A Simple Model for Sharing Knowledge Among Heterogeneous Sensor Data

Gustavo Monte
Universidad Tecnológica Nacional
Facultad Regional Del Neuquén
Plaza Huincul, Argentina
gustavo.monte@ieee.org

Damian Marasco
Universidad Tecnológica Nacional
Facultad Regional Del Neuquén
Plaza Huincul, Argentina
ndm922@hotmail.com

Ariel Agnello
Universidad Tecnológica Nacional
Facultad Regional Del Neuquén
Plaza Huincul, Argentina
arielagn@hotmail.com

Ruben Bufanio
Universidad Tecnológica Nacional
Facultad Regional Del Neuquén
Plaza Huincul, Argentina
ruben.bufanio@speedy.com.ar

Norberto Scarone
Universidad Tecnológica Nacional
Facultad Regional Del Neuquén
Plaza Huincul, Argentina
scarone_norberto@hotmail.com

Pablo Liscovsky
Universidad Tecnológica Nacional
Facultad Regional Del Neuquén
Plaza Huincul, Argentina
pliscovsky@frn.utn.edu.ar

***Abstract*—This work presents a simple model that expose the information embedded into a sensor signal allowing to share it independently of the signal nature. Today's highly interconnected world requires a representation of sensor signals that let efficient sharing of embedded information. The proposed model is a state matrix that combine two important aspects of any signal: Its value inside a range and its behavior over the time. From this state matrix is possible to obtain a self-learning model observing the state transition probabilities and the time lapse in each state to deduce signal normality-abnormality that allows to infer a better perception of reality.**

Keywords—Edge processing, self-learning, data fusion, sensor signal representation, Smart Cities.

I. INTRODUCTION

Nowadays, 55% of the world's population lives in urban areas, a proportion that is expected to increase to 68% by 2050 [1]. therefore, cities are becoming highly complex systems and intelligent paradigms are required for efficient and safe administration.

All actions and decisions made by a system are based on signals from sensors. Sensor-driven digital systems convert perception into information on which operators and systems act [2]. An erroneous perception can generate serious accidents, environmental impact and economic losses. The process of extracting information and knowledge from sensor data in a heterogeneous environment is a challenging task. The simple fact of detecting normality or abnormality of a signal requires having learned its behavior in response to various stimuli.

It could be thought that smart systems and sensors are a natural consequence of the overwhelming increase in processing power and connectivity [3]. But a system does not become intelligent for the simple fact of processing faster and having connectivity, but rather, with the extra capacity, creating and implementing algorithms to reach a state that increases reliability, extracts knowledge from signals and generate elements of judgment for correct decision making. Complex systems, like Smart Cities, take information from sensors and transducers of diverse nature and non-conventional sensors like traffic density obtained from a camera. To feed decision-making processes it is necessary to relate signals that, in advance, is not well known how a perturbation in one signal could affects others, which a priori was not taken as related to the system.

Recent advances in Big Data have notably contributed to the development and consolidation of Smart Cities paradigm [4]. "Big Data" generally refers to large and complex sets of data that represent digital traces of human activities mixed with sensor data [5]. Digital data, generated from a variety of devices, are growing at shocking rates [6],[7]. Until now it is not clear how and where this huge amount of information should be processed. During last years, machine learning techniques have been widely adopted in several massive and complex data-intensive fields such as medicine, astronomy, biology, industry, smart cities, and so on.

As data sets grow and become more complex, the implementation of traditional learning algorithms turns out to be more difficult. The goal of this paper is to provide a simple model that fits any data set. The main idea of the signal representation is to provide an information structure and knowledge regardless of the type of signal. Therefore, it allows to relate behaviors among diverse signals, quite common in complex systems. The algorithms and techniques for representing signals proposed in this work should be incorporated at the point of acquisition in embedded systems to analyze behavior and to extract information and knowledge, in accordance with the Edge Computing paradigm [8]. From this inference, it is possible to estimate signal states, statistics, self-learning patterns, and the class of normal/abnormal or illogical behavior and share knowledge of the system to the cloud.

