FuzzyIoT - Platform for descriptive analysis of sensor data streams

Francisco J. Martínez Mimbrera
University of Jaén
fjmimbre@ujaen.es

Alicia Montoro Lendínez
University of Jaén
amlendin@ujaen.es

José L. López Ruiz
University of Jaén
llopez@ujaen.es

Miguel Damas Hermoso
University of Granada
mdamas@ugr.es

Macarena Espinilla Estévez
University of Jaén
mestevez@ujaen.es

Abstract

Data generated by sensors are often inaccurate due to calibrations, battery failure or network transmission errors among other issues. Fuzzy protoforms have been demostrated to be a successful tool for modelling data with uncertainty by providing both linguistic descriptions of data which offer more relevant information and hidden information. In the context of fuzzy protoforms, data are modeled by fuzzy sets whose degree of truth is defined by membership functions. In the new Internet of Things (IoT) paradigm, there is an explosion of data streams generated by smart devices with sensors which contribute to the Sustainable Development Goals (SDGs). However, currently, there is no software capable of modeling the information from the devices in a flexible, simple and understandable way. In this contribution, a software tool called FuzzyIoT is presented for the descriptive analysis of sensor data streams through fuzzy protoforms. This software generate linguistic descriptions through the definition of ad hoc protoforms of data exported by different file of sensor data through the previous definition of a set of protoforms. A case study focusing on body temperature is illustrated to show the usefulness of the tool, highlighting the main alignment of the SDGs with this case study.

Keywords: Protoforms, linguistic description, fuzzy sets, sensor data stream, Internet of Things, SDGs

1. Introduction

Nowadays, the information generated by the different objects connected to the internet is often inaccurate due to poor calibration, lack of battery or network transmission errors, among other issues López-Medina et al., 2020. All this leads to the usual sensor errors that usually appear in the datasets, such as missing values, out-of-range values, etc. Ferri-Ramírez et al., 2004. In addition to the above, the information provided by sensors often presents problems when it comes to handling and interpreting it correctly. For this reason, there are many types of uncertainty problems, for example, what for some may be a high value, for others may be a low value, as the boundary between one and the other is not clear. This is why new ways of handling and modelling this information are needed.

A review of the current market for data processing tools shows that there are a multiple tools for descriptive and predictive data processing, both from free software and from large companies. These tools are used to make descriptions and predictions of datasets and help decision making. Examples of these include: Pentaho [1], PowerBI [2],Microstrategy [3], Knowage [4], etc. These tools have modules for extract, transform and load (ETL) processes and classic descriptive and predictive data processing, but they do not have methods capable of accurately handling the heterogeneous data produced in problems where uncertainty is involved and will therefore provide worse data when it comes to providing a result for decision making.

Based on the contribution proposed in the literature Albin-Rodriguez et al., 2021; Albın-Rodrıguez et al., 2020; Espinilla et al., 2018; Lopez-Medina et al., 2020; López-Medina et al., 2019; Medina et al., 2017; Peláez-Aguilera et al., 2019 on the use of fuzzy logic techniques to model uncertainty in problems with heterogeneous information, it is evident that the use of fuzzy protoforms to the descriptive analysis

[1]https://www.hitachivantara.com/en-us/home.html
[2]https://www.powerbi.microsoft.com/es-es/
[3]https://www.microstrategy.com/
[4]https://www.knowage-suite.com/site/

URI: https://hdl.handle.net/10125/102825
978-0-9981331-6-4

of data streams from sensors Cornejo and Medina, 2022; Fontenla-Seco et al., 2021; Ramos-Soto and Martin-Rodilla, 2021, the uncertainty can be reduced and also mitigate the effect of errors in data collection produced by sensors to overcome the above problems.

Regarding software proposals for descriptive analysis of sensor data, in the context of energy consumption of buildings over time, in Martinez-Municio et al., 2018 proposed a framework to define models that capture this casuistry, gathering a set of time series of electrical consumption by means sensor in the IoT approach and linguitic protoforms. FLINTSTONES was proposed in Estrella et al., 2014 as a software suite for dealing with multicriteria decision making problems within linguistic contexts with uncertainity. Recently, in Polo-Rodriguez et al., 2021 fuzzy processing of audio event streams was included in the IoT boards by means of temporal restrictions defined by protoforms to describe activities of daily living.

After reviewing the literature, it was observed that there was no descriptive data analysis tool for handling sensor data stream that used linguistic protoforms to model uncertain information and perform descriptive analyses.

