UNIVERSITÁ DI PISA
DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

Computational Systems for Spatio-Temporal Pattern Analysis Based on Stigmergy

DOCTORAL THESIS

Author
Alessandro Lazzeri

Tutor (s)
Prof.ssa Nicoletta De Francesco
Prof.ssa Gigliola Vaglini
Ing. Mario G.C.A. Cimino

Reviewer (s)
Prof. Witold Pedrycz
Prof. Bruno Lepri

The Coordinator of the PhD Program
Prof. Marco Luise

Pisa, Sept. 2016

XXIX Cycle

Acknowledgments

First and foremost, I would like to thank my tutors, Prof. De Francesco, Prof. Vaglini, and Prof. Cimino, for their precious help and guidance during these three challenging and valuable years. I am grateful to Prof. Pedrycz, for the kind support within this research during the abroad period.

I would like to thank the two reviewers, Prof. Lepri and Prof. Pedrycz, for the useful comments, which allowed the improvement of this thesis and suggested new directions for the research.

My deepest gratitude goes to my family, my sweet girlfriend Sabrina and my great friend Giorgio, because making them proud is my greatest strength.

Finally yet importantly, I would like to thank all my friends, my colleagues, the students, and everyone which has spent just a few minutes supporting, suggesting, or simply listening what I did during my Ph.D.

Anyway, if you are not in the above list, thank you for reading this.

Summary

LARGE amount of data are currently produced by an incredible diversity of applications. In order to provide relevant information to the users is necessary to identify important patterns and recognize these patterns when they occur again. In general, at the core of the analysis of human centric data is the construction of two possible types of model: (i) knowledge based models, explicitly designed at the business level in terms of logical or mathematical rules, determined by a domain expert; (ii) data-driven models, i.e., systems that can learn from prototypical data via machine learning or statistical algorithm. Nevertheless, modeling and reusing application contexts remains a difficult task. An important lesson learned is that the algorithms performing the parametric data aggregation must use a limited number of states, be highly adaptable and handle variability. The data-driven approach discussed in this work takes inspiration from the emergent paradigm, in which context information is augmented with locally encapsulated structure and behavior. Emergent paradigms are based on the principle of self-organization of data, which means that a functional structure appears and stays spontaneous at runtime when local dynamism in data occurs.

More precisely, we adopt marker-based stigmergy, i.e., a biologically inspired mechanism performing scalar and temporal information aggregation. In biology, stigmergy is an indirect communication mechanism, while in computer science stigmergy can be employed as a dynamic, agglomerative, computing paradigm because it embodies the time domain. Stigmergy focuses on the low level processing, where individual samples are augmented with micro-structure and micro-behavior, to enable self-aggregation in the environment.

We present the use of stigmergy computation in different applications with spatio-temporal data, showing the feasibility and the capability of the approach to be adopted in heterogeneous fields. To support the user in the parameterization process, we designed an adaptation mechanism based on a bio-inspired evolutionary algorithm. At the final stage of the architecture development, we compared the proposed approach with state-of-art techniques on a classification task. Experimental studies completed for real-world data show that results are promising and consistent with human analysis.

Summary of PhD Achievements

The research activity carried out during the 3-years Ph.D. program has been focused on the development and the application of a computational model based on stigmergy for pattern analysis.

The first year of the program focuses on two main activities:

- an initial examination of the scientific literature in the field of swarm intelligence and in particular the application of stigmergy in computer science;
- the design and construction of a model for the pattern analysis in smart environments, such as (a) indoor elderly monitoring (in collaboration with ISTI of the Italian National Research Council, see pub. 1) and (b) vendor rating problem in collaborative networks (in collaboration with DESTEC, during the Tuscan Region Project PMI 3.0, see pub. 2).

Spatio-temporal pattern analysis is usually addressed with explicit and probabilistic models of the application domain constructed on the knowledge of the context. Instead, the developed model is bio-inspired and based on the emergent paradigm. In particular, it has four processes:

(i) The Marking transforms each sample of input data in a *mark*. The *mark* is a representation of the sample in the *stigmergic space*. The attributes of the mark are the sample, the mark intensity and the mark extension. The mark also evaporates over time slowly disappearing from the stigmergic space.

(ii) The Trailing aggregates marks in a structure named *trail*. The marks can interact if sufficiently close to each other, in other words if the marks are in proximity in time and space they increase in intensity. The trail has a temporal dynamics, specifically if not reinforced by subsequent marks tends to evaporate over time, reducing the intensity and disappearing from stigmergic space.

(iii) The Prototyping, the trail is brought back to a form named the *prototype*.

(iv) Finally, the Similarity compares pairs of prototypes at any given time through a similarity measure. With this mechanism, it is possible to determine set of interesting prototypes, e.g. processing sets of classified input data; and retrieve similar pattern in unclassified data.

The second year, an algorithm based on stigmergy and flocking behavior has been developed for the coordination of swarms of small aerial drones in target search (see pub. 8). Here, the drone releases stigmergic marks in a computational environment, which are locally used by other drones to coordinate the exploration. The mark function is to attract drones to explore interesting areas of the map. The flock behavior helps maintain cohesive formations of drones and thus facilitates the local perception of the marks in the computational environment.
Then, both the model for pattern analysis and the swarm controlling algorithm have been enriched with an adaptation module to support the user in the tuning of the parameters. The adaptation module has been tested in the following applications:

- the two application (a) and (b) developed during the first year (see pub. 6)
- the detection of traffic congestion in urban roads (in collaboration with the IIT of the Italian National Research Council of Pisa, during Project SMARTY, see pub. 7);
- the algorithm for the coordination of the swarms of drones (see pub. 9);
- the analysis of time series regarding technological indicators of European Regions (in collaboration with DESTEC, during Project MIT-UNIPI, see pub. 10);

The third year, the multi-layer architecture based on stigmergy for the analysis of temporal patterns has been developed. Previous applications manifested the need of performing multiple comparisons with different patterns and then to produce a concise evaluation. The Multi-Layer Stigmergic architecture is based on the concept of Stigmergic Receptive Field, which adopts stigmergy to compare couple of patterns. Stigmergic Receptive Fields can be organized into a multilayer connectionist system, and adapted to contextual behavior by means of the adaptation module.
The Multi-Layer architecture has been developed during the 6-months abroad period at the University of Alberta (Edmonton, Canada) under the supervision of Prof. Witold Pedrycz. The architecture has been applied for detecting topic discussions in microblogs and compared with state of the art techniques.

List of publications

International Journals

1. Barsocchi, P., Cimino, M.G.C.A., Ferro, Lazzeri, A., Palumbo, F., and Vaglini, G. (2015, October). Monitoring elderly behavior via indoor position-based stigmergy. *Pervasive and Mobile Computing*. (Vol. 23, pp. 26-42). Elsevier.
2. Appio, F., Cimino, M.G.C.A., Lazzeri, A., Martini, A., and Vaglini, G., Fostering Distributed Business Logic in Open Collaborative Networks: an integrated approach based on semantic and swarm coordination, *Information and Systems Frontiers*, in press.
3. Cimino, M.G.C.A., Lazzeri, A., Pedrycz, W., and Vaglini, G., Using Stigmergy to Identify Event-Specific Social Discussion Topics, submitted to international journal.
4. Alfeo, A.L., Cimino, M.G.C.A., Lazzeri, A., and Vaglini, G., Detecting urban road congestion via parametric adaptation of position-based stigmergy, submitted to international journal.
5. Alfeo, A.L., Cimino, M.G.C.A., Lazzeri, A., Lega M., and Vaglini, G., Swarm coordination of small UAVs for target search with imperfect sensors, submitted to international journal.

International Conferences/Workshops with Peer Review

6. Cimino, M.G.C.A., Lazzeri, A., and Vaglini, G., (2015, June). Improving the Analysis of Context-Aware Information via Marker-Based, in Proc. Springer LNAI *International Conference on Artificial Intelligence and Soft Computing* (ICAISC 2015), pp. 1-12, Zakopane, Poland, 2015, Springer.
7. Cimino, M.G.C.A., Lazzeri, A., and Vaglini, G., (2015, July). Enabling swarm aggregation of position data via adaptive stigmergy: a case study in urban traffic flows, in Proc. IEEE *The Sixth International Conference on Information, Intelligence, Systems and Applications* (IISA 2015), pp. 1-6, Corfu, Greece, IEEE.
8. Cimino, M.G.C.A., Lazzeri, A., and Vaglini, G., (2015, July). Combining

stigmergic and flocking behaviors to coordinate swarms of drones performing target search, in Proc. IEEE *The Sixth International Conference on Information, Intelligence, Systems and Applications* (IISA 2015), pp. 1-6, Corfu, Greece, IEEE.

9. Cimino, M.G.C.A., Lazzeri, A., and Vaglini, G., (2016). Using Differential Evolution to improve pheromone-based coordination of swarms of drones for collaborative target detection, in Proc. INSTICC *The 5th International Conference on Pattern Recognition Applications and Methods*, (ICPRAM 2016), pp. 605-610, Rome, Italy, Springer.

10. Alfeo, L., Appio, F., Cimino, M.G.C.A., Lazzeri, A., Martini, A., and Vaglini, G., (2016). An adaptive stigmergy-based system for evaluating technological indicator dynamics in the context of smart specialization, in Proc. INSTICC *The 5th International Conference on Pattern Recognition Applications and Methods*, (ICPRAM 2016), pp. 497-502, Rome, Italy, Springer.

List of Figure

List of Tables

Contents

CHAPTER 1

Introduction

1.1 Background and Motivation

Large amount of data are currently produced by an incredible diversity of fields ranging from economy, medical surveillance, climate forecasting, biology, road traffic management, genetics, microblogs, or robotics. One of the main problems is to identify important patterns and recognize these patterns when they occur again. For example: the reduction of the speed of multiple vehicles can identify a traffic congestion; the changing of the movement patterns of a person at home can help identifying a degenerative disease; enterprises have to track their key performance indicators to assess profitability of the business; governments direct investments according to the evolution of social and economic indicators; discussions on microblogs evolve and change from a topic to another as people continuously share their opinions about ongoing events. When facing the analysis of such data, there are numerous facets of complexity due to the high dimensionality, and in combination with the difficulty to define an adequate similarity measure based on human perception [1].

In general, at the core of aggregation of human centric data is the construction of two possible types of model: (i) knowledge based models, explicitly designed at the business level in terms of logical or mathematical rules, determined by a domain expert; (ii) data-driven models are systems that can learn from prototypical data via machine learning or statistical algorithm. Nevertheless, modeling and reusing application contexts remains a difficult task. An important lesson learned is that the algorithms performing the parametric data aggregation must use a limited number of states, be highly adaptable and handle variability [2] [3].

Generally speaking, knowledge-based models belong to the cognitivist paradigm [4]. In this paradigm, the system is a descriptive product of a human designer, whose knowledge has to be explicitly formulated for a representational system of symbolic information processing. It is well known that knowledge-based systems are highly context-dependent, neither scalable nor manageable. With respect to knowledge-based models, data-driven models are more robust in the face of noisy and unexpected inputs, allowing broader coverage and being more adaptive. The data-driven approach discussed in this work takes inspiration from the emergent paradigm [4], in which context information is augmented with locally encapsulated structure and behavior.

Emergent paradigms are based on the principle of self-organization of data, which means that a functional structure appears and stays spontaneous at runtime when local dynamism in data occurs [5] [6].
The objective of the research presented in this thesis is to assess the use of stigmergy in supporting the development of computational systems for spatio-temporal pattern analysis. More in details, the objective is to take inspiration from social insects to provide data with a dynamical structure which enables local interaction and aggregation capability. This work adopts marker-based *stigmergy* [7], which is a biologically inspired mechanism performing scalar and temporal information aggregation. Stigmergy [8] is an indirect communication mechanism that occurs in biological systems [9]. In computer science, marker-based stigmergy can be employed as a dynamic, agglomerative, computing paradigm because it embodies the time domain [6]. Stigmergy focuses on the low level processing, where individual samples are augmented with micro-structure and micro-behavior, to enable self-aggregation in the environment. Here, we employ stigmergy as a mean for the aggregation of (i) data produced by a group of humans or artificial agents and (ii) multiple data produced by a single individual over time. The core principle behind stigmergy is that an agent releases *marks* in the environment while performing actions. The *marks* interact in the environment forming an aggregated structure named *trail*. The *trail* stays in the environment for a certain period while evaporating overtime and progressively disappearing. The *trail* influences the agent as a condition in the choice of which action to perform. According to Heylighen [10], it is possible to recognize some level of stigmergy in almost all evolved process that require some coordination between sequences of actions. The subsequent actions are actually stimulated by the marks left by previous actions, which remains observable and manageable for a certain period in the environment. The *trail* acts like a registry and a map, indicating which actions previous agents have performed and thus indicating which action has to be performed in the immediate future. Moreover, the information contained in the trail is locally shared in the environment, thus implying two important advantages: (i) there is no need for direct forms of communication between agents and (ii) it is possible to perform complex sequences of actions without keeping consistent memory resources occupied or adopting planning techniques [10]. From this perspective, the analysis of the current state of the trail in the environment is the mean to keep trace of the history of the system. Moreover, it is possible to define relevant configuration of the system and link them to their corresponding trails. The comparison of couple of trails permits to identify the state of the system. This process is managed by a computational unit named *Stigmergic Receptive Field* (SRF). With this approach, we avoid a definite and direct modelling of the application domain; rather the single data sample embodies a domain-agnostic micro-behavior, which provides a context-independent aggregation mechanism. Second, to achieve new levels of abstraction without encoding explicit knowledge, the principles of connectionism are also applied. More specifically, SRFs are organized into a multilayer connectionist system, and adapted to contextual behavior by means of the DE algorithm. Thus, the novelty of the undertaken study relates to the structure of an SRF and a way in which such receptive fields are being formed and adapted [56] [141]. The concept of receptive field is the most prominent and ubiquitous computational mechanism employed by biological information processing systems [124]. In our approach, it relates to an architectural style consisting of a collection of general purpose local models (archetypes) that detects a micro-behavior of the time series. The use of SRFs can be proposed as a more general and effective way of designing micro-pattern

detection. Moreover, SRF can be used in a multilayered architecture, thus providing further levels of processing to realize a macro analysis.

As example of stigmergy in biology, the colony of ants uses stigmergy in the foraging task: the ants move randomly in the environment, an ant finds some source of food and returns to the colony releasing a pheromone in the environment, and finally the other ants follow the pheromone toward the source of food. In general, this simple mechanism permits the exploration of an area and the detection of patterns of targets in the environment. Recently, the advances in sensing technology encouraged the use of unmanned vehicles for military intelligence, reconnaissance, surveillance, traffic monitoring, forest fire localization, scientific surveys in dangerous conditions, border and harbor patrol, search and rescue, wildlife tracking, undersea research, and so on [132]. Indeed, unmanned vehicles offer more potential, because they can perform tasks in highly inhospitable environments, where access to humans is limited or impossible. Moreover, in tasks such as search and rescue, to explore every available location on the area is also an inappropriate strategy. A more effective approach is to achieve a quick "survey" of the area, identifying key locations as quick as possible, and to better investigate only key locations that provided some circumstantial evidence. To complete this task with a single vehicle, both structure and control logic should be highly costly in terms of design, construction and maintenance. Moreover, a unique vehicle is vulnerable, because a single hardware or software fault may affect the whole system, and it is difficult to predict. Hence, a number of considerations support the use of a swarm-based approach for the analysis of patterns. An important requirement is to avoid centralized control approaches, which frequently lead to exponential increases in communication bandwidth requirements and software complexity [133]. To solve problems cooperatively while maintaining scalability, application designers are investigating swarm intelligent methodologies. The main inspiration for swarm comes from the observation of social animals, such as ants, bees, birds, and fishes, exhibiting a sort of collective intelligence, which appears to achieve complex goal through simple rules and local interactions [134].
This thesis presents an algorithm for the coordination of unmanned vehicles for pattern analysis. The stigmergy, inspired by insects, and the flocking behavior, inspired by birds, are exploited to enable self-coordination in the navigation of unstructured environments. Such environments typically contain a number of obstacles and a certain number of targets placed according to different patterns. More specifically, stigmergy implies the use of an attractive digital pheromone. The pheromone coordinates the vehicles in the exploration of area with potential undiscovered targets. When a vehicle detects a new piece of target, it releases the attractive pheromone. Other vehicles in the neighborhood can sense and follow the pheromone gradient to cooperate in detecting the pattern of targets. The flocking behavior mimics the flying formation of birds and schools of fishes: vehicles travel in a compact formation, with same heading, and keeping a safety distance. To introduce the swarm algorithm, we present a case study with unmanned small drones.

The proposed mechanisms, the SRF and the swarm algorithm, work if structural parameters, such as the mark attributes, are correctly tuned. For example, a very large and persistent mark may cause growing trails with no stationary level, because of a too dominant and long-term memory effect. A very small and volatile mark may cause poor mark aggregation. For tuning such parameters we adopt an adaptation mechanism based

on Biological Inspired Algorithm (BIA). BIAs implement search mechanisms applicable to problems that cannot be efficiently solved using exact and analytical techniques [11]. After the literature review, Differential Evolution [12] (DE) seems to be a very promising algorithm and it has been selected to tackle the adaptation of the stigmergy parameters. DE is a stochastic optimization algorithm based on a population of agents, suitable for numerical and multi-modal optimization problems. In the last decades, DE has been applied to many applications and research areas, including parameterization of stigmergy-based systems [6]. DE exhibits excellent performance both in unimodal, multimodal, separable, and non-separable problems, when compared with other similar algorithms, such as genetic algorithms and particle swarm optimization.

In this work, the principle of stigmergy is applied in six different applications: analysis of key performance indicators in collaborative enterprises, indoor monitoring of elderly people, detection of traffic congestions in urban roads, evaluation of technological indicators in European Regions, detection of target in open environment with swarm of aerial drones, and identification of social discussions in microblogs.

1.2 Research aim and Contribution

The objective of the research presented in this thesis is to assess the use of stigmergy in supporting the development of computational systems for spatio-temporal pattern analysis. More in details, the objective is to take inspiration from social insects to provide data with a dynamical structure which enables local interaction and aggregation capability.
Therefore, the main research aims are to:

- Investigate the problems related to the analysis of patterns when facing spatio-temporal dynamics.
- Study the implications of using the marker-based stigmergy to perform data manipulation, aggregation, and comparison.
- Develop models based on the marker-based stigmergy for spatio-temporal pattern analysis.
- Use the model in different contexts and application problems to amend the model with multiuse and flexible modules, and develop a full architecture that supports the use of stigmergy for pattern analysis.
- Evaluate the effectiveness of using such an approach by applying the architecture to real-world applications, assess the validity of the model, and compare with state of the art techniques.

The methods used in conducting the research presented use a rigorous survey of the literature both in the application of stigmergy in pattern analysis and in the fields of application of the proposed architecture. In particular, the principle of stigmergy is applied in six different applications: analysis of key performance indicators in collaborative enterprises, indoor monitoring of elderly people, detection of traffic congestions in urban roads, evaluation of technological indicators in European Regions, detection of target in open environment with swarm of aerial drones, and identification of social discussions in microblogs.

The main contributions of the thesis can be summarized as follows:

- An overview of marker-based stigmergy with respect to the applications in pattern analysis problems. While stigmergy principles have been exploited for a wide range of specific computational problems, this paradigm has not been extensively applied to pattern analysis problems.
- The introduction of a novel multi-layer and multi-channel architecture based on stigmergy for pattern analysis. The architecture is presented via five real-world applications. More in particular, the applications are presented in chronological order, following the development and the revision of each module. For each application, the role of the modules and the parameters is carefully described, in order to support the reusability and adaptability to other domain of applications.
- An algorithm for swarm-based pattern analysis. The algorithm coordinates swarms of vehicles in an unexplored and unstructured environment. The algorithm takes inspiration from marker-based stigmergy and the flocking behavior of birds. The former directs the exploration via the release of a digital pheromone, which attracts vehicles toward of zones with potential targets, while the latter keeps the swarm dense enhancing local interactions with the digital pheromone. The results obtained by simulations on real world scenario supports the applicability of the approach as a mean to quick survey an area.
- The adaptation employs an evolutionary algorithm to assist the parameterization of the multilayer architecture and the swarm algorithm. More precisely, the adaptation is partially-supervised, which means that the evolutionary algorithm supports the parameterization via a small training set.

1.3 Outline of the Dissertation

This dissertation has five Chapters:
Chapter 1 introduces the background and the motivation of the study.
Chapter 2 explores the literature in: (i) the principles regarding the stigmergy and the applications in computer science; (ii) the application of Biological Inspired Algorithms for problem optimization, in particular the Differential Evolution algorithm; (iii) specific domains in which the proposed approach has been applied, such as: topic detection and visualization in microblogs, urban traffic congestions via GPS data, target detection using small unmanned aerial vehicles, and an overview of similarity measures for temporal patterns.
Chapter 3 presents five of the six applications in five subsections. Each subsection introduces and states one of the problems, then the stigmergic architecture is properly applied and described in all its functional modules, and finally the results are presented.
Chapter 4 presents the Multilayer Stigmergy-Based Architecture for temporal data analysis and its application to the problem of identification of discussion topic in microblogs. Here, we compare the performance of the proposed approach with state-of-the-art techniques.
Finally, Chapter 5 draws the conclusions.

CHAPTER 2

Related Works

This Chapter has three parts: in Chapter 2.1 we present the state of the art regarding stigmergy, both from its biological foundation and its application in computer science; in Chapter 2.2 we present and compare three biological inspired algorithms for parameters adaptation. Then, we focus on Differential Evolution algorithm, which will be applied in the adaptation process of the proposed approach; finally, in Chapter 2.3 we present the state of the art in specific domain fields regarding the applications in which the proposed approach has been applied.

2.1 Computing with Stigmergy

"Stigmergy is an indirect, mediated mechanism of coordination between actions, in which the trace of an action left on a medium stimulates the performance of a subsequent action" [10]. Stigmergy has been discovered by Grassé studying the behavior of termites [8]. The word stigmergy comes from two Greek words: *stigma* which means "mark", and *ergon* which means "action". Stigmergy is at the basis of how insects of very limited intelligence, without apparent communication, manage to collaboratively tackle complex projects, such as building a nest [13]. Grassé noticed a particular degree of coordination in the actions of the insects which allowed the development of complex structure in the nest building process. The explanation of the phenomenon relies on a set of simple rules: the termite moves a mud-ball in a random location and releases some pheromone on it; the pheromone attracts termites, so the more intense is the pheromone in a location the more are the termites moving mud-balls to that precise location. Over time the sequence of actions performed by the colony of insects leads to the construction of pillars, arches, tunnels and chambers. However, stigmergy occurs also in human-behavior [7]. One of the best-known examples of the stigmergic self-organization of the human society is the "invisible hand" of the market. Here, the actions of buying and selling leave a trace by affecting the price of the transacted commodities. Then, the price is the stimuli for following transactions [10].

According to Heylighen [10], the principle that stays behind the stigmergy has its core in the concept of *action*. The *action* is a causal process that produces a change in the state of the world [10]. The *agent* is the performer of the action; however some extension of stigmergy can even do away without the concept of agent, e.g., chemical reactions [10]. In agent systems, the *action* has its antecedent in the *condition*, which is a state of both the agent and the environment. If the *condition* is met the agent activates

the *action*. This process can be deterministic or stochastic, i.e., it can happen with a certain probability. Moreover, multiple *actions* can coexist in multi-agents system, sometimes with opposite effects. In this case, an *action* can inhibit or balance another *action* leading to a sort of stable configuration of the system. The environment [7], also described as a medium in a more extended definition by Heylighen [10], has a key role in the process. The environment is the space in which the agents perform their actions, but it also has an active role as the mediator of the communication between the agents. The environment maintains memory of the action performed by an agent as a *mark* and aggregates subsequent *marks* in a structure named *trail*. The *trail* is the main component of the *condition*, which leads to the *action* of the agent. However, the environment keeps the *trail* at disposal for the agent for a limited amount of time. In fact, the *trail* evaporates overtime disappearing from the environment. This is a natural mechanism to maintain updated the information in the environment.

In our work we take in consideration a particular type of stigmergy, the *marker-based* stigmergy, which has been described by Van Dyke Parunak [7]. The author distinguishes between four categories of stigmergy, described by two binary features, which are summarized in Table 1. The first distinction regards whether the *condition* depends on special markers released in the environment, such as pheromones, or whether the *condition* depends on the current state of the solution. The author refers to the former as *marker-based* stigmergy and the latter as *sematectonic* stigmergy. The second distinction is whether the dimension of the signal is a scalar, *quantitative* stigmergy, or whether is a set of discrete options, *qualitative* stigmergy.

Table 1 Categories of stigmergy.

	Marker-Based	**Sematectonic**
Quantitative	Gradient-following in a single pheromone field	Ant cemetery clustering
Qualitative	Decisions based on combinations of pheromones	Wasp nest construction

The previously described behavior of the termites in the nest building process is an example of *quantitative marker-based* stigmergy. The termites follow the gradient of the pheromone to pile the mud-ball together. When foraging, ants use a similar procedure based on gradient following to locate a source of food and to track back the path toward the colony. Multiple pheromones with different gradient foster and inhibit several activities of the colony regulating the actions of the insects. This is referred as *qualitative marker-based* stigmergy. In the ant cemetery, the ants dispose the corpse according to the distribution of the corpses already in the cemetery. Here, the enabler of the action (move a corpse to a location in the cemetery) depends on the state of the environment (the distribution of the corpse). This is an example of *quantitative sematectonic* stigmergy: the decision is quantitative because depends on the single density of the corpse distribution. Finally, the construction of a wasp nest is an example of *qualitative sematectonic* stigmergy. Wasps decide where to add the next cell to their nests based on which of several templates best characterizes the current local shape of the nest, thus making a qualitative distinction.

In computer science, marker-based stigmergy occurs when marks are left in an environment to enable self-coordination [6]. One of the first applications of stigmergy in computer science is the class of ant algorithms [14], which has been used in solving several computational problems [15], such the traveling salesman problem, scheduling problem, structural and concrete engineering, digital image processing, electrical engineering, clustering, and routing. Stigmergy has also been applied in autonomous robot control systems. Here, the principle is still the achievement of the self-organization of multi-robot system in which the reaching of the goal is possible thanks to the interaction of a great number of entity that works together without deep planning capability or central coordination, but with local interactions and simple rules mechanism. For example, clustering tasks have been modeled with ant behavior like strategies [16], [17] [18], while Valckenaers et al. [19] applied stigmergy principles to model the problem of control and coordination in manufacturing environment. Moreover, stigmergy has been applied for the control of coordination of aerial drones ([20], [21], [22], and [23]) and in road traffic management ([24], [25], [26], [27], [27], [28]) (both detailed in Chapter 2.3).

While stigmergy principles have been exploited for a wide range of specific computational problems, this paradigm has not been extensively applied to pattern analysis problems [29]. Brueckner *et al.* [30] presented an algorithm for pattern detection and classification based on stigmergy, which has been used for the detection and classification of synthetic image. However, the authors focused on the spatial patterns and not on the temporal ones. Differently, our approach can be employed as a powerful computing paradigm exploiting both spatial and temporal dynamics, because marker-based stigmergy intrinsically embodies the time domain. The purpose of our approach is to overcome explicit top-down domain-dependent representations of data, which are more efficient to be computed but more inefficient to be managed in the entire lifecycle. By using stigmergy as a computational paradigm, the collective properties or interactions between data can be described with a domain-independent spatiotemporal logic. Moreover, the mapping provided is not explicitly modeled at design-time and then not directly interpretable. This provides a kind of information blurring of the human data, and can be also exploited to solve privacy issues [6] [31].

2.2 Adaptation via Biological Inspired Algorithm

Many parameterization problems may be solved by *search* methods, which are procedures that look for a solution by trying out many attempts until a satisfactory result is obtained. Biologically inspired algorithms (BIAs) implement search mechanisms applicable to problems that cannot be efficiently solved using exact and analytical techniques [11]. Indeed, it is apparent from Table 1 that each case employs a different quality metrics. Then, an optimization method using a “black box” approach, which is not based on formal properties of the quality function, may be effective. Due to their random nature, BIAs can find near-optimal solutions rather the optimal solution.
BIAs optimize a problem by iteratively trying to improve a population of candidate solutions with regard to a given measure of quality, or *fitness*. Solutions are improved by means of stochastic transformation mechanisms inspired by biology, such as reproduction, mutation, recombination, selection, survival, swarm, movement, in an

environment whose dynamics are represented by the quality measure.
Three key processes characterize a BIA: the generation of the initial population, the evaluation of the fitness value and the generation of a new population. At first, the members of the population are initialized randomly or injecting some initial solutions. Each member represents a solution of the problem. The second step, the members are evaluated by means of a fitness function. The fitness is a measure of goodness of the solution and it as the means of comparing two members of the population and select which one is better. The third key process is the generation of a new population via the perturbation of the current population. The second and the third step iterate until a stop condition is met. There are two main categories of stop condition: (i) static, e.g., a fitness value to reach, a number of generations to compute, etc.; (ii) dynamic, e.g., a percentage of improvement of the fitness between generations, convergence of the population members, and so on. In general, all BIAs have two parameters to be set which are the number of members of the population and the stop condition, which usually is the number of generation to compute.

Since the mid-sixties many BIAs have been proposed, and many efforts have also been devoted to compare them. In the last decade, most notably the following three classes of methods attracted attention: Genetic Algorithm (GA) [32], Differential Evolution (DE) [12], and Particle Swarm Optimization (PSO) [33].

In GA [32], the main idea is to mimic the natural selection of genetic features during the reproduction of living organism. In GA a member of the population is represented as a chromosome. The members are ranked according to the fitness value and the new generation is computed in three steps: selection, crossover and mutation. First, the chromosome are selected to become the parents of the new generation depending on their rank (e.g., the best in the rank has greater probability to be a parent); second, the crossover operator combines the chromosome of the selected parents to generate a new chromosome. With such procedure, chromosomes with better fitness are taken more frequently and are more likely to survive to the next generation. Finally, the mutation injects some modification in the new chromosome to avoid stagnation and increase diversity.

In PSO [33], a member is a particle and the population is a swarm of particles. The particle has a position, a velocity and the historical best position. The position is a solution in the search space of the problem, the velocity indicates where and how the particle is moving, and the best position is the individual best position that the particle has ever visited. The evaluation of the position is computed with the fitness function. The swarm has a global best position, which is the best position visited by all particles. During the initialization of the swarm, the position and the velocity are chosen randomly. The new position of the particle depends on the previous position and velocity. If the new position improves the individual best and/or the swarm best the values are respectively updated. Based on these values, which represents the experience of the particles and the swarm, the velocity for each particle is updated.

DE [12] considers a member of the population as a D-dimensional vector, in which D is the search space. The key difference of DE from GA and PSO is in how the new generation is computed. Each candidate member of the old generation is compared with a trial solution, which is a combination of several members of the population. More in

details, the process has three steps: mutation, crossover and selection. In the mutation, the mutant vector is calculated as a combination of three random members of the population. Then, the trial vector is the result of the crossover between the candidate member and the mutant vector. The crossover is a random selection in which each element of the trial vector is taken from the candidate or the mutant. Finally, the selection is the comparison between the candidate and the trial according to the fitness value. The best one passes to the next generation, while the other is discarded.

Table 2 [11] summarizes the qualitative properties of the three algorithms. The key difference is in the mechanism to produce the new generation from the old generation. In GA, the members of the population need to be ranked according to the fitness value and the best members are more likely chosen as parents of the new generation. For this reason, GA tends to produce solution in clusters around some "good" solutions. Moreover, because of the ranking the time to compute all the generation scales non-linearly with the number of members of the population.
In PSO, the new position of the particle is generated from the old position and the velocity. Thus, the particle can end in every possible location of the search space. In addition, the best particle found by the swarm influences the update step of the velocity that can lead to a premature convergence of the algorithm.
DE has similar exploration capability of PSO because the generation of the new solution is computed from three random members of the population. Moreover, the best member of the population does not take part in the generation, thus the algorithm does not suffer of premature convergence.

Table 2: An excerpt of the properties of the algorithms GA, PSO, and DE [11].

Property	GA	PSO	DE
Require ranking of solution	Yes	No	No
Influence of population size on solution time	Exponential	Linear	Linear
Influence of best solution on population	Medium	Most	Less
Average fitness cannot get worse	False	False	True
Tendency for premature convergence	Medium	High	Low
Density of search space	Less	More	More
Ability to reach good solution without local search	Less	More	More

Vesterstrom and Thomsen [34] compared DE, PSO and a Simple Evolutionary Algorithm (SEA) on a set of 34 benchmark problems. The new generation in SEA is computed as follow: each member of the population can be subjected to mutation or crossover (or both) with a certain probability. The mutation operator is the Cauchy distribution with an annealing scheme, while in the arithmetic crossover the offspring is the weighted mean of the parents, according to a random weight. The set of benchmark problems include: both unimodal and unimodal functions with correlated and uncorrelated variables; two noisy problems; a problem with plateaus. Problem dimensionality varied from 2 and 30, and an extension to 100-dimensionality were provided to assess the performance on more difficult problems. Vesterstrom and Thomsen [34] concluded that DE is the best performing algorithm in their experiments. However, DE has slower convergence in the two noisy problems, and in one of the benchmark function stagnates in a sub-optimal solution (same problem occurred for PSO). As pointed out by the authors, such problems can be solved modifying the DE strategy or adjusting its parameters. In general, DE is robust and able to reproduce the same results consistently over many trials, in particular for 100-dimensionality

problems. PSO results more sensitive to the random initialization of the population and to parameters changing than other algorithms. However, PSO is the fastest in converging, followed by DE.

It is apparent from previous discussion that DE is a simple, robust, fast and efficient adaptive scheme for global optimization. In addition, it has few parameters to set, and the same settings can be used in different problems. For this reason, it was selected to design the adaptation subsystem. Next subsection is devoted to DE and its different variants.