The following sections present the proposed model. In sections II, the model is described in detail, highlighting the efficient information extraction for any kind of signal. The signal representation is described and the learning and monitoring stages to detected behavior patterns, especially those associated with normality. Using supervised learning it is possible to train the model with specific patterns associated also with abnormal conditions.

Section III shows experimental results for different sensor signals and datasets. The learning and monitoring signals will be shown, highlighting the benefits of the model for signals of different nature. In section IV are the conclusions.

II. INFERENCE ALGORITHMS

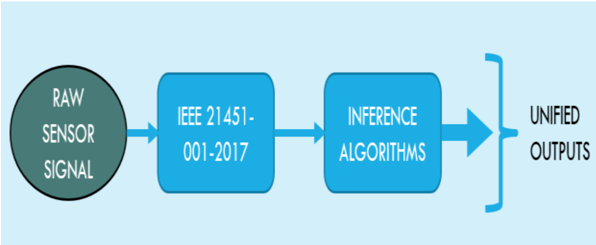

Fig. 1. Synthesis of signal processing model . From digital samples to segments at the output of IEEE 21451-001, then this output is processed to obtain unified information.

A. *Sensor signal model*

The goal of this model is to unify the information and knowledge embedded in the sensor signal as shown in Fig. 1.

The sensor signals have embedded the physical system information they represent; therefore, the trajectory of these signals is in accordance with the dynamics of the associated system. The analysis of these trajectories allows inferring the system behavior they represent, validating their representation and detecting possible anomalous states.

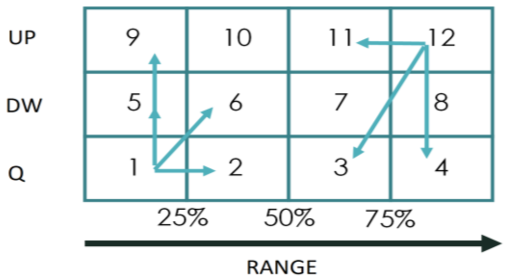

Fig. 2. Representation of the signal as a matrix of behavior vs range. DW means signal going down, UP going up and Q at rest. The arrows indicate possible transitions from states 1 and 12 as an example.

It is proposed to represent the signal according to two important aspects: its value and its trend. The signal is classified in 12 states. The range of the signal is divided into four parts and each of them in three conditions: signal rising, falling and at rest, as shown in Fig. 2. This representation will be called **Δ** matrix. Let us remember that sensory signal carries the dynamics of the physical system it represents. Therefore, the state's dynamic, the transition probabilities and the time lapse in each state identify the system that the signal represents.

The digitization technique employed to determine the rows of the **Δ** matrix is the one described into the IEEE 21451-001-2017 standard [9], which considers sensor data as a bounded sequence of segments instead of samples. The representation of a signal by uniform sampling is not suitable for understanding and does not facilitate knowledge inference processes. Basically, it is due to one reason: it is unknown which samples carry information and which are redundant. All signals have some degree of redundancy [10]. If oversampling condition is satisfied, the value of the sample loses weight in determining the information. In other words, redundancy has been generated. The IEEE 21451-001 standard eliminates redundant samples while maintaining signal structure. This representation provides a simpler and more direct inference platform for extracting information about behavior that determines the rows of **Δ** matrix in Fig. 2.

If the evolution of the signal is observed on the **Δ** matrix, stochastic variables are generated that capture the essence of the signal behavior, especially in cyclic processes. The main observed variables are:

$\boldsymbol{P_{ij}}$ = Transition probability from state **i** to **j**.

$\boldsymbol{T_{i\_min}}$ = Minimum time lapsed in state **i**.

$\boldsymbol{T_{i\_max}}$ = Maximum time lapsed in state **i**.

$\boldsymbol{T_{i\_ave}}$ = Average time lapsed in state **i**.