Therefore, in this contribution a software tool entitled Fuzzy IoT is proposed to the descriptive analysis of heterogeneous data from sensors and through the application of fuzzy protoforms. The advantage of this software is to model the uncertainty and failures of the collected data, producing better results and much more interpretability of the data to achieve an improvement in intelligent environments. In addition, a use case of a real problem with sensor data is presented, applying the proposed tool to show its functionality

In addition, this tool allows the creation of models that define fuzzy protoforms applicable to multiple contexts in IoT systems. Currently, these systems are designed with the purpose of improving our society and contributing in a direct way to the Sustainable Development Goals (SDGs)[5] Espinosa et al., 2021.

In order to carry out this paper, in Section 1 the motivation of this contribution is introduced. In Section 2 the preliminaries associated are presented with the linguistic protoforms, which will be the theoretical base to develop this proposal. In Section 3 the analysis, design and development of the software tool are exposed together with a case study presented in Section 4 where alignment with the SDGs is also discussed. Finally, Section 5 presents the conclusions of this contribution and future work.

[5]https://unstats.un.org/sdgs/indicators/indicators-list

2. Fuzzy background

The purpose of a fuzzy protoform is to integrate and improve the expressiveness of the linguistic model proposed by the expert or data sources to model the uncertainty.

A fuzzy protoform in the context of Internet of Things can be defined as Peláez-Aguilera et al., 2019:

$$P_0(\bar{s}_i) : (Q_k, L_i, T_j), \quad (1)$$

where:

1. $\bar{s}_i$ represents the pair of values sent by a sensor, such as $\bar{s}_i = \{s_i, t_i\}$, where s_i is the value given by the sensor and t_i is the timestamp in which this value has been produced.

2. L_i defines the fuzzy linguistic term to evaluate the data stream. With L_i we refer to the degree of belonging to a fuzzy term defined in a linguistic variable (low, adequate, high etc).

3. T_j defines the fuzzy temporal window where the term L_i is aggregated. For the aggregation of L_i over T_j for a given value s_i is given by the function $L_i T_j(s_i)$.

4. Q_k defines a fuzzy quantifier to evaluate the impact on the linguistic term L_i together with the fuzzy time window T_j. The fuzzy quantifier applies a transformation $\mu_{Q_k} : [0,1]- > [0,1]$ to aggregate the temporal degree of $\mu_{Q_k}(V_r \cup T_k(\bar{s}_i))$

Fuzzy protoforms can be combined to provide a better interpretation of the results using fuzzy logical aggregation operators as Weber, 1983:

- Fuzzy Negation Operator, which is represented as the complement in the following function: $\neg P_0(S_D) = 1 - P_0(S_D)$

- Fuzzy Union Operator, which is represented by the T-Norm $P_0 \cup P_l(SD) = P_0(SD) \cup P_l(SD)$. The semantics proposed by the join operator is the minimum, $P_0 \cup P_l(SD) = min(P_0(SD)), P_l(SD))$.

- Fuzzy Intersection Operator, which is represented by the T-CoNorm $P_0 \cap P_l(SD) = P_0(SD) \cap P_l(SD)$. The proposed semantics for the join operator is the maximum, $P_0 \cap P_l(SD) = max(P_0(SD)), P_l(SD))$.

3. Software analysis, architecture and programming

There are a multitude of sensor devices that generate data streams in the context of the Internet of Things. These data streams are inaccurate and difficult to interpret. This paper is presented a platform that allows a descriptive analysis of the data through linguistic protoforms. The developed platform allows to model the information in a flexible, clear and understandable way.

To do so, first, the functional and non-functional requirements of the proposed tool are presented, then the proposed architecture for the software tool is proposed and, finally, the software languages and tools used to develop the proposal are indicated.

3.1. Functional and non-functional requirements

The functional requirements for the proposed tool for descriptive analysis of sensor data streams using fuzzy protoforms are the following.

A username and password will be required to access the tool, so that each researcher can work with their private datasets. If the user or password is not correct, it will not be possible to access the system. In addition, users can only be created by the administrator user. And the administrator will give the password to each user the first time and after that, once the user is logged in, the user will be able to change their password.