Differential Evolution Algorithm

DE has been used in several domains for optimization and parameterization tasks: bioinformatics, chemical engineering, pattern recognition and image processing, artificial neural networks, signal processing, electrical power systems, electromagnetism, propagation, and microwave engineering, and finally control system and robotics.
DE starts with the initialization of a population of N members. Each member is a v-dimensional vector in which each element is a parameter of the optimization problem. The fitness function $f(v)$ evaluates the goodness of the vector v as a solution of the problem.
The new generation is computed member by member. Naming target vector the current member the three steps are as follows:

1. Mutation
   The mutant vector is the combination of three randomly selected vector of the population exclude the target vector, as:
   $$v_{mutant} = v_1 + F \cdot (v_2 - v_3). \quad (1)$$
   The differential weight $F \in (0, 2]$ is a scaling factor between the vectors;
2. Crossover
   The trial vector is constructed through a crossover operation which combines elements from the target vector and from the mutant vector, according to the crossover rate $CR \in [0, 1]$;
3. Selection
   Finally, the trial vector is compared with the target vector via the fitness function; the best of the two enters the new generation, while the other is discarded;

Many variants of the DE algorithm have been designed, by combining different structure and parameterization of mutation and crossover operators [12] [35] [36]. Any specific DE strategy is formally described as *DE / x / y / z*, where:

- *x* defines the base choice (v_1 in Equation 1) of the mutant vector (v_{mutant}) between:
  - *rand*, random vector as in Equation 1, which usually explores more, but requires more generations to converge;
  - *best*, the best population individual v_{best}, which converges faster, but risks to be trapped in local minima:
    $$v_{mutant} = v_{\text{best}} + F \cdot (v_1 - v_2). \quad (2)$$

- *rand-to-best*, a combination of the above strategies (weighted sum of *F*):

$$v_{mutant} = v_{\text{target}} + F \cdot (v_{\text{best}} - v_{\text{target}}) + F \cdot (v_1 - v_2); \quad (3)$$

- *y* is the number of differences in mutation carriers. Taking as example Equation 2:
  - *1*: same as Equation 2;
  - *2*:

$$v_{mutant} = v_1 + F \cdot (v_2 + v_3 - v_4 - v_5); \quad (4)$$

    where v_2 , ..., v_5 are always random vectors from the population;

- *z* is the type of crossover:
  - *bin* (binomial), in which *CR* is the probability that an element of the vector is taken from the target or from the mutant vector;
  - *exp* (exponential), in which, starting from a random element of the vector, the mutation proceeds sequentially in a circular manner. It stops with probability *CR* after each item, or if all the elements are changed;

Mezura-Montes et al. [35] presented an empirical comparison of several DE variants in solving global optimization problems. The authors implemented eight variants of DE and tested on 13 benchmark problems. The first four variants vary in the base choice (*rand* and *best*) and in the type of crossover (*rand* and *bin*): *DE/rand/1/bin*, *DE/best/1/bin*, *DE/best/1/exp*, *DE/rand/1/exp*. Two variants use a arithmetic combination similar to the *rand-to-best* strategy, and namely are: *DE/current-to-best/1* and *DE/current-to-rand/1*. One variant with multiple differences in the mutation operator: *DE/rand/2/dir* (the *dir* strategy incorporates information about the objective function); finally, a variant with combined discrete-arithmetic recombination: *DE/current-to-rand/1/bin*. The 13 benchmarks problems with dimensionality 30 were selected according to two main features: modality and separability. Three stopping criteria for DE were chosen: (i) reaching a fitness value with an error of 10^{-12} from the global optimum or (ii) computing 120,000 fitness evaluations (i.e., tested vectors) or computing 2000 generations. DE has population *N*=60, CR ∈ (0, 1) by step of 0.1, and F ∈ [0.3, 0.9] randomly change at each new generation.
In unimodal and separable problem the most competitive variant is *DE/best/1/bin*, followed by *DE/rand/1/bin*, *DE/current-to-rand/1/bin*, and *DE/rand/2/dir*. It is worth noting that *DE/best/2/exp* obtained poor performance in all functions, thus the recombination type *binomial* works better than *exponential* in unimodal and separable functions. For unimodal/nonseparable and multimodal/separable problems *DE/best/1/bin* is the only variant able to solve the problems, followed by *DE/rand/1/bin* and *DE/rand/2/dir*. While for multimodal and nonseparable problems the best performance has been achieved by *DE/rand/1/dir*, followed by *DE/rand/1/bin* and *DE/best/1/bin*.
To sum up, according to [35] *DE/best/1/bin* is a competitive DE variant for a wide set of benchmark problems. In addition, the *binomial* crossover outperforms the *exponential* in almost every experiment. Zaharie [36] furtherly investigated the performance of DE with binomial and exponential crossover and varying the crossover rat CR. More in detail, given D the dimensionality of the vector, the target vector v_{target}, the mutant vector v_{mutant}, the trial vector v_{trial} a function i_{rand} which return a random integer, and a function rand which return a random real, Table 3 and

Table 4 presents the binomial and exponential crossover respectively. In both crossover variants at least one element of the mutant vector ends in the trial (Table 3 line 4 *j=k* condition and line 3 **do…while** in
Table 4), and greater is the CR rate greater is the chance that more elements of the mutant end in the trial. However, while in the binomial crossover each element can be taken from the mutant vector independently from what happened to the previous, in the exponential crossover the mutation is circular, and thus elements change in a sequential order and once the process stops no more elements will change. In other words, in the binomial crossover the CR rate explicitly is the probability for a component to be replaced with a mutated one, while in the exponential crossover CR is used to decide how many elements will be mutated. As demonstrated by [36], the dependence between the mutation probability and the crossover is linear in the binomial crossover and nonlinear in the exponential crossover, and thus for the same CR value the mutation probability is smaller for the exponential crossover than the binomial. For this reason, to obtain similar result when using exponential crossover the CR rate has to be greater than when adopting the binomial crossover [12].

Table 3 The algorithm of the binomial crossover.

```
 1: BinomialCrossover (v_target, v_mutant)
 2: k ← irand({1, . . . , D})
 3: for j = 1, D do
 4: if rand(0, 1) < CR or j = k then
 5:     v_trial j ← v_mutant j
 6: else
 7:     v_trial j ← v_target j
 8: end if
 9: end for
10: return v_trial
```

Table 4 The algorithm of the exponential crossover.

```
 1: ExponentialCrossover (v_target, v_mutant)
 2: v_trial ← x; k ← irand({1, . . . , D}); j ← k; L ← 0
 3: do
 4:     v_trial j ← v_mutant j; j ← j + 1; L ← L + 1
 5: until rand(0, 1) > CR or L = n
 6: return v_trial
```

The population size spreads from a minimum of 2*n* to a maximum of 40*n* [37]. A large population increases the chance of finding optimal solution but it is very time consuming.
Several DE strategies have been designed, by combining different structure and parameterization of mutation and crossover operators [35] [36]. As noted by [38], practitioners mostly prefer to use a classical DE variant like DE/1/rand/bin. The differential weight F ϵ [0,2] mediates the generation of the mutant vector. F is usually set in [0.4-1), and a frequent used starting value is 0.8 [35]. There are different crossover

methods in DE. A competitive approach is the binomial crossover [36]. With binomial crossover, a component of vector is taken with probability CR from the mutant vector and with probability 1-CR from the target vector. A good value for CR is between 0.3 and 0.9, and a frequently used starting value is 0.7 [37].

The interested reader is referred to [38] for exhaustive information. As an example, in [39]the authors used DE/1/rand/bin to coordinate multiple drones navigating from a known initial position to a predetermined target location. Here, DE is set up with N=50, F=1.05 and CR=0.85. The algorithm was defined to terminate in 200 generations, but it usually converges in 30 iterations. Our problem sensibly differs, because the target position is unknown, and our approach is independent of the initial position.
In [40] the authors confront DE and Particle Swarm Optimization (PSO) for co-operative distributed multi-robot path planning problem. As for [39] initial position of the robots and final position are known. Here, both centralized and decentralized formulations are proposed. In the centralized approach, DE minimizes the distance for the next step of each robot. In this case all information of the position of each robot, the next position, and the possible collision are provided to DE. In the decentralized formulation, each robot runs DE for itself considering the information of neighbor robots. Authors conclude that decentralized approach needs less time in comparison to the centralized; moreover the performance is comparable to PSO. In our approach, we consider to use DE offline to find a proper and general purpose parameter tuning for the swarms. Moreover, in our formulation drones have a limited computing capability, and then an online execution of DE is not feasible.
In [41] DE/1/rand/bin is used with F=0.7, CR=1.0, N=120 for 250 generations, and DE/1/best/bin is used with F=0.8, CR=1.0, N=150 for 200 generations to tune the behavior of a robot in wall-following task. Here it seems that DE/1/best/bin is able to find a slightly better solution than DE/1/rand/bin. However, authors used different parameters settings (F, N and number of generations) for each variant, thus a comparative analysis is difficult. In our approach we focus on DE/1/rand/bin variant and evaluate several combinations of CR and F.

2.3 Domain Specific Spatio-Temporal Pattern Analysis

This Chapter summarizes the literature in the main fields in which the proposed architecture has been experimented. The Chapter has four parts, which present the state of the art in: (i) the detection and visualization of discussion topics in microblogs, such as Twitter; (ii) the detection of vehicles congestion in urban roads; (ii) the detection of target using swarms of small aerial vehicles; (iv) finally, a brief summary of similarity measures for temporal pattern, which will be used for the comparison in Chapter 4.

Topic Detection and Visualization in Microblogs

The purpose of topic detection in microblogs is to generate terms clusters from a large number of tweets. Most topic detection algorithms are based on clustering algorithms. Although traditional text clustering is quite mature, the topic detection method for microblog should focus more attention on temporal aspects. Furthermore, strictly connected to topic detection are the term cloud visualization models. In this section, we highlight the most relevant works in both aspects.

Weng and Lee in [42] present an approach to discover events in Twitter streams based on wavelet analysis or EDCoW. In their approach, the frequency of each word is sampled over time, and a wavelet analysis is applied to this time series. Subsequently, the bursty energy of each word is calculated by using autocorrelation. Then, the cross-correlation between each pair of bursty words is calculated. Finally, a graph is built from the cross-correlation, and relevant events are mined with graph-partitioning techniques. Xie *et al.* [43] proposed TopicSketch for performing real-time detection of events in Twitter. TopicSketch computes in real-time the second order derivative of three quantities: (i) the total Twitter stream; (ii) each word in the stream; (iii) each pair of words in the stream. Then, the set of inhomogeneous processes of topics is modeled as a Poisson process and, by solving an optimization problem, the distribution of words over a set of bursty topics is estimated. In both [42] and [43], the event is characterized as a bursty topic, which is a trivial pattern in microblogging activities: something happens and people suddenly start writing about it. In contrast to [42] [43], our approach computes the similarity between time series without relying on the burst as a special pattern. Thus, the similarity based on stigmergy is more general and flexible. Other approaches are based on a variety of patterns [44] [45] [46]. Yang and Leskovec [44] developed the K-Spectral Centroid (K-SC) clustering, which is a modified version of classical K-means for time series clustering. K-SC uses an ad-hoc distance metric between time series, which takes in consideration the translation and the shifting to match two time series. More specifically, the authors use long-sized time frames for their analysis, up to 128 hours (with one hour as time unit) to detect six common temporal patterns on news and posts. Similarly, Lehmann *et al.* [45] study the daily evolution of hashtags popularity over multiple days, considering one hour as a time unit. They identify four different categories of temporal trends, depending on the concentration around the event: before and during, during and after, symmetrically around the event, and only during the event. Stilo and Velardi detect and track events in a stream of tweets by using the Symbolic Aggregate approXimation (SAX [46]). The SAX approach consists of three steps: first the word streams in a temporal window are reduced to a string of symbols; then a regular expression is learned to identify strings representing patterns of collective attention; finally, complete linkage hierarchical clustering is used to group tokens with similar strings occurring in the same window. The authors evaluate the approach discovering event in a one-year stream (with one day as time unit) and manually assessing the events on Google. In particular, five patterns of [46] correspond to six patterns of [44]. This emphasizes a further evidence of the variability of temporal patterns in microblogging. However, while [44], [45], and [46] focus on a coarse temporal scale (hour or day), established at the macro (i.e., application) level, our approach adopt a fine temporal scale (minute) to detect the micro patterns occurring in social discussions, carrying out the analysis of the macro-dynamics without adopting specific patterns at design time.

In [47] Xu *et al.* studied the competition among topics through information diffusion on microblogs, as well as the impact of opinion leaders on competition. They developed a system with three views: (i) timeline visualization with an integration of ThemeRiver [48] and storyline visualization to visualize the competition, (ii) radial visualizations of word clouds to summarize the relevant tweets; (iii) a detailed view to list all relevant tweets. The system was used to illustrate the competition among six major topics during the 2012 United States presidential election on Twitter. Here, in contrast to our approach, there is no integration of the temporal information in the word cloud which is used as a mean of summarization of the content of the tweet. In [49] Lee developed a

cloud visualization approach called SparkClouds, which adds a sparkline (i.e. a simplified line graph) to each tag in the cloud to explicitly show changes in term usage over time. Though useful improvements to tag clouds, it does not show term relations and thus it is not able to visualize time-varying co-occurrences. Lohmann *et al.* extends the tag clouds by an interactive visualization technique referred to as time-varying co-occurrence highlighting [50]. It combines colored histograms with visual highlighting of co-occurrences, thus allowing for a time-dependent analysis of term relations. Cui *et al.* in [51] presented TextFlow, which implies three methods of visualization to describe the evolution of a topic over time: (i) the topic flow view, which shows the evolution of the topics over time; it highlights the main topics and the merging and splitting between topics; (ii) the timeline view, which places document relevant to a selected topic in temporal order; (iii) the word cloud view, which shows the word in different font sizes. In contrast to our approach, the word cloud has no temporal or relational information. Other approaches focus the analysis on the user interest, such as ThemeCrowds [52], which displays groups of Twitter users based on the topics they discuss and track their progression over time. Twitter users are clustered hierarchically and then the discussed topic is visualized in multiple word clouds taken at different interval of time. Tang *et al.* used a term cloud to represent the interest of the social media users [53]. They presented a SVM classifier to rank the interest of the users, using the user keywords as a score. The score is applied to the font size of the keyword in the term cloud. Raghavan *et al*. develop a probabilistic model, based on a coupled Hidden Markov Model for users' temporal activity in social networks. The approach explicitly incorporates the social effects of influence from the activity of a user's neighbors. It has better explanatory and predictive performance than similar approaches in the literature [54]. Liang *et al.* observed that (i) the rumor publishers' behaviors may diverge from normal users, and (ii) a rumor post may have different responses than a normal post. Based on this distinction, they proposed an ad hoc feature and trained five classifiers (logistic regression, SVM with RBF kernel function, Naive Bayes, decision tree, and K-NN). Results show that the performance of rumor classifier using users' behavior features is better than that based on a conventional approach [55]. In contrast, in our approach we avoid the explicit modeling of the user interest. In this work, we adopt a new modeling perspective, which can be achieved by considering an *emergent* design approach [56]. With an emergent approach, the focus is on the low level processing: social data are augmented with structure and behavior, locally encapsulated by autonomous subsystems, which allow an aggregated perception in the environment. Emergent paradigms are based on the principle of the self-organization of the data, which means that a functional structure appears and stays spontaneous at runtime when local dynamism occurs.

Traffic Congestion Detection using GPS trace of the vehicles

Taking into account the technology involved in traffic state estimation, a number of methods have been developed. In [57] probe-vehicle data is used to determine kernel-based traffic density estimation. The method first models the traffic data with Gaussian density (centered in the sample position with predefined mean and variance) to extract the kernel parameters. Then, distance between their localized cumulative distributions is measured and optimized, in order to extract the weights of Gaussian kernels in the estimated distribution function. The approximation density function by optimized kernels' weights is finally used to estimate the mobile vehicles density in a specific time

and space. In [58] the traffic flow is analyzed by means of GPS and GIS integrated system. In this approach roads are split up into segments, and mean car speed in it is estimated using loop detectors and taxi as probe vehicles, therefrom an approach based on Federated Kalman Filter and D-S Evidence Theory is used, to join such data. Finally, authors propose a curve-fitting method aimed to estimate mean speed in an urban road. It uses least-square method in order to fit data coming from GPS. In [59] the authors pursue a road-segment average traffic velocity estimation, achieved through two different approaches: vehicle tracking and curve-fitting. Experiments show how a tracking-based method usually bears higher estimate accuracy but slower operational speed with respect to a model-fitting method. In [60] two subsequent GPS samples are used to define a vehicle track by means of the A* algorithm. The combination of tracks velocities passing through the road segment determines the average velocity of the current segment. In [61] an algorithm is proposed to estimate the traffic flow state by using the minimum GPS samples via a curve fitting method. The algorithm takes into account sample frequency, the road type, and the road section length. A spatial and temporal classification of road traffic state based on GPS data is proposed in [62]. Spatial classification aims to represent steady traffic, while temporal classification reflects traffic speed. Authors use GPS samples to calculate vehicles delay distribution over a road segment in order to classify the traffic. Time-location data is converted to spatiotemporal data and then classified using threshold-based quadrant clustering. Authors compare quadrant classifier with maximum likelihood and maximum a priori classifiers.

In [24] traffic congestion forecast is realized via stigmergy. Here vehicle flow is measured via fixed on-road sensors and traffic-density is processed via digital pheromone. Another type of service is the recommendation of a path to avoid congestion. In [25] the authors proposed the DSATJ system, which computes alternative optimum path to avoid traffic jam. Here, digital pheromone evaporation and deposit on a virtual space mapping the roads is managed. The traffic jam is detected via upper bound on the pheromone value. Moreover, diversion of traffic on the roads which had been jammed was represented by normalization of pheromone. While this approach takes advantage from distributed computation, it requires that every vehicle involved in the analysis declares its destination and starting point.

In [26] a traffic lights control system based on swarm intelligence is presented. Here, control methods are divided into macroscopic and microscopic levels, and are based on stigmergic evaluation of traffic flow, by using pheromones deposits characterized by evaporation/diffusion dynamics.

In [27] the authors assume the following types of stigmergy: long term, short term, and anticipatory. The main differences lie in how and when the vehicles' position information is stored. Long term stigmergy is archived in a central storage, and provides stochastic traffic congestion information to vehicles. Short term stigmergy occurs while vehicles are sharing current data, and drivers can choose their routes more dynamically, on the basis of such real time information. Anticipatory stigmergy implies that vehicles can declare their destination, in order to distribute pheromones in advance and use them during routing task. Here, aprioristic knowledge on the phenomena is then required. The authors conducted several simulations on traffic analysis to compare the effectiveness of the different kinds of stigmergy. The results demonstrate that only if the traffic network is static, the combination of long term and short term stigmergy overcome the other kinds of stigmergy.

While in [27] the road is considered as a monolithic structure, in [28] roads are divided

into segments. Here, congestion evasion strategies based on digital pheromone are investigated. More specifically, every vehicle deposits digital pheromone in the virtual environments, and takes into account the pheromone state to decide upon its subsequent route. The authors propose an algorithm for decentralized, self-organizing, traffic flow improvements adapting the mechanisms of the ant pheromone.

Target detection with Swarms of Small Drones

The goal of this section is to briefly characterize the main approaches and results in the literature on stigmergic mechanisms coordinating swarms of small robots to perform target search or similar tasks. The published works in the field can be distinguished into three categories: (a) using a *physical* substance as a pheromone, which is necessarily transmitted in an indirect way between robots, by means of the physical environment; (b) using a *digital* pheromone, transmitted via *direct communication* between robots; (c) using a *digital* pheromone, transmitted via an *indirect communication* between robots. The latter is the category of our approach.

In [20] the authors use a swarm of robots releasing *physical substance* as a repulsive pheromone, for environment exploration. In particular, robots act combining three basic behaviors, with decreasing priority: wall avoiding, pheromone coordination, and random walk. Actually there are various approaches in the literature using physical pheromones, because they do not require a computational structure. Although real pheromones are not usable with aerial vehicle, they can be simulated. Thus, this type of research can be interesting to digitally model new types of stigmergy.

An example of stigmergic coordination between drones using direct communication is presented in [21], where the author focuses on automatic target recognition. Potential target are marked by drones, which also communicate the *gossiped pheromone* to nearby drones, with probability inversely proportional to the distance from the source. The proposed stigmergic schema employs also repulsive pheromone, as a negative feedback, when a predefined number of drones identify the same target. A disadvantage of such scheme is that the bandwidth required goes into an exponential explosion as the population grows. Moreover, to avoid redundancy in target evaluation each UAV has to maintain in memory the state of each potential and confirmed target. In this way, the direct communication in the swarm should be strongly limited [22].

A swarm coordination schema with indirect coordination is proposed in [23]. Here the coordination of a swarm of vehicles is based on digital pheromones maintained in an artificial space called *pheromone map* and composed by an arbitrary graph of place agents, such as intermediate control nodes. There are two classes of agents, namely the walkers and the avatars, which deposit, withdraw, and read pheromones. A walker agent aims to make movements and action decisions, whereas avatars collect location information to make estimates when sensor information is not available. The schema has been applied to a range of scenarios, among which target acquisition. An important problem of this approach is that the exploration depends on the initial state of deploy of the swarm. Moreover, this model does not consider complex targets but only simple targets with no reciprocal relationships.

Similarity measures for Temporal Patterns

The time series analysis is a fundamental field for the study of temporal data which can

be collected in both academic and business scopes. A time series is a collection of chronological observation, generally of high dimensionality, numerous features, continuous, and periodically updated with new samples. The temporal dynamic is often numeric and continuous, and it is considered as a whole set rather than an aggregation of samples [1]. Moreover, in Figure 1, time series are subjected to distortions such as noise, offset translation, amplitude scaling, longitudinal scaling, linear drift and discontinuities [63]. Noise is an unwanted component of the time series, which is usually due to modifications of the data occurring during the processes of capture, storage, transmission, processing or conversion. E.g., recording studios have soundproof rooms to reduce the noise from the outside and obtain high quality soundtracks. The offset translation occurs when two time series have similar shapes, but there is a constant scalar component of difference. E.g., enterprise in the same market sector may have the trend of the turnover but with different range depending on the number selling points or employees. The amplitude scaling is when two sequences are alike, but one has been stretched or compressed in the scalar axis. E.g., swimming suits and motorbikes have a seasonal trend which rises during spring and summer. The shapes are very similar, however, the amplitude for the sold swimming suits is higher because the price is affordable and more people can buy one. Similarly, the longitudinal scaling is a stretch or the compression of the shape of the time series in the temporal axis. The linear drift is a distortion that occurs in many domains subjected to trends. Consider for example the selling of a seasonal good (e.g., sun screen lotion) in two cities which different rate of birth. The seasonality of the selling has the same shape, but due to different of the growth of the population one city also has a linear drift. The discontinuities are often produced by calibration artifacts of the sensor or outliers in the data.

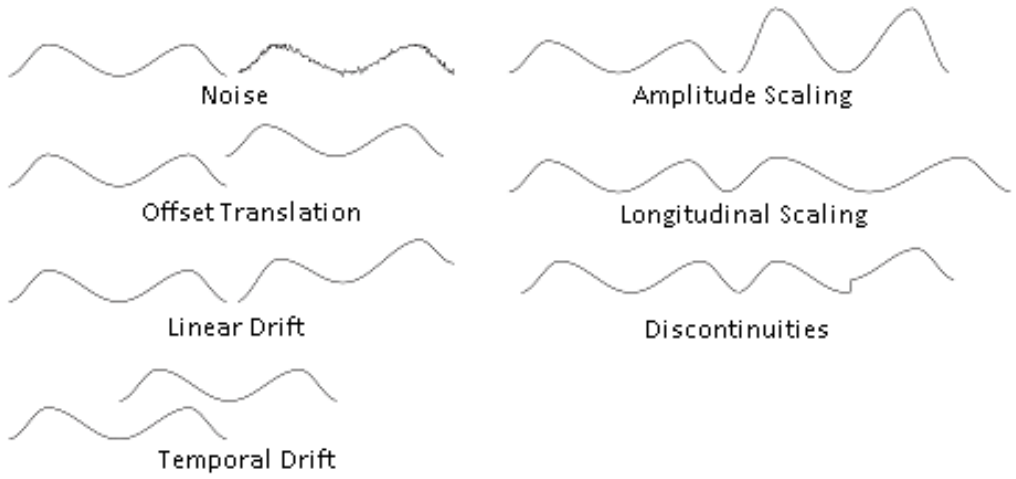


Figure 1 Distortion types in temporal data.

For these reasons, usual techniques relying on *exact match making* can't be adopted, and approximate techniques have to be considered.

At the basis of the time series analysis it is necessary to define a measure of similarity or distance to make comparison between time series. The similarity measure should be consistent with the intuitive notion of shape and provide the following properties [64]:

- It should provide recognition of perceptually similar objects, even though they are not mathematically identical.

- It should be consistent with human intuition.
- It should emphasize the most salient features on both local and global scales.
- A similarity measure should be universal in the sense that it allows to identify or distinguish arbitrary objects, that is, no restrictions on time series are assumed.
- It should abstract from distortions and be invariant to the previously presented set of transformations [63].

Depending on the organization and the processing of the data it is possible to perform a *whole sequence matching* or a *subsequence matching* [1]. In the former, the comparison matches entirely the two time series. E.g., to determine if the time series of the temperature during August 2015 has the same shape of the temperature of August 2016. Here the match takes both the entire length of the two sequences and compare them. In the latter, the objective is to find a shorter sequence in a longer one. E.g., to find all the temperature changes of 2°C occurred in less than 1 hour during the day. Here, the match takes a short sequence (the temperature change) and check the occurrence in a longer sequence (the temperature during the day).
Another distinction between similarity measures is how the elements of the time series are compared the one with the other. The *lock-step* measures compare each *i*-th element of the first time series with the *i*-th element of the second. Instead, the *elastic* measures allow a more flexible comparison of the elements (one-to-many/one-to-one/one-to-none) [65].
It is possible to distinguish between 4 categories of similarity measures [64]. The shape-based distances compare the overall shape of the time series. The edit-based distances compare two time series according to the minimum number of modifications needed to transform one series into another one. The feature-based distances extract a set of features from a time series which describes the salient aspects of the series and then compares with the same set of features of another time series with a similarity/distance function. The objective of the structure-based similarity is to find a higher-level structure in the time series and then to compare them on a more global dimension. According to [64] it is possible to further divide the structure-based similarity into two specific subcategories. The model-based distances aims to fit a model to the various time series and then, the distance is computed by comparing the parameters of the model for each couple of time series. The compression-based distances analyze how well two series can be compressed together. Similarity is reflected by higher compression ratios.

Shape based similarity have been the most widely used measures for time series analysis. The Euclidean distance offers numerous advantages: fast to compute (linear), easy to implement, and no parameters needed. However, the main disadvantage of the Euclidean distance is the mapping between points of the two time series, which is fixed one-to-one. Thus, it is hard to detect offset translation and temporal drift transformation of the time series. The same problems occur for generalization of the Euclidean distance, namely the Minkoski distance.
Given Two time series $x = \{x_1,\ldots,x_P\}$ e $y = \{y_1,\ldots,y_P\}$ of length *P* the Euclidean distance between x e y is:

$$d_E(x,y) = \sqrt{\sum_{i=1}^{P}(x_i - y_i)^2} \qquad \textbf{5}$$

Instead, the Minkoski distance is calculated as:

$$d_M(x,y) = \sqrt[q]{\sum_{i=1}^{P}(x_i - y_i)^q} \quad \textbf{6}$$

where q is a positive integer.

The Pearson's correlation coefficient is usually adopted to measure the correlation between two statistical variables. However, it can be similarly employed as a measure between time series. The coefficient assumes value near 0 in absence of correlation, near 1 in presence of positive correlation, and -1 in presence of negative correlation. The Pearson's coefficient is the ratio between the covariance of the time series and the product of the respective squared mean errors:

$$cc = \frac{\sum_{i=1}^{P}(x_i - \mu_{xi})(y_i - \mu_{yi})}{S_x S_y}$$
$$\mu_{xi} = \frac{1}{P}\sum_{i=1}^{P} x_i, \mu_{yi} = \frac{1}{P}\sum_{i=1}^{P} y_i \quad \textbf{7}$$
$$S_x = \sqrt{\sum_{i=1}^{P}(x_i - \mu_{xi})^2}, S_y = \sqrt{\sum_{i=1}^{P}(y_i - \mu_{yi})^2}$$

The Short Time Series distance, STS, is the sum of the squared difference of the slopes of the two time series. The main advantage of this measure in comparison with the previous is that it embodies the temporal information of the time series. The STS is computed as follows:

$$d_{STS}(x,y) = \sqrt{\sum_{i=1}^{P}\left(\frac{y_{i+1} - y_i}{t_{i+1} - t_i} - \frac{x_{i+1} - x_i}{t_{i+1} - t_i}\right)^2} \quad \textbf{8}$$

with t the time instant of samples x_i e y_i.

The similarity measure more popular among complete sequences is the time warping, which is based on Dynamic Time Warping (DTW) [66]. This technique generalizes the alignment of two time series A and B composed by n and m elements respectively. The first operation is to calculate the X matrix of dimensions (n-x-m), where the elements $x_{i,j}$ with $i \in n$ e $j \in m$ are the Euclidean Distance between a_i and b_j. Given the X matrix, the warping path W is the route of minimum cost where the cost of each step w_k is an element $x_{i,j}$, and $max(m,n) \leq k < m+n-1$. The path has to satisfy three conditions:

(1) the boundaries, the first step is $w_1=(1,1)$ and the last $w_k=(m,n)$;
(2) the continuity, $w_k=(p,q)$ e $w_{k-1}=(p',q')$ where $p-p' \leq 1$ and $q-q' \leq 1$;
(3) the monotonicity, $p-p' \geq 0$ and $q-q' \geq 0$;

Here, the first step to find the minimum cost path is to compute the matrix Y of size n-x-m through the cumulated distances:

$$y(1,j) = \sum_{s=1}^{j} x_{1,s}, j \in [1,m] \quad (9)$$

$$y(i,1) = \sum_{s=1}^{i} x_{s,1}, i \in [1,n] \quad (10)$$

$$y(i,j) = x_{i,j} + \min\big(y(i-1,j), y(i,j-1), y(i-1,j-1)\big), i \in [2,n], j \in [2,m] \quad (11)$$

Then, given matrix Y, the minimum cost path (warping) starts from the element $y(n,m)$ to $y(1,1)$ using a greedy technique (lowest cost) to select each step of the path. Finally, the sum of the elements of the path is the *warping time distance* between the two series.

The main principle of the edit based measures is that the distance between two time series depends on the minimum number of modifications (insertion, deletion, and substitution) needed to transform a series in another. The Longest Common Subsequence [67] is an edit based measure which allows to measure the cost to adapt a time series to another without changing the sequence of the elements, but allowing the possibility of having unmatched elements. This is the main advantage of the approach because it can handle the presence of the outliers (discontinuities) very well without increment the editing cost. In formula:

$$LCSS_{\delta,\varepsilon}(x,y) = \begin{cases} 0, \text{ if x or y are empty} \\ 1 + LCSS_{\delta,\varepsilon}(H(x),H(y)), \text{ if } |x_n - y_m| < \varepsilon \text{ and } |n-m| < \delta \\ \max(LCSS_{\delta,\varepsilon}(H(x),y), LCSS_{\delta,\varepsilon}(x,H(y))), \text{ otherwise} \end{cases} \quad (12)$$

In which x = $\{x_1,\ldots x_i,\ldots,x_m\}$, y = $\{y_1,\ldots y_j,\ldots,y_n\}$, H(x) = $\{x_1,\ldots x_i,\ldots,x_{m-1}\}$, H(y) = $\{y_1,\ldots y_j,\ldots,y_{n-1}\}$.

Another edit based measure is the Edit Distance on Real Sequence (EDR) [68]. Here, differently from the LCSS, the EDR assigns a certain edit cost if the difference between two elements is greater than a parameter threshold δ. Then, the alignment of the time series is adjusted giving a penalty directly proportional to the *gap* between the two time series. More precisely:

$$dist_{edr}(x_i,y_i) = \begin{cases} 0, \text{ if } |x_i - y_i| \le \delta \\ 1, \text{ if } x_i \text{ or } y_i \text{ is a gap} \\ 1, \text{ otherwise} \end{cases} \quad (13)$$

$$EDR_{gap}(x,y)=\begin{cases}0, \text{ if x or y are empty}\\ EDR_{gap}(H(x),H(y)), \text{ if } dist_{edr}(x_1,y_1)=0\\ \min(EDR_{gap}(H(x),H(y))+dist_{edr}(x_1,y_1),\\ EDR_{gap}(H(x),y)+dist_{edr}(x_1,gap),\\ EDR_{gap}(x,H(y))+dist_{edr}(gap,y_1)), \text{otherwise}\end{cases} \quad \textbf{14}$$

Finally, the Edit Distance with Real Penalty (ERP) [69] is a version of the EDR in which there is no threshold, but the alignment cost is measured by using the Euclidean distance. In formula:

$$dist_{erp}(x_i,y_i))=\begin{cases}|x_i-y_i|, \text{ if } x_i \text{ and } y_i \text{ not gap}\\ |x_i-g|, \text{ if } y_i \text{ is a gap}\\ |y_i-g|, \text{ if } x_i \text{ is a gap}\end{cases} \quad \textbf{15}$$

$$ERP_{gap}(x,y)=\begin{cases}\sum_1^m|x_i-g|, \text{ if y is empty}\\ \sum_1^n|y_i-g|, \text{ if x is empty}\\ \min(ERP_{gap}(H(x),H(y))+dist_{erp}(x_1,y_1),\\ ERP_{gap}(H(x),y)+dist_{erp}(x_1,gap),\\ ERP_{gap}(x,H(y))+dist_{erp}(gap,y_1)), \text{otherwise}\end{cases} \quad \textbf{16}$$

Measures based on feature rely on the extraction of a set of features which are descriptor of the time series, e.g., the coefficients from the Discrete Fourier Transform (DFT) [70] or the Discrete Wavelet Transformation (DWT) [71]. Janacek et al. [72] show that a likelihood ratio for DFT coefficients outperforms the Euclidean distance. Vlachos et al. [73] present a combination of period-gram and autocorrelation functions which permits the selection of the most important periods of a series. Papadimitriou et al. [74] propose a tracking of the local correlation extending Vlachos [73]. Concerning symbolic representations, Mannila and Seppnen [75] use random vector to represent symbols in a time series. Thus, the sum of the vectors weighted by the temporal distance is the representation of a sequence of symbols. Instead, Flanagan [76] uses weighted histograms of consecutive symbols as features. The similarity search based on Threshold Queries (TQuEST) [77] uses a given threshold parameter τ in order to transform a time series into a sequence of threshold-crossing time intervals. It has, however, been shown to be highly specialized with mitigated results on classical datasets [78]. The WARP method proposed by Bartolini et al. [79] is a Fourier-based approach.

