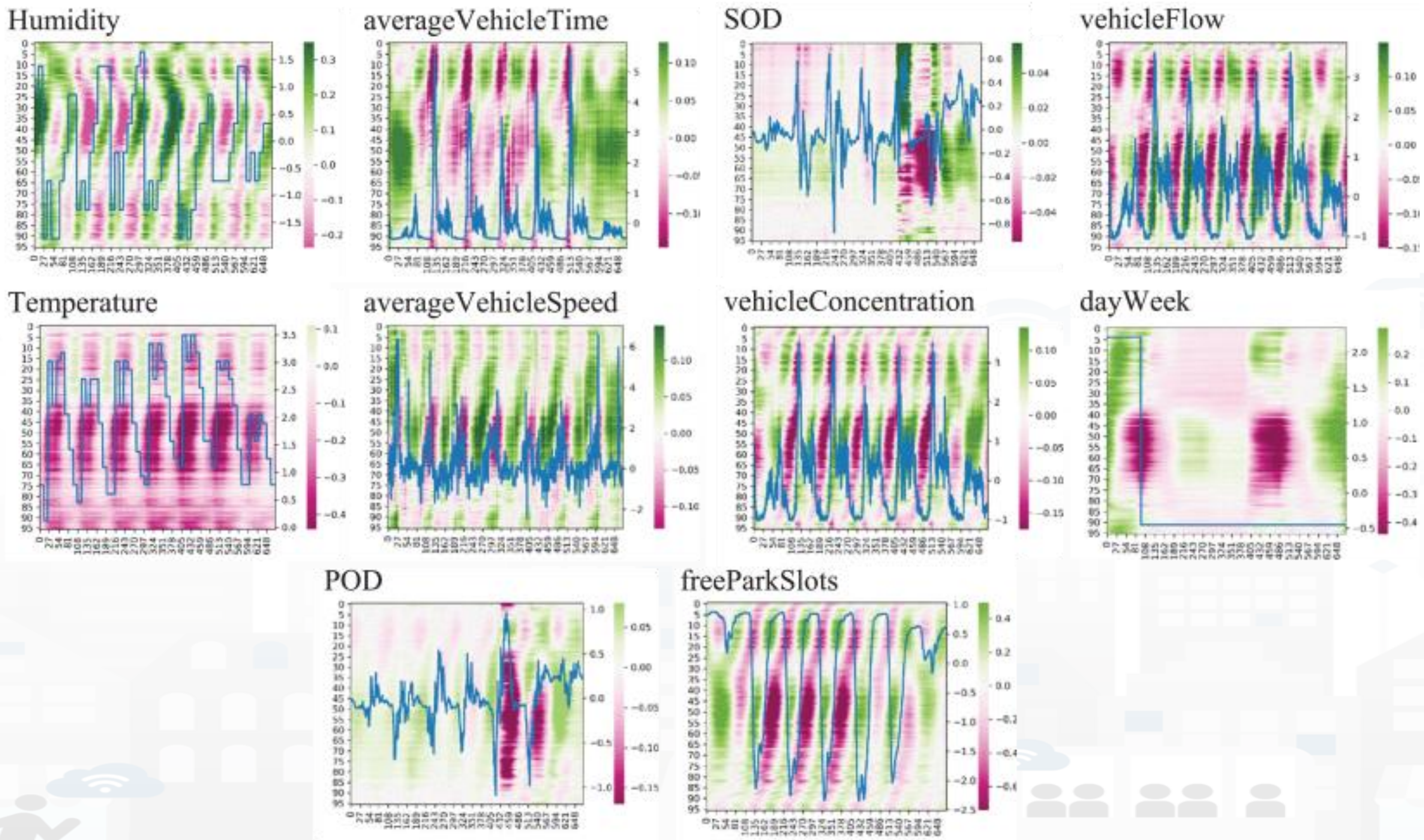


(a)



(b)

Interpretability



IEEE Access

- S. Bilotta, L. A. Ipsaro Palesi and P. Nesi, "Predicting Free Parking Slots via Deep Learning in Short-Mid Terms Explaining Temporal Impact of Features," in IEEE Access, vol. 11, pp. 101678-101693, 2023, doi: 10.1109/ACCESS.2023.3314660
- https://ieeexplore.ieee.org/document/10247516

IEEE Access

Received 2 August 2023, accepted 7 September 2023, date of publication 12 September 2023, date of current version 21 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3314660

RESEARCH ARTICLE

Predicting Free Parking Slots via Deep Learning in Short-Mid Terms Explaining Temporal Impact of Features

STEFANO BILOTTA, LUCIANO ALESSANDRO IPSARO PALESI, AND PAOLO NESI, (Member, IEEE)
Distributed Systems and Internet Technologies Laboratory, Department of Information Engineering, University of Florence, 50121 Florence, Italy

Corresponding author: Paolo Nesi (paolo.nesi@unifi.it)

This work was supported in part by Ministero Industria Università e Ricerca (MIUR), University of Florence; and in part by the National Center on Sustainable Mobility, Centro Nazionale MObilità SosTenibile (MOST).