To estimate the transition probabilities $\boldsymbol{P_{ij}}$, a twelve-by-twelve matrix **Π** is formed by accumulating the state transitions as the signal evolves. Once the sensor signal has gone through its cyclic operating states, the transition probabilities are estimated as the quotient of the number of transitions and the total number that occur in a row (1).

$$\boldsymbol{P}_{ij} = \frac{c_{ij}}{\sum_{j=1}^{12} c_{ij}} \quad (1)$$

$$i = 1,2,3,\ldots,12 \;;\; j = 1,2,3,\ldots,12$$

The **Π** matrix of 144 real numbers captures the signal behavior and, evaluating the sequence of transition probabilities, it is possible to determine if the evolution of the state changes in the matrix **Δ** corresponds to normal behavior. Everything that is not normal is classified into logical and illogical. A pattern is illogical when it is physically impossible for the pattern to exhibit that behavior. These events are detectable by the state transitions in the **Δ** array. For example, when the minimum time between two non-contiguous cells is violated.

B. *Learning and monitoring stages*

During the learning stage, the transition probabilities are estimated. Two signal learning models are proposed. The first one is the basic model named **BSLM** (**B**asic **S**ignal **L**earning **M**odel). Figure 3. outlines the **BSLM**. The inputs TH MAX and TH MIN are the thresholds for signal monitoring.

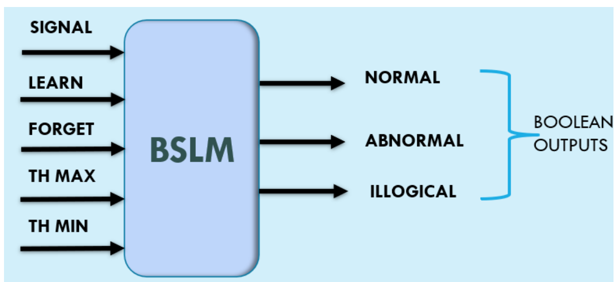

Fig. 3. Block diagram of the Basic Signal Learning Model. It learns one signal model using supervised learning.

The **BSLM** model uses supervised learning then *learn* and *forget* inputs must be provided. When the input *learn* changes from false to true, the **BSLM** model starts accumulating the state transitions in **Π** array. This process continues until *learn* changes from true to false. This input must be false when the signal has traversed all possible states a repeated number of times. The **Π** matrix is estimated using (1) and stored in memory. Only one model can be learned, and the outputs are true/false.

The second model is the **ASLM** (**A**dvance **S**ignal **L**earning **M**odel), Fig. 4.

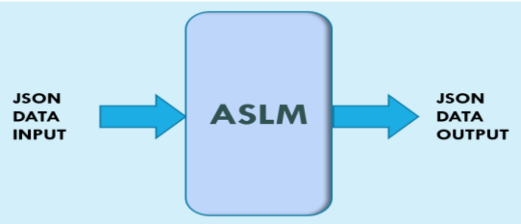

Fig. 4. Block diagram of the Advance Signal Learning Model. The inputs and output are formated in JSON. It can learn in supervised and unsupervised way. Multiple models can be stored.

This model uses **JSON (J**ava**S**cript **O**bject **N**otation format for input and output. is a lightweight data-interchange format. It is easy for humans to read and write. It is simple for machines to parse and generate [11], and it is flexible and adaptable for new data , while maintaining compatibility with pre-existing data structures. This model can learn multiples models using supervised or unsupervised learning.

During the monitoring stage, two indicators are computed that have as input the sequence of probabilities observed as the signal evolves. The first indicator is the estimation of the mean probability of the observed path using the recursive mean formula [12], as shown in (2).

$$pm(k) = \frac{pm(k-1)\,(k-1)+p(k)}{k} \tag{2}$$

Where $pm(k)$ is the mean probability at time k and $p(k)$ is the probability of the observed transition.

The second indicator, short-term estimation, rewards when the transition occurs over the maximum probability and penalizes when it does not. The indicator u accumulates rewards and penalties by saturating in TH_{MAX} and TH_{MIN} according to the following expression:

$$u(k) = \begin{cases} u(k-1) + p(k) \\ if \;\; p(k) = p_{max}(k) \\ \\ u(k-1) - \left(p_{max}(k) - p(k)\right) \\ if \;\; p(k) < p_{max}(k) \end{cases} \tag{3}$$

$$if\ u(k) > TH_{MAX} \xrightarrow{then} u(k) = TH_{MAX}$$

$$if\ u(k) < TH_{MIN} \xrightarrow{then} u(k) = TH_{MIN}$$

Where $p(k)$ and is the observed probability and $p_{max}(k)$ the maximum probability in the row of the **Π** matrix where the transition occurred. The variable $u(k)$ responds quickly to uncommon signal patterns as it will be shown in the experimental section. The sensitivity is controlled by the threshold values.