Model-based distances use prior knowledge about the temporal dynamics to improve the similarity measure. Popular approaches use HMM with continuous output values or ARMA models are common choices [80]. Ge and Smyth [81] combine HMMs and piece-wise linear representation. Bicego et al. [82] use the similarity-based paradigm where HMM is used to determine the similarity between each object and a predetermined set of other objects.

Keogh et al. [83] define a distance measure based on the Kolmogorov complexity called Compression-Based Dissimilarity Measure (CDM). The underlying idea is that concatenating and compressing similar series should produce higher compression ratios than when doing so with very different data. This approach appears particularly efficient for clustering; it has been applied to fetal heart rate tracings [84]. Similarly, Degli Esposti et al. [85] propose a parsing-based similarity distance in order to distinguish healthy patients from hospitalized ones on the basis of various symbolic codings of ECG signals. By comparing the performances of several data classification methods, this distance is shown to be a good compromise between accuracy and computational efforts.

Comparison of Distance Measures

In conclusion, when facing an analysis task in time series domain there is no unique choice in the selection of a similarity measure. In general, the most adequate similarity measure highly depends on the nature of the data to analyze as well as the required application-specific properties. Here, the experience of the analyst plays a key factor. However, as suggested by Esling et al. [64], there are a few guidelines according to the properties of the problem:

- If the time series are relatively short and visual perception is a meaningful description, shape-based methods seem to be the appropriate choice;
- If the application is targeting a very specific dataset or any kind of prior knowledge about the data is available, model-based methods may provide a more meaningful abstraction;
- Feature-based methods seem more appropriate when periodicities are the central subject of interest and causality in the time series is not relevant;
- if the time series are long and little knowledge about the structure is available, the compression based and more generally structure-based approaches have the advantage of being a more generic and parameter-free solution for the evaluation of similarity;

Even with these general recommendations and comparisons for the selection of an appropriate distance measure, the accuracy of the similarity chosen still has to be evaluated. The accuracy of distance measures is usually evaluated within a 1-NN classifier framework. It has been shown by Ding et al. [78] that, despite all proposals regarding different kinds of robustness, the forty year old DTW usually performs better.

CHAPTER 3

Spatio-Temporal Pattern Analysis Based on Stigmergy

In this Chapter five of the six applications are presented. This Chapter follows the chronological development of the Stigmergic Architecture from its early stage to the complete version, which will be described in the following Chapter 4. First, we present the KPI analysis in collaborative networks (Chapter 3.1), in which multiple enterprises adopt stigmergy to collectively analyze and compare their KPI preserving their privacy. The application to elderly monitoring (Chapter 3.2) shows how stigmergy can be applied for the aggregation of human mobility data. Then, we enrich the architecture with the adaptation via Differential Evolution for supporting the user in the tuning of the parameters. We present two applications: the detection of urban road traffic congestions (Chapter 3.3) and the evaluation of Regional technological indicators (Chapter 3.4). Finally, we present an algorithm based on stigmergy for the coordination of swarms of aerial drones for target detection and its adaptation with Differentia Evolution (Chapter 3.5).

3.1 KPI Analysis in Collaborative Networks

Problem Statement

The 2000s were characterized by ICT advancements enabling new collaborative partnerships modes and the concept of *Collaborative Networked Organization* (CNO), which further generalizes VO. A CNO is an organization whose activities, roles, governance rules, are manifested by a network consisting of a variety of entities (e.g., organizations and people). Such entities are largely autonomous, geographically distributed, and heterogeneous in terms of their operating environment, culture, social capital and goals. But they collaborate to better achieve common or compatible goals, thus jointly generating value, and whose interactions are supported by computer network. Since not all forms of collaborative partnership imply a kind of organization of activities, roles, and governance rules, the concept of *Collaborative Network* (CN) further generalize the collaborative partnership [86], [87], [88]. In the meanwhile, a progressive opening of the companies boundaries enabled what has been defined the *Open Innovation paradigm* [89], in which externally focused, collaborative innovation practices were adopted.

With the explosion of diverse types of information in OCNs in general, and in OCNs in particular, analytics technologies that mine structured and unstructured data to derive

insights are now receiving unprecedented attention [90] [91]. Today's analytics must be operated firms wide, deep, and at a strategic level [90]. A wide range of unstructured data from firms' internal as well as external sources is available [92], enabling a broader set of industry partners to participate. In OCNs, under this model, all entities collaborate and co-develop high value analytics solutions. Well [93] properly frames them under the label "collaborative analytics" namely, a set of analytic processes where the agents work jointly and cooperatively to achieve shared or intersecting goals. They include data sharing, collective analysis and coordinated decisions and actions. Collaborative analytics, while encompass the goals of their conventional counterparts, seek also to increase visibility of important business facts and to improve alignment of decisions and actions across the entire business [93] [92].

Information reduces uncertainty in OCNs [94] and aids in integrating flows and functions across working groups such as partners (e.g., [95]). This reduction of uncertainty is useful as it saves organizational time and cost by minimizing alternate decisions that arise due to uncertainty. Furthermore, the flow of information is important for managing interactions and negotiations during collaboration activities and for combining the work of individual agents. Agents exchanging information in OCNs should confront with two characteristic: 1) *trails*, in order to identify new business opportunities and organizations to partner with; trails vanish over time realizing temporal evolution dynamics of OCNs; 2) *information perturbation*, as enabler of collaboration as privacy and unveiling sensitive information of highly competitive value; our context may be assimilated to the partial-information problem formulated by Palley and Kremer [96], in which the agent only learns the rank of the current option relative to the options that have already been observed. It is clear that information is something which is capable of having a value attached to it and can be considered to be an economic good [97]. In order to protect the economic value of information, it can be provided by using a privacy-preserving mechanism.

With regards to the *privacy-preserving collaborative analytics*, a workflow model is also used to assemble data flow together while preserving each individual flow. To maximize the usability of data flow without violating its market value, a suitable *data perturbation* technique is proposed, enabling collaborative analytics. Indeed relevant marketing concerns largely prevent data flow in collaborative networks. More specifically, business data is perturbed via digital *stigmergy*, which is a processing schema based on the principle of self-aggregation of marks produced by data. Stigmergy allows protecting data privacy, because only marks are involved in aggregation, in place of actual data values. There are two basic features which allow stigmergy to protect data flows in OCN. The first is the decentralization of control in decision making: each member has a partial view of the process which is insufficient to make the decision. Second, members are not statically organized but can dynamically move between different virtual enterprises.

Well [93] defined formally the term collaborative analytics, as "a set of analytic processes where the agents work jointly and cooperatively to achieve shared or intersecting goals". Such processes include data sharing, collective analysis and coordinated decisions and actions. Collaborative analytics, while encompass the goals of their conventional counterparts, seek also to increase visibility of important business facts and to improve alignment of decisions and actions across the entire business [93].

The focus here is not on specific KPIs: the technique is suitable for any business measurements that need to be aggregated handling company's data.

The problem in general can be brought back to comparing providers' performance. In

practice, a collective comparison is related to the "to share or not to share" dilemma Figure 2, an important reason for the failure of data sharing in collaborative networks.

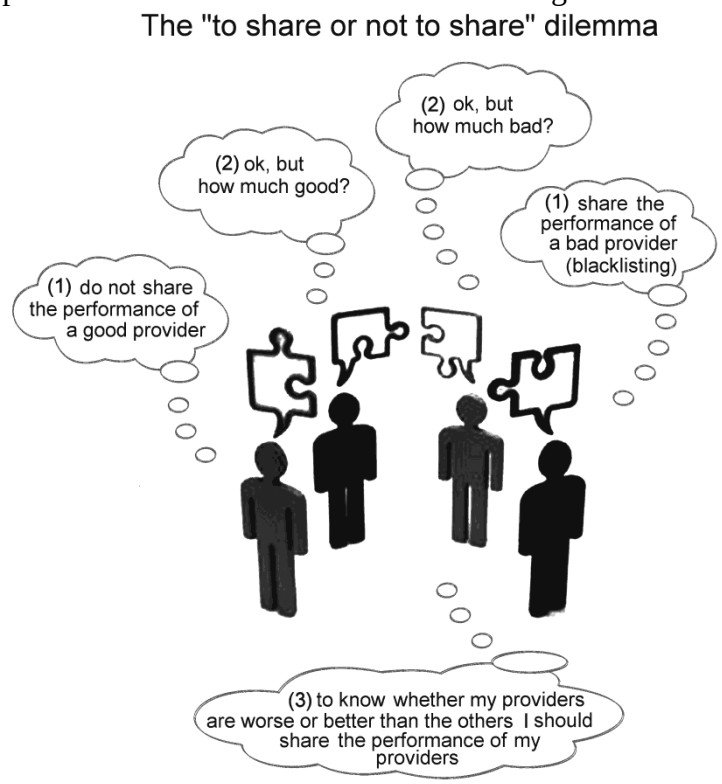


Figure 2 A representation of the "to share or not to share" dilemma between a group of buyers.

In the dilemma, a typical buyer does not like to share the performance of his good providers (keeping a competitive advantage over its rivals) and likes to share the performance of a bad provider (showing his collaborative spirit). However, each buyer knows a subset of the providers available on the market. The fundamental question of a buyer is: how much are my providers good/bad? To solve this question, providers' performance should be shared. This way, buyers with good providers would lose the competitive advantage. Given that nobody knows the absolute ranking of his providers, to share this knowledge is risky and then usually it does not happen.

In the literature, this problem is often characterized as "Value System Alignment" [98]. Values are shared beliefs concerning the process of goal pursuit and outcomes, and depend on the standard used in the evaluation. An example of value model is the economic value of objects, activities and actors in an e-commerce business. There are a number of methodologies and ontologies to define value models supporting BPs [99]. CN are typically formed by heterogeneous and autonomous entities, with different set of values. As a result, to identify partners with compatible or common values represents an important success element. However, tools to measure the level of alignment are lacking, for the following reasons: (i) the collection of information to build a model can be very difficult; (ii) the models are not easy to maintain and modify; (iii) if there are many interdependencies between values, the calculation becomes very time consuming because often it demands a record of past behavior that might not be available.

Our solution comes from perturbing business data via digital stigmergy. Stigmergy allows masking plain data by replacing it with a mark, a data surrogate keeping some

original information. Marks enable a processing schema based on the principle of self-aggregation of marks produced by data, creating a collective mark. Stigmergy allows protecting data privacy, because only marks are involved in aggregation, in place of original data values. Moreover, the masking level provided by stigmergy can be controlled so as to maximize the usability of the data itself.

The Collaborative Analytics System (called hereafter "System" for the sake of brevity) is the main pool located on a shared server and coordinating pools of registered buyers. Each buyer's pool is located on a private server.

The main goal of the data flow is to create a public collective mark by aggregating buyers' private marks. This aggregation process protects buyers' mark from being publicized. More specifically, at the beginning the System randomly extracts a buyer and generates a fictitious collective mark. A fictitious mark is a mark created from artificial data that mimics real-world data, and then cannot be distinguished from an actual mark in terms of features. The collective mark is then anonymously sent to the extracted buyer, who adds his private mark to it and ask the System for the next buyer. The system will answer with a randomly extract next buyer. Then, the buyer sends anonymously the collective mark. This way, the collective mark is incrementally built and transferred from a buyer to another one, under orchestration of the System. Each buyer is not aware of his position in the sequence. This is because the first extracted buyer receives a fictitious collective mark, and because the sender is always anonymous. The last extracted buyer will be provided with a fictitious buyer by the system. Such fictitious buyer actually corresponds to the System itself. After receiving the collective mark, the System subtracts the initial fictitious mark, thus obtaining the actual collective mark, which is then processed (so as to extract some common features) and sent to all buyers. By comparing the collective mark with his private mark, each buyer will be able to assess his position with respect to the collective performance. The results of this process can be used by to select a partner whose performance is higher than the collective performance.

Application of the Stigmergic Architecture

Let us consider a *real value* – such as a price, a response time, etc. – recorded by a firm as a consequence of a business transaction. As discussed in previous section, to publicize the plain value with the associated context may provide advantages to other firms over the business competition. In this context, data perturbation techniques can be efficiently used for privacy preserving. In our approach a real value is represented and processed in an information space as a *mark*. Thus, marking is the fundamental means of data representation and aggregation. In Figure 3 the structure of a single triangular mark is represented. Here, a real value x_j, recorded at the time t by the j-th firm, is represented with dotted line as a mark of intensity $I(t)(x)$ in the firm's private space. A triangular mark is characterized by a central (maximum) intensity I_{MAX}, an extension ε, and a durability rate θ, with $\varepsilon>0$ and $0< \theta <1$, where ε and I_{MAX} are the half base and the height of the triangular mark, respectively.

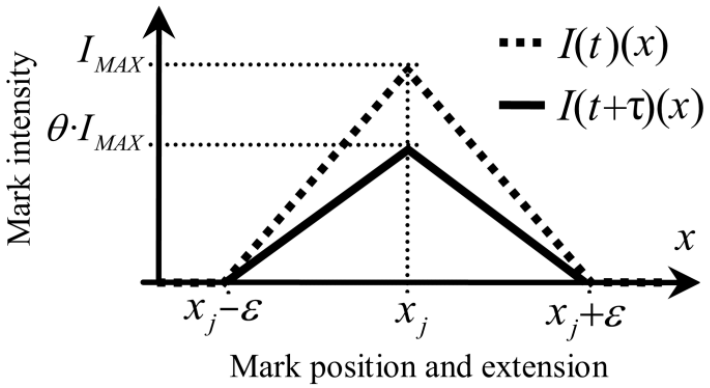


Figure 3 A single triangular mark released in the marking space by a marking agent (dotted line), together with the same mark after a temporal step (solid line).

Figure 3 shows, with a solid line, the same mark after a period τ. In particular, the mark intensity spatially decreases from the maximum, corresponding with the recorded value x_j, up to zero, corresponding with the value of $x_j \pm \varepsilon$. In addition, the intensity released has a durability rate, θ, per step, as represented with the solid line. More precisely θ corresponds to a proportion of the intensity of the previous step. Hence, after a certain decay time, the single mark in practice disappears.

Let us consider now a series of values, $x_j^{(t)}$, $x_j^{(t+\tau)}$, $x_j^{(t+2\tau)}$, ..., recorded by a firm as a consequence of a series of business transactions. Marks are then periodically released by *marking agents*. Let us suppose that each firm has a private marking space and a private marking agent. The decay time is longer than the period, τ, by which the marking agent leaves marks. Thus, if the company holds very different values in the series, the marking agent releases marks on different positions, and then the mark intensities will decrease with time without being reinforced. If the company holds an approximately constant value, at the end of each period a new mark will superimpose on the old marks, creating a lasting mark. More formally it can be demonstrated that the exact superimposition of a sequence of marks yields the maximum intensity level to converge to the stationary level $I_{MAX}/(1-\theta)$ [5]. For instance, with $\theta = 0.75$ the stationary level of the maximum is equal to $4 \cdot I_{MAX}$. Analogously, when superimposing N identical marks of different companies, we can easily deduce that the intensity of the *collective mark* grows with the passage of time, achieving a collective stationary level equal to N times the above stationary level.

Figure 4 shows four private marks (thin solid lines) with their collective mark (thick solid line) in three different contexts, created with $I_{MAX} = 10$, $\varepsilon = 0.3$, $\theta = 0.75$. In Figure 4(a) the private marks have a close-to-triangular shape, with their maximum value close to $I_{MAX}/(1-\theta) = 4 \cdot I_{MAX} = 40$. It can be deduced that, in the recent past, record values were very close and almost static in the series. As a consequence, also the collective mark has a shape close to the triangular one, with a maximum value close to $N \cdot 40 = 160$. We say *reference private marks* and *reference collective mark* when marks are exactly triangular, because they produce the highest marks. Figure 4 (b) shows a sufficiently static context, where record values in the recent past were not very close and not very static. For this reason, private marks have a rounded-triangular shape and the collective mark has a Gaussian-like shape. Finally, Figure 4 (b) shows an actual market context, where private and collective marks are very dynamic.

The first important observation is that Figure 4 (a) and Figure 4 (b) do not present

privacy problems, because all companies have similar performance because their providers are equivalent. In Figure 4 (c) there is dynamism but also a structural difference between companies: two of them have better performance. Here, the reference private marks and the reference collective mark are also shown, with dashed lines and located at the barycenter of the collective mark. It is worth noting that the contrast between marks and reference marks is a quite good indicator of the position and the dynamism of each company in the market. The two best companies are at the right of the reference private mark. Furthermore, all companies are in a dynamic context, because the shape of their marks is far from the triangular one. Finally, comparing the shapes of the reference collective mark and the collective mark, it can be also deduced the amount of overall dynamism.

We can associate some semantics to the parameters of a mark. A very small extension ($\varepsilon \to 0$) and a very small durability rate ($\theta \to 0$) may generate a Boolean processing: only almost identical and recent records can produce collective marking. More specifically to increase the extension value implies a higher uncertainty, whereas to increase the durability value implies a higher merging of past and new marks. A very large extension ($\varepsilon \to \infty$) and a very large durability rate ($\theta \to 1$) may cause growing collective marks with no stationary level, because of a too expansive and long-term memory effect. Hence, the perturbation carried out by stigmergy can be controlled so as to maximize the usability of the data itself while protecting the economic value of information.

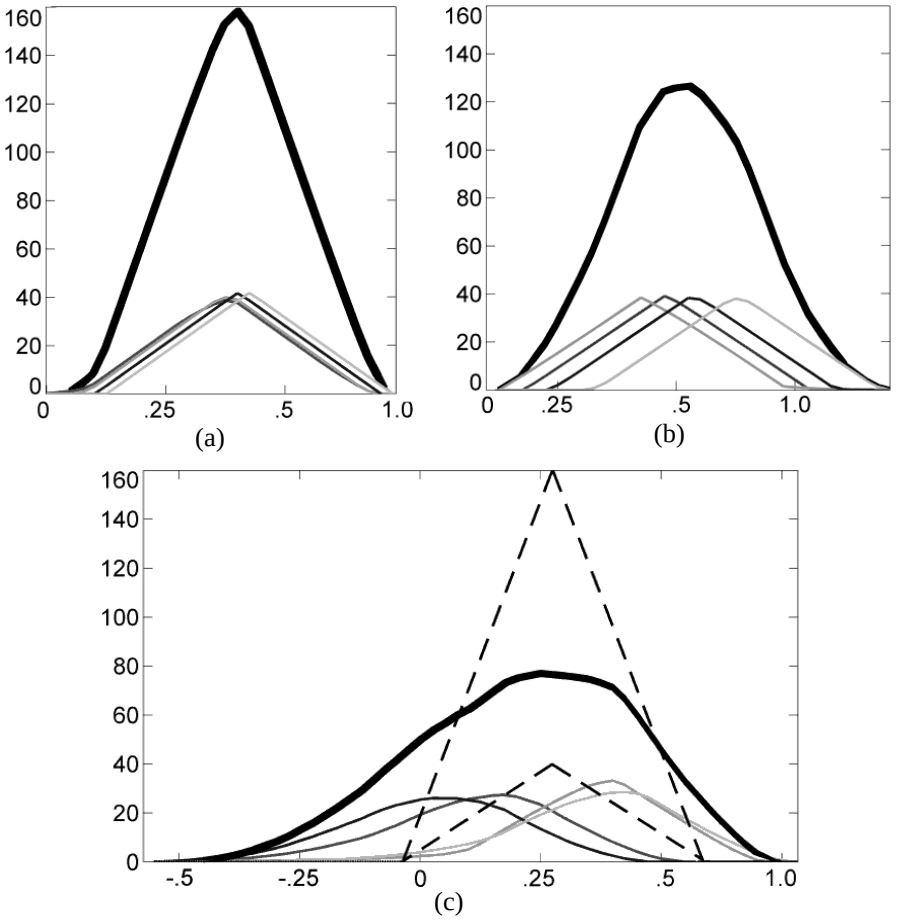
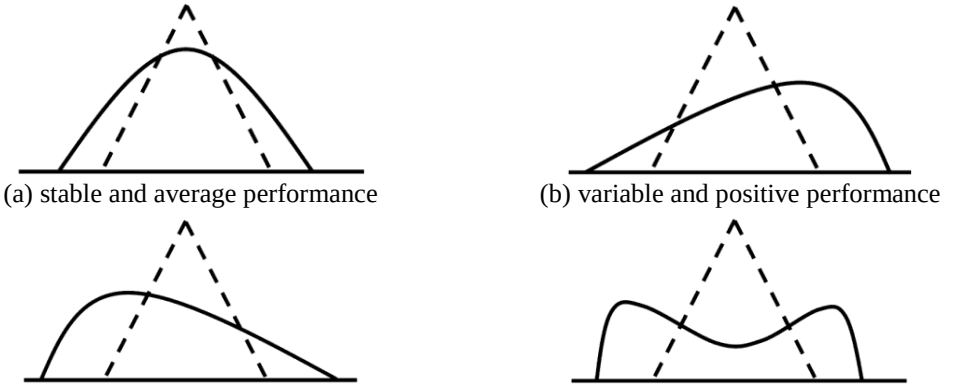


Figure 4 Four private marks (thin solid lines) with their collective mark (thick solid line) in different contexts: (a) very static; (b) sufficiently static; (c) dynamic with reference marks (dashed line). I_{MAX} = 10, ε = 0.3, θ = 0.75.

To summarize the approach, Figure 5 shows the classification of four recurrent patterns in marking, based on the proximity to a triangular shape and to a barycentric position of the mark (solid line) with respect to the reference mark (dashed line).
Exploiting the above observations, in the following, we discuss how a different type of agent can recognize the patterns of Figure 5: the *analytics agent*. Basically, the analytics agent is responsible for assessing the similarity and the integral difference of a mark with respect to the corresponding reference mark, as represented in Figure 6. More formally, given a reference mark, *A*, and a mark, *B*, their similarity is a real value calculated as the area covered by their intersection (colored dark gray in the figure) divided by the area covered by the union of them (colored light and dark gray). The lowest similarity is zero, for marks with no intersection, whereas the highest is one, for identical marks. The barycentric difference is the normalized difference between the right and the left areas of the mark with respect to the barycenter of the reference mark.


(a) stable and average performance (b) variable and positive performance

(c) variable and negative performance (d) very dynamic and balanced performance

Figure 5 Classification of four recurrent patterns in marking, based on the proximity to a triangular shape and to a barycentric position of the mark (solid line) with respect to the reference mark (dashed line).

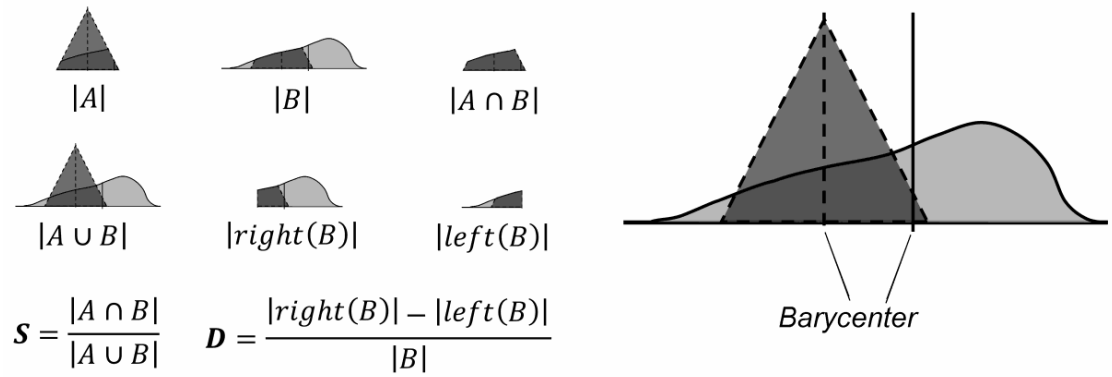


Figure 6 Representation of Similarity (S∈[0,1]) and barycentric Difference (D∈[-1,1]) of a mark (B) with respect to the corresponding reference mark (A).

Thus, the proximity to a triangular shape can be then measured by the similarity, whereas the barycentric position of the mark with respect to the reference mark can be assessed by means of the barycentric difference, as represented in Figure 7.

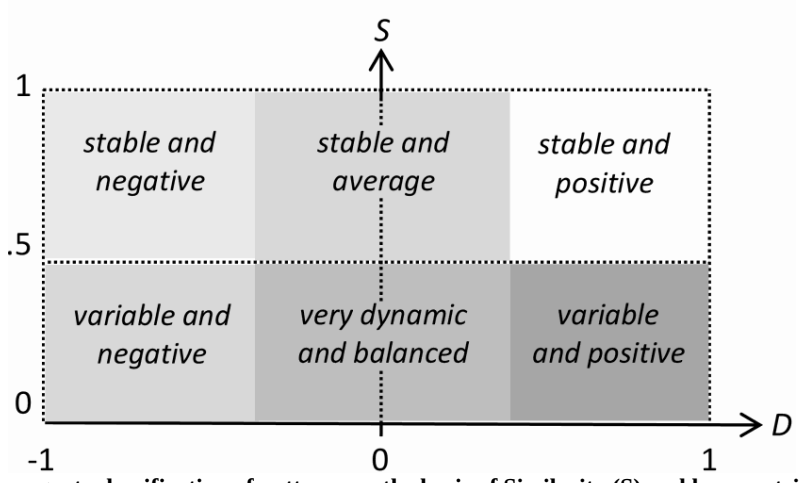


Figure 7 Analytics agent: classification of patterns on the basis of Similarity (S) and barycentric Difference (D).

Results

In this section we adopt e the KPI productivity as an example of partners' performance, and we show a numerical example of processing of such KPI, performed by the marking agent and the analytics agent. The numerical example is based on the publicly available dataset *Belgian Firms*[1], containing 569 records each characterized by four attributes: capital (total fixed assets), labour (number of workers), output (value added) and wage (wage cost per worker) [100]. Starting from raw data, the KPI *productivity* has been first calculated as output divided by labour. Then, 7 clusters representing provider companies have been derived by using the Fuzzy C-Means algorithm. Subsequently, 4 buyers have been supposed, and each buyer has been connected to three providers.

Figure 8 shows the output of the marking agent in terms of private marks (solid gray lines), collective mark (solid black line), and reference marks (dotted lines), with different extension values: (a) $\varepsilon = 30$ for all buyers; (b) $\varepsilon = 60$ for B1 and $\varepsilon = 30$ for the others. In the figure, the buyer B1 has been highlighted with a larger thickness. It can be noticed that the different extension values sensibly modifies the shape, and then the perturbation, of the buyer's private mark.

[1] http://vincentarelbundock.github.io/Rdatasets/doc/Ecdat/Labour.html

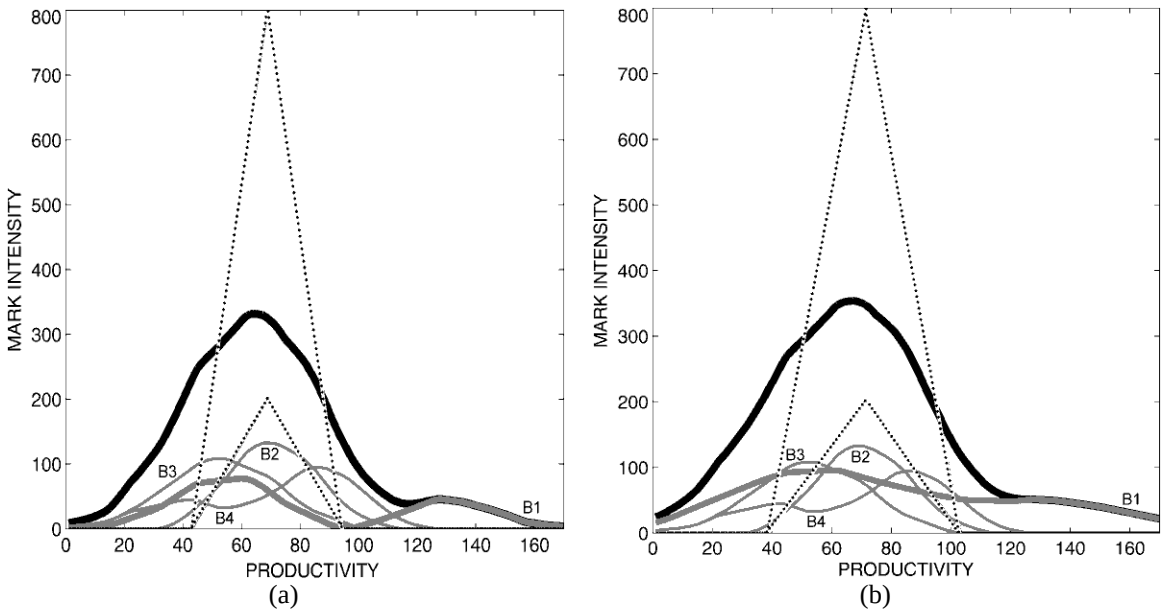


Figure 8 Belgian firms scenario: four buyers' private marks (solid gray lines), collective mark (solid black line), and reference marks (dotted lines), with different extension values: (a) ε = 30 for all buyers; (b) ε = 60 for the buyer B1 (with larger thickness) and ε = 30 for the others.

Table 5 shows the patterns recognized by the analytics agent. It is worth noting that, despite the different level of perturbation that affected the buyer B1, there are no differences in the Performance patterns detected.

Table 5 Performance patterns of each buyer, with respect to Similarity (S) and barycentric Difference (D) for the Belgian Firms scenario.

	S	D	Performance pattern
B1	0.26	-0.07	dynamic and balanced
B2	0.73	-0.08	stable and average
B3	0.37	-0.58	variable and negative
B4	0.31	-0.20	dynamic and balanced

(a)

	S	D	Performance pattern
B1	0.32	-0.03	dynamic and balanced
B2	0.77	-0.01	stable and average
B3	0.36	-0.64	variable and negative
B4	0.39	0.15	dynamic and balanced

(b)

3.2 Indoor Monitoring of Elderly People

Problem Statement

Recent prospects of the world population show clear trends tending towards more elderly people and single households [101], which have substantial effects on public and private health care, emergency services, and the individuals themselves. In this context, Ambient Assisted Living (AAL) is currently one of the most important research and development areas. It aims at applying ambient intelligence technology to enable people with specific demands and elderly to live in their preferred environment longer and safer [102]. The possibility of monitoring the health status of elderly people living alone in their houses is a core service of AAL scenarios [103]. This possibility optimizes the prevention of emergencies, which can have important effects on public and private healthcare services. Examples of emergencies are falls, leading to immobilization, cardiac arrest, or helplessness: when unnoticed for hours, they may lead to severe follow-up complications. Age-related chronic diseases, such as dementia, depression, cardiac insufficiency, or arthritis, can be faced in a proactive and preventive way in order to let patients take advantage of more adequate assistance services [104] [105].

Among the possible physical conditions to be monitored by an automatic monitoring system in AAL scenarios, emergency situations and chronic diseases are the most relevant. These two kinds of situations can be clearly distinguished on the basis of different perspectives. Prompt detection and timely notice are fundamental requirements of an emergency, which usually occurs in a short time. In contrast, a disease is characterized by a gradual detection of long-term deviations from the typical behavior or by critical trends in the user's vital parameters. Active user involvement (e.g. pressing buttons on wearable alarm devices) can be appropriate while dealing with an emergency, but it is not acceptable for disease situations, which are initially characterized by a lack of noticeable symptoms and then by the absence of an emotional involvement that could activate decision-making.

In the literature of AAL, a number of systems have been developed for automatic detection of in-home situations, embracing two broader paradigms: Context Awareness and Ambient Intelligence [106] [107] [108]. In AAL systems, data sources can be different parametric sensors (e.g., location, movement, interaction, vital data, etc.), which feed a multi-level and hierarchic processing [106]. There are many possible parameters, derived from sensors that can be intrusive and whose management costly. Moreover, their tracking is often related to specific situations to detect. In practice, monitoring elderly behavior in real-time requires choosing a trade-off between what to monitor and how to notice it. Actually, much work still has to be done before such systems can be used on a regular basis. One of the most important lessons learned from these efforts is that the parametric aggregation must use a limited amount of states, be highly flexible and able to handle uncertainty [104] [109] [110].

When dealing with uncertainty, the logic of automatic detection is intrinsically different between emergency and disease.

An emergency event can either be true or false. As a Boolean event, the single emergency has no doubt in the human experience. Uncertainty arises in an automatic system from the question whether or not the system perception correctly classified the event. The system assessment can be based on the frequency that an event is actually true or false over a high number of cases. Instead, in a disease situation, uncertainty is an aspect of human experience, characterized by a ‘‘gray zone’’ where the situation can

be classified as both true and false. The system assessment can be based on the similarity degree of an event to the two classes, based on the proximities of an observation to the different truths. The final point is that emergency modeling involves conventional single-valued classification, whereas disease modeling involves multi-valued classification. Another relevant difference between emergency event and disease situation is related to the standardization versus personalization of the logic. When detecting emergency events, it is relatively easy to reuse logic for many individuals.
In contrast, there are significant differences between individuals when detecting disease situations.
In this work we propose a novel approach to unusual behavior detection, with a focus on disease situations. In the literature of behavioral monitoring and health state assessment, a great standardization effort has been done by means of the so-called Activities of Daily Living (ADL). ADLs are daily activities carried out by individuals, such as feeding, dressing, sleeping, walking, watching TV, etc. [111] [112], which act as a basis to represent habits of healthy people. Health professionals can thus refer to the ability or inability to perform ADLs as a measurement of the functional status of people with disabilities. Most AAL research is currently carried out with the purpose of allowing software systems/agents to detect ADLs on the basis of suitable processing, reasoning and manipulation of sensors data. One of the most important sources of information to infer ADLs is the position of the elderly moving in his home during daily living activities. While in outdoor scenarios Global Positioning System (GPS) constitutes a reliable and easily available technology, in indoor scenarios GPS is largely unavailable. For this reason, several systems have been proposed for indoor localization. These algorithms fuse information coming from different sources in order to improve the overall accuracy [113] [114]. In the literature, each solution has advantages and shortcomings, which, in most cases, can be summarized in a trade-off between precision and installation complexity (and, thus, costs). The proposed approach to unusual behavior detection aims at overcoming the inherent imprecise output of a localization system implementing a source-agnostic software infrastructure. The main goal is to be able to handle position information coming from any kind of localization system with different accuracies and precisions. In order to assess the - proposed solution in dealing with different localization systems, we modeled the localization error of real indoor localization systems in order to use it as a test of reliability and insensitiveness to perturbations on the input data.
For instance, in [115] stigmergy has been used for tracing the intelligent navigation of people in ubiquitous computing environments. Here the idea of digital pheromone trials is adopted with the purpose of finding the optimal route from the history of people's behavior. Although the problem considered is different with respect to our problem, the paper is an example of implementation of the emergent paradigm with stigmergy: to find the answer to a problem in people's emergent behavior, rather than in the environment via a cognitivist strategy.
In this work, we show how an emergent approach can be implemented by discussing a multi-level scheme for the detection of disease situations, structured into four levels of information processing. The first level is managed by a generic localization system with different accuracies and precisions. The second level is in charge of a marking subsystem leaving marks in the environment in correspondence to the position of the person. The accumulation of marks creates an aggregated mark, observed by a perception subsystem, which compares the current aggregated mark with a reference aggregated mark. The reference aggregated mark is a stigmergic track, sampled in a

period determined by a relative and a healthcare professional, in which the elderly is stable in her health conditions. Finally, a detection subsystem processes the similarity in order to extract indicators of a behavioral change. For a better distinction of the progress of unfolding deviation events, we adopt a smoothed activation function for behavioral changes. Figure 9 shows an UML activity diagram of the overall information processing. Here, the localization and the monitoring systems are represented as a rectangle. The monitoring system is in turn made of the subsystems: environment, marking, perception, and detection. Activities (represented by gray oval shapes) are connected via data flow (dashed arrow) through input/output data objects (white rectangles). The black circle represents the initial step, while the black circle with white border represents the final step. More specifically: (i) localization takes a person (in our scenario, the elderly) and provides his position; (ii) marking takes a position and produces a mark; (iii) aggregating takes marks and provides aggregated marks; (iv) comparing takes reference and aggregated marks and provides similarity between them; (iv) activation takes similarity and produces events; (v) collecting takes events and produces event collections; (vi) and finally scoring takes event collection and provides anomaly.