ABSTRACT Looking for available parking slots has become a serious issue in urban mobility, since it influences traffic and emissions. This paper presents a set of metrics and techniques to predict the number of available parking slots in off-street parking facilities. This study deals with deep learning model solutions according with a mid-term prediction of 24 hours, every 15 minutes. Such a mid-term prediction can be useful for citizens who need to plan a car transfer well in advance and to reduce as much as possible any computational effort. Since most solutions in literature are focused on 1-hour ahead prediction, the proposed solution has been also tested in these conditions. The proposed solution is based on Convolutional Bidirectional LSTM models. Results have been compared in terms of precision metrics based both on occupancy and free slots. The paper also provides a framework to pass from an assessment model based on occupancy to models based on free slots and vice-versa. The obtained results have improved those already available in literature. A formal study has been conducted to perform feature relevance analysis by using explainable AI technique based on gradient and integrated gradient and proposing new heatmaps which highlighted the difference from LSTM and Bidirectional LSTM, feature relevance (base line, weather, traffic, etc.) and the impact of seasonality on predictions, namely the temporal relevance of features. The comparison has been performed on the basis of data collected in garages in the area of Florence, Tuscany, Italy by using Snap4city platform and infrastructure.

INDEX TERMS Smart city, available parking lots, prediction model, machine learning, deep learning, explainable AI.

I. INTRODUCTION

Traffic management and sustainable mobility are central topics for intelligent transportation systems (ITS) so as to monitor and reduce vehicular traffic congestion [1], [2] and emissions [3], [4], [5]. Services providing available parking slots (in real time or as predictions) are becoming relevant for urban mobility management due to the increment of vehicles which need to park in cities. Drivers do waste a considerable amount of time while trying to find a vacant parking lot, especially during peak hours and in specific urban areas (e.g., hospitals, stations, parks, sport stadium). Searching for available parking spots can be a time-consuming task that simultaneously increases traffic congestion, thus leading to a peak of 25-40% of the traffic flow [6], [7] and greenhouse gas pollution.

Parking slots can be located on the street (they are called *on-street parking*) or in parking garages with gates (named as *off-street parking*). Searching for an available parking space has a harmful impact on both transportation system efficiency within the urban tissue and sustainability. Actually, any car parking searching activity generates unnecessary traffic workload and may affect the environment negatively due to increased vehicle emissions. These issues are surely valid

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

LIME

LIME is a model-agnostic interpretability method that explains individual predictions by approximating a complex model locally with a simple interpretable model. Focus on a small area around a specific prediction to create explanations using simpler models (e.g., linear models or decision trees).

Select Instance: Choose the data point for which you want an explanation.

Perturb Data: Create slightly modified versions of the input.

Predict: Use the model to make predictions on these perturbations.

Weight Samples: Assign weights to perturbations based on their proximity to the original input.

Fit Local Model: Train a simple interpretable model (e.g., linear regression) on the weighted data to approximate the prediction locally.

Generate Explanations: Use the simple model to identify feature contributions.

LIME

Advantages and Limitations

Advantages:

Flexible: Supports various interpretable models (e.g., linear, trees).

Easy to Implement: Straightforward methodology and tools available.

Human-Readable: Produces intuitive explanations for non-experts.

Limitations:

Stability Issues: Explanations can vary depending on the perturbations generated.

Computational Cost: Requires retraining the local model for each prediction.

Interpretability of Perturbations: Artificial perturbations may not always make sense in the context of the data.

Scalability: May struggle with high-dimensional data or very large datasets.

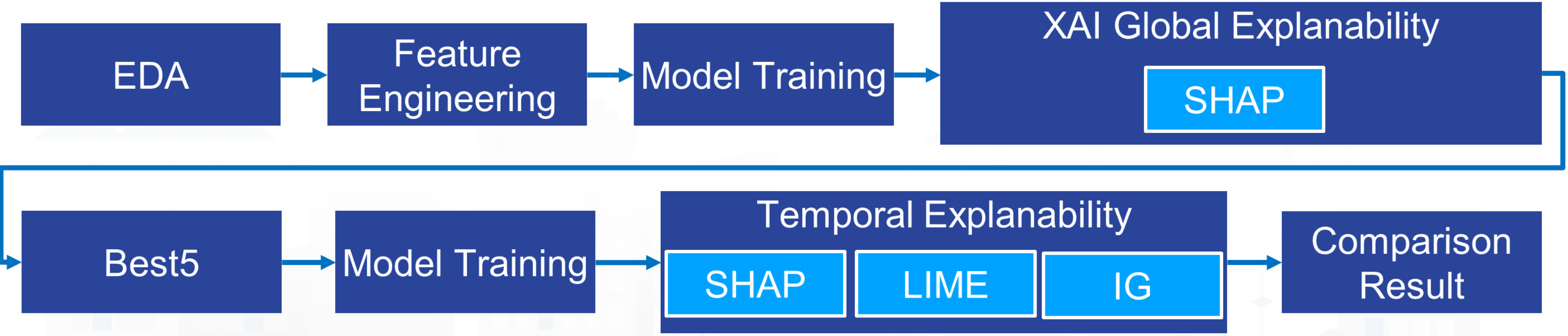
XAI methodologies

Evaluate **XAI methodologies** on the time series analysis in the AI models, focus is on **global XAI aspects**