The system will have a main screen with a menu showing the possible actions to be performed. The actions which users can perform are as follows:

- Defining elements which are part of a framework (Number of sensors and type, linguistic variables, temporary windows and protoforms)
- Load datasets
- Define framework of the problem to be solved and obtain results

In the definition of elements which are part of a framework, it must be possible to define the following work variables of the system:

- Number of sensors. For each sensor, it is necessary the sensor ID, the type of value (Binary/Continuous) and the type of values that it generates. The binary type sensor can have values 0 or 1, while the continuous type sensor will have to establish a maximum and a minimum of values accepted by the sensor.
- Linguistic variable. It is possible to define the linguistic variable to be used to model and treat the information generated by the each sensor. The linguistic variable will have a name and a set of linguistic terms, each linguistic term defined by its membership function. It is established as a restriction that the membership function to be used will be of the trapezoidal type, although by the same method triangular ones can be used due to the fact that these are a particular case of a trapezoidal function.
- Temporary window. This component defines the temporal windows used to evaluate a segment of the sensor data stream. The temporal window will have a name and a membership function expressed in time units (ms., seconds, minutes or hours). Furthermore, it will be possible to work in two different ways: by sampling over the time samples of the dataset or by reading the whole dataset.
- Protoforms. Protoforms can be defined using a name and a threshold. The linguistic variable and the temporal windows that the protoform is supposed to model the sensor data stream. The threshold is a value for which the protoform returns values above that threshold, the rest is zero, which can be interpreted as a fuzzy alpha-slice.
- Protoform combination rules. Rules can be defined to combine different protoforms to obtain complex protoforms. Three types of fuzzy operators that can be used to combine protoforms: Negation, Union and Intersection.

When a dataset is loaded, it is possible to select the dataset produced by the sensors to be worked with it. To do this, files in CSV format can be imported, which correspond with the following format: timestamp, idsensor, value.

It must also be taken into account that the file may contain errors. These errors shall be displayed to the user after the upload and the user shall be given the possibility to continue or to re-upload another file. Some of the upload error messages that should appear are: i) Timestamp is not in the correct format, ii) Sensor ids do not match those previously defined and iii) there are null values.

The dataset of the problem to be solved and the protoform(s) to be used will be selected. On one hand, for the definition of the problem framework to be solved and, on the other hand, for the results to be obtained. Moreover, the results will be displayed graphically and the possibility will be given to export them to CSV format.

Having reviewed the main functional requirements of the tool, the non-functional requirements of the tool are described below.

First, the access to the system must be possible from any of the current browsers (Firefox, Chrome, Edge, Safari, etc). Following with the web design should be responsive (adaptable to any size of device automatically) as far as possible.

In addition, the tool will be multi-language (Spanish and English), it must be developed using free software tools and libraries and it will be accessible from the server of the ASIA research group of the University of Jaén through the URL [6].

Finally, the solution must be deployed using Docker container technology so that it is easily portable between servers and a possible failure does not affect the rest of the server and the database must be backed up daily on an external NAS server dedicated for this purpose.

3.2. Architecture of the software tool.

The Figure 1 shows the general architecture proposed for the development of the solution.

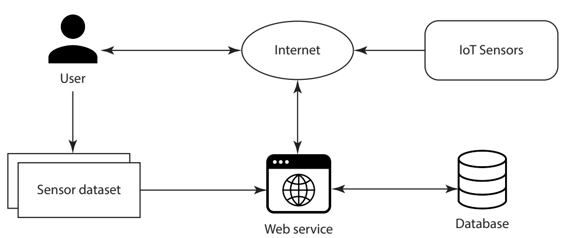

Figure 1. Proposed architecture for the FuzzyIoT platform

As can be seen in Figure 1, the architecture to set up our solution consists of a web server plus a relational database to store data. It is also necessary to have an internet connection to access the solution and import sensor data streams. The user is the one in responsible of interacting with the system through the internet and the web browser. Each component of the architecture is described in detail below.

- Web server. A web server can be described as a computer program which is responsible for processing an application on the server using bidirectional or unidirectional connections to the client machine, after which a response is generated on the client side. Web servers can process files that are written in different programming languages such as PHP, Python, Java and others. These are converted into static HTML files and sent to the client browser of the web users.

- Relational Database Server. A relational database server, also known as a database server or RDBMS (Relational DataBase Management Systems), is a type of server software which allows us to carry out the organisation of information through the use of tables, indexes and records. It is mainly used for data storage.

- IoT Sensor Data. In this case, it is contemplate 2 possible data input streams. Directly from a previously downloaded sensor dataset or from a sensor database in the cloud. The application will be prepared for both cases and according to the requirements it must be prepared for both scenarios.