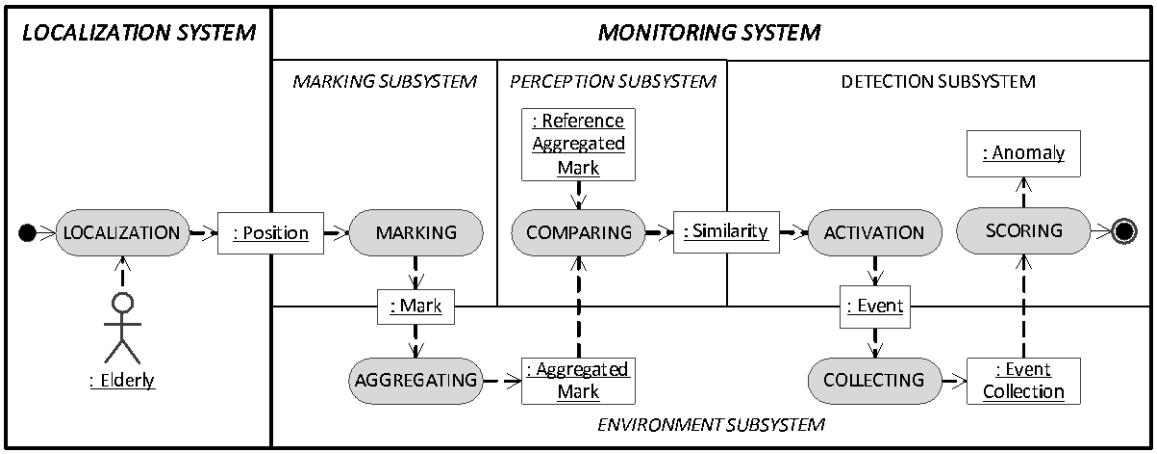


Figure 9 UML activity diagram of the macro activities of the proposed approach to anomaly detection.

The system does not characterize specific disease situations: it alerts relatives or healthcare professional on a behavioral change that might be better investigated in person. This strategy is characterized by the use of an unobtrusive positioning system, a very standardized logic, and a broad-spectrum monitoring. We prove that the proposed technique is quite insensitive to the different levels of localization error.

Error modeling of indoor localization systems

Localization is a key component for achieving context-awareness. Recent years have witnessed an increasing trend of location-based services and applications. In most cases, however, location information is limited by the accessibility to Global Navigation Satellite Systems (GNSS), largely unavailable for indoor environments. Recently, an international competition on localization systems for Ambient Assisted Living (AAL) scenarios has been created (EvAAL - Evaluating AAL Systems through Competitive Benchmarking) [116]. The objective of this competition is to reward the best indoor localization system from the point of view of AAL applications [116].

In this work we selected the three best localization systems presented at EvAAL, namely CPS [117], n-Core [118], and RealTrac [119], and we used the estimated positions by these systems as an input to the monitoring system. The description of their systems is as follows:

- CPS is a device-free localization and tracking system, where people to be located do not carry any device. It is based on a wireless sensor network that uses a tomographic approach to localize the users. A static deployed wireless network measures the Received Signal Strength (RSS) on its links and locates people based on the variations caused by the movements of people.
- The n-Core localization system exploits the RSS and the LQI (Link Quality Indicator) measures between a mobile unit worn by the user and a static deployed ZigBee wireless network. The system applies a set of locating techniques to estimate the position of each mobile unit in the monitored environment. These locating techniques include signpost, trilateration, as well as a fuzzy logic.
- The RealTrac localization system exploits the time-of-flight (ToF) and the RSS measures between a mobile unit worn by the user and a static deployed wireless network. In particular, ToF and RSS measures are processed by the server using a particle filter that also takes into consideration the structure of the building, the air pressure value and the inertial measurement unit data.

In order to evaluate how the proposed stigmergy-based technique performs when applied with a real localization system, we modeled the localization error introduced by the selected localization systems and we apply it to the real positions of the monitored users. The selected traces (pair of coordinates) analyzed to model the localization error introduced by the localization systems were collected during the EvAAL competitions. In particular, the traces are related to a typical AAL scenario where the user is alone in the house, he moves along a path from one room to another, including some waiting points, where the user stands for at least 5 seconds. Table 6 shows the statistics of the localization systems evaluated during the EvAAL competition. In particular, the mean, the variance, and the percentiles of the localization error (the distance between the real point where the actor is and the estimated coordinates) are shown.

Table 6 Performance statistics: mean, variance, and percentiles in meters of the localization error for the selected systems during the EvAAL

	Mean error (m)	Error variance (m)	First quantile (m)	Second quantile (m)	Third quantile (m)
CPS	0.5333	0.0562	0.3658	0.5196	0.7129
n-Core	1.0133	0.2863	0.5939	0.9471	1.2869
RealTrac	1.3525	0.5714	0.8700	1.1891	1.7340

In order to model the errors that the selected localization systems undergo, we analyze the distribution of the error values of each system. To test the behavior of the localization error, we analyzed about 400 position's measures for each system. Table 7 reports the values of the skewness (a measure of the asymmetry of the probability distribution) and of the kurtosis (a measure of the shape of a probability distribution) for

the error along the x and y direction, respectively. As shown in Table 7 the skewness and the kurtosis values are near to 0 and 3, respectively (except for the n-CORE system along the y direction). This indicated that the probability distribution function (hereafter PDF) of the error along the x and y direction should follow the Gaussian distribution.

Table 7 Skewness and kurtosis values of the localization error for the selected systems during the EvAAL competition.

	Axis	Skewness	Kurtosis
CPS	x	0.19	2.34
	y	-0.35	3.09
n-Core	x	0.0815	2.1434
	y	0.7479	4.0761
RealTrac	x	-0.22	2.55
	y	-0.19	2.56

Figure 10, Figure 11, and Figure 12 show the scatter plots of the estimated coordinates together with the histogram bar plots evaluated along the x-y directions, for each localization system. From this preliminary analysis we supposed that a bivariate Gaussian distribution could model the error distribution of the selected localization systems. Indeed, we superimpose to each scatter plot a bivariate Gaussian distribution with mean and covariance shown in Table 8 and estimated from the traces collected during the EvAAL competition.

Table 8 The parameters chosen for the bivariate Gaussian distributions.

	μ_x	μ_y	Covariance	
CPS	-0.0274	0.0314	0.2283	
			0.0320	0.1116
n-Core	0.0362	0.4608	0.6240	-0.1026
			-0.1026	0.4783
RealTrac	0.0239	-0.1469	1.1083	-0.1115
			-0.1115	0.7500

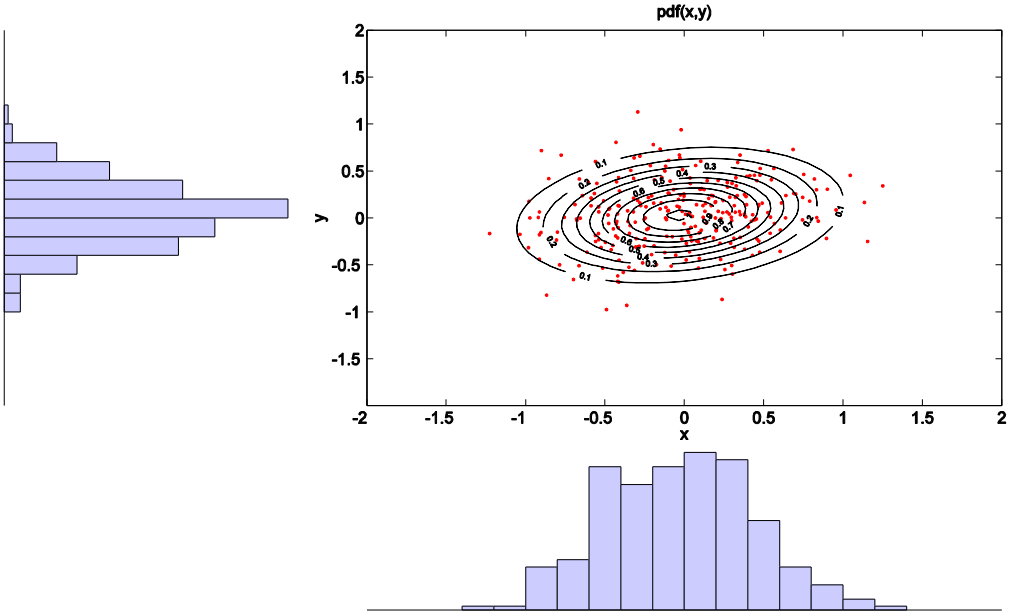


Figure 10 Scatter and histogram bar plots of the CPS localization system.

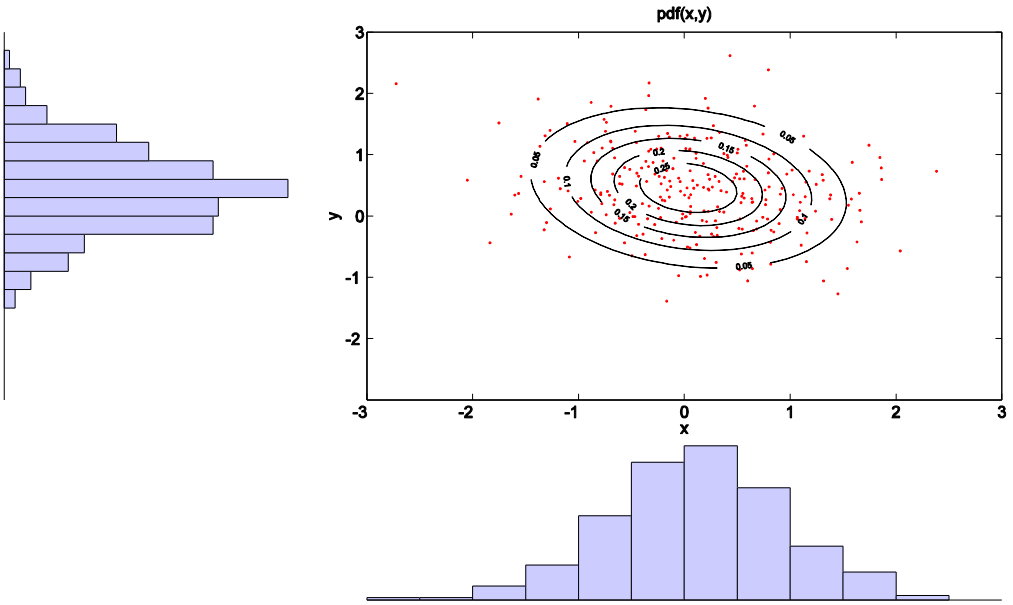


Figure 11 Scatter and histogram bar plots of the n-Core localization system.

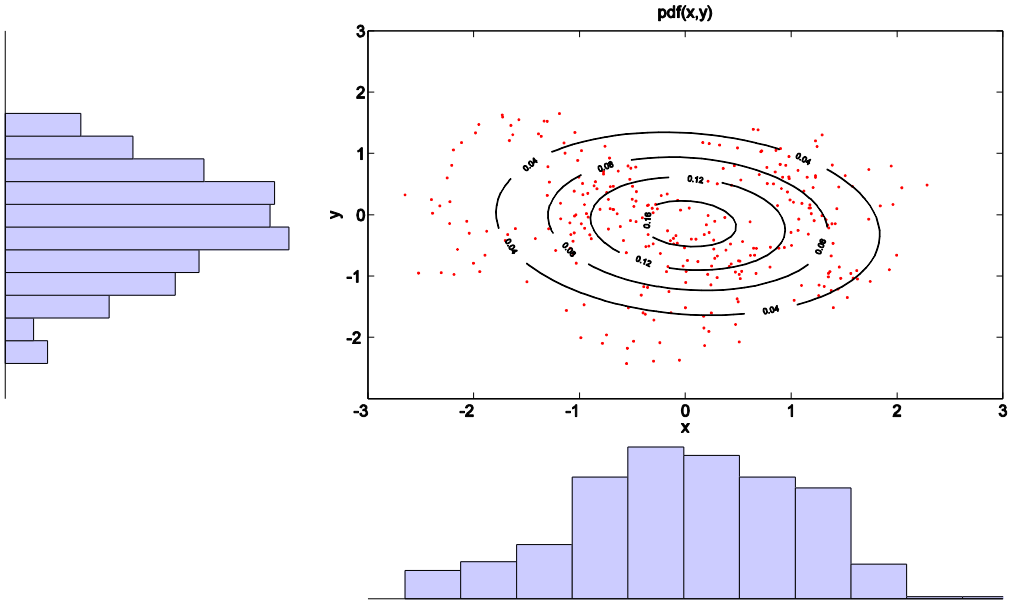
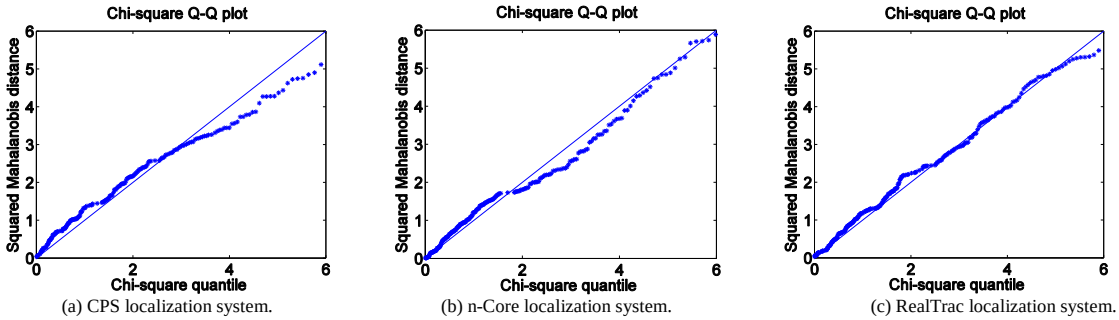


Figure 12 Scatter and histogram bar plots of the RealTrac localization system.

If a set of variables is distributed as a multivariate normal, then each variable must be normally distributed. However, when all individual variables are normally distributed, the set of variables may not be distributed as a multivariate normal [120]. Hence, testing each variable for univariate normality only is not sufficient. The best-known method of assessing the degree to which multivariate data deviate from multinormality is the Mardia's tests [121]. This method allows testing the null hypotheses that the traces are compatible with the assumption of multinormality, based on sample measures of multivariate skewness and kurtosis. Moreover, we tested the assumption of multinormality by using also other two tests: the Kolmogorov-Smirnov [122] and the Doornik-Hansen [123] normality test. The Kolmogorov-Smirnov statistic quantifies a distance between the empirical distribution functions of two samples, while the Doornik-Hansen is a powerful alternative to the Shapiro-Wilk test and, like the Mardia's test, is based on the skewness and kurtosis of multivariate data.

In Figure 13 the chi-square quantile-quantile plot is shown. For each localization system, we plot the squared mahalanobis distances against corresponding quantiles of the limiting chi-square distribution. If data are distributed as a multivariate normal, then the points should fall on a straight line with slope one and intercept zero. Outliers can be visually detected; indeed, they will show up as points on the upper right side of the plot for which the Mahalanobis distance is notably greater/lesser than the chi-square quantile value.



(a) CPS localization system. (b) n-Core localization system. (c) RealTrac localization system.

Figure 13 Quantile–quantile plot of the squared Mahalanobis distance versus the corresponding quantiles of the chi-square distribution.

Although the localization errors exhibit distributions which slightly deviate from a multivariate normal distribution (Figure 10, Figure 11, Figure 12, and Figure 13), we verified that the localization errors have been corroborated to be bivariate Gaussian distributed. Indeed, we successfully test, at 0.025 significance level, that the error traces and the bivariate Gaussian distributions (with the parameters estimated in Table 8) pass the Mardia, the Doornik-Hansen, and the Kolmogorov-Smirnov tests. From here on, we will assume that the bivariate Gaussian assumption of the localization error is valid.

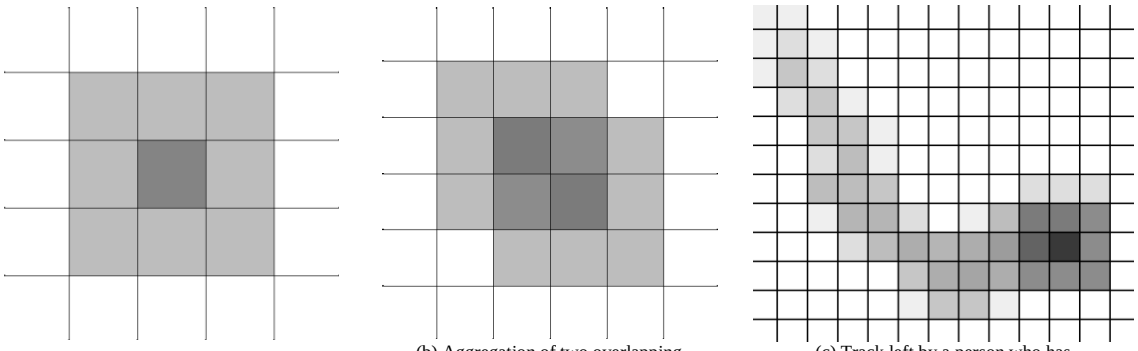
Application of the Stigmergic Architecture

In this Section, we report on the monitoring system, designed according to the emergent paradigm. More specifically, we adopt the principles of the marker-based *stigmergy*, which, in social insect colonies, employs chemical markers (pheromones) that the insects deposit on the ground in specific situations. Multiple deposits at the same location aggregate in strength. Members of the colony who perceive pheromones of a particular flavor may change their behavior. Pheromone concentrations in the environment disperse in space and evaporate over time, because pheromones are highly volatile substances.

Marker-based stigmergy can be employed as a powerful computing paradigm exploiting both spatial and temporal dynamics, because it intrinsically embodies the time domain. Moreover, the provided mapping is not explicitly modeled at design-time and then it is not directly interpretable. This offers a kind of information blurring of the human data, and can be enhanced to solve privacy issues. Furthermore, analog data provided by marker-based stigmergy allows measurements with continuously changing qualities, suitable for multi-valued classification.

The marking process

In the *marking* process, a mark structure is encapsulated by a marking subsystem. A marking subsystem takes as an input coordinates generated by the localization system at the micro-level and leaves marks in a computer-simulated spatial environment, thus allowing the accumulation of marks. Consider the entire localization error model as the input for the marking subsystem implies a lot of statistical processing that needs to be done prior to deploying the actual AAL monitoring system and will impact the applicability of the proposed system in new environments, where different localization solutions may be available. While the marking process need to be transparent and not tied up to a specific localization system.

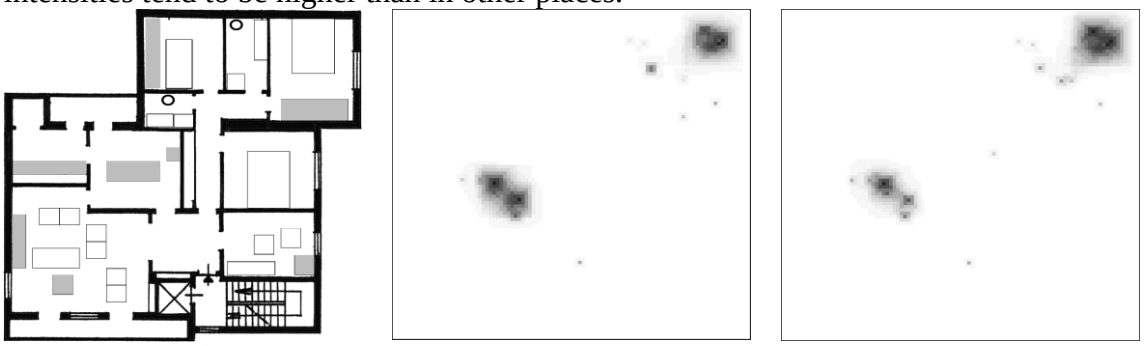


(a) Single mark. (b) Aggregation of two overlapping marks. (c) Track left by a person who has stopped after a brief walk.

Figure 14 Basic scenarios of the marking process.

Figure 14 shows some basic scenario of the marking process. More specifically, Figure 14 (a) shows the structure of a single mark. The levels of mark intensity are represented by different gray gradations: the darker the gradation is, the higher the intensity of the mark. The highest intensity of the mark, I_{MAX}, is in the middle, which corresponds to the position of the person when the mark is left. Mark intensity decreases with the number of squares from the position of the person, of a percentage σ (called *spatial decay*) for each square. Further, mark intensity has a *temporal decay*, which means a percentage τ of decrease after a period of time. Hence, an isolated mark after a certain time tends to disappear.

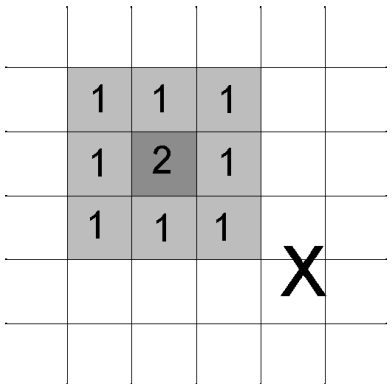
Marks are periodically left by the marking subsystem, with frequency *v*. The time that a mark takes to disappear is longer than the period used by the marking subsystem to release a new mark. Hence, if the user is still in a specific position, new marks at the end of each period will superimpose on the old marks, thus increasing the intensity up to a stationary level. It can be demonstrated that the exact superimposition of a sequence of marks yields the maximum intensity level to converge to the stationary level I_{MAX} / τ [5]. If the person moves to other locations, consecutive marks will be partially superimposed and intensities will decrease with the passage of time without being reinforced. Figure 14 (b) shows two consecutive and overlapping marks, and Figure 14 (c) shows the track left by a person who has stopped after a brief walk. The area with the highest intensity (on the right bottom) corresponds to the place where the person is still. The track with lower intensity is the area where the person was moving. Thus, when the person is still, the superimposition of marks causes their intensities sum up, and then the resulting intensities tend to be higher than in other places.



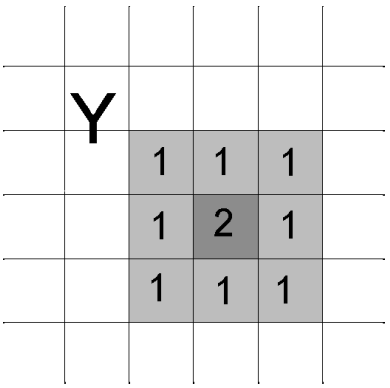
(a) Pilot scenario: layout of the apartment. (b) Stigmergic track in a normal day. (c) Stigmergic track in a day with disease progression.

Figure 15 Two scenarios of marking process in a real-world apartment with an elderly with some risk of disease progression.

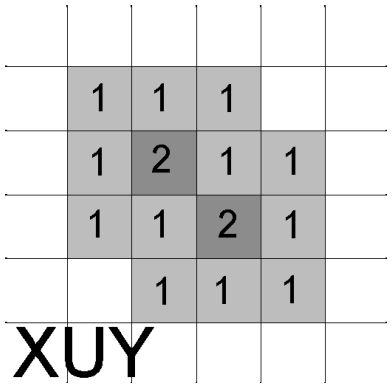
The stigmergic track can then be considered as a short-term and a short-size action memory. The marking level allows capturing a coarse spatiotemporal structure in the domain space, which hides the complexity and the variability in data. As a real-world pilot scenario, Figure 15 (a) shows the layout of an apartment where an elderly with some risk of disease progression has been monitored. Here, a black or gray region represents non-walkable areas (e.g., wall, wardrobe, TV, etc.), whereas a white region represents an area where the person can walk or stay (e.g., floor, bed, armchair, etc.).
Figure 15 (b) shows the stigmergic track generated in the morning of a normal day, for two hours and a half, up to 10.30 A.M. Here, a relevant intensity is located on two points of the apartment, which might correspond to Point of Interest (POI). Figure 15 (b) shows the stigmergic track generated in the same day of the week for the same timetable, when the person had disease symptoms. Here, top-right track is larger and the bottom-left track is smaller than the corresponding track in the normal day of Figure 15 (b).



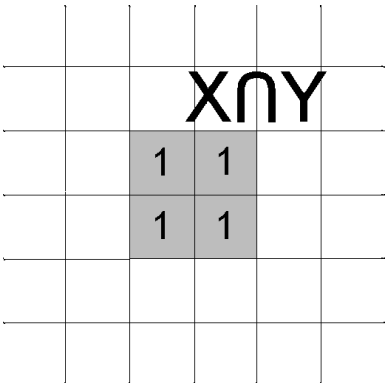
(a) $|X| = 1\times 8 + 1\times 2 = 10$.



(b). $|Y| = 1\times 8 + 1\times 2 = 10$



(c) $|X \cup Y| = |\{\max(X,Y)\}| = 16$.



(d) $|X \cap Y| = |\{\min(X,Y)\}| = 1\times 4 = 4$.

Figure 16 An illustrative example of similarity between two consecutive marks,
$S = |X \cap Y| / |X \cup Y| = 4/16 = 0.25$.

In the above example, it is clear that a stigmergic track provides comprehensive information that can be handled to automatically detect behavioral changes without explicit activity modeling, with simple processing, and preserving privacy. An in-depth (cognitivist) investigation reveals that the above-mentioned disease's symptoms are sleep changes and loss of energy, causing more sleep and late breakfast, with a shift of about 20 minutes with respect to the normal day. Indeed, the two major tracks are placed on the bed (top-right) and on the living room (bottom-left).

The perception process

At the second level there is the *perception* process, consisting in the sensing of the track accumulated in the environment at the macro-level. Here, we take advantage of stigmergy (computed at the first level) as a means of information aggregation of the human spatiotemporal tracks. Indeed, the process of information aggregation is a vehicle of abstraction, leading to the emergence of high-level concepts beyond occurring fluctuations. Such fluctuations can be caused by the underlying localization system, but mostly by physical, mental, cultural, and lifestyle differences between individuals. The perception subsystem performs a comparison, called *similarity*, between the current and a reference track. In other terms, similarity aims at sensing the variation of the current behavior situation with respect to what was judged a normal behavior. The normal behavior of the elderly is established in a period of stable health conditions by a relative and a healthcare professional. Of course, since the normal behavior can change in the long-term, the related reference track can be updated when necessary.

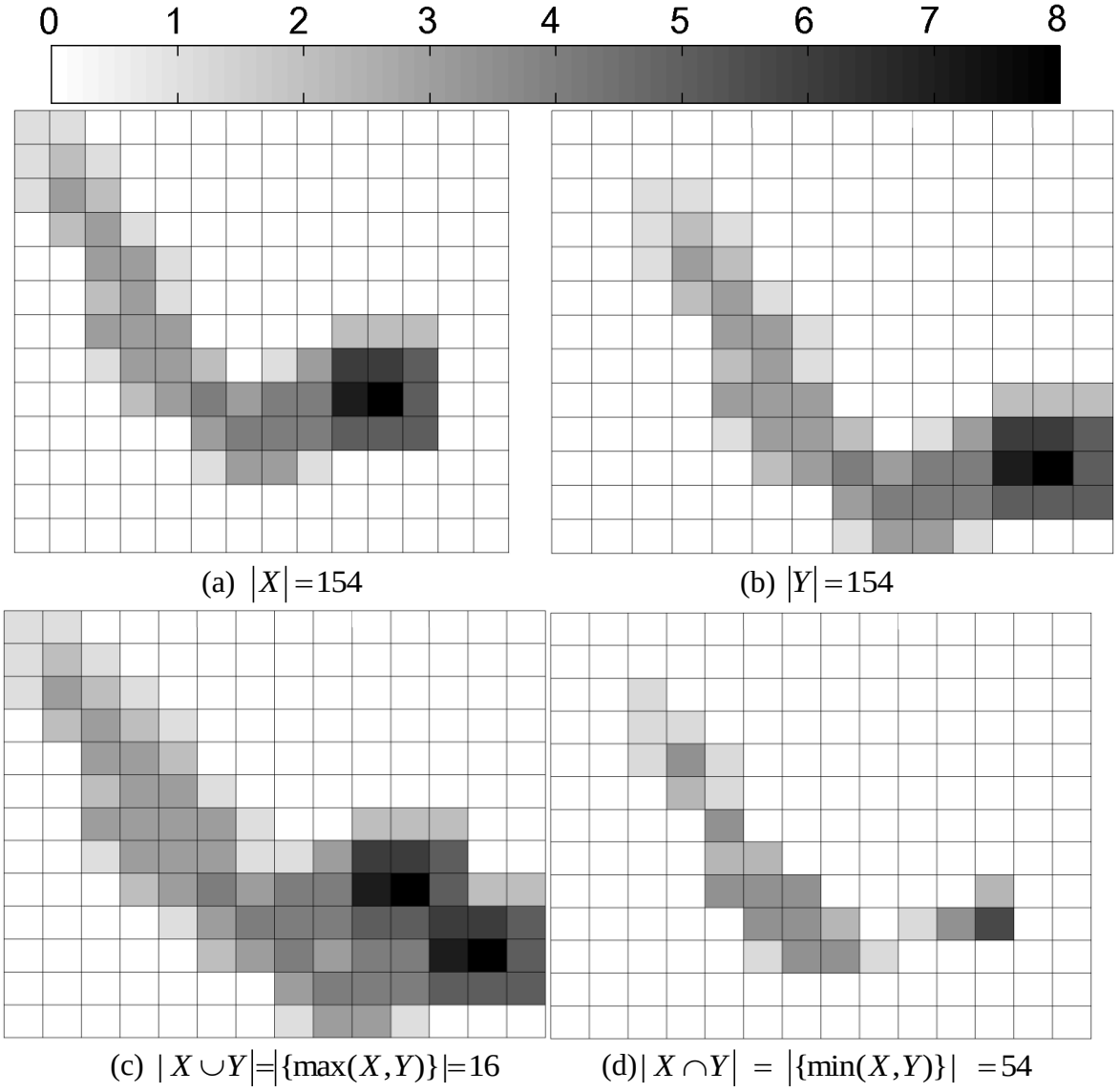


Figure 17 An illustrative example of Similarity between tracks, $S = |X \cap Y| / |X \cup Y| = 54/256 = 0.21$

Figure 16 shows a three-dimensional representation of the similarity implemented by the perception subsystem. Formally, given two marks *X* and *Y*, their similarity is a real value calculated as the volume covered by their intersection ($X \cap Y$ in the figure) divided by the total volume (the union of them). The lowest similarity is zerofor tracks with no intersection, whereas the highest is one for identical marks.

Figure 17 shows an illustrative example of similarity between tracks. More specifically, Figure 17 (a) shows the track of Figure 17 (c), whereas Figure 17 (b) shows the same track shifted two cells right and two down. Figure 17 (c) and Figure 17 (d) shows their intersection and union, respectively.

Figure 18 shows the similarity values in a time frame of about 6 hours, between a track in a normal day of the week and a track in the same day of the week with a disease progression. A sample of the two tracks in a specific instant of time is shown in Figure 15 (b) and Figure 15 (c) Here, the reduction in the similarity values in the interval 90-130 is due to the differences discovered in Figure 15 (b) and Figure 15 (c).

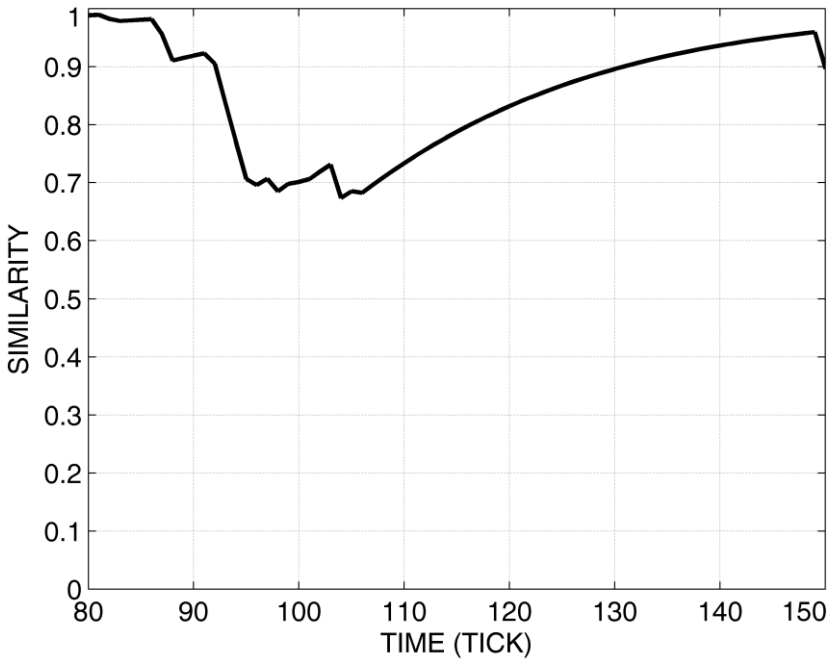


Figure 18 Similarity function for the two marks represented in Figure 17 over a time frame of about 6 hours.