Time series
- With seasonality
- Without short-term seasonality
- With outliers

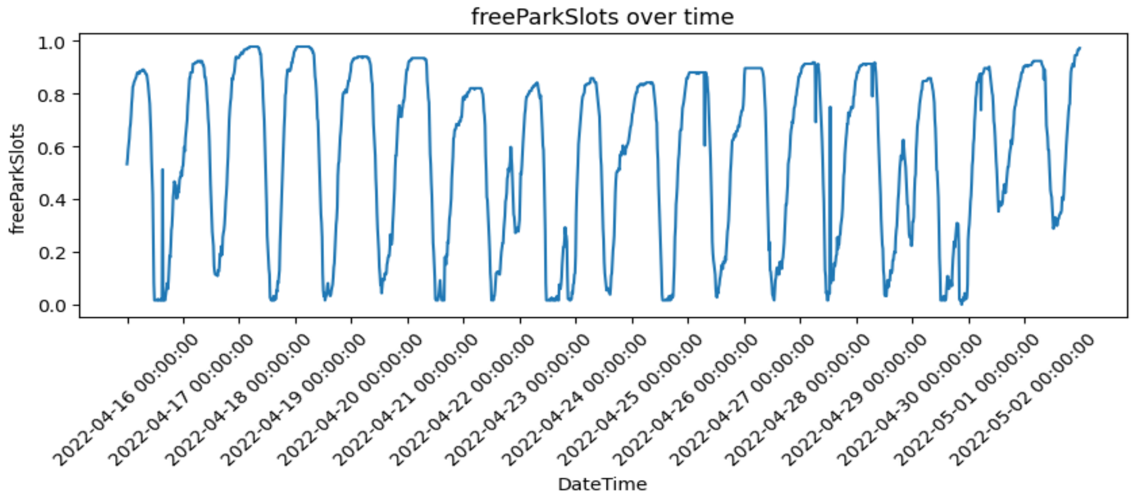
eXplainability methodologies
- Shapley Values
- Integrated Gradients
- LIME

Workflow


EDA
Feature Engineering
Model Training
XAI Global Explanability
SHAP
Best5
Model Training
Temporal Explanability
SHAP
LIME
IG
Comparison Result

Time-series with seasonality

Cardinality	Sampling rate (minutes)	Target variable
1738	15	freeParkSlots

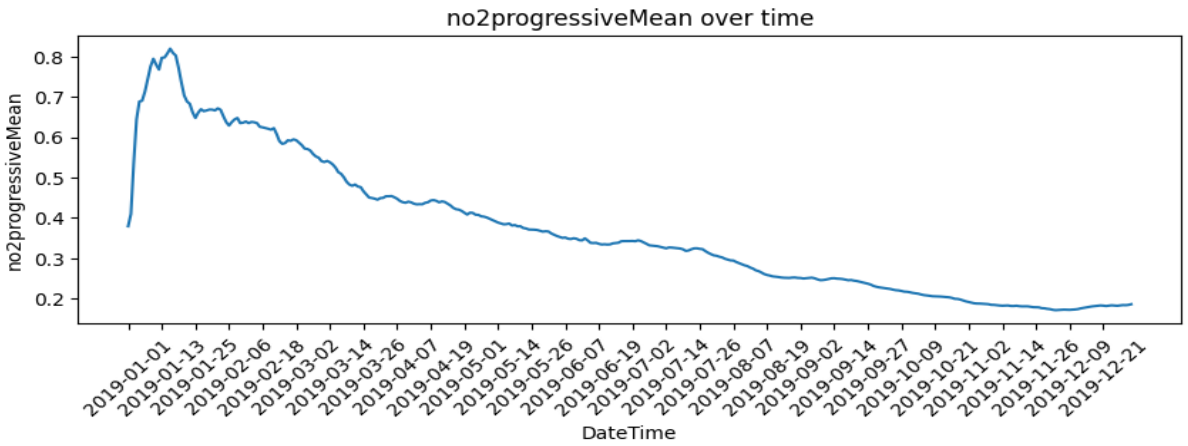


$$f(freeParkSlots_{t-1}, SOD_{t-1}, POD_{t-1}, temperature_{t-1}, humidity_{t-1}) \rightarrow freeParkSlots_t$$

Bilotta, S., Palesi, L. A. I., & Nesi, P. (2023). Predicting free parking slots via deep learning in short-mid terms explaining temporal impact of features. ***IEEE Access***.

Time-series without short-term seasonality

Cardinality	Sampling rate (minutes)	Target variable
2144	1440	no2ProgressiveMean

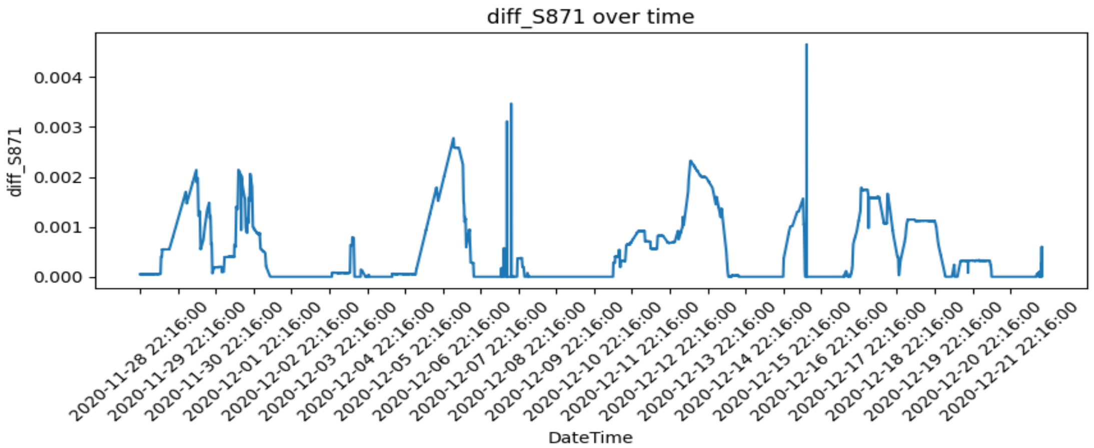


$$f\,(no2progressiveMean_{t-1}, NOxDomestic_{t-1}, TMean_{t-1}, numberOfVehiclesCumulated_{t-1}, Month_{t-1}) \rightarrow no2progressiveMean_t$$