The evolution of the state transition can be contrasted with different pre-trained models. These models can include normality models and specific patterns of interest and run in parallel.

III. EXPERIMENTAL RESULTS

The model was evaluated with real signals and from databases. Fig 5. shows the monthly average temperature of land and ocean of this planet from year 1850 up to 2015 [13]. This signal will be used as a first example for learning and monitoring.

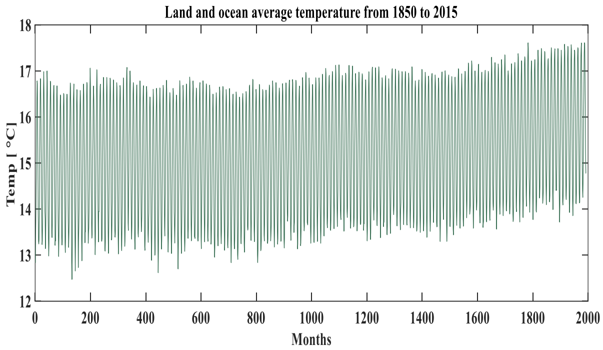

Fig. 5. Monthy average temperature of land and ocean of Earth planet from year 1850 up to 2015.

Observing the graph of average temperatures, an increase in maximum and minimum temperatures in recent years is clearly seen. Let's train the model with the first years to quantify with the indicators this visible anomaly.

The model was trained with the first twenty years. Fig. 6. shows the **Π** matrix as a result of the training.

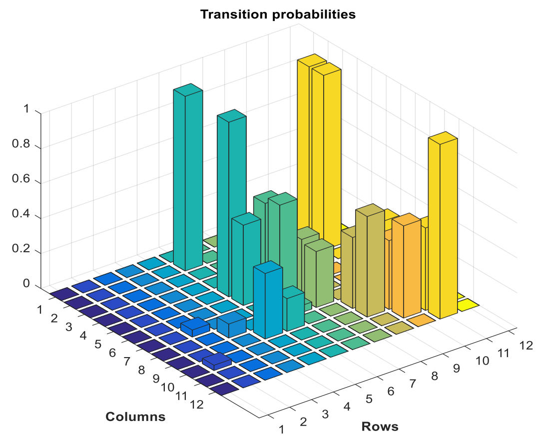

Fig. 6. Π matrix after training using the first twenty years of the Fig 5 signal.

The mean probability $pm(k)$ computed after training is shown in Fig. 7. The increase of the maximum and minimum values of temperature observed during the last years, causes the decrease of the average probability.

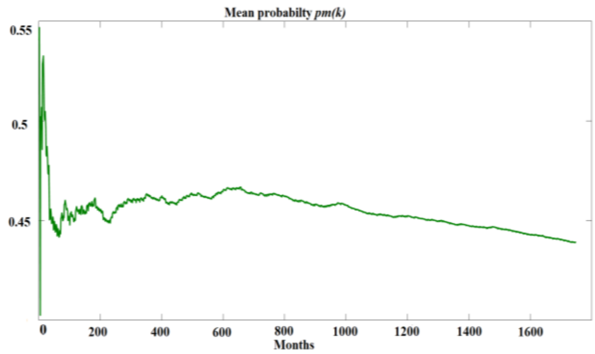

Fig 7. Mean probabilty *pm(k)* for the signal in Fig. 5

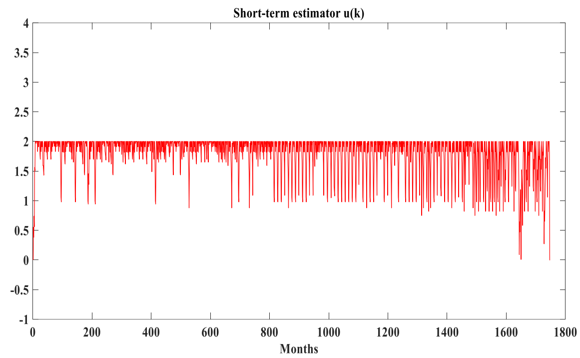

Fig 8. Short term estimator $u(k)$ for the signal in Fig. 5. TH_{MAX} =2, TH_{MIN}=-2. Note that there are three regions considering how often $u(k)$ aparts from TH_{MAX}.

