A two-phase algorithm for recognizing human activities in the context of Industry 4.0 and human-driven processes

Borja Bordel[1], Ramón Alcarria[1], Diego Sánchez-de-Rivera[1]

[1] Universidad Politécnica de Madrid,
Madrid, España
bbordel@dit.upm.es, ramon.alcarria@upm.es, diegosanchez@dit.upm.es

Abstract. Future industrial systems, a revolution known as Industry 4.0, are envisioned to integrate people into cyber world as prosumers (service providers and consumers). In this context, human-driven processes appear as an essential reality and instruments to create feedback information loops between the social subsystem (people) and the cyber subsystem (technological components) are required. Although many different instruments have been proposed, nowadays pattern recognition techniques are the most promising ones. However, these solutions present some important pending problems. For example, they are dependent on the selected hardware to acquire information from users; or they present a limit on the precision of the recognition process. To address this situation, in this paper it is proposed a two-phase algorithm to integrate people in Industry 4.0 systems and human-driven processes. The algorithm defines complex actions as compositions of simple movements. Complex actions are recognized using Hidden Markov Models, and simple movements are recognized using Dynamic Time Warping. In that way, only movements are dependent on the employed hardware devices to capture information, and the precision of complex action recognition gets greatly increased. A real experimental validation is also carried out to evaluate and compare the performance of the proposed solution.

Keywords: Industry 4.0; pattern recognition; Dynamic Time Warping; Artificial Intelligence; Hidden Markov Models

1 Introduction

Industry 4.0 [1] refers the use of Cyber-Physical Systems (unions of physical and cybernetic processes) [2] as main technological component in future digital solutions, manly (but not only) in industrial scenarios. Typically, digitalization has caused, at the end, the replacement of traditional work mechanisms by new digital instruments. For example, workers in the assembly lines were substituted by robots during the third industrial revolution.

However, some industrial applications cannot be based on technological solutions, being human work still essential [3]. Hand-made products are an example of applications where the presence of human works is essential. These industrial sectors, any case, must be also integrated into fourth industrial revolution. From the union of

Cyber-Physical Systems (CPS) and humans acting as service providers (active works), humanized CPS arise [4]. In these new systems, human-driven processes are allowed [5]; i.e. processes which are known, executed and managed by people (although they may be watched over by digital mechanisms).

To create a real integration between people and technology, and move the process execution from the social subsystem (humans) to the cyber world (hardware and software components), techniques for information extraction are needed. Many different solutions and approaches have been reported during the last years, but nowadays pattern recognition techniques are the most promising one.

The use of Artificial Intelligence, statistical models and other similar instruments have allowed a real and incredible development of pattern recognition solutions, but some challenges are still pending.

First, pattern recognition techniques are dependent on the underlying hardware device for information capture. The structure and learning process changes if (for example) instead of accelerometers we consider infrared sensors. This is very problematic as hardware technologies evolve much faster than software solutions.

And, second, there is a limit to the precision in the recognition process. In fact, as human actions turn more complicated, more variables and more complex models are required to recognize them. This approach generates large optimization problems whose residual error is higher as the number of variables increases; which causes a decreasing in success recognition rate [6]. In conclusion, mathematics (not software, thus, not dependent on the implementation) force a certain precision for the pattern recognition process given the actions to be studied. To avoid this situation, a lower number of variables should be considered, but this also reduces the complexity of actions that may be analyzed; a solution which is not acceptable in industrial scenarios where complex production activities are developed.

Therefore, the objective of this paper is to describe a new pattern recognition algorithm addressing these two basic problems. The proposed mechanism defines actions as a composition of simple movements. Simple movements are recognized using Dynamic Time Warping (DTW) techniques [7]. This process is dependent on the selected hardware for information capture; but DTW are very flexible and updating the pattern repository is enough to reconfigure the entire algorithm. Then, complex actions are recognized as combinations of simple movements through Hidden Markov Models (HMM) [8]. These models are totally independent from hardware technologies, as they only consider simple actions. This two-phase approach also reduces the complexity of models, increasing the precision and success rate in the recognition process.

The rest of the paper is organized as follows: Section 2 describes the state of the art on pattern recognition for human activities; Section 3 describes the proposed solution, including the two defined phases; Section 4 presents an experimental validation using a real scenario and final users; and Section 5 concludes the paper.

2 State of the art on pattern recognition

Many different pattern recognition techniques for human activities have been reported. However, most common proposal many be classified into five basic categories [9]: (i) Hidden Markov Models; (ii) the Skip Chain Conditional Random Field; (iii) Emerging Patterns; (iv) the Conditional Random Field; and (v) Bayesian classifiers.