Bellini, P., Bilotta, S., Cenni, D., Collini, E., Nesi, P., Pantaleo, G., & Paolucci, M. (2021). Long Term Predictions of NO 2 Average Values via Deep Learning. In **Computational Science and Its Applications–ICCSA 2021**: 21st International Conference, Cagliari, Italy, September 13–16, 2021, Proceedings, Part VIII 21 (pp. 595-610). Springer International Publishing.

Time-series with outliers

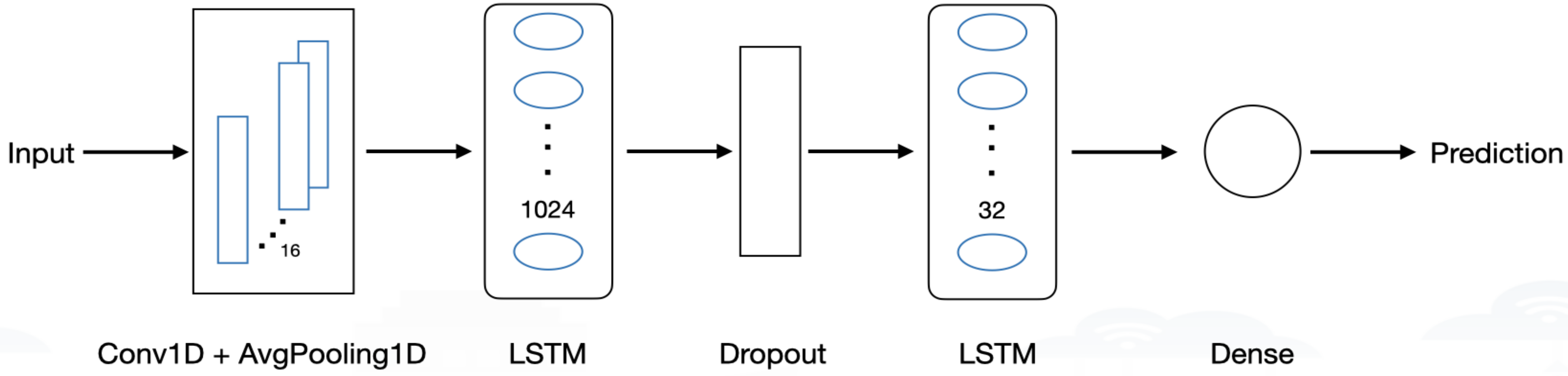
Cardinality	Sampling rate (minutes)	Target variable
313183	1	diff_S871



$$f(diff\_S871_{t-1}, S484_{t-1}, S904C_{t-1}, S4304_{t-1}, KOHrampa2caricoprodotti_{t-1}) \rightarrow diff\_S871_t$$

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," **2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)**, Oxford, United Kingdom, 2021, pp. 9-16, doi: 10.1109/BigDataService52369.2021.00007.

Model Architecture

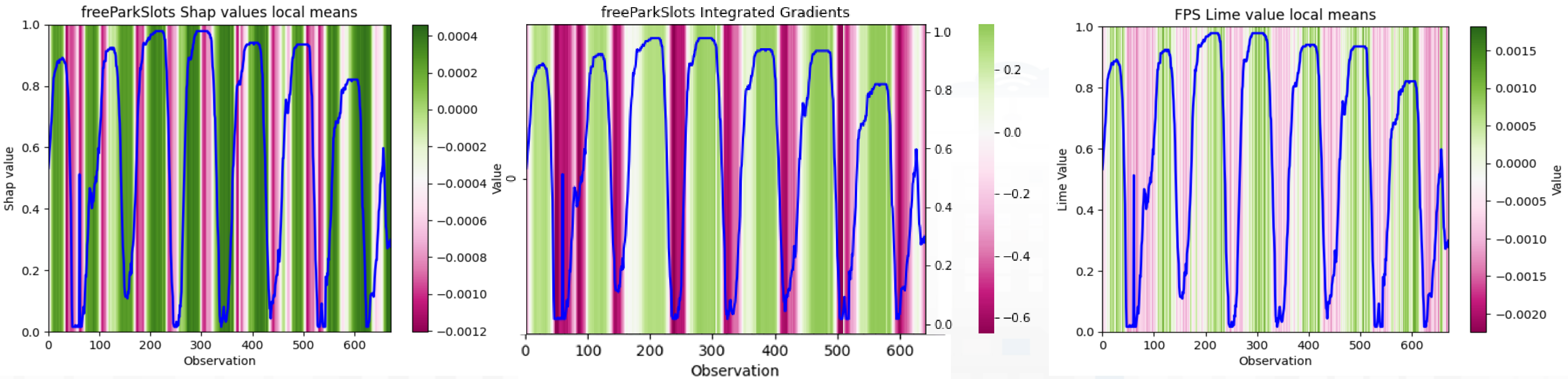


Convolutional network for locality patterns detection
Recurrent network for temporal patterns detection