The short-term estimator $u(k)$ provides more information about the moment of occurrence of the abnormalities. Moreover, it responds quickly when the signal apart from normality. The Fig. 8 shows this estimator for the signal in Fig. 5.

If the behavior of the signal is normal, the indicator u(k) tends to stay close to the threshold TH_{MAX}. As abnormal behavior appears due to the low probability of occurrence, the indicator begins to deviate from the TH_{MAX} threshold and tends towards the TH_{MIN} threshold. In Fig. 8. three regions determined by the tendency to move away from TH_{MAX} are observed. The worst abnormality condition occurs in month 1650 which corresponds to year 2007.

A flow signal from an industrial process will be taken as a second example. Fig 9 shows the normalized signal corresponding to a cyclic process. The model was trained using two cycles that correspond to an interval of 3000 minutes.

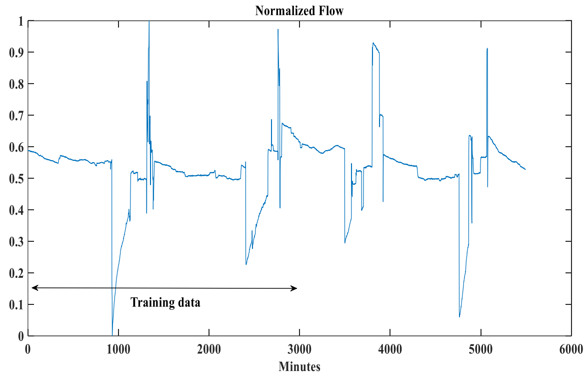

Fig 9. Normalized flow from a cyclic industrial process. The model was trained with the first 3000 minutes.

Fig. 10 shows the $\mathbf{\Pi}$ matrix for the flow signal.

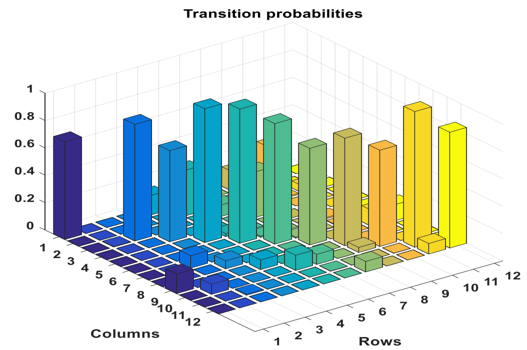

Fig 10. Π matrix after training using the 3000 minutes of Fig. 9 signal.

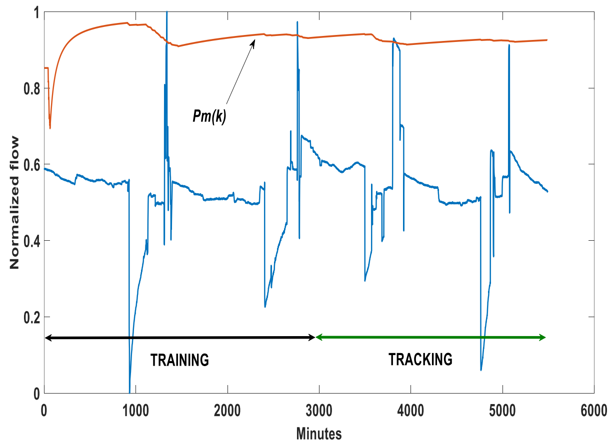

Fig 11. Normalized flow and transition probability $pm(k)$. From minute 3000 onwards the flow signal was never entered into the model, even so the probability pm(k) remains close to the training. values.

Fig. 11. shows the mean probability $pm(k)$ for the whole signal. The tracking signal, which was never presented to the system, is similar, but not the same, generating an average path probability that remains close to the learning value.