3.3. Programming languages used

This section shows the languages and tools used to develop the proposed tool.

In order to provide a solution for the production deployment of the solution architecture, a container-based technique (Dockers)[7] has been chosen.

Many other researchers' work coexists on this server, so a solution needed to be found that would allow it to coexist with all the other applications without interfering with them. The tool can be accessed through the ASIA domain http://asia.ujaen.es/herrmientaTFMfjmimbre/.

Following on from the architecture outlined above, the proposed web server, the database and, finally, the programming languages and development IDEs to be used are mentioned below.

As a container server it has been used Dockers and as a web server it has been proposed to use Apache Web Server, as it is the most widely used worldwide to process and serve contents on the web [8].

As a database server, MariaDB has been used, which is a relational database management system that was created as a fork of the MySQL project and uses GPL (General Public License) for its distribution.

For the development of the tool, it has been proposed to use different programming languages, each one for a different part of the solution, which is divided into Front and Back.

For the implementation of the front end of the web it has been proposed to use 5 and CSS 3 and for the

[6] http://asia.ujaen.es/herramientaTFMfjmimbre/

[7] https://docs.docker.com/.

[8] https://httpd.apache.org/

implementation of the back end of the web system it has been proposed to use PHP 7 and for data processing, fuzzy logic, temporary windows, protoforms etc, to use Python 3 and a set of support libraries to facilitate certain tasks.

In this solution PHP is used to execute operations to collect data, connect to the database, launch Python scripts etc.

Python[9] is an interpreted, multi-paradigm programming language, as it supports object-oriented, imperative and, in a minor way, functional programming. It can be used on multiple platforms without having to change anything in the code.

The libraries that have facilitated the task have been Pandas, Numpy, Scikit-Learn, Scikit-Fuzzy. Specifically, Scikit-Fuzzy [10] is a collection of algorithms and structures for handling fuzzy logic intended for use in the Python language. It is developed by the SciPy community. In this case, it will be used to facilitate the handling of fuzzy information in sets and variables.

Scikit-Learn is a collection of algorithms for data processing and machine learning. It is available for the Python programming language and in this case it will be proposed to use it to process data from IoT sensor data streams.

NumPy [11] is a library for handling and supporting large multidimensional vectors and matrices, along with a large collection of high-level mathematical functions to operate with it, which is written in Python.

Pandas [12] is a data analysis and manipulation library for the Python programming language. It was written as an extension to the NumPy library. In particular, it provides the programmer with data structures and operations to manipulate numerical tables and time series. It is free software distributed under the BSD licence.

Finally, the development IDE used was Visual Studio Code. Visual Studio Code[13] is an editor for writing source code intended for programmers.

The source code generated in the construction of the FuzzyIoT is available in a github repository[14].

4. Illustration of the software through a case study

In this section, a case study is presented to show how the Fuzzy IoT software tool works. To do so, first the case study is described and then the linguistic protoform is visualised through the tool. Finally, due to the importance of achieving the 17 Sustainable Development Goals (SDGs), the main alignment of the SDGs with this case study is presented.

4.1. Case study description.

The problem presented with the tool is to analyse whether an ICU patient has had a very high temperature during the last 5 mornings, in order to see if the medication used first thing in the morning has had an effect or if it is counterproductive.

For this purpose, the data collected by the temperature sensor worn on the patient's finger, which constantly monitors the patient at 15-minute intervals, has been used. For the case study, a 5-day sample set has been used. Therefore, the expression "The temperature has been very high in the morning" using a fuzzy protoform, will be processed by the tool with a dataset from a sensor.

To do this, first define the set of sensors to be worked with, being in the case of the study an element $S = \{S_{D1}\}$. This sensor produces a temperature data steam, which is expressed by pairs of values, where the first one is the value given by the sensor, body temperature, and the second one is a timestamp of when this value was produced, as shown in the Figure 1.

Table 1. CSV file sensor value pairs.

Sensor	Timestamp	Body temperature ($^{\circ}C$)
S1	1631397600	35.5
S1	1631398500	35.6
S1	1631399400	35.5
S1	1631400300	35.7
S1	1631401200	36
S1	1631402100	36.2
S1	1631403000	36.3
S1	1631403900	36
S1	1631404800	35.7
S1	1631405700	35.5
S1	1631406600	36
S1	1631407500	36
S1	1631408400	35.5
S1	1631409300	35.8
S1	1631410200	35.9
S1	1631411000	36.2

The dataset corresponds to 480 intervals of 900

[9] https://www.python.org/
[10] https://pythonhosted.org/scikit-fuzzy/overview.html
[11] https://www.numpy.org/
[12] https://www.pandas.python.org/
[13] https://code.visualstudio.com/
[14] https://github.com/franmartinezmimbrera/TFM

seconds, with corresponding temperature values, which is exactly 5 days of 24 hours of temperature taking.