In fact, most authors propose the use of Hidden Markov Models (HMM) to model human activities. HMM allow modeling actions as Markov chains [10][11]. Basically, HMM generate hidden states from observable data. In particular, the final objective of this technique is to construct the sequence of hidden states that fits with a certain data sequence. To finally define the whole model, HMM must deduct from data the model parameters in a reliable manner. Figure 1 shows a schematic representation about how HMM work. When human activities are recognized, the actions composing the activities are the hidden states, and sensor outputs are data under study. HMM, besides, allow the use of training techniques considering prior knowledge about the model. This training is sometimes essential to "induce" all possible data sequences required to calculate the HMM. Finally, it is very important to note that simple isolated HMM can be combined to create larger and more complex models.

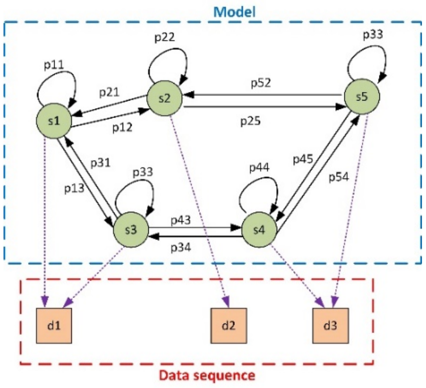

Fig. 1. Graphical representation of an HMM

HMM, nevertheless, are useless to model certain concurrent activities, so other authors have reported a new technique named, Conditional Random Field (CRF). CRF are employed to model those activities that present concurrent actions or, in general, multiple interacting actions [12][13]. Besides, HMM requires a great effort on training to discover all possible hidden states. To solve these problems, Conditional Random Field (CRF) employs conditional probabilities instead of joint probability distributions. In that way, activities whose actions are developed in any order may be easily modeled. Contrary to chains in HMM, CRF employs acyclic graphs, and enables the integration conditional hidden states (states that depend on past and/or future observations).

CRF, on the other hand, are still useless to model certain behaviors, so some proposals generalize this concept and propose the Skip Chain Conditional Random Field (SCCRF). SCCRF is a pattern recognition technique, more general than CRF, that enables modeling activities that are not sequence of actions in nature [14]. This technique tries to capture long-range (skip chain) dependencies; and may be understood as the product of different linear chains. However, calculating this product is quite heavy and complicated, so this technique is usually too computationally expensive to be implemented in small embedded systems.

Other proposals employ higher level description techniques such as Emerging Patterns (EP). For most authors, EP is a technique describing activities as vectors of parameters and their corresponding values (location, object, etc.) [15]. Using distances between vectors it is possible to calculate and recognize actions developed by people. Finally, other authors have successfully employed secondary techniques such as Bayesian classifiers [16], which identify activities making a correspondence between human activities and the most probable sensor outputs while these actions are performed, considering all sensors are independent. Decision trees [17], HMM extensions [18], and other similar technologies have been also studied in the literature, although these proposals are sparse.

Among all described technologies, HMM is not the most powerful one. However, it fits perfectly with Industry 4.0, where actions are very complex but very structured and ordered (according to company protocols, efficiency policies, etc.). Besides, fast feedback is required (sometimes even real-time), in order to guarantee human-driven processes operate correctly before a global critical fail occurs. Thus, computationally expensive solutions are not a valid approach, and we are selecting HMM as main base technology. In order to preserve its lightweight character and, at the same time, being able to model complex activities, we introduce a two-phase recognition scheme which enable dividing complex actions in two simpler steps.

3 A two-phase pattern recognition algorithm

In order to (i) make independent the pattern recognition process from employed hardware devices to capture information, (ii) enable the recognition of complex actions, and (iii) preserve the lightweight character of the selected models, the proposed solution presents an architecture with three different layers (see Figure 2).

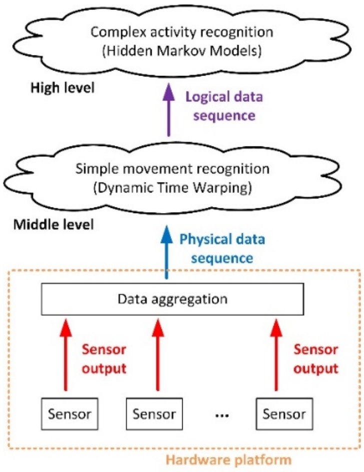

Fig. 2. Architecture of the proposed pattern recognition solution

The lowest layer includes the hardware platform. Monitoring devices such as accelerometers, smartphones, infrared sensors, RFID tags, etc., are deployed to capture information about the people behavior. The outputs of these devices create physical data sequences whose format, dynamic range, etc., are totally dependent on the selected hardware technologies.