The detection process

The anomaly detection is handled at the third processing layer, called *detection*, involving the discovery of patterns from the similarity provided by the perception layer. The detection subsystem provides a domain-related output. More specifically, we apply to similarity a smoothed activation function. The term "activation function" is taken from the neural sciences and it is related to the requirement that a signal must reach a certain level before a processing layer fires to the next layer. A smoothed activation

function allows achieving a better distinction of the critical phenomena during unfolding deviation events, with a better detection of progressing levels of the anomaly. For this purpose we employ the *s-shaped* activation function to the similarity output. An example of activation function is shown in Figure 19, with α = 0.7 and β = 0.8, considering the following definition:

$$f(x)=\begin{cases}0, & \text{if } x \le \alpha \\ 2\times\dfrac{(x-\alpha)^2}{(\beta-\alpha)^2} & \text{if } \alpha \le x \le \dfrac{\alpha+\beta}{2} \\ 1-2\times\dfrac{(x-\alpha)^2}{(\beta-\alpha)^2} & \text{if } \dfrac{\alpha+\beta}{2} \le x \le \beta \\ 1, & \text{if } x \ge \beta \end{cases} \quad \textbf{17}$$

Figure 20 shows the resulting output when applying the activation function of Figure 19 to the similarity of Figure 18. As an effect of the s-shape activation function, values lower than α are further decreased, whereas values higher than β are further amplified, in order to evidence major dissimilarity. Each sample of the s-shaped similarity of Figure 20 is then associated to one of two event classes: *Positive* (i.e., behavioral deviation) and *Negative* (i.e., no deviation). The actual anomaly is established by the *Daily Positive Rate*, an online anomaly score defined as the percentage of positive samples with respect to the total samples of the day. The Daily Positive Rate supports the decision process of the healthcare professional with a kind of augmented perception, because it increases when the normal behavior of the elderly is affected by significant deviations. The next section provides real-world examples and related illustrations of the different processing layers.

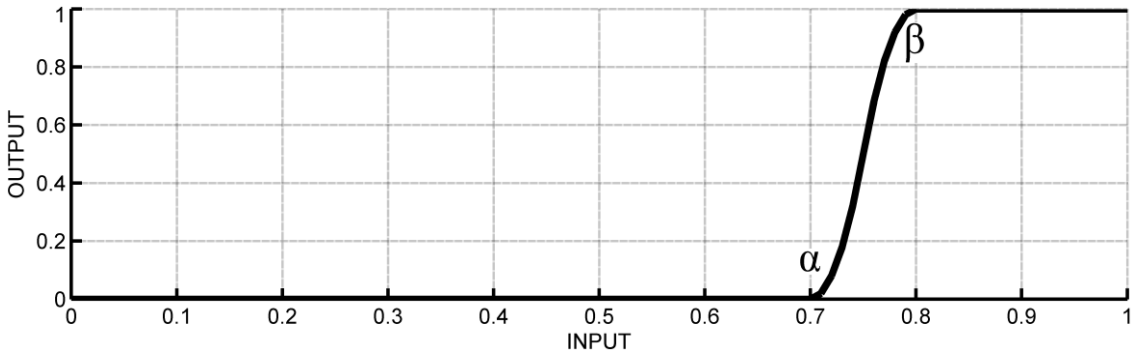


Figure 19 S-shape activation function with α = 0.7 and β = 0.8.

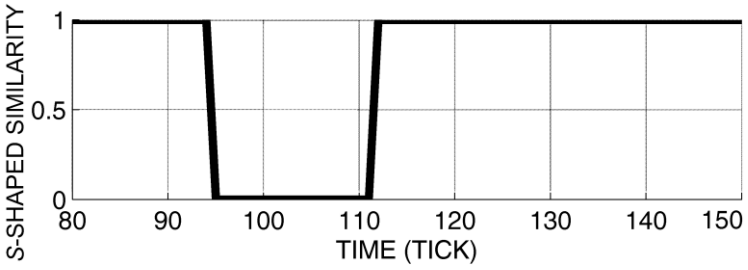


Figure 20 S-shaped similarity.

Results

This Section shows a real-world application on an elderly with some risk of disease progression, living alone in the apartment of Figure 15 (a). The experimentation aims at assessing the effectiveness of the system in recognizing a reliable measurement of the deviations from the routine indoor activities, by applying different localization systems. In order to assess the effectiveness of our system, human-based observation and system-based detection must be connected for each sample of the considered time period. In this way, the results provided by the system could be interpreted thanks to data and metadata provided by the human observation. The following subsections explain how experimental data were collected and how the system assessment was carried out.

Despite of the advantages of ambient assisted living, identifying human paths in home remains a difficult issue due to privacy-related concerns. In order to preserve the privacy of the patient and to avoid interfering with his daily life, positioning and behavioral data were reported by a relative of the patient. The relative was in charge of observing every week off-line video tracks of the elderly, living alone and independently. The video tracks were provided by a collection of cameras, placed one per room. More specifically, first, continuous video acquisition sessions were made, each lasting for a week. For each week, a list of behavioral deviations occurred in the session was extracted by the relative. After some sessions were collected, two sessions were selected by the relative: a week considered as a healthy period (i.e., with no significant behavioral deviations), and a week with some behavioral deviation. For the two selected weeks, the position of the elderly in the apartment was sampled by the relative, every 5 minutes (i.e., 12 samples per hour) and with 0.6 meters of uncertainty. Thus, we were supplied at most with 24 x 12 = 288 actual position samples per day. When the patient is away from home, there are no position samples available and then there is no system output. To assess the robustness of our system, different indoor positioning systems were simulated by taking into account the actual position samples. In order to evaluate how the proposed monitoring system performs with different localization systems, we applied the different localization techniques modeled in previous section to the real position of the monitored users. This way, we derived three cases having exactly the same mobility scenario, but differing on the localization error. Indeed, the purpose of the experimentation is to show that our monitoring system is not sensitive to the particular localization technology.

Table 9 Main parameters set in the tuning session.

	Symbol	Value
Maximum mark intensity	I_{MAX}	20
Size of the cells grid	-	100 × 100 cells
Spatial decay	σ	10 cells (≈1.4 m)
Temporal decay	τ	0.1 (single mark life ≈30 periods)
Marking period	v-1	5 min
S-shape parameter	α	0.4
S-shape parameter	β	0.4

The healthy session was used as a tuning session, whereas the other session was used for testing. Hence, during the first session, the main parameters were set. Table 9 shows the main parameter values used. More specifically, the spatial decay was set to 10 cells, which is about 1.4 meters in the discretized marking space of 100 x 100 cells. Since we want to be independent of the localization systems we chose the mark size equal to the maximum localization error of the analyzed systems. This value represents an upper bound of the localization error along the x direction of the RealTrac system.
The setting process of this parameter is quite simple: it depends on the expected position uncertainty of the localization systems. Indeed, 1.4 meters proved large enough to cover such uncertainty, while being sufficiently small to distinguish the closest point of interests. The maximum mark intensity is usually provided with the same scale value of the spatial decay (i.e., 20) in order to reduce the approximation errors in computing numbers that have different orders of magnitude. However, the maximum mark intensity is a very insensitive parameter. Temporal decay was set so as to provide sufficient historical memory in a single mark: 30 periods (i.e., 150 minutes). This duration proved sufficient to cover the largest behavioral deviations occurring with disease progression. Indeed, the setting of this parameter is a two-step process. First, the healthcare professional establishes how disease progression is commonly manifested, and the relative contextualizes such candidate symptoms on the elderly to monitor. This allows establishing an order of magnitude of the parameter value. Second, behavioral deviations are simulated in order to assess a more precise value. In this stage, the low sensitivity of the parameter on the set value is also assessed. More specifically, with the *CPS* localization system, the system performance (F-measure) calculated with a temporal decay of 0.09, 0.10 and 0.11 are 0.729, 0.728, 0.727 respectively. We set the two s-shape parameters to the same value because both the healthcare professional and the relative established that there are no transients in the expected deviations. In the second phase, significant deviation events are artificially generated to assess a more precise value, and the low sensitivity on the set value has been verified.
More specifically, with the *CPS* localization system, the exact value 0.4 was selected after a very few alternatives determined as follows: the lower bound must be higher than the baseline, whereas the highest bound must fit the most significant deviation events. The baseline is computed by the similarity between two corresponding healthy days of the week, e.g., two Mondays, because it represents the noise generated by micro behavioral differences during a healthy period. Finally, the system performance (F-measure) calculated with α and β equals to 0.3, 0.4, and 0.5 are 0.72, 0.73, and 0.69, respectively.

For the reasons above discussed, the approach followed is semi-supervised, because samples are assumed only for normal data. Essentially, a basic setting is established via heuristic supported by the healthcare professional, whereas close-to-optimum values are established by using simulation and sensitivity. This approach avoids configuring a system specialized on specific behavioral deviations. Indeed, disease cases are usually rare, and then the use of them for training would cause over fitting. Although we tested the system on a specific disease, the behavioral deviations simulated can occur in many diseases. It is worth noting that the anomaly is not detected via the output of the activation function: it is assessed via the Daily Positive Rate, an anomaly score combining many activation events, which is discussed in the next subsection.

Table 10 Behavioral deviations observed in the testing session.

Id	Day	Start-end	Duration	Description of the observed behavioral deviation
1	Tue	185–204	20	He had a shorter lunch
2	Wen	096–100	5	He woke up later in the morning
3	Wen	150–158	9	He had a shorter lunch
4	Wen	181–230	50	He had a longer nap and shorter tasks in the afternoon
5	Thu	098–107	10	He woke up later in the morning
6	Thu	150–153	4	He had a shorter lunch
7	Thu	195–240	46	He had a longer nap and shorter tasks in the afternoon
8	Fri	092–095	4	He woke up later in the morning
9	Fri	108–119	12	He carried out less housekeeping tasks
10	Fri	170–173	4	He had a longer nap in the afternoon
11	Fri	190–216	27	He carried out less tasks in the afternoon
12	Fri	246–250	5	He had a shorter dinner
13	Sat	103–107	5	He woke up later in the morning
14	Sat	158–164	7	He had a shorter lunch
15	Sat	188–242	55	He had a longer nap and shorter tasks in the afternoon
16	Sun	109–113	5	He woke up later in the morning
17	Sun	119–122	4	He did not carry out self-care tasks
18	Sun	153–156	4	He did not carry out self-care tasks
19	Sun	241–246	6	He had a shorter dinner

The goal of this Section is to measure a match between the behavioral deviations annotated by the human observer in Table 10 and the correspondent results provided by the system. For this purpose, each output sample was considered as a point belonging to one of two classes: Positive (*P*, i.e., behavioral deviation) and Negative (*N*, i.e., no deviation).

Considering only the period when the patient was at home, the total number of available samples in the experimentation week was 1822, of which 1544 were negative samples and only 278 were positive samples. Class unbalancing is common in the AAL domain, because behavioral deviations must be discovered at the early stage when there are few symptoms.

Table 11 shows the confusion matrices generated by using the three localization systems. Here, each column counts the instances in a predicted class (i.e., *P*' and *N*'), while each row represents the instances in an actual class (i.e., *N* and *P*). Thus, the diagonal cells (gray) count the number of correct classifications made for each class, and the off-diagonal cells (white) count the errors made. The former are called True Positives (*TP*) and True Negatives (*TN*), whereas the latter False Positives (*FP*) and False Negatives (*FN*), considering correct predictions as true, and wrong predictions as false.

Table 11 Confusion matrices.

(a) CPS				**(b) n-Core**				**(c) REALTrac**			
	Prediction				Prediction				Prediction		
		P'	*N'*			*P'*	*N'*			*P'*	*N'*
Actual	P	216 (TP)	62 (FN)	Actual	P	221 (TP)	57 (FN)	Actual	P	221 (TP)	57 (FN)
	N	99 (FP)	1445 (TN)		N	114 (FP)	1430 (TN)		N	134 (FP)	1410 (TN)

Table 12 shows the most important offline assessment indicators of the system. More specifically, *accuracy* is the classification rate, which is the number of correctly classified samples with respect to the total number of samples. This indicator is often high when dealing with a small (positive) class against a large (negative) class, because the latter dominates the ratio despite of the results on the positive class.
Two better indicators are *precision* and *recall*, meaning the number of behavioral deviations correctly returned, with respect to the total returned and to the total actually occurred, respectively. As both indicators are important, in Table 12 a combination of them, called *F-measure*, is also reported. It is worth noting that the system assessment is very insensitive to the different levels of noise (localization error) coming from the different localization systems.

Table 12 Offline assessment indicators.

	Accuracy $\frac{TP+TN}{TP+FP+TN+FN}$	**Precision** $\frac{TP}{TP+FP}$	**Recall** $\frac{TP}{TP+FN}$	**F-measure** $\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$
CPS	0.912	0.686	0.777	0.728
n-Core	0.906	0.660	0.795	0.721
REALTrac	0.895	0.623	0.795	0.698

Table 13 Online assessment indicators.

Day	**Positive rate =** $\frac{TP+FP}{TP+FP+TN+FN}$								
	CPS			**n-Core**			**REALTrac**		
	TP	**FP**	**Positive rate**	**TP**	**FP**	**Positive rate**	**TP**	**FP**	**Positive rate**
Mon	0	2	0.008	0	3	0.012	0	9	0.036
Tue	15	15	0.115	15	20	0.134	14	21	0.134
Wed	46	9	0.191	47	9	0.194	46	11	0.198
Thu	52	7	0.242	53	10	0.258	52	13	0.266
Fri	34	28	0.215	35	30	0.226	39	34	0.254
Sat	53	8	0.212	54	9	0.219	53	11	0.222
Sun	16	30	0.228	17	33	0.248	17	35	0.257

Table 13 shows true positive, false positive, and the positive rate as an online assessment indicator, averaged per day of a week. Again, the result obtained is very insensitive to the different levels of noise coming from the different localization systems, and consistent with what is shown in Table 12. Indeed, starting from Monday, the number and the duration of the behavioral deviations observed in the testing session are increasing up to Sunday.

In order to directly assess how the output of the system was computed during the testing session, Figure 21, Figure 22, and show the outputs of the perception (analogical black signal), the detection (digital black signal), and the human observation (digital gray signal), when the error model of the analyzed localization systems is applied on each day of the week. Here, it is possible to realize that the system manifests an "inertial" character: chains of behavioral deviation events are better recognized than isolated ones. This is consistent with the memory effect of the marking process. Moreover, it is

possible to realize that events shorter than the event duration considered in the tuning stage are more difficult to be recognized. Actually this is not a problem: for a given localization system, different duration events may be recognized by different instances of the monitoring system with different parameters tuning.

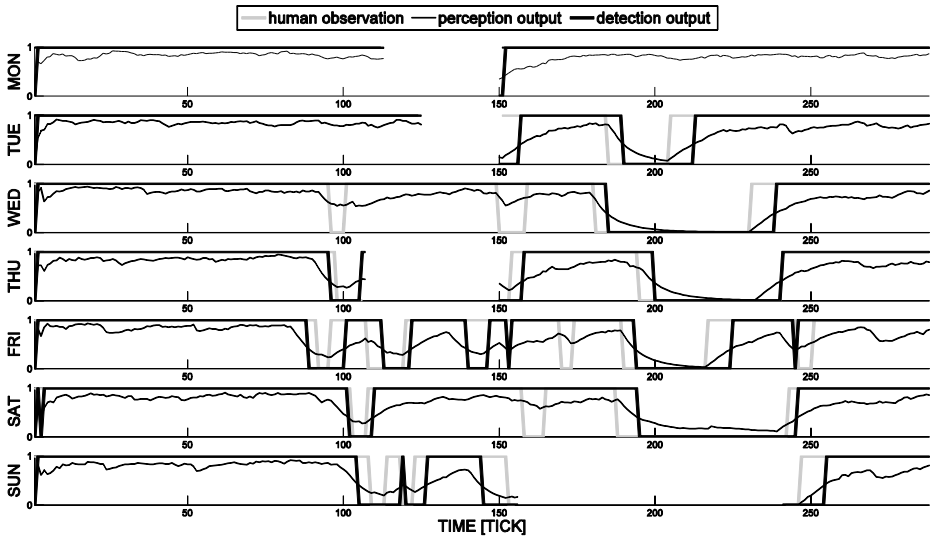


Figure 21 Outputs of the perception, detection, and of the human observation when the error model of the CPS localization system is applied on each day

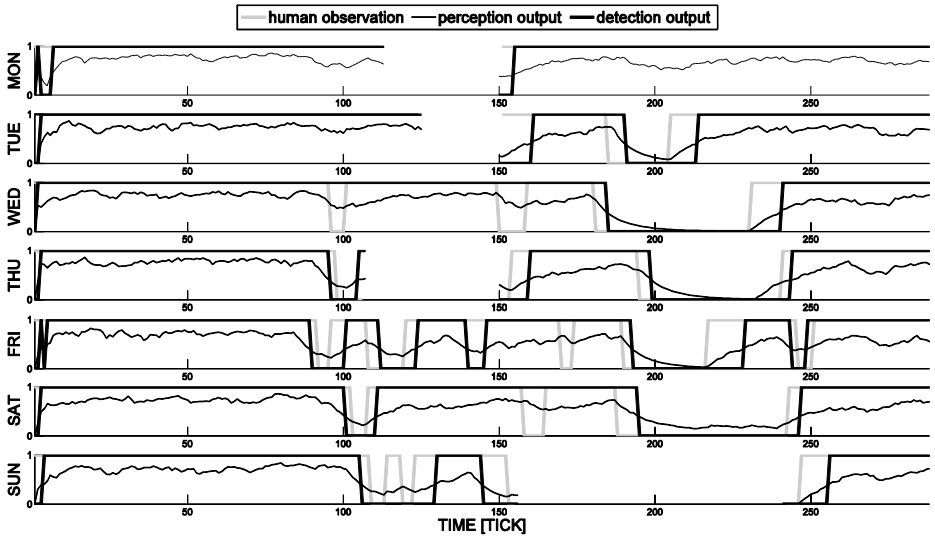


Figure 22 Outputs of the perception, detection, and of the human observation when the error model of the n-Core localization system is applied on each

Marking is a general purpose data processing mechanism that may fit phenomena of different sizes and nature such as: fluctuations of a localization system, transitions between points of interest of an apartment, human mobility patterns, human diseases patterns, and so on. This work focuses on the application of stigmergy to behavioral dynamics in indoor mobility generated by human diseases. For this purpose, the design is not specialized to fit the error of a specific localization system. By experimenting that the used approach is not sensitive with respect to benchmark localization systems, we obtain that localization error is naturally compensated. Similarly, the design of the marking process is not specialized to fit the structure of the apartment: having marks

large enough to distinguish the closest point of interests it is sufficient for our purposes. Different stigmergic layers may be designed for each (sub-)dynamics. However, this would lead to an expensive, reductionist and cognitivist design. For this reason, in this work the stigmergic processing is focused on human mobility patterns, showing that the approach is intrinsically able to tackle other kinds of uncertainty.

Currently, we are designing and developing an adaptive scheme in the marking process and the detection process, in contrast with the current adaptable scheme. More specifically, one of the problems to solve when optimizing parameters is that optimization encompasses all available scenarios at once and concerns the tuning of all parameters over the overall training set, which should be spread across the entire space of diseases. This optimization scheme is usually referred to as global tuning, and leads to increasing difficulties from the practical perspective, due to fitting different scaled spatiotemporal behavioral deviations. An alternative is local modelling. For example, the design paradigm of the *receptive fields* is based on local tuning, more precisely on sub-models that focus predominantly on some selected regions of the entire diseases domain. An overall model is then formed by combining such local models. This modular layer may provide a topology offering a considerable level of flexibility, as the resulting receptive fields can be highly diversified according to the distribution of the different behavioral deviations [124].

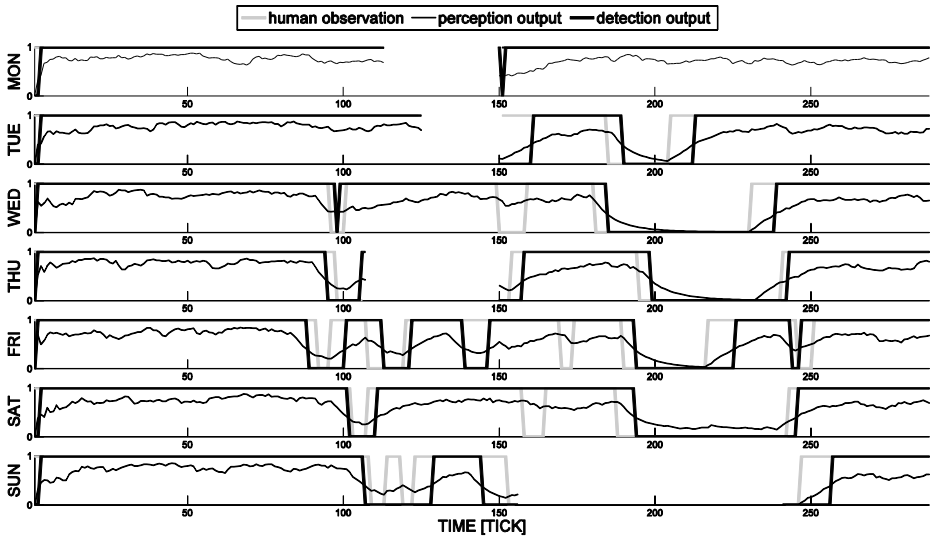


Figure 23 Outputs of the perception, detection, and of the human observation when the error model of the REALTrac localization system is applied on each

3.3 Spatial and Temporal Detection of Urban Traffic Congestions

Problem Statement

To reduce traffic congestion is one of the main issues in the Smart City strategy [125]. Recently more and more advanced sensor techniques for traffic measurement have attracted a number of researchers. The available technology can be basically grouped into two categories: on-vehicle devices and roadside infrastructures. The former mainly employs the Global Positioning System (GPS) and its variants to report real-time information on a vehicle position, whereas the latter typically involves dedicated equipment, such as loop detectors and cameras. Actually, any use of roadside infrastructures is essentially constrained to highways, freeways and primary arteries, and then cannot be used for urban traffic estimation. For this reason, the on-vehicle positioning device is considered as the reference data source in this work. Hence, the availability of vehicle position data is supposed to be a requirement in our approach.
In this work we present a new design of swarm aggregation of vehicle positions based on marker-based stigmergy [7] [6]. We use marker-based stigmergy as a computing paradigm for exploiting both spatial and temporal dynamics that characterize urban traffic. Basically, in our approach each vehicle of a monitored urban network releases periodical marks in a computational environment, according to its position. Marks may aggregate in the environment reinforcing in strength, whereas lose intensity evaporating over time. As a consequence a marks concentration, called track hereafter, appears and stays spontaneously while many stationary vehicles and high density roads occur. For a better scalability and a better distinction of unfolding congestion events, we adopt activation interfaces at the input-output of the stigmergic layer. Here, the term "activation" is taken from neural sciences and it is related to the requirement that a signal must reach a certain level before a processing layer fires to the next layer.
The proposed mechanism works if structural parameters, as mark extension, for example, are correctly tuned for the given application context [126]. Determining such correct parameters is not a simple task since different urban areas have different traffic flux and density. For this purpose, we adopt a tuning mechanism based on differential evolution for adapting parameters to the specific urban area.

Application of the Stigmergic Architecture

Given an urban street network, which can be modeled as a *directed graph*, let us consider the least number of paths such that every link belongs to at least one path. As an example, Figure 24 shows a static view with two paths of the Pisa center urban street network (Italy). In the dynamic view of the system, each path can be modeled as a linear segment, because the position of each vehicle in the path can be measured by the on-road position from the initial point of the directed path.

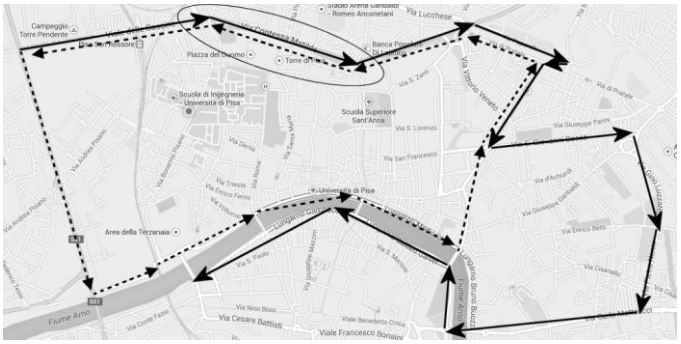


Figure 24 The Pisa center urban area considered for pilot experiments, with two sample paths.

Let us assume that the geo-position $g_{v,t}$, of each vehicle v at the time t in a given urban area is periodically sampled and provided as input to the system. When a traffic congestion event E_k occurs, this is characterized by temporal begin and end, at the instant $\underline{t}_k$ and $\overline{t}_k$, respectively. In addition, at each sampling instant $t \in [\underline{t}_k, \overline{t}_k]$, the on-road positions of the queue head and tail $\overline{s}_k^t$ and $\underline{s}_k^t$ are periodically detected. In conclusion, the output of the system is characterized by a series of traffic congestion occurring events:

$$E_k^{DETECTED} \equiv \{[\underline{t}_k, \overline{t}_k], [\underline{s}_k^{\underline{t}}, \overline{s}_k^{\underline{t}}], ..., [\underline{s}_k^{\overline{t}}, \overline{s}_k^{\overline{t}}]\} \quad \textbf{18}$$

To model the quality of the output, let us distinguish between the actual and the detected (estimated) congestion event. The real event is characterized by the same format but slightly different values, since the detection process is never perfect:

$$E_K^{ACTUAL} \equiv \{[\underline{\tau}_k, \overline{\tau}_k], [\underline{\sigma}_k^{\underline{\tau}}, \overline{\sigma}_k^{\underline{\tau}}], ..., [\underline{\sigma}_k^{\overline{\tau}}, \overline{\sigma}_k^{\overline{\tau}}]\} \quad \textbf{19}$$

Given the above definitions, a fitness function of the system can be also determined, to evaluate how the system approximates the *k*-th event:

$$f_k = \frac{\left|\underline{\tau}_k - \underline{t}_k\right| + \left|\overline{\tau}_k - \overline{t}_k\right|}{\left|\overline{\tau}_k - \underline{\tau}_k\right|} + \sum_{\underline{i} \equiv \min(\underline{\tau}, \underline{t})}^{\overline{i} \equiv \max(\overline{\tau}, \overline{t})} \frac{\left|\underline{\sigma}_k^i - \underline{s}_k^i\right| + \left|\overline{\sigma}_k^i - \overline{s}_k^i\right|}{\left|[\underline{i}, \overline{i}]\right| \left|\overline{\sigma}_k^i - \underline{\sigma}_k^i\right|} \quad \textbf{20}$$

where the left addend represents the absolute differences on the start and end times, normalized with respect to the time interval, whereas the right addend represents the average absolute differences on the head and tail of the queues, normalized with respect to the queues length and the number of samples. It is worth noting that f_k=0 for a perfectly detected event and that in general f_k is a positive real number. The overall quality of the model is then measured by the average fitness on all events:

$$Fit = 1/K \cdot \sum_k f_k \quad \textbf{21}$$

Actually the system may generate: (i) detected events which do not correspond to actual events (false positive); (ii) undetected actual events (false negative). To find the match between actual and detected events we choose at each step the pair that minimizes *Fit*. Unmatched events are also entirely considered as contribution.
Finally, given the above definitions, the problem is to detect all the traffic congestion events with the lowest fitness.

The input activation interface

The role of the input activation interface is to take vehicle positions $p_{v,t}$ (generated from $g_{v,t}$ via projection on a linear path) to establish whether they should be processed by the stigmergic layer or not and the marks intensity. For this purpose, the interface adopts the

concept of *hypothetical track*, placed on the vehicle position to assess whether two hypothetical tracks generated on two consecutive positions of the vehicle overlap, whether a mark can be released. Figure 25 shows a scenario of two overlapping hypothetical tracks centered on the vehicle positions, with the form of isosceles trapezoid, whose height, upper and lower bases are 1, β and 2β, respectively. Here, the vehicle covered the distance δ between $p_{v,t-1}$ and $p_{v,t}$. It can be demonstrated that, when the two hypothetical tracks overlap, the ordinate of the cross point of their diagonal edges, called intensity coefficient γ, is:

$$\gamma_{v,t} = \min\{1,\ 2-\delta/\beta\} \in [0,1] \qquad 22$$

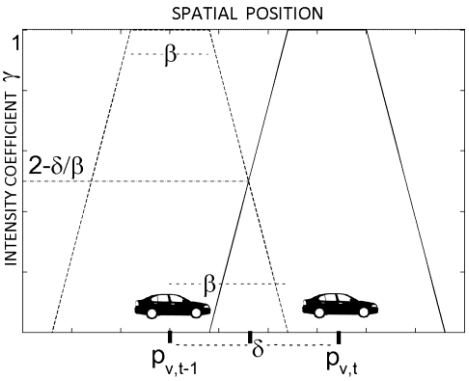


Figure 25 A scenario of the input activation interface with hypothetical tracks.

The input activation interface works according to the following rule: when two consecutive hypothetical tracks overlap, then the stigmergic layer is activated and supplied with the pair ($p_{v,t}$, $\gamma_{v,t}$).

The stigmergic layer

Figure 26 shows a mark (with solid line) released by the vehicle v_1 in the path P_k, at the position $p_{v1,k}$, which is characterized by a central (maximum) *intensity* $\gamma_{v,t} \cdot I$, an *extension* $\varepsilon > 0$, and an *evaporation* $\theta \in [0,1]$. Figure 27 shows, with a gray line, the same mark after a step of marking. More precisely, θ corresponds to a proportion of the intensity of the previous step. Hence, after a certain decay time, the single mark in practice disappears. The decay time is longer than the step of marking. Thus if the vehicle is still, a new mark superimpose on the old marks, creating a track, whose intensity will reach a stationary level. In contrast, if the vehicle speed is sufficiently high, the mark intensities will decrease with time without being reinforced.

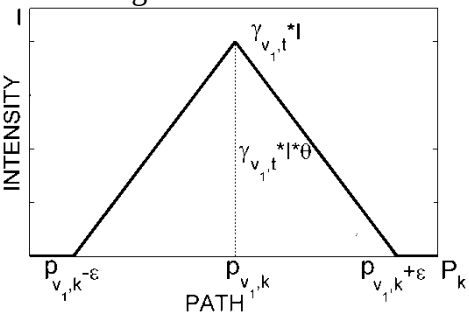


Figure 26 A single mark released in the marking space (solid line), together with the same mark after a step of decay (gray line).

Similarly, two vehicles sufficiently close and still superimpose their marks. Figure 27 shows an example of track (the overlying non-triangular shape) generated by two vehicles releasing marks (the two underlying triangular shapes) at different instants of time. More precisely: vehicle v_1 released, at the previous time t-1, a mark which accordingly evaporated by a factor θ, whereas vehicle v_2 released a mark at the current time, t, close to the mark of the vehicle v_1.

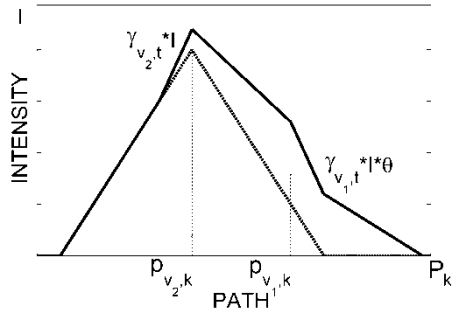


Figure 27 Two marks released by two close vehicles (triangular shapes) with the corresponding track (overlying non-triangular shape).

It can be realized from Figure 26 and Figure 27 that mark extension and evaporation take into account the mobility and the proximity of vehicles, and that the track is a kind of short term memory of the activities on a path. It is worth nothing that a single and stationary vehicle (e.g. parking) can produce a stationary mark with poor intensity with respect to a queue of vehicles, which in contrast can lead to a high aggregation of marks.

The output activation interface

The process of information aggregation leads to abstraction and emergence of high-level concepts beyond occurring micro-fluctuations. The output activation interface allows achieving a better distinction of the critical phenomena during unfolding traffic congestion, with a better detection of the progressing levels. For this purpose we apply a sigmoidal activation function to the track intensity:

$$\Sigma(I_k) = 1/(1+e^{-\alpha(I_k-\phi)}) \quad \textbf{23}$$

Figure 28 shows the activation function with inflection point ϕ = 120 and different values of α. As an effect of the activation, values of the intensity higher than ϕ are further amplified to evidence major congestion effects, whereas values lower than ϕ are further decreased to hide minor queues. In both cases, micro fluctuations are smoothed. The parameter α controls the inflection slope, and then the width of the "gray zone", to deal with uncertainty in data: a high value makes the activation Boolean (suitable for stable events) whereas a low value enhances the multi-class or "fuzzy" character of the output, which is useful to reduce information hiding when upper processing layer are available. As an example, Figure 29 shows vehicle positions, track intensity and congestion degree for the road highlighted with an oval in Fig. 1.

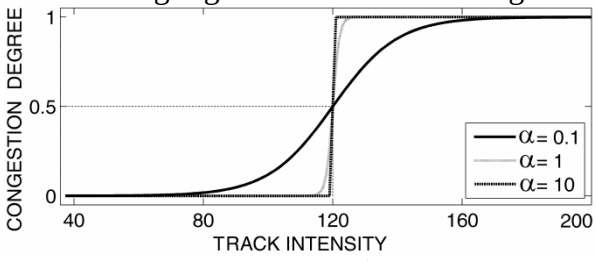


Figure 28 Sigmoidal activation function with ϕ = 120 and different values of α.

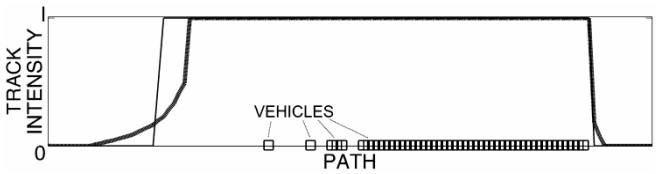


Figure 29 Vehicles positions, track intensity (thick line), congestion degree (thin line), for the road highlighted with an oval in Fig.1.