The short-term estimator $u(k)$ is shown in Fig.12. The strong point of the u(k) estimator is the speed of response to unlikely transitions. If the patterns are the learned ones, most of the time the estimator saturates at the threshold value TH_{MAX}. The detection of abnormal states is controlled by the value of the threshold and the time it remains in the negative threshold.

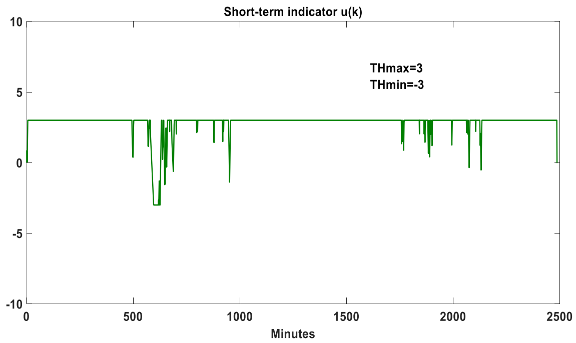

Fig 12. Short-term indicator $u(k)$ for the normalized flow of Fig. 8. Note that most of the time the value stays close to TH_{MAX} for the training and tracking stages.

In Fig.13. Gaussian distribution noise was added with σ=0.01 starting from minute 4000. It is observed how the average probability begins to decrease with the inclusion of noise.

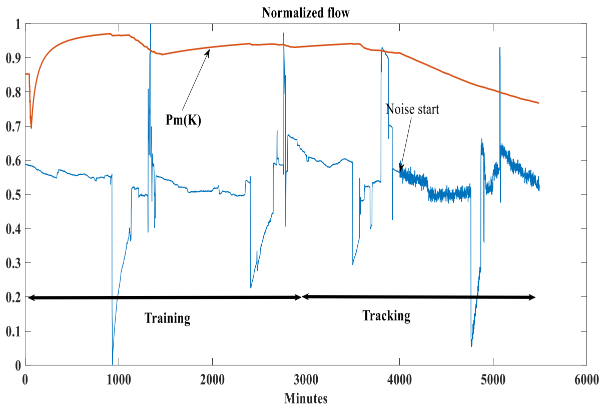

Fig 13. Normalized flow with Gaussian noise added starting from minute 4000. Note how $pm(k)$ starts to decrese from this instant.

Fig. 14. shows the short-term indicator $u(k)$ for the noisy signal. The goodness of the $u(k)$ indicator is that it tags the time instant in which the signal begins to depart from the learned model.

It is clear from Fig. 14 that, if the signal exhibits its typical trajectory, most of the time it is in the positive threshold. However, if the signal trajectory includes cells with low probability, most of the time it reaches the negative threshold. Therefore, this indicator allows a quick discernment between normality and abnormality.

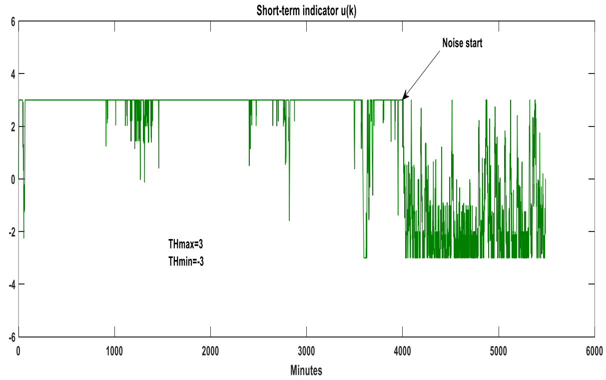

Fig 14. Short time indicator $u(k)$ for the normalized flow signal with Gaussian noise starting from minute 4000. This indicator tends to the lower threshold almost inmediatly.

IV. Conclusions

An information representation model of sensory signals is proposed. Unlike the uniform sampling of a signal, which only considers the value, this model uses the value and its behavior with respect to the past. A deeper understanding of the signal is achieved on which a normality/abnormality model is built. This representation is plausible for all signals and provides a highly versatile unified platform, especially when dealing with signals of a diverse nature.