For this case study, the protoform "Very high body temperature in the morning" is defined. This protoform will be formed by the linguistic variable "Body temperature" with its linguistic term "very high": $P(S_{D1}) = VD1 \cup TD1$, where $VD1$ is the linguistic variable associated with sensor $D1$ and $TD1$ is the temporal window associated with sensor $D1$.

Once the protoform to be applied to the dataset has been presented, each of its component parts is defined. Therefore, the first step is to define the linguistic variable "Body temperature" $VD1$.This variable will be associated to the data coming from the sensor $D1$.

The linguistic variable consists of linguistic terms, which are represented by membership functions. In this case, the linguistic variable "Temperature" will be represented by 4 linguistic terms: Low, $Normal$, $High$, $VeryHigh$, whose membership function is trapezoidal with the following semantics expressed in degrees centigrade (see Figure 2):

- Low (34, 34, 36, 36.8)
- Normal (36, 36.8, 36.8, 37.4)
- High (37.2, 37.8, 37.8, 38.5)
- Very High (38.1, 41.2, 43, 43)

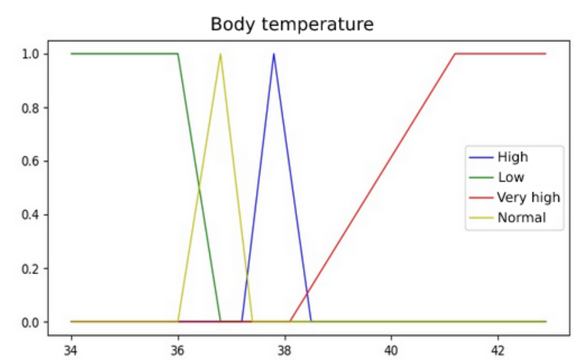

Figure 2. Linguistic variable body temperature together with its associated linguistic terms.

Having defined the linguistic variable together with its fuzzy terms, it is now necessary to define the time window over which the fuzzy protoform is processed. To do so, we will define a linguistic term with its membership function that represents that time window $\mu TD1(\Delta t_i)$. In this case it will be called "During the morning". This time window will take as values L_1, L_2, L_3, L_4=(28800, 39600, 46800, 57500), being t^j=86400 (24 hours).

The membership of a pair of values $\bar{s}_i$ to a fuzzy term value $VD1$ in a temporary linguistic term $TD1$, of the body temperature sensor S_{D1}, is defined by the intersection operator that merges both degrees of membership such and therefore for our use case the resulting equation to be applied is the following:

$$VD1 \cap TD1(\bar{s}_i) = \mu VD1_{VeryHigh}(s_i) \cap \mu TD1(\Delta t_i) \quad (2)$$

To aggregate the resulting membership information of a subset of measurements given by the sensor data stream associated with a fuzzy temporal window T with $\mu TD(\Delta t_i)$ ¿0, the fuzzy join operator is used to obtain a degree of membership of a fuzzy term in a temporal linguistic term and as indicated in the equation:

$$VD1 \cup TD1(S_D1) = \cup(VD1 \cap TD1(\bar{s}_i)) \in [0, 1] \quad (3)$$

In Figure 3 graphically illustrates the result of applying the protoform to a subset of 24 hours (out of the 5 days of the set). This subset consists of the 24 hour range from [1631484000,35:5] to [1631569500,36.4]. The minimum values obtained by applying the linguistic variable to the data in the fixed time window are underlined in green.

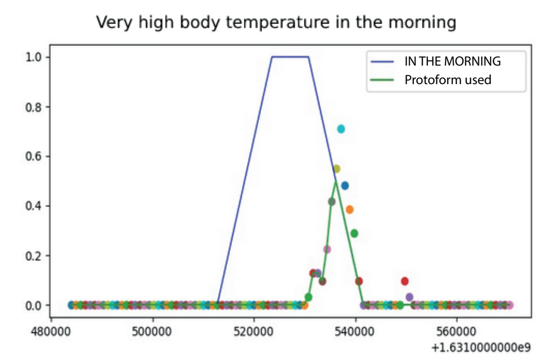

Figure 3. Protoform very high body temperature in the morning.