The design of the parametric adaptation

The swarm aggregation designed in previous section involves a number of structural parameters to be set appropriately for each given application context. Determining such correct parameters is not a simple task since different urban areas have different traffic flux and density. Manual tuning is very time-consuming, human-intensive and error-prone. Moreover, it depends on the intuition and experience, which are typically undocumented and therefore non-reproducible. Hence, means for automated parameter tuning are required [126]. In this section, we first report on the role of each parameter in biasing the processing, and then we adopt a supervised data-driven parametric optimization based on differential evolution.
In the literature, the approaches for setting a set of parameters can roughly be divided into *model-free* versus *model-based* procedures [127]. The main difference is that model-based procedures build a model interpreting the relation between the algorithm and its parameter values derived from the human experience. Model-free procedures are more lightweight and faster in execution, but have no extrapolation potential (black-box approach). In general, the no-free-lunch theorem of optimization states that a general-purpose universal optimization strategy is impossible, and the only way one strategy can outperform another is by specializing it to the structure of the specific problem under consideration. A well-known method of specialization is to apply constraints to the search space. Given the complexity of the search space, a population-based method, called evolutionary algorithm (EA), is commonly applied [127] [128]. More specifically, we adopt Differential Evolution (DE), a method based on vector differences and therefore primarily suited for numerical optimization problems. Since EAs are meta-heuristics, they have parameters to be tuned. However they show effectiveness with default values when sufficient domain constraints are applied.

Model-based analysis

Table 14 summarizes the main structural parameters. The hypothetical track extension (β) depends on the traffic statistics of the specific urban area. Let us consider, for instance, a speed limit of 50 km/h and a sustainable average speed of 25 km/h = 416.6 meters/min. To set β = 208.3 meters implies that two consecutive hypothetical tracks do not overlap when car speed is higher than 25 km/h. The mark intensity (I) is the maximum intensity of a mark. The value of this parameter influences directly the intensity level of the track and then the triggering at the output activation interface, as well as the lifetime of a mark. For example, with an evaporation of 0.5 and I=5, after 3 steps the mark intensity falls under 1, and then in practice disappears.

Table 14 Structural parameters

Parameter	*Description*	*Section*
$\beta > 0$	hypothetical track extension	II.B
$I > 0$	mark intensity	II.C
$\varepsilon > 0$	mark extension	II.C
$0 < \theta < 1$	mark evaporation	II.C
$\phi > 0$	inflection point	II.D
$\alpha > 0$	inflection slope	II.D

The mark extension (ε) implies the distance of interaction between marks, and it is measured in units. The mark of a vehicle in a unit should interact with both the next and the previous occupied units. Thus, a lower bound can be 1 unit = 10 meters. In addition, considering a 100 meters congestion, vehicle marks in the middle should interact at most with the queue head and tail. This is allowed with an extension of 5 units = 50 meters. Hence, values lower than 1 may prevent mark aggregation, whereas values higher than 5 allow interaction between vehicles too far from each other in any urban context, thus increasing the system error on the start and the end positions of a queue. The mark evaporation (θ) affects the lifetime of a mark. Short-life marks cannot aggregate, increasing the system error on the temporal start of the event, long-life marks cause saturation, increasing the error on the temporal end of the event. Hence, considering a sampling period of 1 minute, mark lifetime should be higher than 2 minute ($\theta = 0.5$) and lower than 5 minutes ($\theta = 0.75$). The inflection point (ϕ) is in the domain of the mark intensity, and the inflection slope (α) is a multiplicative factor of the transient dynamics, and can be set according to the structural mark parameters. It is apparent from Fig. 5 that the maximum value for α is 10, which causes an almost-Boolean transition. Considering ϕ, a single vehicle still in a long-duration congestion produces a stationary mark intensity whose maximum value is $I / (1-\theta) = 5/(1-0.5)=10$ [5]. Thus, the minimum value of ϕ is 10 because the single vehicle does not represent a congestion event. Let us consider $\theta = 0.75$ and $\beta = 5$ to compute the track intensity with 5 overlapping triangular marks at the head or tail of the queue. With $\delta = 0$ (vehicles in queue) the intensity coefficient (Formula 4) is 2. Thus, the track intensity of the 5 marks is $2 \cdot (1 + 4/5 + 3/5 + 2/5 + 1/5) \cdot 5 / (1 - 0.75) = 120$. In practice, 120 is the maximum value of ϕ because it is the track intensity that should be detected at the queue extremity. In conclusion, $\phi \in [10,120]$.

The model-free parametric tuning

The parametric tuning uses DE algorithm to optimize the parameters of the system with respect to the fitness defined in (4). In DE algorithm, a solution is represented by a real *n*-dimensional vector, where *n* is the number of parameters to tune. DE starts with a population of *N* candidate solutions, injected or randomly generated. At each iteration and for each member (target) of the population, a mutant vector is created by mutation of selected members and then a trial vector is created by crossover of mutant and target. Finally, the best fitting among trial and target replaces the target.

Results

A Java-based system architecture for the proposed approach has been developed and experimented, under a research program co-founded by a regional government. The stigmergic aggregation module and the tuning modules have been developed under the Matlab[2] and the Repast[3] frameworks. A traffic simulator based on Java and the Google Maps API has been developed to feed the system. To generate traffic data, as a pilot urban area we considered about 8 km of the network of Figure 30. In two hours of simulation, 116 congestion events occurred.

For the setting of CR and F we took into account the model-based analysis of Section III.A, thus using human experience. We call the approach "HU+DE". More exactly, we constrained the parameters to the following values or interval-values: $\beta = 208.3$; $I = 5$; $\varepsilon \in [1, 5]$, $\theta \in [0.5, 0.75]$, $\phi \in [10,120]$, $\alpha \in (0, 10]$. For each experiment, 5 trials have been carried out, with 30 generations. We also determined that the resulting fitness values are well-modeled by a normal distribution, using a graphical normality test. Hence, we calculated the 95% confidence intervals. Table 15 shows the fitness, in the form "mean ± confidence interval", for the considered values of the parameters *CR* and *F*. Here it can be observed that the variation of both parameters does not significantly affect the performance of the algorithm, and that the best performance (represented with boldface style in Table 15) is achieved with *CR*=0.7 and *F*=0.8.

Table 15 Optimization parameters Setting with the HU+DE approach: *F* (with *CR*=0.7); *CR* (with *F*=0.8)

F	*Fit*
0.7	35.16 ± 0.47
0.8	$\mathbf{33.72 \pm 0.72}$
0.9	35.91 ± 0.89

(a)

CR	*Fit*
0.6	35.44 ± 0.80
0.7	$\mathbf{33.72 \pm 0.72}$
0.8	34.81 ± 0.31

(b)

We carried out three experiments with different approaches, to optimize the aggregation parameters: (a) "HU+DE", i.e., a DE constrained with the same model-based analysis used for the setting of *CR* and *F*, and then using human experience; (b) "DE", with the following model-free parameters bounds: $\varepsilon \in [1, 10]$, $\theta \in [0.35, 0.9]$, $\phi \in [1, 500]$, $\alpha \in (0, 20]$; (c) "HU", i.e., a manually-parameterized experiment, using default values for each parameter: $\varepsilon = 3$; $\theta = 0.675$, $\phi = 23$; $\alpha = 1$. Table III shows the optimal parameters setting and the related best fitness of each experiment. To provide an absolute quality measure, the table also shows separately the average absolute errors on the start and end times, and the average absolute errors on the head and tail of the queues. It can be observed that the model-free approach ("DE") significantly improved the quality of the detection with respect to the model-based ("HU"), from 65.5 to 36.0, and that the hybrid approach ("HU+DE") provided further improvements. Table 16 also shows that the major impact of the parametric adaptation is on the temporal error (from 28.4 to 9.5 minutes), whereas the impact on the spatial error is not significant (about one meter).

[2] http://www.mathworks.com
[3] http://repast.sourceforge.net/

Table 16 Human-driven vs./with DE-driven parameterization

Approach	Parameterization					Performance	
	ε	θ	α	ϕ	Fit	*Avg Time Err (min.)*	*Avg Position Err (mt.)*
HU+DE	4	0.579	8.5	25	33.(	9.5	36.4
DE	4	0.611	18.5	26	36.(	12.3	34.9
HU	3	0.675	1	23	65.5	28.4	35.7

Figure 30 shows the fitness versus the number of generations for the HU+DE and DE approaches. We observe that the fitness function gets stable around a small number of generations (about 30). It is also apparent that DE optimization sensibly compensates the leverage of human knowledge.

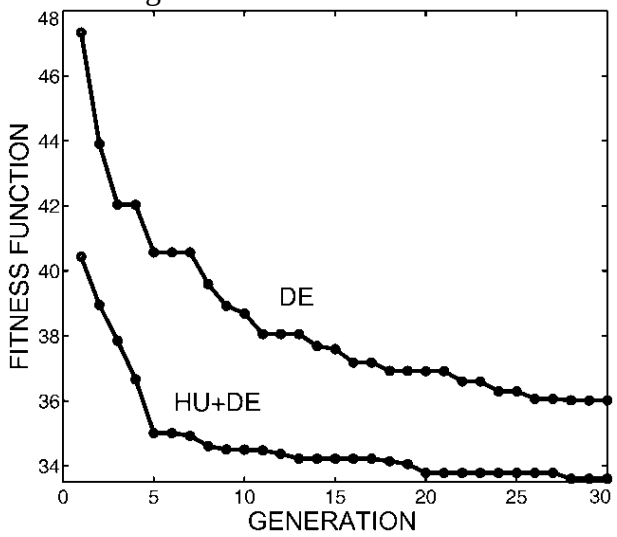


Figure 30 Fitness function versus generation, for HU+DE and DE approaches.

3.4 Evaluation of Technological Indicators in Smart Specialization

Problem Statement

After years of economic crisis and the resulting reduction of resources available for research and development investments, Smart Specialization has immediately become a very relevant concept to get these two questions answered [129]. It represents an important chance for a progressive economical restart. In order to develop a policy-prioritization logic to foster regional growth is important to have a deep knowledge of the potential evolutionary pathways related with the existing dynamics and the structures at regional level [130]. In this light, each region should start this process using as standpoints the knowledge-based sectors in which already presents a consistent 'critical mass' or, at least, capabilities that refer to a future potential exploitable with right and focused investments.

In the last decade, several causes have determined the increasing need for rationalization of resources within regions. The crucial ones are the increased globalization, mainly pursued by multinational enterprises, the economic crisis involving all EU regions with different magnitudes and the diffusion of a new wave of general purpose technologies.

This situation calls for a deep rethinking of the overall approach to regional development; policy-makers and experts largely agree on the fact that the new economic boost should originate exploiting and enhancing the specific potential and competitive advantage of each region through focused innovation policies. On this line, the European Commission has established a program labelled 'Smart Specialization', consisting in a set of policies and guidelines aimed to promote the efficient and effective use of public investment in research and development (R&D).

Smart Specialization is defined as "*an industrial and innovation framework for regional economies that aims to illustrate how public policies, framework conditions, but especially R&D and innovation investment policies can influence economic, scientific and technological specialization of a region and consequently its productivity, competitiveness and economic growth path. It is a logical continuation in the process of deepening, diversifying and specializing of more general innovation strategies, taking into account regional specificities and inter-regional aspects, and thus a possible way to help advanced economies (as well as emerging economies) to restart economic growth by leveraging innovation led / knowledge-based investments in regions*" [131]. From one hand, this approach requires the concentration of R&D resources in few domains; from the other hand, a consisting part of literature underlines the importance of industry diversification in promoting innovation. In this light, the dichotomy specialization-diversification has become topical.

The long term aim of this work is exploring whether - and to what extent - different policies of 'technological specialization' and 'technological diversification' pays off in term of wealth creation at regional level. Then, we want to provide policy makers with computerized support in the analysis of innovation-relevant trends. To properly move into that direction, we start looking at this problem by analysing the trends of the aforementioned indicators for 268 EU-27 regions over 35 technological domains in the period 1990-2012, in order to obtain a model that can efficiently recognize significant events. For this purpose, we have designed and developed a software system. In contrast with conventional knowledge-based design, our approach is biologically-inspired and based on stigmergy as a mechanism of self-organization of information. Moreover, the performance of such a model is contrasted with a supervised adaptation based on differential evolution.

In Figure 31 we present the terminology via an ontology diagram. Concepts are enclosed in white ovals and connected by properties (represented as black directed edges). A property that cannot be directly sensed (i.e., instantiated) is represented as an abstract property, shown by a dashed edge.

More specifically, it is known that diversification and specialization of *Patents applied in a Region measure the Innovation* of the region itself. Thus, it is important for a *Policy Maker to analyse Trends of Innovation,* to properly address the investments. Such trends cannot be directly sensed nor associated to the Innovation. For this purpose, there are three important *indicators which quantify Innovation*: *specialization (S), related variety (R),* and *unrelated variety (U).* The study of such *Trends* by the *Policy Maker* is fundamental to recognize scenarios of interest, namely the ways in which special situations may develop. Example of scenarios of interest are: (i) R or U decreases, while S increases; (ii) R or U decreases, while S is stable; (iii) R or U increases, while S is stable;(iv) R or U increases, while S increases.

The problem is to detect variations of an indicator in terms of increase, decrease or stability. In this work, we adopt an emergent modeling perspective. With an emergent approach, the focus is on the low level processing. In Figure 31 we also present the

terminology related to the approach. More specifically, an *Indicator Value enables* the release of a *Mark*. *Marks aggregate in Tracks*, depending on their spatiotemporal local dynamics. Emergent paradigms are based on the principle of the self-organization of the data, which means that a functional structure, the *Track*, appears and stays spontaneous at runtime when local dynamism occurs. A particular *Track* representing only the main characteristics of such local dynamics is the *Prototype*. It is the *Dissimilarity* which *compares Prototypes* generated at different times, in order to *assess the Trend*. Finally, the *Evolution* process *adapts Mark*, *Track*, *Prototype* and *Dissimilarity* to properly fit the temporal dynamics of the indicators. The Evolution process represents the application of biologically-inspired patterns to adapt parameters. The approach iteratively tries to improve a population of candidate parameters with regard to a given measure of quality, or fitness. Solutions are found by means of stochastic transformation mechanisms inspired by biology, such as reproduction, mutation, recombination, selection, in an environment where competition is represented by the quality measure.

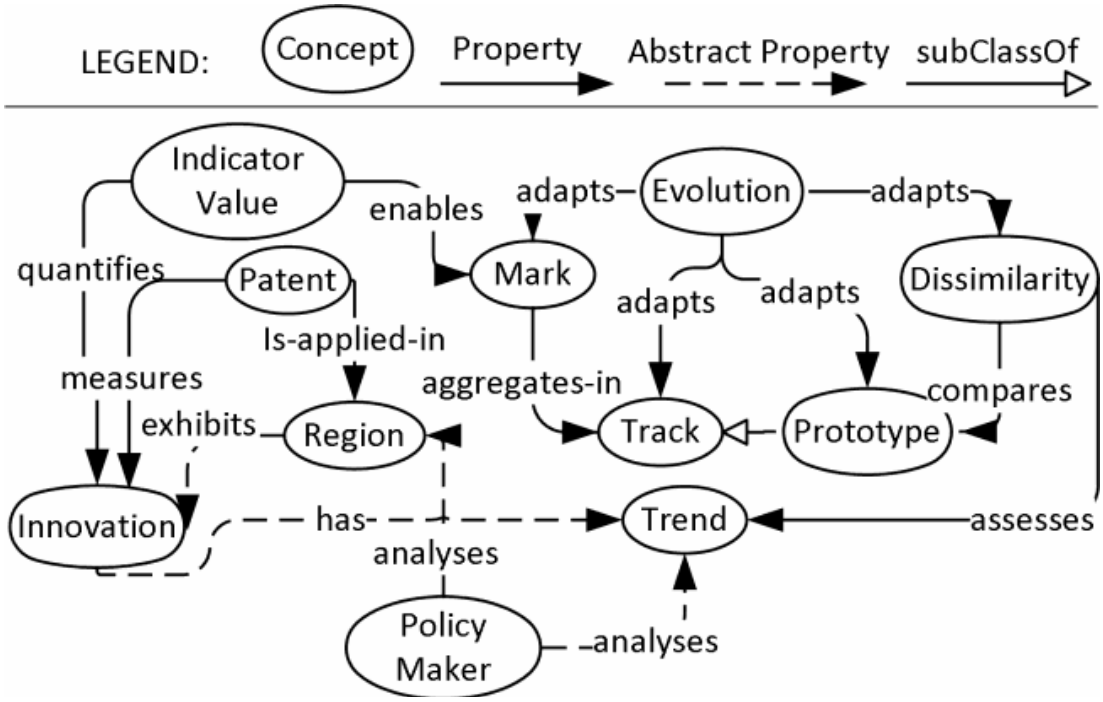


Figure 31 Ontological view of the approach.

Application of the Stigmergic Architecture

Figure 31 illustrates the system architecture, made of four main subsystems, namely *marking*, *trailing*, *prototyping*, and *dissimilarity*.

At the input/output interfaces of each subsystem, an *unbiasing* module can be used for a better efficiency and alignment of the processing layers. This lets an input signal to reach a certain level before a processing layer passes it to the next layer, and allows a better distinction of the critical phenomena during unfolding events, with a better detection of the progressing levels.

The marking subsystem transforms input data into *marks*, whereas the trailing subsystem aggregates and evaporates marks as a *track* in the stigmergic space. The prototyping subsystem provides a simplified version of the track. It is a vehicle of abstraction, leading to the emergence of high-level information. The dissimilarity

subsystem evaluates the difference between consecutive prototypes in order to extract trend information of the indicators. The proposed mechanism works if structural parameters are correctly adapted for the given application context. Determining such correct parameters is not a simple task since different indicators may have different dynamics. For this purpose, we adopt a tuning mechanism based on differential evolution algorithm. In the next subsections each module and subsystem is precisely described, by using a pilot data sample.

The Unbiasing Module

In Figure 32 the U indicator of a region for 4 years is shown, in dashed line. To unbias the input signal we adopt the *s-shaped function*, having the following behaviour: input values smaller (larger) than $(\beta - \alpha)/2$ are lowered (raised); values smaller (larger) than α (β) assume the minimum (maximum) value, i.e., 0 (1). In biologically inspired subsystems, this function is an abstraction representing the active zone of a signal generated by a subsystem. In Figure 32 we represent the output of the unbiasing, with α_M =0.2, β_M =0.8, in thick line.

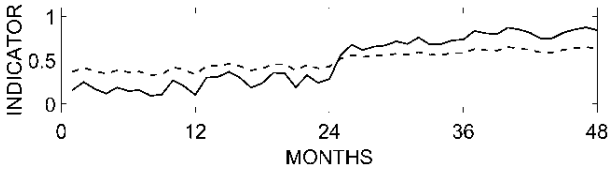


Figure 32 An example of application of unbiasing.

The Marking Subsystem

The Marking takes an unbiased sample $\tilde{d}(t)$ of a normalized input time series *D*, and releases a *mark* in a marking space whose codomain is called *intensity*. The mark has four structural attributes: the center position $\tilde{d}(t)$, the intensity I, the *mark extension* ε, and the *mark evaporation* θ. In Figure 34 (a) we show, in thick line, the mark released by the sample $\tilde{d}(2)$ of our pilot time series. The mark shape is an isosceles triangle: its center is $\tilde{d}(2)$=0.25, its height I=1, and its base has length 2ε=0.5.

The Trailing Subsystem

The evaporation θ is the temporal decay of the mark. After each step the mark intensity decreases by a percentage θ. Thus, evaporation leads towards a progressive disappearance of the mark. Anyway, subsequent marks can reinforce previous mark in the environment if their shapes overlap. In Figure 34 (a) we also show the mark $\tilde{d}(2)$ after an evaporation step, in thin line. In Figure 34 (b) we show in thin line three consecutive marks, their apex coordinates (x, y). We also show the final *track*, T_3 at time t=3, in thick line, as the sum of the track intensities

The Prototyping Subsystem

The Prototyping subsystem takes as input the output track of the trailing subsystem, T_t. This input is first unbiased, as $\tilde{T}_t$. The prototype P_t is then generated as a triangular shape, with base width 2ε, saturation height $I_{max}=I/(1-\theta)$ [5], and center $p(t)$.

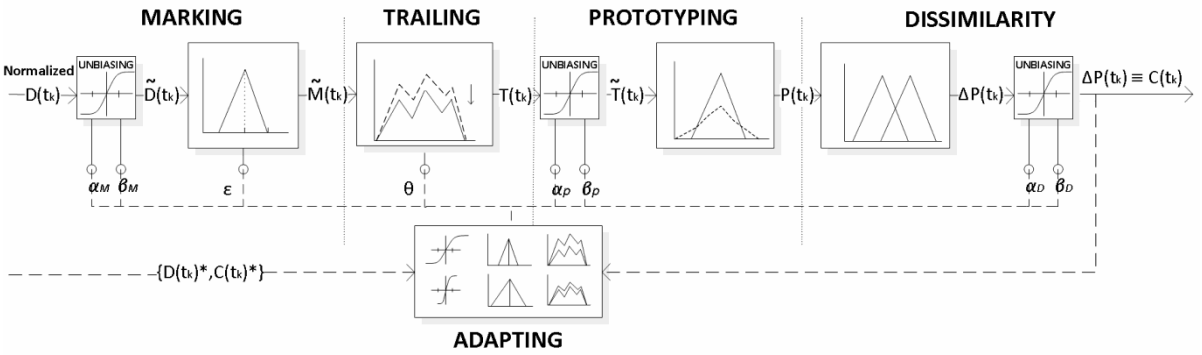


Figure 33 Architectural overview of our data analysis system based on Stigmergy

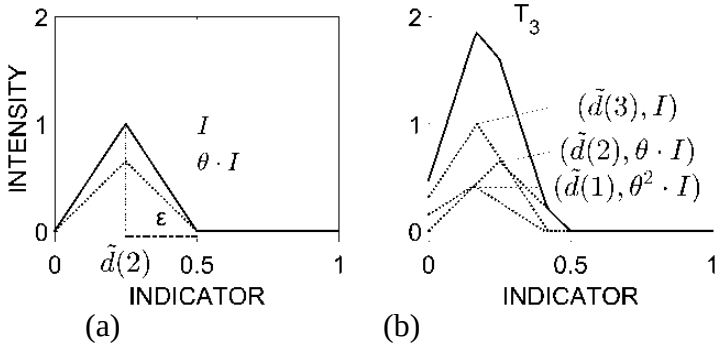


Figure 34 Single mark shape (a) and aggregation of three marks (b) in a track

Figure 35 shows in dotted line the track T_3 of Figure 34 (b), the unbiased track $\tilde{T}_3$ in dashed line, and the corresponding prototype P_3 centered in p_3 in solid line.

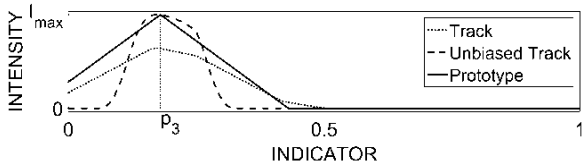


Figure 35 An example of prototyping.

The center p_t of the prototype is the position that maximizes the similarity between the unbiased track and the prototype itself. The similarity between two shapes is $S(P_t, P_{t-24}) = P_t \cap P_{t-24}/P_t \cup P_{t-24}$. It is the ratio between the intersection and the union of the shapes. In Figure 36, $S(P_{48}, P_{24}) = 0.0097$.

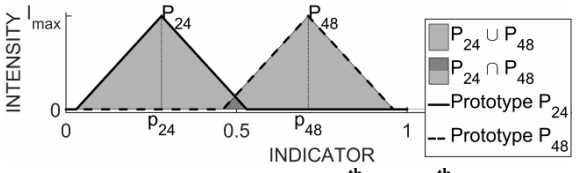


Figure 36 Similarity between two prototypes sampled at the 24th and 48th months of the time series, respectively.

The Dissimilarity Subsystem

This subsystem calculates the complement of the similarity between two prototypes generated at two instant of times, with a positive (negative) sign if the barycenter of the most recent track is larger (smaller) than the previous: $\Delta P = (1 - S(P_t, P_{t-2}) \cdot sgn(p_t - p_{t-2})$.
In Figure 36 $\Delta P = 0.9903$. In the subsystem, an unbiasing module is finally used, in order to provide one of three different classes as output: -1, +1, or 0, when the time series decreases, increases, or it is otherwise considered stable, respectively. In Figure 36, $\widetilde{\Delta P} = +1$, meaning that the indicator increases.

The Adapting Subsystem

The overall system uses 8 structural parameters (summarised in Table 17) to be appropriately adapted. Since different indicators in different Regions have different dynamics, manual adaptation is very time-consuming, human-intensive and error-prone [3]. In this section, we first describe the role of each parameter in the processing, and then we adopt a supervised optimization based on DE, an evolutionary technique for numerical optimization problems [6].

Table 17 System Parameters

Module	Params	Human Expert	Range
marking	α_M β_M ε	(0.2; 0.8; 0.2)	(0, 1)
trailing	θ	(0.65)	(0, 1)
prototyping	α_P β_P	(0.15; 0.75)	(0, I_{max})
dissimilarity	α_D β_D	(0.35; 0.65)	(0, 1)

The *mark extension* (ε) controls the distance of interaction between marks. If it is close to 0, the marks cannot interact with each other, and there is no patterns reinforcement. If the mark extension is close to 1, all marks reinforce each other without distinction between patterns. The *mark evaporation* (θ) affects the lifetime of a mark. Short-life marks evaporate to fast preventing aggregation and pattern reinforcement. Long-life marks cause an early saturation of the track, thus all tracks become similar to each other. Finally, α and β affect the system as previously described in the unbiasing module.

The adaptation uses DE algorithm to optimize the parameters of the system with respect to the fitness computed using a training set. Let Z be the set of Regions in the dataset, $D_z(t_k)$ the input signal for Region z, $c_z(t_k) \in \{-1,0,1\}$ the output of the system. Let us consider $\{D_z(t_k)^*, C_z(t_k)^*\}$ the training set. We compute the average fitness among Regions as the Mean Squared Error (MSE) between the outputs of the system calculated for the training inputs and their corresponding training outputs:

$$f(Z) = \sum_{r=1}^{R} \sum_{k=4}^{Y} (c_z(t_k)^* - c_z(t_k))^2 \Big/ R \cdot Y \qquad 24$$

Results

We propose a case study based on a data set that contains the three annual indicators S, U, R, (described in previous section) monitored for 15 years for 200 European Regions. The dataset contains 9000 samples. In order to reduce data to a canonical size, the following normalization is first applied: $x_n = (x_i - x_{min})/(x_{max} - x_{min})$. Since the original data samples are subject to significant sampling error, we also performed a granulation process. More specifically, for each year we grouped regions via k-nearest neighbor algorithm. For each group we computed the annual mean μ and the standard deviation σ. We also determined that the resulting indicator samples are well-modelled by a normal distribution, using a graphical normality test. Finally, monthly samples have been derived considering normal distribution with mean and variance μ/12 and $\sigma^2/12$, respectively.

To choose the best value of *CR and F*, we first performed trials with *CR* in [0.3, 0.6, 0.9] and *F* in [0.4, 0.6, 0.8]. For each experiment, 5 trials have been carried out, by using the 20% of the dataset as a training set, and the remaining 80% as a testing set. We also determined that the resulting MSE samples are well-modeled by a normal distribution, using a graphical normality test. Hence, we calculated the 95% confidence intervals. Table 18 shows the results in the form "mean ± confidence interval". The best performance has been with CR=0.6 and F=0.6. In general, we observed that fitness function gets stable after 15 generations only.

Table 18 95% confidence interval of the MSE for the best setting of differential weight (F) and crossover rate (CR).

		F		
		0.4	0.6	0.8
CR	0.3	0.022 ± 0.0002	0.012 ± 0.00002	0.018 ± 0.0002
	0.6	0.019 ± 0.0002	**0.009± 0.0001**	0.011 ± 0.00001
	0.9	0.014 ± 0.00006	0.013 ± 0.00007	0.013 ± 0.00007

In order to assess the effectiveness of the approach, we adopted a 5-fold cross-validation. Indeed, each evaluation is also dependent on the data points, which end up in the training and test sets. For each trial, the training and test sets consist, respectively, of randomly extracted 20% and 80% of the original data. We carried out each trial 5 times. Table 19 summarizes, for indicator U, the results in terms of mean and standard deviation of the MSE for each trial. The low values of the MSE, for all trials and for both training and testing sets, highlight the effectiveness of the system in terms of both performance and generalization properties. We replicated the same experiments and achieved similar performances for the other indicators.

Table 19 MSE for each trial extracted via 5-fold cross-validation, averaged over 5 repetitions.

	MSE (mean ± stdev)	
Trial	**Training Set**	**Testing Set**
1	0.011 ± 0.010	0.018 ± 0.004
2	0.010 ± 0.010	0.020 ± 0.003
3	0.009 ± 0.006	0.020 ± 0.008
4	0.008 ± 0.008	0.020 ± 0.005
5	0.010 ± 0.007	0.022 ± 0.008

Finally, to highlight the great benefits of the adaptation subsystem, we also computed the MSE for the worst case of Table 19 (i.e., Trial 5), by using manual adaptation: this implied an MSE of 0.106, which is very higher than 0.022.

3.5 Target Detection with Small Aerial Drones

Problem Statement

Small drones are unmanned aerial vehicles small enough to be man-portable. Recent advances in sensing technology encourage the use of small drones for military intelligence, reconnaissance, surveillance, traffic monitoring, forest fire localization, scientific surveys in dangerous conditions, border and harbor patrol, search and rescue, wildlife tracking, and so on [132]. Indeed, small drones can be equipped with self-localization and sensing capabilities, to be used for information gathering tasks, such as target search or environmental sensing. More specifically, in target search small drones offer more potential, because they can perform tasks in highly inhospitable environments, where access to humans and large drones is limited or impossible, or in environments where medium and large drones can cause significantly more damage. Moreover, in tasks such as search and rescue, to accurately scan every available location on the area is also an inappropriate strategy. A more effective approach is to achieve a quick "survey" of the area, identifying key locations as quick as possible, and to better investigate only key locations that provided some circumstantial evidence. To complete this task with a single drone, both structure and control logic should be highly costly in terms of design, construction and maintenance. Moreover, a unique drone is vulnerable, because a single hardware or software fault may affect the whole system, and it is difficult to predict. Hence, a number of considerations support the use of small drones.
To take advantage of the above strategy, an important requirement is to avoid centralized control approaches, which frequently lead to exponential increases in communication bandwidth requirements and software complexity [133]. To solve problems cooperatively while maintaining scalability, application designers are investigating swarm intelligent methodologies. The main inspiration for swarm drones comes from the observation of social animals, such as ants, bees, birds, and fishes, exhibiting a sort of collective intelligence which appears to achieve complex goal through simple rules and local interactions [134]. The main benefits of swarm drones are: (i) robustness, for the ability to cope the loss of individuals; (ii) scalability, due to the ability to perform well with different group size; (iii) flexibility, thanks to the ability

to cope with a broad spectrum of different environments and tasks. For this purpose, each individual of the swarm: (i) acts with a certain level of autonomy (ii) performs only local sensing and communication; (iii) operates without centralized control or global knowledge, and (iv) cooperates to achieve a global task.

We adopt DE algorithm to optimize the behavior of a swarm of aerial drones carrying out collaborative target detection [135]. In our approach, stigmergy and flocking are exploited to enable self-coordination in the navigation of unstructured environments. Such environments typically contain a number of obstacles such as trees and buildings. Targets are placed according to different patterns. More specifically, stigmergy implies the use of an attractive digital pheromone to locally coordinate drones. When a drone detects a new piece of target, it releases the attractive pheromone. Other drones in the neighborhood can sense and follow the pheromone gradient to cooperate in detecting the pattern of targets. With respect to [135] in this study we sensibly improve the performance of the algorithm, by adopting the parametric DE-based adaptation, by introducing repulsive pheromone and scattering mechanisms. Basically, flocking implies a set of three behavioral rules, named separate, align, and scatter. It maintains the drones in groups enhancing the stigmergy when occurring. In contrast, the repulsive pheromone helps the drones to avoid multiple exploration of the same zone. Indeed, the drone releases a repulsive pheromone whereas it does not sense a target. Finally, scattering rules makes a drone to turn by an angle when it tails another drone.
In the literature there are two design methods to develop collective behaviors in swarm systems: behavior-based design and automatic design [134]. The former implies the developers to implement, study, and improve the behavior of each single individual until the desired collective behavior is achieved. This is the approach adopted in [135] . The latter is usually used to reduce the effort of the developers. Automatic design methods can be furtherly divided in two categories: reinforcement learning and evolutionary robotics [134]. The first implies a definition at the individual level of positive and repulsive reinforce to give reward to the individual. In general, it is usually hard for the developer to decompose the collective output of the swarm in individual rewards. Evolutionary robotics implies evolutionary techniques inspired by the Darwinian principle of selection and evolution. Generally in these methods each swarm consists of individuals with the same behavior. A population of swarms is then computed, where each population member has a particular behavior. A simulation is made for each member and a fitness function is computed. Then through a mutation and crossover procedure a new generation is computed. This process iteratively repeats improving the performance of the swarm population.

Application of the Control and Coordination Algorithm

We refer to the time unit as a *tick*, which is an update cycle of both the environment and the drones.
Here, we suppose each small drone with the following equipment: (a) wireless communication device for sending and receiving information from a ground station; (b) self-location capability, e.g. based on global position system (GPS); (c) a sensor to detect a target in proximity of the drone; (d) processor with limited computing capability; (e) a sensor to detect obstacles.

The environment structure and the pheromone

We consider a predefined area that contains a set of targets to be identified. The environment is modelled by a digital grid corresponding to the physical area. The grid has C^2 cells, each identified by (x, y) coordinates with x, y ∈ [1,…,C]. The actual size of the area and the granulation of the grid depend on the domain application. Figure 37 shows Pheromone dynamics in an urban scenario. Here, the intensity of the pheromone is represented as a dark color, and each target is represented by an "X". A darker gradation means higher pheromone intensity. At the beginning, the pheromone is in one cell at its maximum intensity, and then it diffuses to neighbors cells. After a certain time the pheromone evaporates, disappearing from the environment. When a drone detects a piece of the overall target, a pheromone of intensity I is released in the cell corresponding to the drone location. Then, at each tick the pheromone diffuses to the neighbors cells according to a diffusion rate $\delta \in [0,1]$. At the same time the pheromone evaporates decreasing its intensity, by an evaporation rate $\varepsilon \in [0,1]$.