Time-series	MAE	MSE	R2
With Seasonality	2.0100000	15.50000	0.995
With no Seasonality in the short-term	0.7200000	0.98900	0.968
With outliers	0.0000093	0.00164	0.939

Experimental result
Time series with seasonality

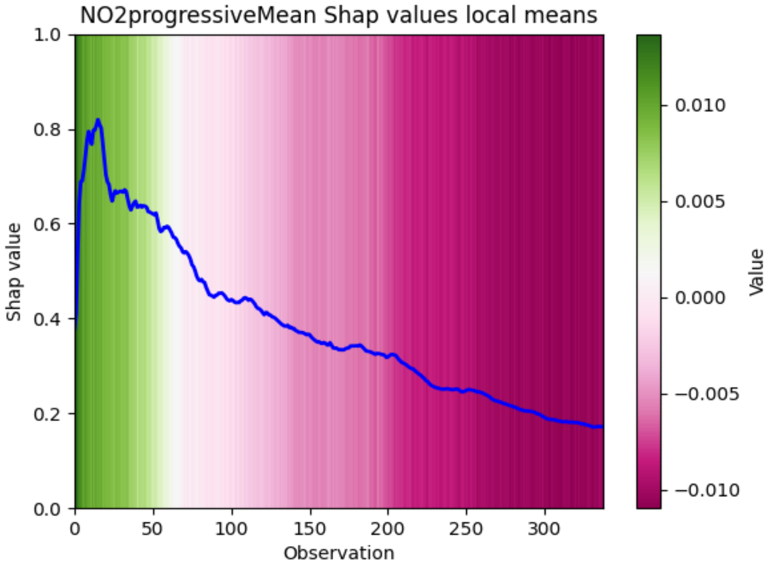
SHAP IG LIME



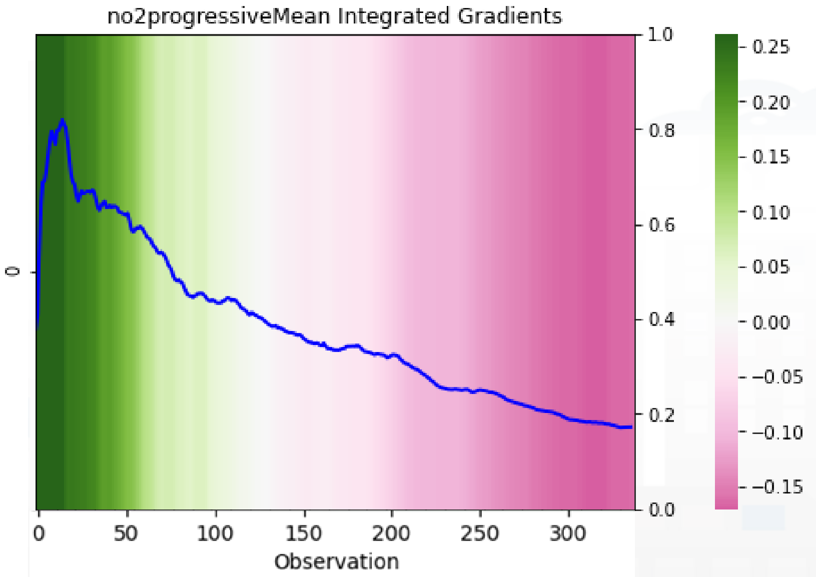
- The methodologies show consistent behavior in capturing the **seasonality** of the target variable of this time-series
- The scales of the results are different while the substance is very similar