These physical data sequences are then processed in the middle layer using DTW techniques. As result, for each physical data sequence, a simple movement or action is recognized. These simple actions are represented using a binary data format to make the solution as lightweight as possible. Software at this level must be modified each time the hardware platform is updated, but DTW technologies do not require a heavy actualization process, and refreshing the pattern repository is enough to configure the algorithm at this level.

Recognized simple movements, then, are grouped to create logical data sequences. These sequences feed a high-level pattern recognition system based on Hidden Markov Models. At this level, software components require a heavy training process, but middle layer makes totally independent the hardware platform and high-level models. Thus, any change in the hardware platform does not enforce an actualization in the HMM, which would be extremely computationally costly. By the analysis of the sequence of simple movements, complex actions are recognized.

Next subsection describes both proposed pattern recognition phases in detail.

3.1 Simple movement recognition: Dynamic Time Warping

In order to recognize simple gestures or movements, a Dynamic Time Warping solution is selected. DTW technologies fulfill the requirements of middle-level software components as they adapt to the underlying hardware platform's characteristics very easily and are quite fast and efficient (so small embedded devices may implement them).

In our solution, human behavior is monitored through a family of sensors $\mathcal{S}$, containing N_s components (1).

$$\mathcal{S} = \{s_i, i = 1, \dots, N_s\} \tag{1}$$

The outputs of these sensors are periodically sampled each T_s seconds; obtaining for each time instant, t, a vector of N_s values (each value from each sensor). This vector Y_t is called "a multidimensional sample". (2)

$$Y_t = \{y_t^i, i = 1, \dots, N_s\} \tag{2}$$

Then, a simple movement Y will a have a duration of T_m seconds and will be described by the temporal sequence of N_m multidimensional samples collected during this time (3). In order to later recognize movements, a pattern repository $\mathcal{R}$ is created containing the corresponding temporal sequences for each one of the K simple actions to be recognized (4).

$$Y = \{Y_t, t = 1, \dots, T_m\} = \{Y^i, i = 1, \dots, N_m\} \tag{3}$$

$$\mathcal{R} = \{R_i, i = 1, \dots, K\} \tag{4}$$

In general, people perform movements in similar but different manners. Thus, transitions may be slower or faster, some elemental actions may be added or removed, etc. Therefore, given a sequence X with N_x samples, representing a movement to be recognized, it must be located the pattern $R_i \in \mathcal{R}$ closer to X; so R_i is recognized as the performed action. To do that it is defined a distance function (5). This distance function may be applied to calculate a cost matrix, required as samples usually don't have the same length neither they are aligned (6).

$$d: \mathcal{F} \times \mathcal{F} \longrightarrow \mathbb{R}^+ , \qquad X^i , r_j^i \in \mathcal{F} \tag{5}$$

$$C \in \mathbb{R}^{N_x \times N_m} \;\; C(n,m) = d(X^n, R_j^m) \tag{6}$$

In positional sensors (accelerometers, infrared devices, etc.) distance function is applied directly to the sensors' outputs (contrary to, for example, microphones whose outputs must be evaluated in the power domain). Although other distance functions can be employed (the symmetric Kullback–Leibler divergence or the Manhattan distance), for this first work we are employing the standard Euclidian distance (7)

$$d\left(X^n, R_j^m\right) = \sqrt{\sum_{i=1}^{N_S} \left(x_i^n - r_i^{m,j}\right)^2} \tag{7}$$

Then, it is defined a warping path $p = (p_1, p_2, \dots, p_L)$ as a sequence of pairs (n_ℓ, m_ℓ) with $(n_\ell, m_\ell) \in [1, N_x] \times [1, N_m]$ and $\ell \in [1, L]$, satisfying three conditions: (i) the boundary condition, i.e. $p_1 = [1,1]$ and $p_L = [N_x, N_m]$; (ii) the monotonicity condition, i.e. $n_1 \leq n_2 \leq \dots \leq n_L$ and $m_1 \leq m_2 \leq \dots \leq m_L$; and (iii) the step size condition, i.e. $p_\ell - p_{\ell-1} \in \{(1,0), (0,1), (1,1)\}$ with $\ell \in [1, L-1]$.