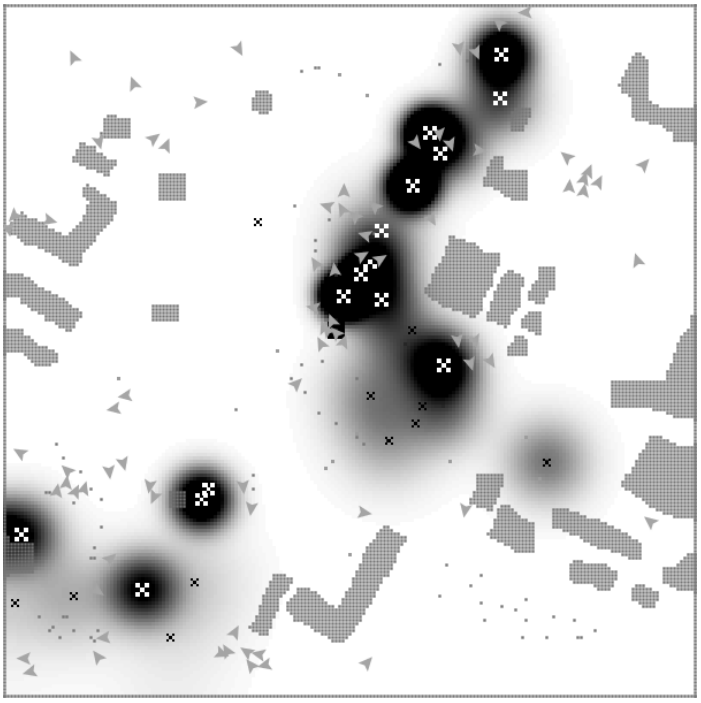


Figure 37 Pheromone dynamics in an urban scenario.

The drone behavior

The drone behavior is structured into five-layer logic, where each layer has an increasing level of priority. Starting from the lowest: the random fly behavior entails the exploration in absence of stimuli/information; the repulsive pheromone-based coordination reduces the redundancy of exploration; the flocking behavior assembles the flock enabling local pheromone interactions; the positive pheromone-based coordination exploits the digital pheromone and directs the swarm toward potential new targets; finally the highest priority level, the objects/boundary avoidance, which

prevents the drone from exiting the area or colliding with obstacles and other drones.
In Figure 38 we present the UML activity diagram of the drone behavior. Here, a rectangular box represents sensors (left) and actuators (right), a rounded-corner box represents an activity (a procedure), solid and dashed arrows represent control and data flows, respectively.

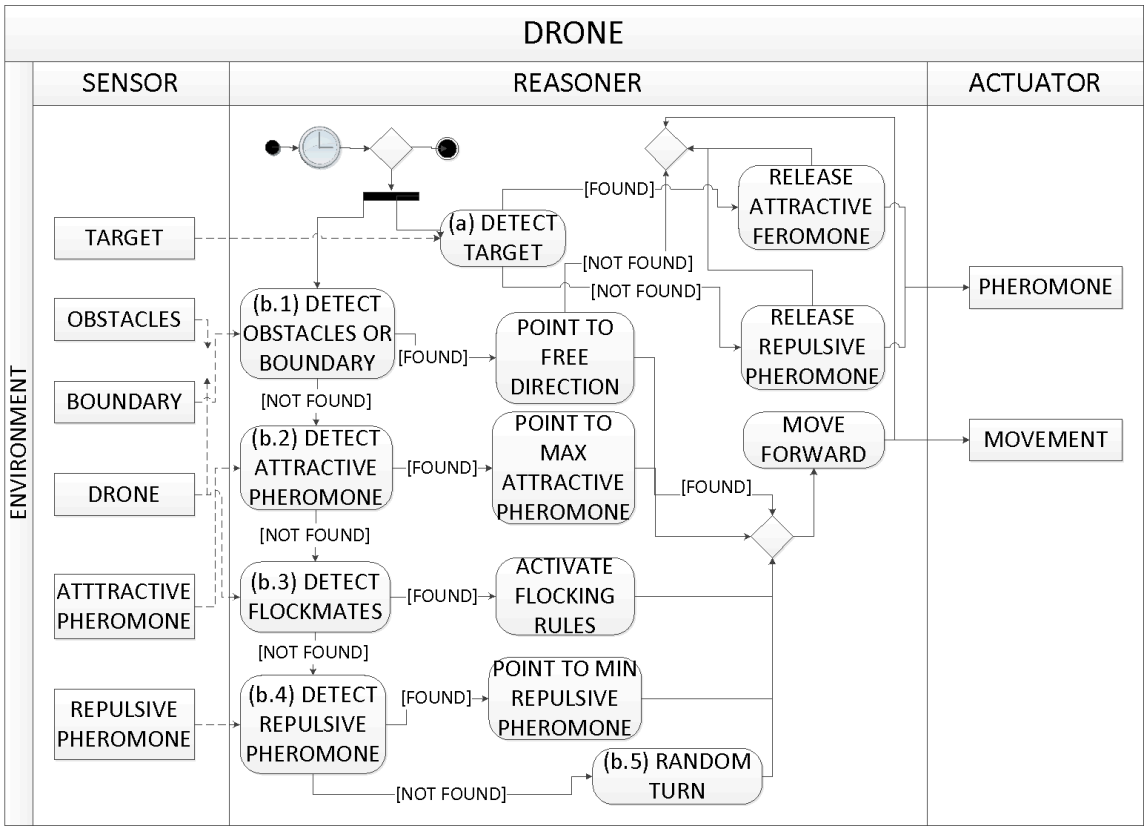
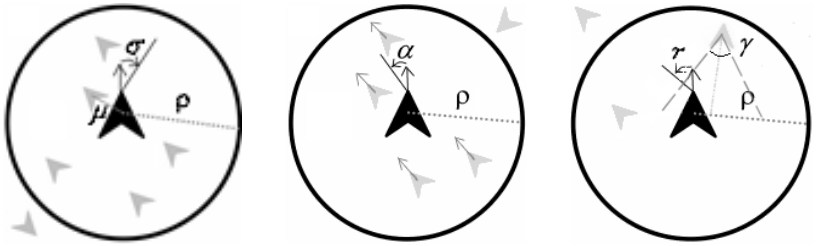


Figure 38 Overall representation of the drone behavior.

Every tick period, represented by the clock, the drone performs two parallel processes: (a) the target detection, which corresponds to the release of the attractive pheromone on the digital grid or rather of the repulsive pheromone; (b) the movement procedure which starts with the obstacle and boundary detection. If a close obstacle (i.e., object or drone) is detected (b.1), the drone points toward a free direction. If a free direction is found, it moves forward, otherwise it stays still. If there are no close objects detected (b.2), the drones try to sense positive pheromone in its neighbor cells and, if detected, points toward the maximum intensity of it. Differently, if no pheromone is detected (b.3), the drone tries to detect surrounding drones in order to stay with the flock. Then, the drone checks if a repulsive pheromone is in nearby cells (b.4). If there is, the drone moves toward the minimum intensity of it. Finally, if there are no surrounding drones (b.5), it performs a random turn and move forward.
At the highest priority (b.1), the drone has to avoid collision and to exit the search area. For this purpose the drone is able to detect objects (i.e. other drones or obstacle) at the *object sensing distance* (o) on its trajectory. If a potential collision is detected the drone chooses the minimum rotation angle to the left or right to avoid the obstacle. If it finds a free trajectory it moves forward, if there is no free direction in 180 degree it stays still one tick. The boundaries of the area are considered as a "wall" of obstacles.

The second priority is to follow the positive pheromone gradient (b.2), provided by the ground station. Given the drone position (i.e. the cell on the grid), the cell with the maximum intensity of positive pheromone and the minimum intensity of the repulsive pheromone (b.4) in its 8 neighbor cells is returned to the drone. Then, the drone follows the maximum pheromone intensity.

The flocking behavior (b.3) consists in three rules, represented by Figure 39. A drone considers as a flock mates all drones in a *flock visibility radius* (ρ). The first rule is the *separation*: more specifically, if the drone senses another drone closer than the *flock mobility distance* (μ), it turns away by a *flock separation angle* (σ). The second rule is the *alignment*: the drone computes the average direction of its flock mates and it adapts its direction by a *flock alignment angle* (α). The third rule is the *scatter*: the follower drone considers the closer following drone in front of itself. If the follower is in the tail sector of the following defined by angle *flock tail angle* (γ) the follower turns away the following by a *flock scatter angle* (τ). These three rules determine the structure of the swarm, permitting the drones to navigate in coordinated group, exploring the environment and, most important, sensing pheromone released by flock mates.



(a) separation (b) alignment (c) scattering

Figure 39 Flock visibility radius and other parameters in flocking behavior.

The fourth priority (b.4) is to move against the repulsive pheromone gradient. The drone received the information of the lowest cell from the ground station (b.2).

Finally, if no one of the previous has been activated (b.5), then the drone is in its basic behavior (random fly). It randomly turns by an angle smaller than the *maximum rand-fly turn angle* (θ).

We implemented our algorithm using *NetLogo*, a leading simulation platform for swarm intelligence [136], and *Matlab* for the adaptation algorithm [137]. We evaluated our algorithm by comparing four different strategies: the adaptive stigmergic and flocking behaviour (S+F*) presented in Section 3; the stigmergic and flocking behavior (S+F) presented in [135]; the stigmergic behavior (S), in which the pheromone grid maintains only the positive pheromone intensity information and the drones do not perform flocking behavior; finally, the random fly behavior (R), in which drones randomly explore the area with no pheromone nor flocking behavior. We tested the swarm algorithm on 6 different scenarios. For each scenario, each of the three strategies has been specifically optimized by the adaptation module. To compare the approaches, 10 trials have been carried out for each considered approach. We also determined that the resulting performance indicator samples are well-modelled by a normal distribution, using a graphical normality test. Hence, we calculated the 95% confidence intervals.

The total number of drones is 80; each drone is represented as a black triangle, arranged in 4 swarms placed in the corners of the map, except for *Illegal Dumps* in which 3 swarms are placed in 3 corners due to the topology. Characteristics of the topology of

targets and obstacles are summarized in Table 21.

Figure 41 represents the initial configuration for all scenarios. The three scenarios *Field*, *Forest*, and *Urban* represent respectively an area of 40,000 m^2. Figure 41 (a) does not contain obstacles at all, but contains 5 clusters of 10 targets each. Figure 41 (b) represents a timber with 1 cluster of 20 targets with tree obstacles. Figure 41 (c) shows an industrial area with 2 clusters of 55 targets, each representing a pollutant lost. The two scenarios *Rural minefield* and *Urban minefield*, represented in Figure 41 (d) and Figure 41 (e), respectively, are two areas of 80,000 m^2 derived from real-world maps near Sarajevo, in Bosnia-Herzegovina, and containing landmines.

Such maps are extracted from publicly available datasets. The last scenario, *Illegal Dump*, is a real-world example of illegal dumping taken from an area of 80,000 m^2 near the town of Paternò, Italy.

Results

Figure 40 shows the evolutions of DE algorithm against generations, with two different values of *CR* and *F*. Here, the best and the mean solution (solid and dashed line, respectively) have been calculated over 5 runs. The optimization is characterized by a very good improvement of the initial solution and by a fast convergence: only 20 generations are needed to reduce the average time by 20%.

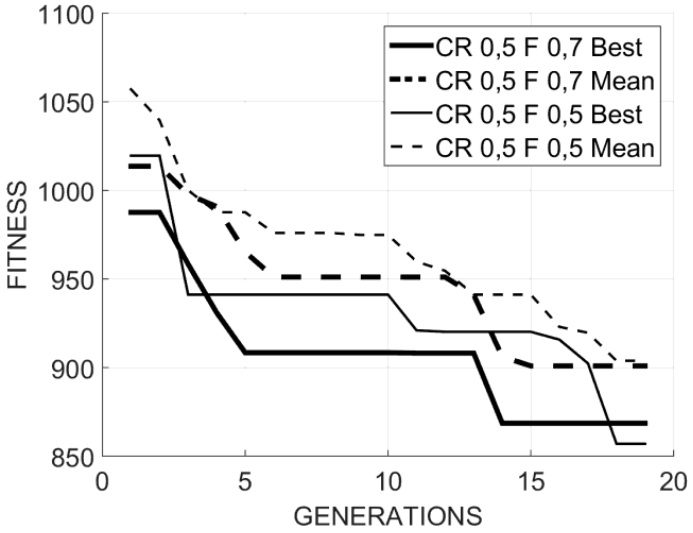
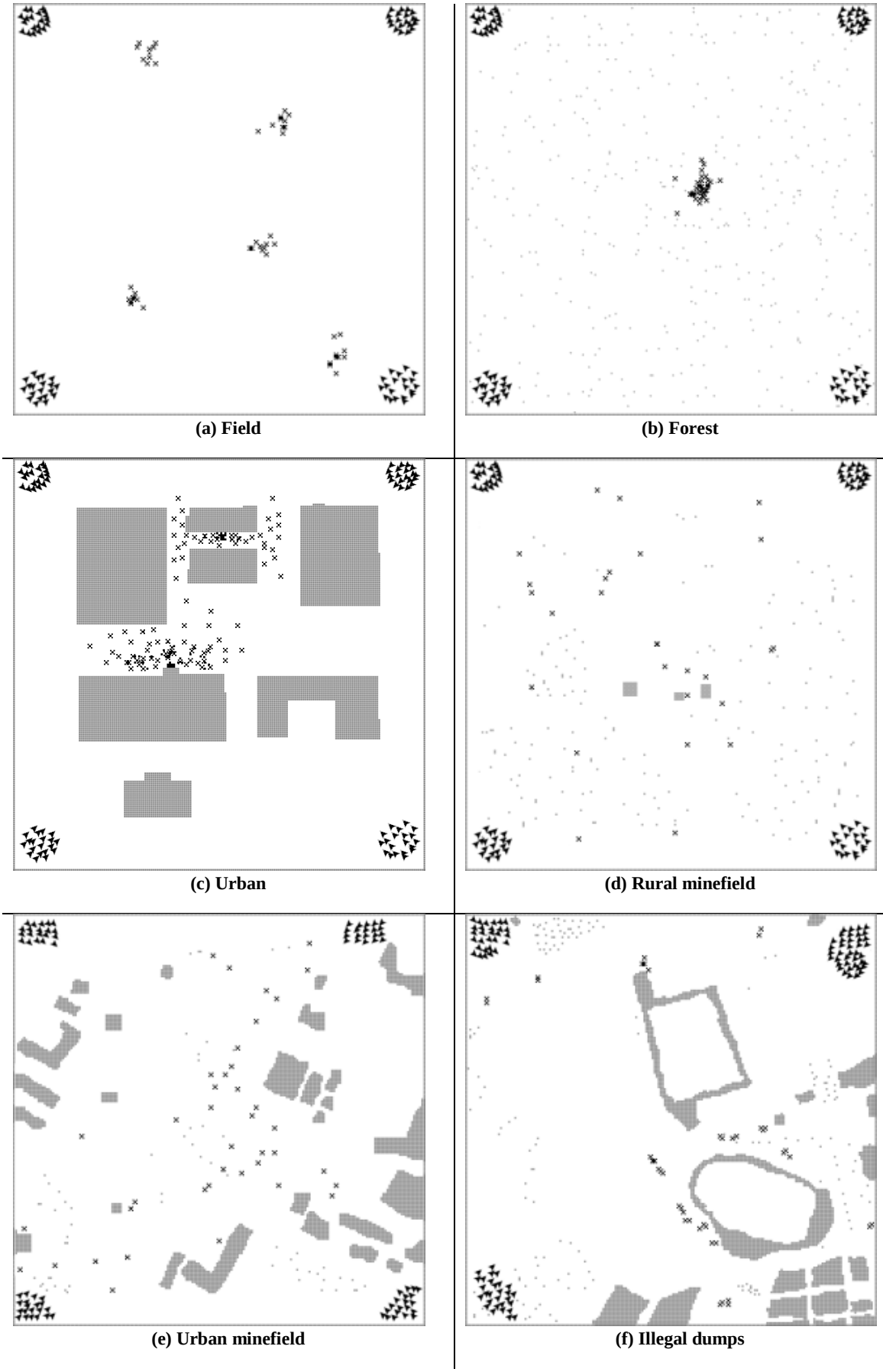


Figure 40. Evolution of the DE algorithm against generations for different DE parameters

More precisely, Table 20 shows that the final best performance is achieved with *CR*=0.5 and *F*=0.7.

Table 20: Final results of the evolution of Figure 5 in numerical terms

F\CR	***0.3***	***0.5***	***0.7***
0.5	933,67±14,16	903,85±29,77	950,80±16,36
0.7	966,60±15,53	**900,65±19,06**	907,15±12,96
0.9	951,40±15,67	930,60±22,36	933,73±54,14



(a) Field **(b) Forest**

(c) Urban **(d) Rural minefield**

(e) Urban minefield **(f) Illegal dumps**

Figure 41: Maps of 3 synthetic and 3 real-world scenarios.

Table 21 summarizes the characteristics and the results for each scenario, in the form "mean ± confidence interval". It is worth noting that the adaptive algorithm 'S+F*' presented in this work outperforms the other algorithms in all scenarios.
More specifically, in *Field* and *Forest* scenarios, the two algorithms 'S+F' and 'S+F*' are comparable. This is due to the simple layout of both scenarios, which does not allow a good exploitation of the "S+F*" features. Indeed, in the other scenarios, with a more complex topology, the advantages of the 'S+F*' strategy are substantial.

Table 21 Features and numerical results of each scenario

Scenario	*N° of targets / clusters*	*Type / n° of obstacles*	*Completion time (ticks)*	
Field	50 / 5	Trees: 0 Buildings: 0	R	1664±220
			S	656±101
			S+F	589±86
			S+F*	**582±121**
Forest	20 / 1	Trees: 400 Buildings: 0	R	1862±356
			S	615±67
			S+F	602±124
			S+F*	**593±146**
Urban	110 / 2	Trees: 0 Buildings: 7	R	2049±148
			S	998±61
			S+F	890±93
			S+F*	**666±100**
Rural Mines	28 / 28	Trees: 281 Buildings: 3	R	1588±216
			S	1570±158
			S+F	**1530±225**
			S+F*	1123±116
Urban Mines	40 / 40	Trees: 54 Buildings: 28	R	1844±140
			S	1733±169
			S+F	**1704±225**
			S+F*	1025±76
Illegal Dumps	42 / 11	Trees: 140 Buildings:19	R	1548±207
			S	971±160
			S+F	934±216
			S+F*	**757±112**

CHAPTER 4

Multi-Layer Architecture for the Identification of Discussion Topics in Microblogs

In this Chapter the Multi-Layer Stigmergic architecture, based on the concept of *Stigmergic Receptive Field*, is presented and applied to a social sensing problem. The Stigmergic Receptive Field is the computational unit which adopts stigmergy to compare couple of patterns. Stigmergic Receptive Fields can be organized into a multilayer connectionist system, and adapted to contextual behavior by means of the DE algorithm.

Problem Statement

Microblogging is a broadcast process that allows users to exchange short sentences. These short messages are important sources of information and opinion about socially relevant events. Microblogging systems such as Twitter, Tumblr, and Weibo are increasingly used in the everyday life. As a consequence, a huge amount of informal, unstructured messages is produced in real time. In the literature, a research challenge is to identify and separate the topics of a specific event.
A commonly encountered approach is to adopt visual representations to summarize the content of the messages [49], [50], [51], [52], [53], [43], [47]. A *term cloud* is a straightforward means to represent the content of the discussion topics of a given event [138]. This is achieved by arranging the most frequent terms at the center of the tag cloud, and positioning less frequent ones on the border, by using a font size proportional to the frequency [139]. In practice, a conventional tag cloud does not include information about the relationship between terms. The purpose of this study is to enrich the cloud representation to generate a *relational term cloud*, by exploiting a similarity measure between terms. Figure 42(a) shows a sub-list of major terms used on Twitter during the terrorist attack in Paris on 13 Nov 2015, by gunmen and suicide bombers. For each term, a simplified example of time series segment, generated by the term occurrences, and periodically sampled as scalar value, is also represented. Here, it can be noticed that the terms *police*, *victim*, *shooting* and *blood*, related to the story, are mostly used in the first part of the event. In contrast, the terms *Allah*, *islamic*, *religion*, related to the religious topic, are mostly used in the middle. Finally, the terms *terrorism*, *Hollande*, *embassy* and *Obama*, related to politics, are used at the end. Figure 42 (b) shows a conventional term cloud, generated on the basis of the global occurrences of each term.

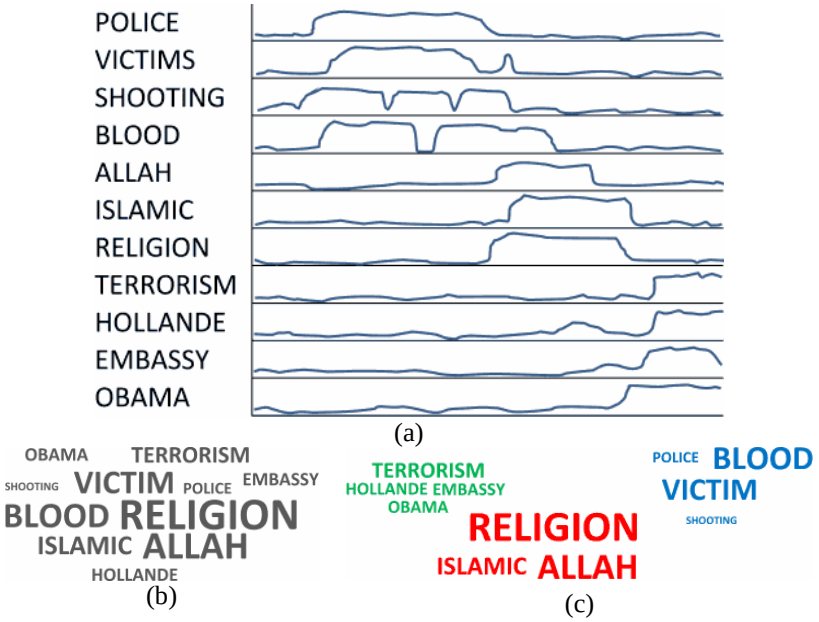


Figure 42 (a) Terms and temporal series, (b) Term Cloud, (c) Relational Term Cloud.

Figure 42 (c) shows a relational term cloud, generated via a clustering process based on a similarity measure between terms. Here, terms belonging to different discussion topics are represented with a different color. A number of time series segments may generate consecutive relational term clouds, showing also the temporal evolution of the major topics. More specifically, in our approach the similarity between two terms is the similarity of the related time series. This means that the terms used in the same discussion topic have similar usage over time, more specifically similar time series [46]. In the literature, the term similarity is usually derived from the co-occurrence of the terms in the same text message [49], [46] , which is the special case of identical time series. In contrast, our notion of similarity between the time series generalizes the co-occurrence, by taking into account other fundamental processes of discussion topics such as the scalar and temporal proximity between terms of different participants. Moreover, our notion of similarity deals with different types of noise, such as ripple, amplitude scaling, drift, offset translation, and temporal drift [140].

To compute the similarity between time series we adopt marker-based *stigmergy*, which is a biologically inspired mechanism performing scalar and temporal information aggregation. Stigmergy is an indirect communication mechanism that occurs in biological systems [9]. In computer science, marker-based stigmergy can be employed as a dynamic, agglomerative, computing paradigm because it embodies the time domain. Stigmergy focuses on the low level processing, where individual samples are augmented with micro-structure and micro-behavior, to enable self-aggregation in the environment. Self-aggregation of data means that a functional structure appears and stays spontaneous at runtime when local dynamics occur.

The proposed mechanism works if structural parameters, such as the mark attributes, are correctly tuned. For example, a very large and persistent mark may cause growing trails with no stationary level, because of a too dominant and long-term memory effect. A very small and volatile mark may cause poor mark aggregation. For tuning such parameters we adopt an adaptation mechanism based on Differential Evolution (DE). DE is a stochastic optimization algorithm based on a population of agents, suitable for numerical and multi-modal optimization problems. In the last decades, DE has been applied to many applications and research areas, including parameterization of

stigmergy-based systems [6]. DE exhibits excellent performance both in unimodal, multimodal, separable, and non-separable problems, when compared with other similar algorithms, such as genetic algorithms and particle swarm optimization.
Overall, in this work, a novel technique is explored to realize a similarity measure between term occurrences in event-specific social discussion topics. The similarity measure is based on stigmergy as a vehicle of both scalar and temporal proximity, and exploits Self-Organizing Maps to carry out a clustering between terms. To manage the considerable level of flexibility offered by stigmergy in recognizing different patterns, we propose a topology of multilayer network based on Stigmergic Receptive Fields (SRFs). We discuss in detail a design strategy of the novel architecture that fully exploits the modeling capabilities of the contributing SRFs.
To study the effectiveness of the proposed approach, we applied it to analyze the discussion topics occurred on *Twitter* during the November 2015 Paris terrorist attack. We compared the quality of the stigmergic similarity with a popular distance measure used for time series, namely the Dynamic Time Warping (DTW) distance [78] [64].

Application of the Stigmergic Architecture

Given a stream of tweets $E \equiv \{e\}$ related to a specific event, let us consider for each tweet *e* the set of contained terms $\{term_i\}$. A score is assigned to each term: $s(term_i, e) = \{1$ if $term_i \in e$, 0 otherwise$\}$. The aggregated score of $\{term_i\}$ in the stream *E* is the sum of the scores of each term in each tweet, namely $s(term_i, E) = \sum s(term_i, e) : e \in E$.
The overall output of the system is a *relational term cloud*. Similarly to a conventional term cloud, the font size of $term_i$ is linearly proportional to $s(term_i, E)$ and scaled to the range [*minFontSize*, *maxFontSize*] defined by the user.
In the relational term cloud, the similarity of the *D* terms is the result of a clustering process based on a similarity measure between terms. The similarity measure between two terms is based on the similarity between the related time series. A time series related to a term is generated considering the discrete time $0 < k \leq K$ of consecutive, non-overlapping temporal slots of fixed length τ. The element $d(k)$ of the time series is the score $s(term_i, E(k))$, where $E(k) \subset E$ is the sub-stream of tweet published in slot *k*.

The Stigmergic Receptive Field

In order to calculate the similarity between two time series we propose a connectionist architecture which relies on bio-inspired scalar and temporal aggregation mechanisms. In the proposed architecture, the input samples are aggregated by functional structures called *trails*. A trail is a short-term and short-size memory, appearing and staying spontaneous at runtime when local dynamism in samples occurs. The trailing process is inspired by *stigmergy* [6], [5] , [140], an insects' coordination mechanism, and is managed by computational units called *Stigmergic Receptive Fields* (SRFs). An SRF provides a (dis-)similarity measure between samples streams. SRFs are organized into a multilayer connectionist system, and adapted to contextual behavior by means of the DE algorithm. Thus, the novelty of the undertaken study relates to the structure of an SRF and a way in which such receptive fields are being formed and adapted [56] [141]. The concept of receptive field is the most prominent and ubiquitous computational mechanism employed by biological information processing systems [124]. In our approach, it relates to an architectural style consisting of a collection of general purpose

local models (archetypes) that detects a micro-behavior of the time series. An example of micro-behavior is *rising topic*, which means that the topic (term) shows a sudden increase of occurrences over time. Since micro-behavior is not related to a specific event, any receptive field can be reused for a broad class of events. Thus, the use of SRFs can be proposed as a more general and effective way of designing micro-pattern detection. Moreover, SRF can be used in a multilayered architecture, thus providing further levels of processing so as to realize a macro analysis.

Basically, the purpose of an SRF is to provide a degree of similarity between a current micro-behavior, represented by a segment of the time series, and an *archetype* micro-behavior, referring to a pure form time series which embodies a behavioral class of a micro event. Thus, in order to reveal the underlying properties of an input time series, it is divided into a sequence of discrete segments and each segment is compared with an archetype via a receptive fields.

Figure 43 shows the structure of a single SRF. Let $\bar{d}(k)$ and $d(k)$ denote the values of two time series at time *k*. More specifically, the data samples of the archetype are denoted by $\bar{d}(k)$, represented in gray color, is a constant segment of the SRF, whereas the data samples of the time series $d(k)$ periodically feed the SRF. The first three processing modules of the SRF are exactly the same for the archetype and the input segment. To avoid redundancy in the figure, these modules for the archetype are represented as gray shadow of the corresponding modules of the input segment.

Before processing, a linear transformation of the original time series called min-max normalization is assumed. It is a linear mapping of the data in the interval [0,1], i.e., $d(k) = [d(k) - d_{MIN}]/ [d_{MAX} - d_{MIN}]$, where the bounds d_{MIN} and d_{MAX} are estimated in an observation time window. To assure samples are positioned in-between 0 and 1, the results are clipped to [0,1].

First, normalized data samples are processed by *clumping*, in which samples of a particular range group close to one another. Clumping is a kind of soft discretization of the continuous-valued samples to a set of levels. Second, with the *marking* processes each data sample produces a corresponding *mark*, represented as a trapezoid. Third, with the *trailing* process, the accumulation and the evaporation of the marks over time create the *trail* structure. Fourth, the *similarity* compares the current and the archetype trails. Fifth, the *activation* increases/decreases the rate of similarity potential firing in the cell. Here, the term "activation" is borrowed from neural sciences: it inhibits low intensity signals while boosts signals reaching a certain level to enable the next layer of processing [142].

Each SRF should be properly parameterized to enable an effective aggregation of samples and output activation. For example, short-life marks evaporate too fast, preventing aggregation and pattern reinforcement, whereas long-life marks cause early activation. The *adaptation* module uses the DE algorithm to adapt the parameters of the SRF with respect to the fitness function. The fitness function is the Mean Squared Error (MSE) between the output the SRF provided for a certain input and the desired output provided in the tuning set for the same input. In figure, the tuning set is denoted by asterisks: it is a sequence of *(input, desired output)* pairs, on the left side, together with a corresponding sequence of *actual output* values, on the right side. In a fitting solution, the desired and actual output values corresponding to the same input are made very close to each other.

Figure 44(a) shows the topology of a stigmergic perceptron. The stigmergic perceptron computes a single archetype from a sequence of input samples (representing a time series segment) by forming a linear combination of SRFs, parameterized for achieving

some desired mapping [124].
More specifically, Figure 44 (a) shows six SRFs. Each SRF is associated to a class in a totally ordered set. Indeed, an ordering relation between the archetypes can be defined considering the barycenters of the related trails. Thus, the output of each SRF is a prefixed natural number in the interval [0,5]. In the output layer, the average of the natural numbers of each SRF is weighted by the related activation. As a final result, it provides a linear combination of the SRFs outputs in the real-valued interval [0,5].

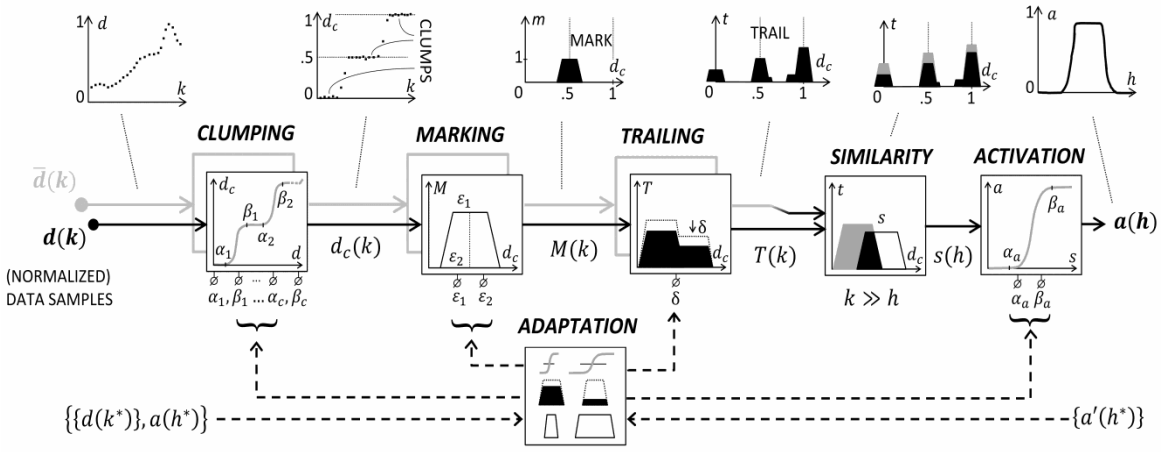
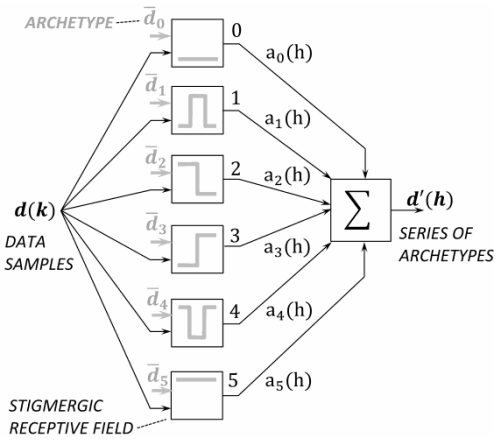
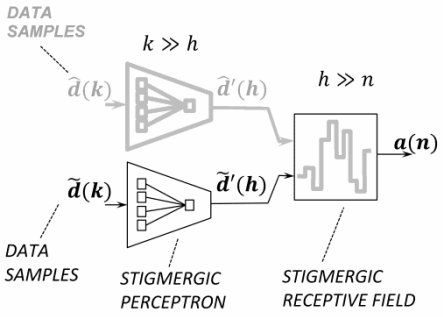


Figure 43 Structure of a Stigmergic Receptive Field.

Figure 44 (b) shows the topology of a multilayer architecture of SRFs. In the first layer, each SRF is fed with the input data series, to provide the degree of similarity to each archetypal pattern. An archetype is a pure stigmergic form generated by a behavioral micro pattern. Thus, an SRF carries out a scalar-temporal micro clustering plus similarity detection, corresponding to micro-pattern detection. The activation of each stigmergic receptive field is then used to generate a higher level time series through the stigmergic perceptron. In the next layer, another SRF is used to provide the degree of similarity between two time series of archetypes, more precisely the current and the reference time series. Here, the adaptation is based on similarity samples provided by an application domain expert. This layer carries out a macro-level similarity between two time series. The next subsections formalize each module; we also include an illustrative example to elaborate on the main features of the ensuing processing.



(a)



(b)

Figure 44 (a) Topology of a stigmergic perceptron. (b) Topology of a multilayer architecture of SRF.

The input/output interfaces of SRF

Figure 45 shows an example of input data for an SRF: it takes the input signal $d(k)$ and the archetype signal $\bar{d}_2(k)$ to return the similarity between them. The archetype $\bar{d}_2(k)$ manifests the pattern for a *falling topic*. In general, micro-patterns of changes over time may manifest in different forms [143], as illustrated in Figure 44 (a): *dead topic (0)*, *cold topic (1)*, *falling topic (2)*, *rising topic (3)*, *trending topic (4)*, and *hot topic (5)*. Thus, we assigned each micro-pattern to a different SRF.