Experimental result
Time series without short-term seasonality

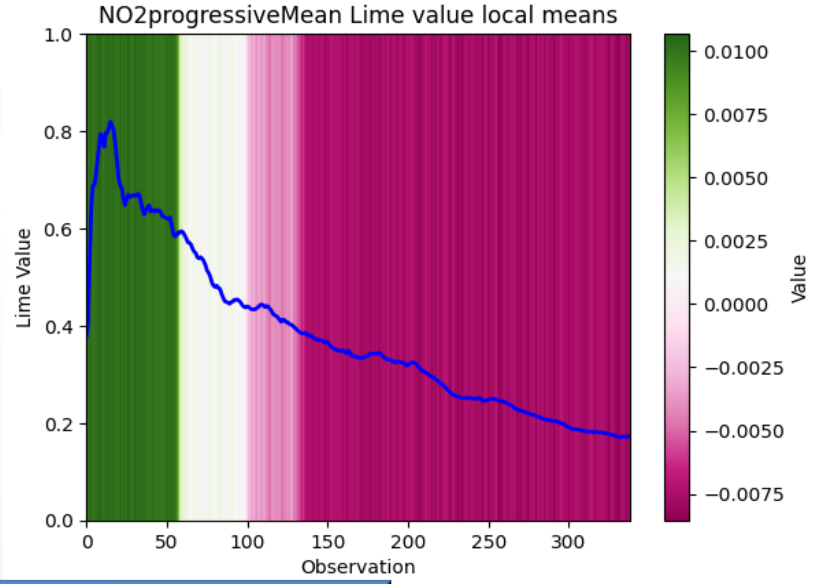
SHAP



IG



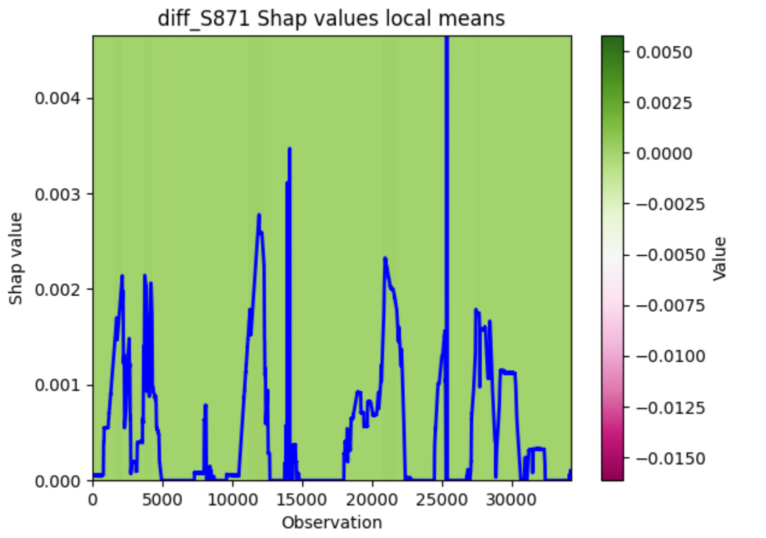
LIME



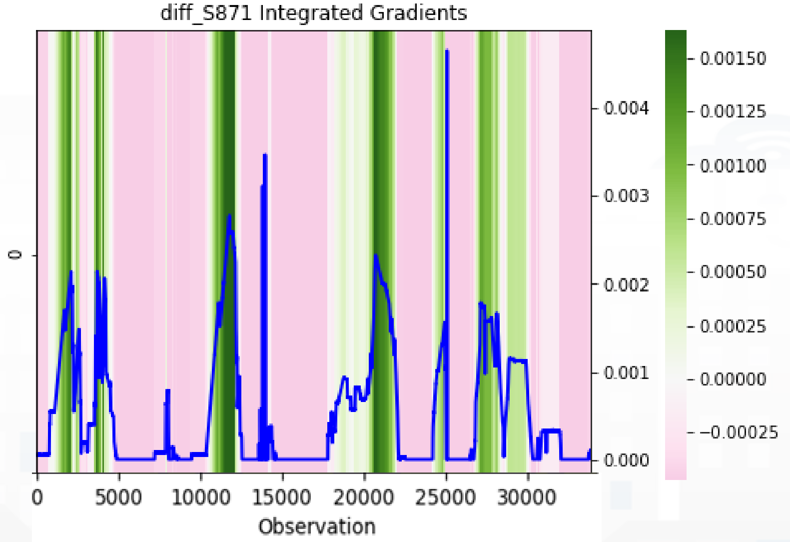
- The methodologies show consistent behavior
- LIME shows very **drastic passage** from green to white and red dividing areas between the signs of impact
- LIME is actual less stable than SHAP and IG seems to be the smoother