Then, the total cost of a warping path p_i is calculated adding all the partial costs or distances (8). With all this, the distance between two data sequences R_i and X is defined as the cost (distance) of the optimum warping path p^* (9).

$$d_{p_i}\left(X, R_j\right) = \sum_{\ell=1}^{L} d(X^{n_\ell}, R_j^{m_\ell}) \tag{8}$$

$$d_{DTW}\left(X, R_j\right) = d_{p^*}\left(X, R_j\right) = min\left\{d_{p_i}\left(X, R_j\right), being\ p_i\ a\ warping\ path\right\} \tag{9}$$

Finally, the simple movement recognized from the data sequence X is that whose pattern R_i has the smallest distance (is the closest) to X. The use of this definition is tolerant to speed variations in the movement execution, to the introduction of new micro-gestures, etc. Besides, as can be seen, when a different hardware technology is deployed, it is enough to update the patter repository $\mathcal{R}$ to reconfigure the entire pattern recognition solution (as no training is required).

3.2 Complex action recognition: Hidden Markov Models

Previously proposed mechanism is very useful to recognize simple actions, but complex activities involve a huge number of variables and require much more time. Thus, DTW tend to become imprecise, and probabilistic models are required. Among all existing models, HMM is the most adequate for industrial scenarios and human-driven processes.

From the previous phase, the universe of possible simple movements to be recognized is $\mathcal{M} = \{m_i , i = 1, \dots, K\}$. Besides, it is defined a state universe $\mathcal{U} = \{u_i , i = 1, \dots, Q\}$, describing all the states that people may cross while performing any of the actions under study.

Then, a set of observations $\mathcal{O} = \{o_i\,, i = 1, \dots, Z_o\}$ (simple movements recognized in the previous phase) is also considered, as well as the sequence of states $V = \{o_i\,, i = 1, \dots, Z_v\}$ describing the action to by modeled by HMM. In this initial case, we are assuming each observation corresponds to a new state, so $Z_v = Z_o$ Then, three matrices are calculated: (i) the transitory matrix A (10) describing the probability of state u_j following state u_i; (ii) the observation matrix (11) describing the probability of observation o_i caused by state u_j independently from k; and (iii) the initial probability matrix (12).

$$A = [a_{i,j}] \quad a_{i,j} = P(v_k = u_j \mid v_{k-1} = u_i) \tag{10}$$

$$B = [b_j(o_i)] \quad b_j(o_i) = P(x_k = o_i \mid v_k = u_j) \tag{11}$$

$$\Pi = [\pi_i] \quad \pi_i = P(v_1 = u_i) \tag{12}$$

Then, the HMM for each complex activity λ_i to be recognized is described by these previous three elements (13).

$$\lambda_i = \{A_i\,, B_i\,, \Pi_\mathrm{i}\,\} \tag{13}$$

Two assumptions are, besides, made: (i) the Markov assumption (14) showing that any state is only dependent on the previous one; and (ii) the independency assumption (15) stating that any observation sequence depends only on the present state not on previous states or observations.

$$P(v_k \mid v_1, \dots, v_{k-1}) = P(v_k \mid v_{k-1}) \tag{14}$$

$$P(o_k \mid o_1, \dots, o_{k-1}, v_1, \dots, v_k) = P(o_k \mid v_k) \tag{15}$$

To evaluate the model and recognize the activity being performed by users, in this paper we are using a traditional approach (16). Although forward algorithms have been proved to be more efficient, for this initial work we are directly implementing the evaluation expression in its traditional form.

$$\begin{aligned} P(\mathcal{O} \mid \lambda) &= \sum_V P(\mathcal{O} \mid V, \lambda)\, P(V \mid \lambda) = \\ &= \sum_V \left(\prod_{i=1}^{Z_o} P(o_i \mid v_i, \lambda) \right) \left(\pi_{v1} \cdot a_{v1v2} \cdot \ldots \cdot a_{v_{zv-1}v_{zv}} \right) = \\ &= \sum_{v1, v2, .., v_{zv}} \pi_{v1} \cdot b_{v1}(o_1) \cdot a_{v1v2} \cdot b_{v2}(o_2) \cdot \ldots \cdot a_{v_{zv-1}v_{zv}} \cdot b_{vzv}(o_{zo}) \end{aligned} \tag{16}$$