The result obtained for this 24 hour subset was "Very High Body Temperature During the morning" with 0.5 degree of belonging. For each of the 15 days that make up the set, the graphical representation of its protoform and the value obtained will be obtained.

4.2. Fuzzy IoT sofware tool

Once the theoretical framework has been defined, it will be presented how this use case is defined using the constructed tool.

The Figure 4 illustrates how the $S1$ sensor has been defined and the sensortemperature.csv dataset, which has been generated by this sensor, has been loaded, using the built tool, in which the timestamp and value pairs for each shot taken by the $S1$ sensor are found. In

ID SENSOR	DESCRIPTION	MAX VALUE	MIN VALUE	INTERVALS	TIME BETWEEN SAMPLES	DELETE
S1	Patient Temperature Sensor	43	34	0.1	900	Delete
S2	Glucose Sensor	400	15	10	3600	Delete
S3	Humidity Sensor	0	0	0	0	Delete
S4	CO2 Sensor	0	0	0	0	Delete

Figure 4. Sensor S1 definition.

Figure 4 shows the data entered into the tool to define the sensor. The software requires a Sensor ID, in this case $S1$, and a description of the sensor. Also defined for each sensor is the maximum value, the minimum value and how often data is taken (time between samples).

In Figure 5 shows how the dataset is uploaded using the tool. To do this, a name must be given to the dataset and the path from where it should be uploaded to the server. In addition, the type of file or database on which this case study works has been selected, in our case a historical temperature data file.

DATA SET NAME	PATH	TYPE	DATABASE	ACTIONS
5 DAYS PACIENT TEMPERATURE	/var/www/html/uploads/sensoretemperatura.csv	HISTORICAL		Delete
1 DAY PACIENT TEMPERATURE	/var/www/html/uploads/sensorito.csv	HISTORICAL		Delete
15 DAYS PACIENT GLUCOSE	/var/www/html/uploads/sensorazucar.csv	HISTORICAL		Delete
5 DAYS CITY GASES	/var/www/html/uploads/SENSORGAS.csv	HISTORICAL		Delete

Figure 5. Upload file data recorded by sensor S1.

In order to define a linguistic variable with the tool, the desired name must be entered. Once chosen, each of the linguistic terms associated with the linguistic variable is created by entering the name of the term and the 4 values that define the trapezoidal membership function as shown in Figure 6. This figure shows how the variable "Body temperature" has been defined.

VAR NAME	ACTIONS
BODY TEMPERATURE	Add Terms Delete
GLUCOSE LEVEL	Add Terms Delete
HEIGHT	Add Terms Delete
HUMIDITY	Add Terms Delete
PRESSURE	Add Terms Delete

Figure 6. Definition Linguistic Variable Body Temperature.

TEMPORAL FUZZY WINDOWS NAME	A VALUE	B VALUE	C VALUE	D VALUE	TJ TIME VALUE	ACTIONS
IN THE MORNING	28800	39600	46800	57600	86400	Modify Delete
OVERNIGHT	0	14400	21600	28800	86400	Modify Delete

Figure 7. Definition fuzzy temporal window "In the morning" together with its associated linguistic terms.

FUZZY PROTOFORM NAME	LINGUISTIC VARIABLE AND TERM	FUZZY TEMPORAL WINDOWS	LINGUISTIC QUANTIFICADOR	ACTIONS
P-456JD	GLUCOSA LEVEL-VERY HIGH	OVERNIGHT		Modify Delete
P-GMBDLN	GLUCOSA LEVEL-VERY LOW	OVERNIGHT		Modify Delete
P-TMADLM	BODY TEMPERATURE-VERY HIGH	IN THE MORNING		Modify Delete

Figure 8. Definition Fuzzy Protoform.