Experimental result
Time series with outliers

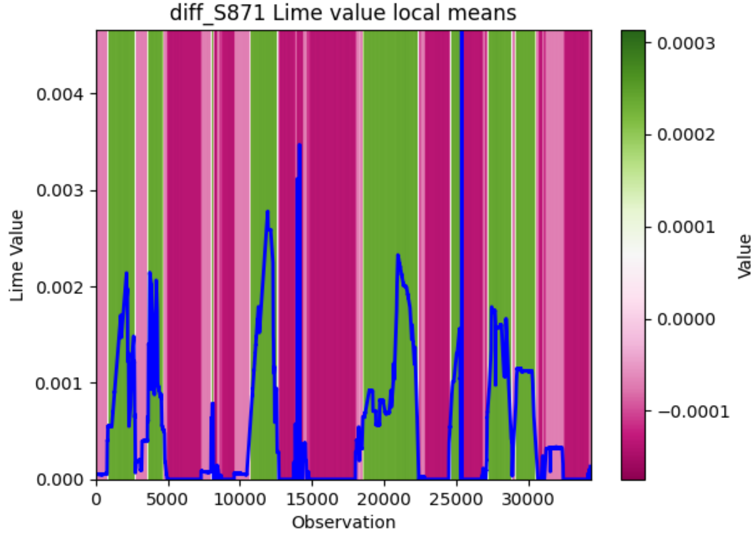
SHAP



IG



LIME

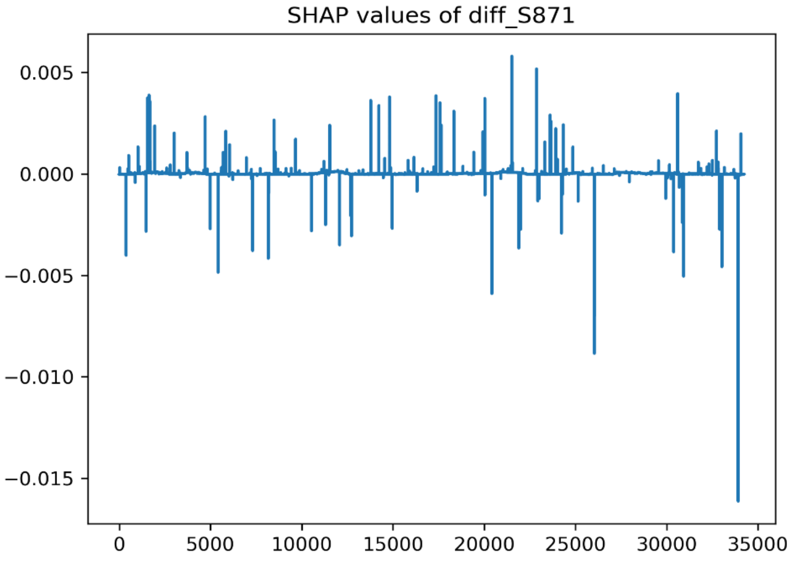


- Difficulty for SHAP to compute feature impact on time-series with outliers (effect of the last negative peck)
- LIME and IG seems to provide more coherent results between each other, highlighting alternating influences on the model output, at a different scale and losing a part of the smoothness

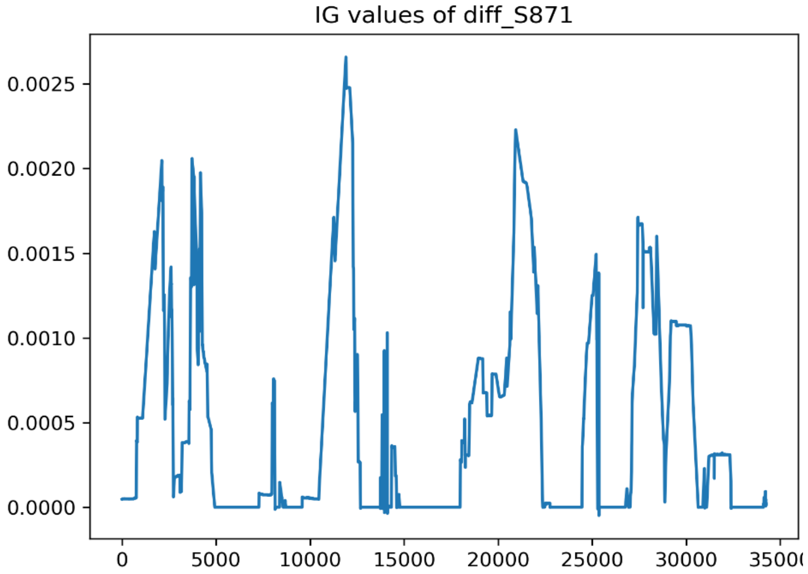
Experimental result

Plot Shap, IG and LIME values for the feature diff_S871 which is one of the variable determining the anomalies/outliers.

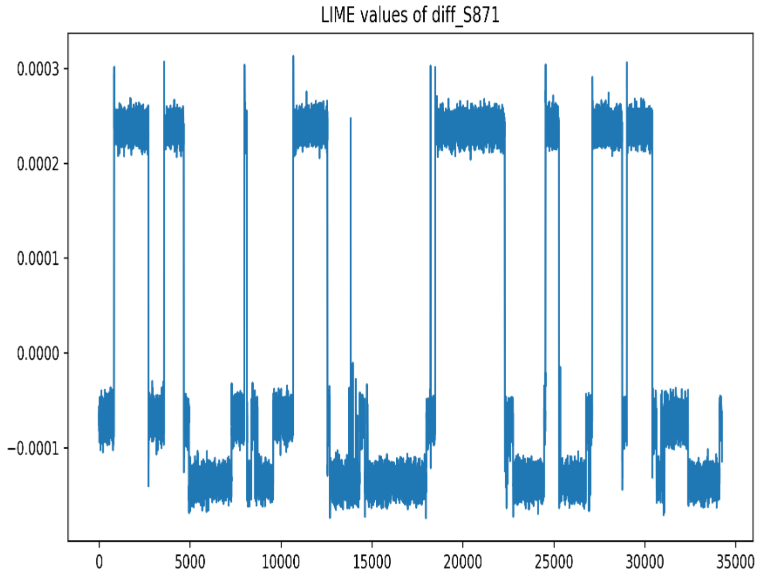
SHAP



IG



LIME



- LIME actually shows again instability and saturation of its behavior with respect to the trend of the variable.
- IG exhibits better behavior in the presence of anomalous values, while SHAP and LIME are more sensitive.

Execution Time

Time-series	XAI methods	Execution time in s
With Seasonality	SHAP	4204.58
	IG	32.95
	LIME	1598.82
With no Seasonality in the short-term	SHAP	60.87
	IG	0.61
	LIME	45.54
With outliers	SHAP	1162.38
	IG	0.81
	LIME	2763.81

Dell Precision 5820 Tower equipped with 12cores from Intel(R) Xeon(R) W-2235 CPU @ 3.80GHz, 32GB of RAM and 1.5TB of space

Conclusions

Time series

- With seasonality
- Without short-term seasonality
- With outliers

With seasonality, Without short-term seasonality: coherent behavior from the XAI methodologies (SHAP, IG, and LIME) compared

Without short-term seasonality, With outliers: SHAP was not able to provide insights on the target variable compared to LIME and IG. Between these two IG was able to provide more coherent explanations with respect to LIME

EXECUTIONTIME
IG resulted faster than SHAP and LIME

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

SNAP4CITY

IEEE Access

- E. Canti, E. Collini, L. A. I. Palesi and P. Nesi, "Comparing Techniques for Temporal Explainable Artificial Intelligence," *2024 IEEE 10th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService)*, Shanghai, China, 2024, pp. 87-91, doi: 10.1109/BigDataService62917.2024.00019