The learning process was also implemented in its simplest way. Statistical definitions were employed for transitory matrix, observation matrix and initial probability matrix. In particular, the Laplace definition of probability was employed to estimate these three matrices from statistics about the activities under study (17-19). The operator $count(\cdot)$ indicates the number of times an event occurs.

$$a_{i,j} = P(u_j \mid u_i) = \frac{count(u_j\, follows\ u_i)}{count(u_j\,)} \tag{17}$$

$$b_j(o_i) = P(o_i \mid u_j) = \frac{count(o_i\ is\ observed\ in\ the\ state\ u_j)}{count(u_j\,)} \tag{18}$$

$$\pi_i = P(v_1 = u_i) = \frac{count(v_1 = u_i)}{count(v_1)} \tag{19}$$

4 Experimental validation: implementation and results

In order to evaluate the performance of the proposed solution, an experimental validation was designed and carried out. An industrial scenario was emulated in some large rooms in Universidad Politécnica de Madrid. The scenario represented a traditional company manufacturing handmade products. In particular, a small PCB (printed circuit boards) manufacturer was emulated.

In order to capture information about people behavior, participants were provided with a cybernetic glove, including accelerometers and a RFID reader [19]. Objects around the scenarios were identified with an RFID tag, so the proposed hardware platform may identify the hand position (gesture) and the objects people interact with.

A list of twelve different complex activities where defined and recognized using the proposed technology. Table 1 describes the twelve defined activities, including a brief description about them.

Table 1. Complex activities' description

Activity	Description
Draw the circuit's paths	The circuit to be printed is designed using a specific software PC program.
Print the circuit design using a plotter	Using plastic sheet and a special printer called plotter, the circuit design is printed.
Clean the copper-sided laminate of boards	Using a special product all dust and particles are removed from the cooper-sided laminate.
Copy the circuit design on cooper boards	The circuit design in the plastic sheet is copied into the cooper laminate using a blast of UV light.
Immerse the boards in the acid pool	To remove all unwanted cooper, the printed board is immersed in an acid bath.
Wash off the cooper using a dissolvent bath	After the acid bath, the remaining cooper surface is washed off in a dissolvent bath.
Layer alignment	PCB are composed of several layer; stacked and aligned during this phase.
Optical inspection	Using a laser, the layer alignment is checked.
Join outer layers with the substrate	Using an epoxy glue, the final and outer layer in the board are joined.
Bond the board	The bonding occurs on a heavy steel table with metal clamps.
Drill the required holes	Holes for components, etc., are bored into the stack board.
Plating	In an oven, the board is finished.

Eighteen people (18) were involved in the experiment. People were requested to perform the activities in a random number. The real order, as well as the order the activities are recognized were stored by a supervisory software process. The global success rate for the whole solution was evaluated, identifying (moreover), the same rate for each one of the existing phases.

In order to evaluate the obtained improvement in comparison to existing similar solutions, the same physical data sequences were employed to feed a standard pattern recognition solution based only on HMM. Using statistical data processing software, some relevant results are extracted.

Figure 4 represents the mean success rate for three cases: the global solution, the first phase (DTW), and the second phase (HMM). Besides, the success rate for the traditional HMM-based approach is also included. As can be seen, the proposed technology is, globally, around 9% better than traditional pattern recognition techniques based on HMM exclusively. Besides, first phase (based on DTW) is around 20% worse than the second phase (HMM) which is meaningful as Dynamic Time Warping techniques are weaker by default.

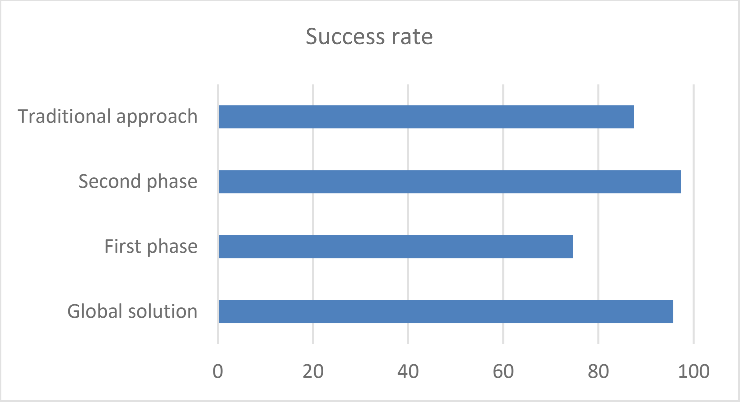

Fig. 4. Mean success rate for the proposed solution

5 Conclusions and future works

In this paper we present a new pattern recognition algorithm to integrate people in Industry 4.0 systems and human-driven processes. The algorithm defines complex activities as compositions of simple movements. Complex activities are recognized using Hidden Markov Models, and simple movements are recognized using Dynamic Time Warping. In order to enable the implementation of this algorithm in small embedded devices, lightweight configurations are selected. An experimental validation is also carried out, and results show a global improvement in the success rate around 9%.