FUZZY PROTOFORMS: P-456JD DATASETS: 5 DAYS PACIENT TEMPERATURE SENSORS: S1

INSERT FRAMEWORK

PROTOFORM	DATA SET	SENSORS	ACTIONS	EXECUTION
P-TMADLM	5 DAYS PACIENT TEMPERATURE	S1	Eliminar	Resolver

Figure 9. Definition Framework applied to the problem

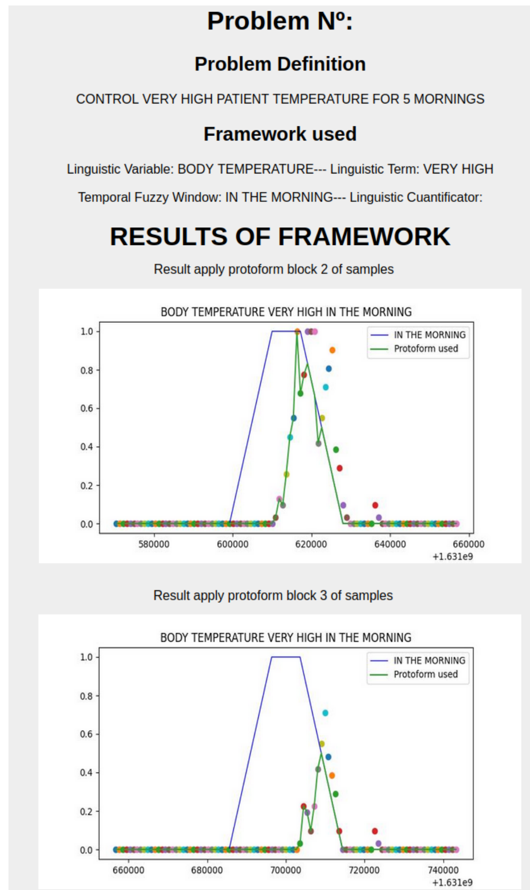

Figure 10. Result of applying the framework to the problem.

To define the temporal window in the proposed FuzzyIoT, a name and the linguistic terms which define it are entered. The Figure 7 shows how the temporal window "In the Morning" has been defined. Using the tool the name of the temporal window and the values of each of the linguistic terms are entered, which conform its function of belonging to this window and, finally

t^j, the time that elapses from the instant T^* until the moment of analysis.

Next, the protoform is defined. For this purpose, Figure 8 shows the definition. First, the name of the protoform ("Very high temperature during the morning") is given and then the variable "temperature" is assigned with its term "very high" and the time window "during the morning". It can be seen how in the tool the definition of the protoform $P(S_{D1})$ has been created.

In Figure 9 the creation of a work frame with previously defined elements to be used (protoform, temporal window, variable, dataset, etc.) is shown.

After applying the protoform to the dataset (the frame previously created in the tool) the graphs obtained in Figure 10 represent the shapes of the variable, the time window and the protoform and the value obtained from applying the protoform to the problem.

4.3. Alignment with the Sustainable Development Goals

In 2015 the United Nations adopted 17 SDGs to promote the end of poverty, protect the planet and improve the lives and prospects of people around the world in order to reach them by 2030[5]. Currently, many of the IoT systems in the literature can contribute directly or indirectly to achieve these goals Espinosa et al., 2021.

In this paper, we have presented a software tool entitled FuzzyIoT as a platform for descriptive analysis of sensor data streams by means linguistic protoforms in the IoT context.

For the presented case study, the contribution with the SDGs can be analyzed due to the fact that the model analyzes in natural language the temperature of a patient in the ICU and to ensure an optimal use of resources (medication), specifically, the study of medication administration. The result of this system, broadly speaking, is the contribution to improving patient health and ensuring optimal use of resources (medication).

To obtain the evaluation, the 231 indicators and 169 targets have been analysed, and it can be concluded that the model strongly contributes to the following goals: i) Goal 3. Ensure healthy lives and promote well-being for all at all ages, ii) Goal 11. Make cities and human settlements inclusive, safe, resilient and sustainable and, iii) Goal 12. Ensure sustainable consumption and production patterns.

In this case, the analysis concludes an alignment with Goals 3, 11 and 12, but it is possible to contribute to other types of SDGs depending on the objective of the IoT system, such as water quality monitoring (Goal 6. Water and sanitation), energy efficiency monitoring (Goal 7. Affordable and clean energy) or even the monitoring of gas emissions (Goal 13. Climate action).

[5]https://unstats.un.org/sdgs/indicators/indicators-list

5. Conclusions

There is an explosion of data streams generated by smart devices with sensors under the new Internet of Things (IoT) paradigm. However, there is currently no software capable of modeling information from devices in a flexible, simple, and clear manner. Fuzzy protoforms have been demonstrated as a successful tool for modelling uncertainty generated by the sensor data. This contribution has presented the software tool entitled FuzzyIoT that is a platform for descriptive analysis of sensor data streams by means linguistic protoforms. To do so, the functional and not-functional requirements have been presented, then client-server architecture has been proposal and the languages that have been used have been indicated. Finally, a case study has been illustrated to show the functionality of the proposed tool for the described data analysis and its alignment with the SDGs has been analysed. Future work in this contribution is focused on combining various linguistic protoforms through aggregation operators and rules.