Knowledge of sensory signals promotes safer and more efficient systems since a better inference of reality is achieved. The model is easy to implement in embedded systems and adapts perfectly to the Edge Computing paradigm. The matrix occupies 144 float numbers for each sensor signal state, a low requirement for current technology. In addition, the implementation of tracking variables consumes low CPU resources, therefore these algorithms can be added to embedded systems without additional requirements.

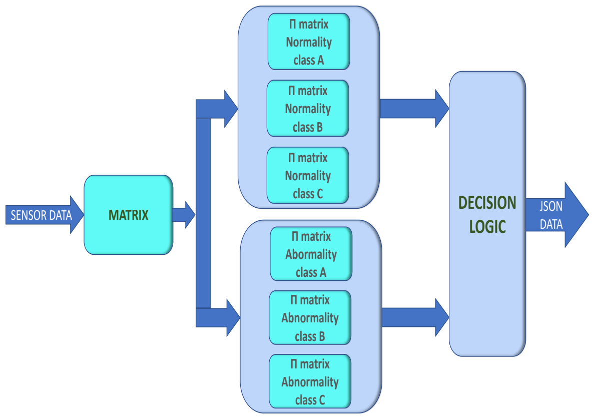

Fig 15. The model applied to classify and subclassify normal and abnormal behavior.

The normal state can be classified to identify different classes within it. For example, from a traffic density sensor, could learn normality on a weekday versus a weekend or a holiday. Also, an hourly based learning can be processed to infer normality and infer the state at each hour of the day.

Also, the abnormality can be subclassified to determine the type of anomaly. In Fig. 15 the block diagram is shown. Decision logic defines the most likely behavior by evaluating $u(k)$ and $pm(k)$.

Currently, work is being done on the correlation among different signals represented with this model. For example, for the traffic density sensor find out how is the relationship between weather models and traffic given that from the model, through the indicator u(k), the moment in which it changes from normality to abnormality is obtained.

REFERENCES

[1] United Nations https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html.

[2] Cristiano Premebida, Rares Ambrus and Zoltan-Csaba Marton. “Intelligent robotic perception systems, Applications of Mobile Robots, IntechOpen, 2018. DOI: 10.5772/intechopen.79742. Available from: https://www.intechopen.com/chapters/62978.

[3] Ahuett-Garza, H., & Kurfess, T. “A brief discussion on the trends of habilitating technologies for Industry 4.0 and Smart manufacturing”, Manufacturing Letters, 2018, Vol. 15, pp. 60-63.

[4] Qiu, J., Wu, Q., Ding, G. *et al.* “A survey of machine learning for big data processing”. *EURASIP J. Adv. Signal Process.* 2016**,** 67, 2016. https://doi.org/10.1186/s13634-016-0355-x

[5] S.E. Bibri. “The IoT for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability”. Sustainable Cities and Society, 38, 2018, pp. 230-253.

[6] https://datareportal.com/reports/digital-2021-global-overview-report

[7] Gerhard P. Hancke, Bruno de Carvalho e Silva, Gerhard P. Hancke, Jr. “The Role of Advanced Sensing in Smart Cities”. Sensors (Basel) 2013 Jan 13 393–425. Published online 2012 Dec 27. doi: 10.3390/s130100393.

[8] W. Shi, G. Pallis and Z. Xu, "Edge Computing [Scanning the Issue]," in *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1474-1481, Aug. 2019, doi: 10.1109/JPROC.2019.2928287.

[9] 21451-001-2017 - IEEE Recommended Practice for Signal Treatment Applied to Smart Transducers. https: //standards.ieee.org/standard/21451-001-2017.html.

[10] M. Carratù, A. Espírito-Santo, G. Monte and V. Paciello, "Earthquake Early Detection as an IEEE1451 Transducer Network Trigger for Urban Infrastructure Monitoring and Protection," in *IEEE Instrumentation & Measurement Magazine*, vol. 23, no. 9, pp. 43-49, December 2020, doi: 10.1109/MIM.2020.9289072.

[11] Robert Grover Brown, Patrick Hwang. Introduction to Random Signals and Applied Kalman Filtering. John Wiley & Sons. 1997.

[12] https://www.json.org/

[13] https://data.world/data-society/global-climate-change-data