- https://ieeexplore.ieee.org/abstract/document/10730341

2024 IEEE 10th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService)

Comparing techniques for TEmporal eXplainable Artificial Intelligence

Edoardo Canti, University Of Florence, DISIT lab, Italy, https://orcid.org/0009-0009-8939-4627
Enrico Collini, University Of Florence, DISIT lab, Italy, https://orcid.org/0000-0002-1304-5545
Luciano Alessandro *Ipsaro* Palesi, University Of Florence, DISIT lab, Italy, https://orcid.org/0000-0001-8992-2084
Paolo Nesi, University Of Florence, DISIT lab Florence, Italy, Paolo.nesi@unifi.it, https://www.disit.org

***Abstract*—Artificial Intelligence models have been employed in various fields, leading to a growing interest in the subject and in the development of the models. The direct involvement of complex AI models in decision-making processes stressed the needs to explain the rationales behind the results, globally and locally for each prediction/result via eXplainable Artificial Intelligence (XAI) techniques. This paper compared three XAI techniques (SHAP, LIME and IG) with aim of using them for temporal explainability of predictive results regarding time-series in order to understand if these methods are able provide temporal explanation of deep learning AI models. The comparison provided has been qualitative and quantitative and addressing computational performance. This work has been partially supported by the CN MOST, national center on sustainable mobility in Italy, on CAI4DSA of FAIR, and has been developed on the Snap4City platform.**

***Keywords*—time-series, XAI, LIME, SHAP, Integrated Gradients**

I. INTRODUCTION

In recent years, technological advancements have facilitated the collection and exploitation of Internet of Things data over time. They are time-series adopted in various scenarios, from Smart Cities to Industry. Decision Support Systems (DSS) exploit this information, grounding the decision on early warning and predictions. Time-series can describe a multitude of behaviors across various domains, including urban mobility (e.g., traffic flow), energy production (solar panels), environment (pollution levels), as well as tourism (people flow), each of which with different kinds of trends. Regarding time-series, the increasing data availability and computing power enabled the development of AI that outperformed traditional statistical methods in various task such as anomaly detection, early warning, forecasting. However, the success of AI in resolving a variety of problems has been met by increasing model complexity and employing black-box models that lack transparency. The process of explaining the rationales behind the models has been focus of multiple research works. Ridley in [1] reported the importance of providing explanations for experts of AI and for the final users towards human-centered explainable AI (HCXAI). More recently, multiple explainable AI methodologies have been proposed, and the overall scenario regarding the explanations process in the AI Life Cycle is quite large. Furthermore, when explaining time-series data, the complexity of patterns such as seasonality, outliers, and noise pose challenges to explainable artificial intelligence (XAI) algorithms and techniques [2].

A. Related work

Decision making systems can leverage data-driven insights derived from time-series analysis to anticipate trends, identify patterns and forecast. Regarding Time-series analysis, AI based solutions demonstrated to be capable of generating optimal results in various fields. These models are complex and often operate as 'black boxes'. Providing explanations of the results would improve the AI trust and problem understanding of Decision makers.

LIME [3] XAI methodology generates a set of neighbors around a specific instance and evaluates the black-box model's predictions on these neighbors. LIME suffers from stability problems: when the same explanation process is carried out multiple times the explanations might be different. Choi et al., [4], assessed a XAI methodology exploiting gradient calculations to discern the influence of input features on model predictions on 3 time-series datasets. The AI models used for the time-series analysis were Long Short-term Memory (LSTM) and Convolutional Neural Network (CNN) based. The authors compared Grad-CAM, Integrated Gradients (IG) and Gradients. The gradient based solution extracts attributions as a local explanation, highlighting the importance of input variables of a sample, based on the predicted output of a model [5]. In other approaches additive attributions get calculated to give each feature value an additive score as in sHapley Additive exPlanations (SHAP) [6]. The results in [7] focused on finding a solution for a digital consulting firm that aimed to find a data-driven approach to understanding the effect of its sales activities on sales contracts. A time-series dataset was trained using Support Vector Regression (SVR), as XAI both LIME and SHAP were applied to capture the temporal. In [7], the explanations produced using LIME and SHAP have been assessed with a survey conducted with 60 people. The results of the human evaluation studies clearly showed that the explanations produced by LIME and SHAP greatly helped humans to understand the predictions. Another important finding was that the majority of the participants preferred LIME explanation over SHAP.

B. Aim of the paper and organization

In this paper, it has been performed a study on the XAI methodologies (LIME, IG, SHAP) across three distinct types of time-series data: those with a normal trend, presence of seasonality, and occurrence of picks to assess how these explanations vary across different characteristics. Through experimentation and analysis, we examine how these explanations of *AI deep learning predictive* models vary according to the temporal patterns in the data, thus helping to provide more informative insights into the application of XAI methodologies in time series analysis, let say *TEmporal eXplainability of AI*, ***TEXAI***. The use of consistent XAI techniques is critical to ensure that the explanations provided are reliable and understandable, enabling greater transparency and acceptability of machine learning models. This work has been partially supported by the CN MOST national center on sustainable mobility in Italy and by

979-8-3503-6638-9/24/$31.00 ©2024 IEEE
DOI 10.1109/BigDataService62917.2024.00019