Acknowledgements

This result has been partially supported through the research projects REMIND project Marie Sklodowska-Curie EU Framework for Research and Innovation Horizon 2020, under grant agreement no. 734355, the Spanish Government by the project PID2021-127275OB-I00.

References

Albin-Rodriguez, A.-P., De-La-fuente-Robles, Y.-M., Lopez-Ruiz, J.-L., Verdejo-Espinosa, A., & Espinilla, M. (2021). Ujami location: A fuzzy indoor location system for the elderly. *International Journal of Environmental Research and Public Health*, *18*(16).

Albın-Rodrıguez, A.-P., Ricoy-Cano, A.-J., de-la-Fuente-Robles, Y.-M., & Espinilla-Estévez, M. (2020). Fuzzy protoform for hyperactive behaviour detection based on commercial devices. *International Journal of Environmental Research and Public Health*, *17*(18), 6752.

Cornejo, M., & Medina, J. (2022). Implicative linguistic summaries. *Studies in Computational Intelligence*, *955*, 21–28.

Espinilla, M., Medina, J., Hallberg, J., & Nugent, C. (2018). A new approach based on temporal sub-windows for online sensor-based activity recognition. *Journal of Ambient Intelligence and Humanized Computing*, 1–13.

Espinosa, Á. V., Ruiz, J. L. L., Mata, F. M., & Estevez, M. E. (2021). Application of IoT in healthcare: Keys to implementation of the sustainable development goals. *Sensors*, *21*(7), 2330.

Estrella, F., Espinilla, M., Herrera, F., & Martinez, L. (2014). Flintstones: A fuzzy linguistic decision tools enhancement suite based on the 2-tuple linguistic model and extensions. *Information Sciences*, *280*, 152–170.

Ferri-Ramírez, C., Ramírez-Quintana, M.-J., & Hernández-Orallo, J. (2004). *Introducción a la Minería de Datos*. Prentice-Hall Pearson.

Fontenla-Seco, Y., Bugarin, A., & Lama, M. Fuzzy temporal protoforms for the quantitative description of processes in natural language. In: *2021-July*. 2021.

Lopez-Medina, M., Espinilla, M., Cleland, I., Nugent, C., & Medina, J. (2020). Fuzzy cloud-fog computing approach application for human activity recognition in smart homes. *Journal of Intelligent and Fuzzy Systems*, *38*(1), 709–721.

López-Medina, M., Espinilla, M., Cleland, I., Nugent, C., & Medina-Quero, J. (2020). Fuzzy Cloud-fog computing approach application for human activity recognition in smart homes. *Intelligence and Fuzzy Systems*, *38*, 709–721.

López-Medina, M., Espinilla, M., Paggetti, C., & Medina Quero, J. (2019). Activity Recognition for IoT Devices Using Fuzzy Spatio-Temporal Features as Environmental Sensor Fusion. *Sensors*.

Martinez-Municio, S., Rodriguez-Benitez, L., Castillo-Herrera, E., Giralta, J., & Jimenez-Linares, L. (2018). Linguistic modeling and synthesis of heterogeneous energy consumption time series sets. *International Journal of Computational Intelligence Systems*, *12*(1), 259–272.

Medina, J., Espinilla, M., Zafra, D., Martinez, L., & Nugent, C. (2017). Fuzzy fog computing: A linguistic approach for knowledge inference in wearable devices. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10586 LNCS*, 473–485.

Peláez-Aguilera, M.-D., Espinilla, M., Fernández Olmo, M.-R., & Medina-Quero, J. (2019). Fuzzy linguistic protoforms to summarize heart rate streams of patients with ischemic heart disease. *Complexity*, *2019*, 1–11.

Polo-Rodriguez, A., Vilchez Chiachio, J., Paggetti, C., & Medina-Quero, J. (2021). Ambient sound recognition of daily events by means of convolutional neural networks and fuzzy temporal restrictions. *Applied Sciences (Switzerland)*, *11*(15).

Ramos-Soto, A., & Martin-Rodilla, P. (2021). Enriching linguistic descriptions of data: A framework for composite protoforms. *Fuzzy Sets and Systems*, *407*, 1–26.

Weber, S. (1983). A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms. *Fuzzy Set and System*, *11*(1), 103–113.