Future works will consider most complex methodologies for data processing, and comparison for different configurations of the proposed algorithm will be evaluated. Besides, the proposal will be analyzed in different scenarios.

Acknowledgments. The research leading to these results has received funding from the Ministry of Economy and Competitiveness through SEMOLA (TEC2015-68284-R) project and the Ministry of Science, Innovation and Universities through VACADENA (RTC-2017-6031-2) project.

References

1. Bordel, B., Alcarria, R., Sánchez-de-Rivera, D., & Robles, T. (2017, November). Protecting industry 4.0 systems against the malicious effects of cyber-physical attacks. In International Conference on Ubiquitous Computing and Ambient Intelligence (pp. 161-171). Springer, Cham.
2. Bordel, B., Alcarria, R., Robles, T., & Martín, D. (2017). Cyber–physical systems: Extending pervasive sensing from control theory to the Internet of Things. Pervasive and mobile computing, 40, 156-184.
3. Neff, W. (2017). Work and human behavior. Routledge.
4. Bordel, B., Alcarria, R., Martín, D., Robles, T., & de Rivera, D. S. (2017). Self-configuration in humanized cyber-physical systems. Journal of Ambient Intelligence and Humanized Computing, 8(4), 485-496.
5. Bordel, B., de Rivera, D. S., Sánchez-Picot, Á., & Robles, T. (2016). Physical processes control in industry 4.0-based systems: A focus on cyber-physical systems. In Ubiquitous Computing and Ambient Intelligence (pp. 257-262). Springer, Cham.
6. Pal, S. K., & Wang, P. P. (2017). Genetic algorithms for pattern recognition. CRC press.
7. Müller, M. (2007). Dynamic time warping. Information retrieval for music and motion, 69-84.
8. Eddy, S. R. (1996). Hidden markov models. Current opinion in structural biology, 6(3), 361-365.
9. Kim, E., Helal, S., & Cook, D. (2010). Human activity recognition and pattern discovery. IEEE Pervasive Computing/IEEE Computer Society [and] IEEE Communications Society, 9(1), 48.
10. Li, Z., Wei, Z., Yue, Y., Wang, H., Jia, W., Burke, L. E., ... & Sun, M. (2015). An adaptive hidden markov model for activity recognition based on a wearable multi-sensor device. Journal of medical systems, 39(5), 57.
11. Ordonez, F. J., Englebienne, G., De Toledo, P., Van Kasteren, T., Sanchis, A., & Krose, B. (2014). In-home activity recognition: Bayesian inference for hidden Markov models. IEEE Pervasive Computing, 13(3), 67-75.
12. Zhan, K., Faux, S., & Ramos, F. (2015). Multi-scale conditional random fields for first-person activity recognition on elders and disabled patients. Pervasive and Mobile Computing, 16, 251-267.
13. Liu, A. A., Nie, W. Z., Su, Y. T., Ma, L., Hao, T., & Yang, Z. X. (2015). Coupled hidden conditional random fields for RGB-D human action recognition. Signal Processing, 112, 74-82.
14. Liu, J., Huang, M., & Zhu, X. (2010, July). Recognizing biomedical named entities using skip-chain conditional random fields. In Proceedings of the 2010 Workshop on Biomedical Natural Language Processing (pp. 10-18). Association for Computational Linguistics.
15. Gu, T., Wu, Z., Tao, X., Pung, H. K., & Lu, J. (2009, March). epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on (pp. 1-9). IEEE.
16. Hu, B. G. (2014). What are the differences between Bayesian classifiers and mutual-information classifiers?. IEEE Trans. Neural Netw. Learning Syst., 25(2), 249-264.
17. Wang, X., Liu, X., Pedrycz, W., & Zhang, L. (2015). Fuzzy rule based decision trees. Pattern Recognition, 48(1), 50-59.
18. Davis, M. H. (2018). Markov models & optimization. Routledge.
19. Bordel Sánchez, B., Alcarria, R., Martín, D., & Robles, T. (2015). TF4SM: a framework for developing traceability solutions in small manufacturing companies. Sensors, 15(11), 29478-29510.