The similarity value, $0 \le s(h) \le 1$, is close to 1 if the signal is very similar to the archetype. Since a single output similarity sample is provided with many input data samples, it follows that $k >> h$. Fig.4 shows that k=20 data samples provide one (h=1) similarity sample.

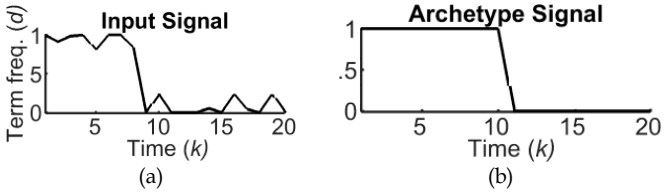


Figure 45 Example of input data for an SRF.

Figure 46 shows the *clumping* process. Here, the input and the output signals are represented with dashed and solid lines, respectively. It is evident that the clumping reduces the ripples and allows for a better distinction of the critical phenomena during unfolding events, with a better detection of the progressing levels. The clumping is

generated by multiple *s-shape* functions, endowed with $2 \cdot c$ parameters $(\alpha_1, \beta_1, ..., \alpha_C, \beta_C) \in [0,1]$. More formally: (a) input values in the interval $[0, \alpha_1]$ are mapped to level 0; input values in the interval $[\beta_i, \alpha_{i+1}]$ are mapped to level i/c, and so on; input values in the interval $[\beta_C, 1]$ are mapped to level 1. Continuous values in the interval $[\alpha_i, \beta_i]$ are softly/hardly transferred into discrete counterparts or levels, depending on the distance between α_i and β_i. All archetypes shown in Figure 44 use two levels, thus *c*=1.

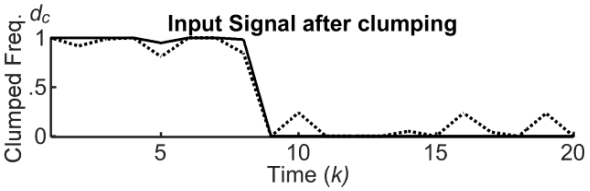


Figure 46 The clumping process, with two levels, $\alpha_1 = 0.25$ and $\beta_1 = 0.75$

In essence, at the output interface the *activation* process is always generated on two levels by a single *s-shape* function. Thus, the activation is set via a pair of parameters (α_a, β_a), as represented in Figure 44.

The Marking process

The marking takes a clumped sample $d_c(k)$ as an input, and releases a mark $M(k)$ to the *trailing*. A mark has three structural attributes: the center position $d_c(k)$, the intensity *I*, and the extensions of the upper/lower bases, $\varepsilon_1/\varepsilon_2$. Fig.6 shows an input signal after clumping, and the release of three trapezoidal marks at the time ticks 0, 2, and 12, with intensity *I*=1, upper and lower bases ε_1=*0*.3 and $\varepsilon_2 = 1.5 \cdot \varepsilon_1$, in a time window of 20 total ticks. More precisely, the sample $d_c(1) = 1$ generates the first mark centered on 1.

The Trailing process

After each step, the mark intensity decreases by a value θ, $0 < \theta < 1$, called evaporation rate. Evaporation leads to progressive disappearance of the mark. However, subsequent marks can reinforce previous mark when overlapping. Fig. 6 shows, at time 2, the release of a second mark on the *trail*, $T(k)$, which is made by the first mark evaporated by a factor $\theta = 0.25$. For this reason, the new trail is lower than 2. Finally Figure 47 shows, at time 12, the evolution over time of the trail. Here, the right trapezoid is higher than the left one, because more marks were recently released on 1 with respect to 0. This example shows that the stigmergic space acts as a scalar and temporal memory of the time series.

The Similarity process

The similarity takes two trails as inputs, namely the archetype trail, $\bar{T}(h)$, and the current signal trails, $T(h)$, to provide their similarity *s(h)*. As a similarity measure we adopt the Jaccard's index [144], which is then used for determining the (dis-)similarity of sample sets or functions. Let *A*(*x*) and *B*(*x*) be two trails defined on the variable *x*; the

similarity is calculated as $|A(x) \cap B(x)| \,/\, |A(x) \cup B(x)|$ where $A(x) \cap B(x) = \min[A(x), B(x)]$, $A(x) \cup B(x) = \max[A(x), B(x)]$. Thus, the similarity is maximal assumes value 1 for identical sets, and decreases progressively to zero with the increase of the non-overlapping portion, as shown in Figure 48.

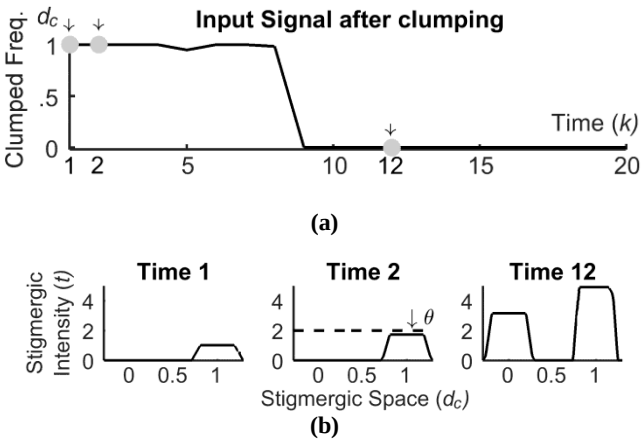


Figure 47 The marking and the trailing processes: (a) time domain, (b) stigmergy domain

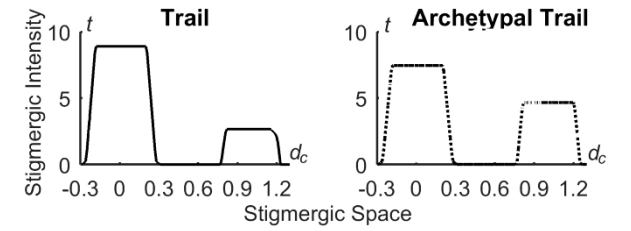
The Stigmergic Perceptron

The Stigmergic Perceptron (SP) compares an input time series fragment $d(k)$ with a collection of archetypes P, using SRFs, and returns an information granule $d'(h)$ reflecting the activation of some close SRFs. Indeed, a total ordering relation between the SRFs is defined, in the stigmergic space. Thus, each SRF is associated to a prefixed integer number in the interval [0,P-1]. Figure 49(a) and Figure 49 (b) show the considered archetypes and the related trails, respectively. Here, the trails show a progressive shift of the barycenter from 0 to 1, which clearly is an ordering relation.
The output information granule is taken as the average of the SRFs natural numbers, weighted by their activation levels:

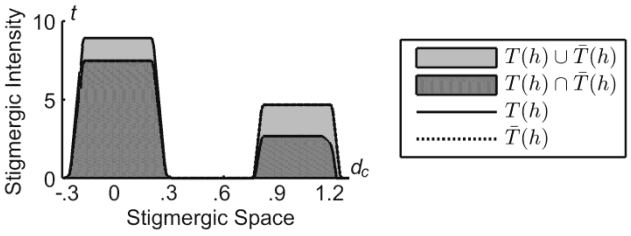
$$d'(h) = \sum_{1}^{P} p \cdot a_p(h) \Big/ \sum_{1}^{P} a_p(h) \qquad \textbf{25}$$

where $a_p(h) \in [0,1]$ is the output of the p^{th} SRF.

$$s(h) = \frac{T(h) \cap \bar{T}(h)}{T(h) \cup \bar{T}(h)} \qquad \textbf{26}$$



(a)



(b)

Figure 48 (a) Two trails and (b) their similarity

Figure 49 (c) shows in solid line the trails of the pilot signal in the six SRFs. Figure 49 (a) top shows also the similarity value between the input trails and the corresponding archetype trails. In the case of Figure 49 (a), the overall output for the pilot time window is $\tilde{a}'(h) = 2$, which identifies the *falling topic* behavior.

The Second Layer SRF

In general, an SRF is able to detect the similarity between two time series segments. At the first layer, SRFs are used to detect the archetype(s) representing the input segment. The resulting information granule represents a sample of a second level time series. The process uses sliding time windows that overlap for the half of their length, to assure a smooth output time series. Thus, the first layer provides a time series of scalar-temporal information granules. Thus, at the second (macro) level, two time series can be compared using another SRF, as represented in Figure 44(b).
The resulting activation can be used as a dissimilarity function, such as $1-a(n)$, where $a(n)$ is the Jaccard's similarity index. Thus, dissimilarity is close to 0 (1) if the two time series show the same (different) temporal dynamics. Finally, a dissimilarity matrix between the *D* time series related to terms can be generated.

The Adaptation with Differential Evolution

The multilayer architecture requires a proper parameterization to generate a suitable similarity measure. More specifically, the temporal parameters τ and *K* regulate the granularity of the analysis. They depend on the span (first and last timestamp), the volume of the stream of tweets *E*, and the dynamics of the events. The parameters of the SRF, namely α_c, β_c, ε (setting ε=ε_1, and fixing the ratio $\varepsilon_1/\varepsilon_2$ to a constant value), δ, α_a, β_a, depend on the parameter *K* and the shape of the archetypal signals. At the first layer, since each SRF is specialized to recognize a specific scalar-temporal pattern, for each

SRF a different adaptation process is carried out by the DE algorithm. At the second layer, the adaptation of the SRF is based on a set of similarity measures between terms established by a group of analysts of the event.

In the DE algorithm, a solution is represented by a real *v*-dimensional vector, where *v* is the number of parameters to tune. Here, this vector has 6 elements $v = (\alpha_c, \beta_c, \varepsilon, \theta, \alpha_a, \beta_a)$. The fitness function is specified as the MSE (Mean Squared Error) between the output of similarity calculated by the module and the corresponding similarity provided for the related tuning inputs:

$$f(Z) = \sum_{h=1}^{|Z|} (a(h^*) - a'(h^*))^2 / |Z| \quad \textbf{27}$$

The objective of the optimization is to minimize the fitness function. In the tuning set, the target values *a(h*)* provided for each SRF are 1 (similar) and 0 (dissimilar). At each generation of the DE, a population $V_g = \{v_{1,g}, \ldots, v_{|V|,g}\}$ of candidate solutions is generated, where *g* is the current generation. In the literature, different ranges of population are suggested [37]. Population size can vary from a minimum of $2 \cdot |V|$ to a maximum of $40 \cdot |V|$. A higher size of the population increases the chance to find an optimal solution but it is more time consuming. At each iteration, and for each member (target) of the population, a mutant vector is first generated by mutation of selected members, and a trial vector is then generated by crossover of mutant and target. Finally, the best fitting member among trial and target replaces the target [12].
The DE algorithm itself can be configured by using three parameters [37]: the population *N*, the differential weight *F*, and the crossover rate *CR*. To avoid premature convergence towards a non-optimal solution, such parameters should be properly tuned. Many strategies of the DE algorithm have been designed, by combining different structure and parameterization of mutation and crossover operators [35]. We adopted the *DE/1/rand/bin* [36] version, which computes the mutant vector as:

$$v_m = v_{r_1,g} + F \cdot (v_{r_2,g} - v_{r_3,g}) \quad \textbf{28}$$

Where $v_{r_1,g}, v_{r_2,g}, v_{r_3,g} \in V_g$ are three randomly selected population members, with $r_1 \neq r_2 \neq r_3$. The differential weight $F \in [0,2]$ controls the amplification of the differential variation. *F* is usually set in the range [0.4-1) [35]. There are different crossover methods for the generation of a trial vector v_b. Results show that a good approach is generally based on the binomial crossover [36]. With binomial crossover, a component of the offspring is taken with probability *CR* from the mutant vector, and with probability 1-*CR* from the target vector. A sound value of *CR* is between 0.3 and 0.9 [36].

Definition of the search space

When some domain knowledge is available, the search space of each parameter can be constrained to improve and speed up the convergence of the SRF adaptation. In [6] the authors pointed out that DE can benefit from: (i) the limitation of the space domain of each parameter, (ii) the injection of a solution based on human-expert knowledge. In terms of limitation to the space domain, SRFs at the first layer should handle more scalar and temporal perturbation (e.g., ripple, amplitude scaling, drift, offset translation, and temporal drift [140]), than the SRF at the second layer. Indeed, the first layer is fed with raw samples, whereas the second layer is fed with archetypal patterns. For this

reason, with respect to the total search space of each layer (i.e., [0,1] and [0,5], respectively) the space domain at the second layer is narrower than at the first one.

More specifically, considering the clumping process, the search space for each SRF at the first and the second layers is $\alpha_c \in [0,0.4]$, $\beta_c \in [0.6,1]$, and $\alpha_c = [0,0.05]$, $\beta_c = [0.95,1]$, respectively. Such intervals have been established by means of a preliminary estimation of the noise at the input time series of each SRF. We remark that the input noise in the two levels differs by a factor of about 10. In the activation process, low/high values of both α_a and β_a facilitate/impede the sudden activation of the SRF, whereas different values of them enable a soft and incremental activation. The search space for each SRF at the first and the second layers is $\alpha_a \in [0.45,0.75]$, $\beta_a \in [0.8,1]$, and $\alpha_a \in [0,0.2]$, $\beta_a \in [4.8,5]$, respectively. Let us consider the marking processes. For the sake of interpretability of the trails, the ratio $\varepsilon_1/\varepsilon_2$=2/3 of trapezoids is fixed. Thus, the parameter $\varepsilon=\varepsilon_1$ regulates the spatial aggregation of the marks, and is a means to absorb scalar perturbation. Indeed, if ε is close to 0, two consecutive marks cannot interact with each other unless the two samples are identical. If ε is close to 1, all marks reinforce each other without distinction between patterns. Thus, the related search space has been set considering the average perturbation and variability in the input signal, at the first and second layer: $\varepsilon \in [0,0.25]$ and $\varepsilon \in [0,3]$, respectively. Considering the trailing process, the evaporation δ is a mean to absorb the temporal variations between samples controlling the duration of the trail. A low value of δ implies a long duration of the trail, and then it becomes possible to aggregate samples occurring in very different instants of time. In contrast, a high value of δ implies that only subsequent samples can aggregate. At the first layer, the related search space has been set to allow a distinction between the archetypes *cold*, *rising*, and *falling*: $\delta \in [0.4,0.9]$. At the second layer, δ should be sufficiently low to allow long-term similarity between the two time series; these values are taken from the range [0.05, 0.3].

Table 22 shows, for each parameter of a first-layer SRF and for each archetype, the value generated by human expert tuning (HE), and the value generated by DE algorithm. In the case of human tuning, the adaptation process adjusted a parameter per time in the same order of the data processing. In contrast, the DE algorithm investigates solutions with a higher resolution than human expert. More precisely, the human expert explored solutions with resolution 0.1, whereas DE can achieve resolution 0.01, which is very time consuming for humans.

Table 22 The human tuning (H) and the DE adaptation for each first layer SRF.

Archetype		*Dead*	*Cold*	*Rising*	*Falling*	*Trending*	*Hot*
α_c	HE	0.2	0.2	0.2	0.2	0.2	0.2
	DE	0.23	0.30	0.06	0.4	0.36	0.23
β_a	HE	0.8	0.8	0.8	0.8	0.8	0.8
	DE	0.93	0.70	0.82	0.71	0.74	0.93
ε	HE	0.2	0.1	0.1	0.1	0.1	0.2
	DE	0.06	0.03	0.03	0.03	0.03	0.06
δ	HE	0.8	0.5	0.5	0.5	0.5	0.8
	DE	0.88	0.46	0.42	0.43	0.46	0.88
..	HE	0.7	0.6	0.5	0.5	0.6	0.7
	DE	0.75	0.60	0.55	0.55	0.60	0.75
β_a	HE	0.8	0.9	0.9	0.9	0.9	0.8
	DE	0.79	0.94	0.90	0.90	0.93	0.79

The HE values reported in Table 22 have been found by using the following heuristic

rules. We adjust one parameter per time, in the same sequence of the data processing. Let us consider the sets of examples labeled, *dead topic* and *hot topic*. In the clumping process, the intervals $\alpha_c \in [0, 0.4]$ and $\beta_c \in [0.6, 1]$ are sufficient to include all the occurrences of a *dead topic* and a *hot topic*, respectively. The values set are the mean values of such intervals, $\alpha_c = 0.2$ and $\beta_c = 0.8$. The same values are set for all the SRFs. In the marking and the trailing process, the mark width ε regulates the spatial aggregation of the marks and it is an important means to absorb scalar noise such as amplitude scaling, offset translation, linear drift and temporal drift [140]. The mark evaporation θ is a sort of short memory of the trail. A low (high) value implies a long (short) -term memory. For *Cold*, *Rising*, *Falling*, and *Trending topic*, which are characterized by variation between levels, the intervals $\theta \in [0.4, 0.6]$ and $\varepsilon \in [0, 0.3]$ guarantee a good distinguishability between trails. The values set are the mean values of such intervals, ε = 0.1 and $\theta = 0.5$. For *Dead topic* and *Hot topic* both parameters can be increased to allow a better distinguishability, ε=0.2 and $\theta = 0.8$. Finally, for the activation α_a and β_a: in general, high values of both are required if there are signals with similar pattern otherwise it is advisable to set low the values to facilitate the activation of the SRF.

Optimal Tuning of V, F and CR

In this Section we compare the performance of DE in terms of fitness, with different sizes of population V, and different values of the parameters CR, F. More specifically, the values considered are: $V \in \{10, 20, 30\}$, $CR \in \{0.2, 0.4, 0.6, 0.8\}$, and $F \in \{0.4, 0.8, 1.6, 2\}$. As a performance indicator, the average fitness of the population after 20 generations has been considered. Table 23 shows the indicator for V=30 and for different values of CR and F. It is apparent that the values of CR and F do not affect significantly the performance. Table 24 shows the average number of generations to achieve 0.5 as average fitness of the population, over 5 trials, for different combinations of CR and F, and for the population size V=30. According to the table, the values (CR, F) = (0.4, 0.8) cause faster convergence times of the tuning. This result is in line with other researches in the literature. Finally, Figure 49 shows the average fitness of the population against the generation, for different values of V. It is apparent that higher values of V, up to 30 members, cause a significant reduction of the fitness. This result is in line with those presented in the literature. Thus, after these experiments the setting V =30, CR = 0.4, and F = 0.8 has been considered.

The SOM-based clustering and the relational term cloud

Given a $D \times D$ dissimilarity matrix between terms, calculated using the stigmergic similarity between time series, the purpose of the clustering process is to visualize the terms on a two-dimensional space, as a *relational term cloud* (RTD).
In an RTD, terms are grouped according to their scalar-temporal usage, thus generating different clusters each characterizing a social discussion topic.
To produce a two-dimensional representation of the dissimilarity matrix we adopt a Self-Organizing Map (SOM), developed with the use of unsupervised learning [145]. A SOM applies competitive learning on the stigmergic dissimilarity matrix to preserve the topological properties of the matrix itself. This makes SOM effective to produce a two-dimensional view of high-dimensional data.

Table 23 The average fitness of the population, over 5 trials, for V = 30, and for different combinations of CR and F.

V=30		CR			
		0.2	0.4	0.6	0.8
	0.4	0.545	0.478	0.494	0.551
	0.8	0.483	**0.473**	0.482	**0.473**
F	**1.2**	0.487	0.475	0.474	0.476
	1.6	0.500	0.475	0.475	0.481
	2.0	0.492	0.483	0.475	0.487

Table 24 The average number of generations to converge to fitness 0.5, over 5 trials, for V = 30, and for different combinations of CR and F.

V=30		CR			
		0.2	0.4	0.6	0.8
	0.4	20.0	17.4	17.0	20.0
	0.8	18.8	**16.8**	17.4	18.0
F	**1.2**	19.2	17.2	17.8	17.6
	1.6	19.0	18.0	17.6	17.6
	2.0	18.6	18.4	18.4	18.6

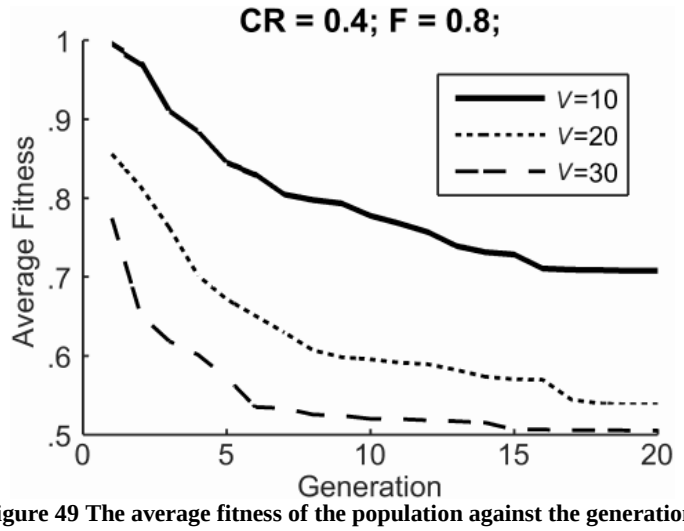


Figure 49 The average fitness of the population against the generation.

The SOM exhibits a single layer of $n_1 \times n_2$ neurons, each made of a D-dimensional array of weights, whose training causes different part of the network to respond similarly to similar input patterns. This behavior is partly inspired by how visual, auditory and other sensory information is handled in separate parts of the cerebral cortex of the brain, and matches with the concept of receptive field. More specifically, in the training of the network the weights of the neurons are initialized to random values. Iteratively (i) each row of the similarity matrix is fed to the network; (ii) its Euclidean distance to all neurons is computed; (iii) the weights of the most matching neurons are then adjusted to be closer to the input.

A common feature of SOM is to reduce the dimensionality of data [146]. It means that information about distance and angle is lost in the training process but proximity relationship between neighbor neurons is preserved: neurons which are close one to

another in the input space should be close in the SOM space. Since each neuron becomes a representative of a discussion topic as a group of patterns in the input data set, discussion topics can be represented on a 2D-space, exploiting the Euclidean distances between neurons. For this purpose, we adopt a Force-directed graph drawing algorithm [147], which positions the nodes of a graph in two-dimensional space. In essence, the algorithm assigns forces among the set of edges and the set of nodes, based on their relative positions, and then uses these forces either to simulate the motion of the edges and nodes or to minimize their energy.

As a result, each discussion topic is represented in a different color, and the relative position between different discussion topics reflects the corresponding proximity between nodes of the SOM. More specifically, each cluster of terms is then represented as a term cloud whose barycenter is placed on the position assigned by a force-directed graph algorithm [148]. In particular, the score of each term $s(term_i, E)$ determines the font size of *i*-th term. The most used terms appear more visible in the picture, and discussion topics which are scalarly and temporally related appear closer.

Figure 50 shows the relational term cloud generated from the overall process. The major terms are located on different topics (clusters), namely *dead*, *attack*, *bataclan*, *terrorists*, and *shootings*. On the bottom-left side, there is a topic regarding the political speech and the emerging reactions. Here, the term *terrorists* was one of the most used overnight, especially in conjunction with the speech of Barack *Obama* and Francois *Hollande*, which began the speech with *solidarity* towards the *brothers* and *Parisians*. Indeed, the last two terms belong to the neighboring cluster. Another political sub-topic raised by the community was the management of *borders*. At the same time an appeal to *Italians* in Paris that may *need*, was spread re-tweeting details of the Italian *Embassy*. The relevance of this topic, represented by a small neighboring cluster, is caused by the filter by Italian language which applied to the posts stream.

The next cluster on the right is focused on people *killed* at the *concert*, *hall*, and the strong connection with *Islamic*, *religion*, which raises the *emergency* level for *Rome* and *London*. Moreover, the community expressed disagreement using the expression "*Allah* is not *great"*. On the top-middle, the absolutely important term is *dead*, widely used overnight when reporting the *ongoing* events, producing *blood* at the *theater* and at the *Stade*. On the right side, there are different discussions topics, on details about the story. The terms *attack*, *terrorist-attacks*, *bataclan*, *victims* are the most prominent and distinguish different aspects. More specifically, on the top-right there is the *shooting* at the *downtown*, at the *restaurant*, with the presence of *hostage* and *victims*. It is worth noting how the terms *shooting* and *shootings* are quite distant in the cloud. The reason is that the term *shooting* (singular) has been mainly used to report the attacks at the *restaurant*, whereas the term *shootings* (plural) has been largely used for all the overnight events. Indeed, the latter is represented as an individual cluster. At the bottom-right, the terms related to *terrorism* and *terrorist-attack(s)* are grouped together. Nearby, the *horror* and the *fear* emerge because of the attack at the *hearth* of a *city* of the *Europe*. Finally, at the bottom, a well-defined discussion topic on the *terror* of a possible *massacre* in *Italy*.

Overall, four major social discussion topics are highlighted: story of the attacks, general debate on terrorism, speech of political leaders, and discussion on religion.

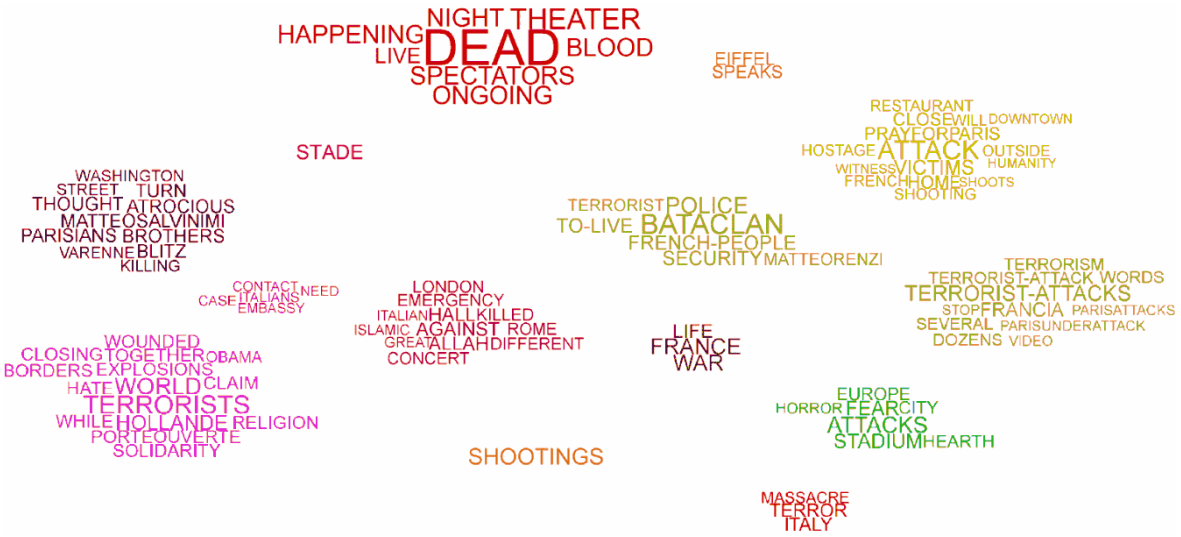


Figure 50 The relational term cloud with some relevant social discussion topics.

Results

The proposed architecture has been tested with different types of socially relevant events, ranging from musical/sport competition to political elections, from weather emergency to terrorist attacks. For the sake of significance, in this section we focus on the record highlighting the most important properties of the approach: a dataset of 188,607 Twitter posts collected during the terrorist attacks in Paris on Nov 13, 2015, between the 9 p.m. of the 13 and the 2 a.m. of the 14[4].

A number of challenges are related to data capturing and filtering of social media data. The challenge related to data capturing lies in gathering, among the sheer amount of social media messages, the most complete and specific set of messages for the detection of a given type of social event. Moreover, since not all the collected messages are actually related to an unfolding event, hence the need of a data filtering step to further reduce the noise among collected messages and retain only the relevant ones. Moreover, techniques are needed in order to analyze relevant messages and infer the occurrence of an event. In this section we briefly summarize the most important steps of the process and of the supporting architecture. The reader is referred to [5] [149].

From the overall collection of tweets related to the event, the first step is to remove the stop words and the historical baseline. The stop words are the most common words universally used in a language, with relevant statistics but without trends in the occurrence of an event, such as *the*, *is*, *at*, *which*, and so on. The historical baseline is a set of sources and related terms generally related to the type of the event rather than to the specific occurrence, such as the posts of press agencies, journalists, and bloggers, since the purpose is to identify terms, which make deviations from the historical baseline [5]. To carry out an accurate text preprocessing, the analysis was made considering only posts in the mother language of the data analyst (i.e., Italian). For the sake of readability, in each figure the terms have been translated to English.

Once the above process has been completed, the top 150 words ranked by frequency have been selected. For each word, the corresponding time series is generated over the collection of tweets, using temporal slots of 1 minute. The resulting set *D* of time series

[4] BBC News, 9 Dec 2015, *Paris attacks: What happened on the night*, http://www.bbc.com/news/world-europe-34818994

has been labeled, in an annotation campaign involving externally enrolled annotators [5]. The task assigned to the annotators was the following: given a time series, assign it to a group of similar time series, considering L groups. The groups are initially empty, and then the process is essentially iterative. For each annotator, the process was repeated three times for L equals to 10, 15, and 20. In case of disagreement between annotators, the membership of a time series to a cluster was decided according to the principle of majority, otherwise the disputed term is discarded. At the end of the annotation process, the collection of time series was divided into L clusters, providing $|D|=102$ time series and their corresponding cluster labels $L^*(i)$.

Clustering Performance

To assess the performance of the clustering we use the B-cubed cluster scoring, which decomposes the evaluation of the clusters estimating the precision and the recall of each item of the dataset [150]. More formally, for a given item i, let us consider $L^*(i)$ and $L(i)$, namely the actual label and the cluster label, respectively. The correctness of the relation between two items i, j reflects the situation that two items are correctly clustered together and they have the same label:

$$Correctness(i, j) = \begin{cases} 1 \text{ if } L(i) = L(j) \ \& \ L^*(i) = L^*(j) \\ 0, \text{ otherwise} \end{cases} \quad \textbf{29}$$

The B-cubed Precision of an item is the proportion of items in its cluster that have the same item's label (including itself). The overall B-cubed *Precision* is the average precision of all items. The B-cubed *Recall* definition is analogous to Precision, replacing the word "cluster" with the word "label":

$$P = avg_{i,j}\left[Correctness(i, j)\right]\Big|_{L(i)=L(j)} \quad \textbf{30}$$

$$R = avg_{i,j}\left[Correctness(i, j)\right]\Big|_{L^*(i)=L^*(j)} \quad \textbf{31}$$

Finally, the B-cubed *F-measure* assesses the performance of the cluster:

$$F_1(R, P) = 2 \cdot R \cdot P / (P + R) \quad \textbf{32}$$

Numerical performance of the clustering process

In this section, we compare the result of the clustering of time series by using three different distance measures and a variant of k-means algorithm (see Chapter 2.3) used in previous literature for time series clustering in social media: our similarity measure based on the Multilayer SRF architecture (M-SRF for short), the Dynamic Time Warping (DTW) distance [78] [64], the Euclidean distance (E), and the K-Spectral Centroid (KSC) clustering developed by Yang and Leskovec [44]. We also test the benefits of min-max normalization of input data on the last three techniques, N-DTW, N-E and N-KSC for short, respectively. Table 25 summarizes the result for different numbers of labels/clusters.

Table 25 Performance of the M-SRF and the DTW distances

|L|	F-Measure ± 95% confidence interval						
	DTW	N-DTW	E	N-E	KSC	N-KSC	M-SRF
10	0.299 ±0.005	0.355 ±0.008	0.297 ±0.007	0.360 ±0.006	0.357 ±0.009	0.349 ±0.009	0.383 ±0.009
15	0.301 ±0.006	0.341 ±0.005	0.309 ±0.007	0.334 ±0.008	0.324 ±0.008	0.327 ±0.008	0.423 ±0.010
20	0.339 ±0.003	0.389 ±0.003	0.344 ±0.004	0.386 ±0.004	0.372 ±0.006	0.370 ±0.006	0.410 ±0.004

A first aspect to highlight is that E and DTW distances exhibit similar performance. This may be ascribed to the fact that the time series do not have temporal drift nor temporal scaling [140]. Thus, the robustness to temporal shift, which is a distinctive feature of DTW, is not exploited. A second aspect is that normalization improves the performance for both the E and the DTW, while there is no significant difference between the KSC and the N-KSC because it is invariant to the total volume of the messages [44]. The KSC has performance in line with the N-E and the N-DTW. The most important result is that the M-SRF outperforms the other measures, thus confirming the effectiveness of our approach. In particular, the M-SRF achieves the best performance using a SOM with 5x3 neurons (i.e. 15 clusters), whereas the DTW and E achieve the best performance with 5x4 neurons (20 clusters).In general, we remark that the performance is not the unique criterion to take into account: interpretability of the cloud is also important. With regard to interpretability, 20 clusters can be considered as an upper bound.

CHAPTER 5
Conclusions

This section draws the conclusion of this Ph.D. thesis. First, we discuss the proposed approach, which adopts marker-based stigmergy as a computational paradigm for spatio-temporal pattern analysis. Then, we sum up the results of each presented application. The final section is devoted to future directions of the research.

5.1 Discussion

In this Ph.D. thesis, we presented a novel approach based on marker-based stigmergy for pattern analysis. More precisely, marker-based stigmergy occurs in biological system, in which pieces of information released in the environment, such as pheromones, enable group coordination for task solution. As a computing paradigm, marker-based stigmergy has been adopted for solving several specific problems such the traveling salesman problem. On the contrary, we proposed a novel technique based on stigmergy for micro-level aggregation of data produced by both human and artificial agents. The ground of our approach is the concept of mark, which is a structure that encapsulates an input data. Marks interact in a computational environment aggregating in a trail structure if close to each other and evaporating over time, losing intensity, if not reinforced. The comparison of a couple trails is the mean for the pattern analysis. This process is managed by the Stigmergic Receptive Field (SRF).
We applied this principle in several contexts with input data of different nature and distinct dynamics which proves the versatility of the technique. It is worth noting that the stigmergy computing is particularly suitable when facing spatial data; in fact, the mark mimics the properties of a chemical pheromone which is a volatile substance that diffuses in a spatial area and has effect for a certain period of time. In contexts such as the monitoring of the position of a person (Chapter 3.2) or vehicles (Chapters 3.3 and 3.5), in which the localization device is subdued to an inaccurate measurement, stigmergy has been proven robust in the detection of shift of the human behavior (Chapter 3.2). However, we also applied the stigmergy computing to different kind of data, such as business key process indicator (Chapter 3.1), number of published messages in social media (Chapter 4), and economical innovation indicators (Chapter 3.4). In fact, the mark structure enables the spatial proximity between data in the

computational environment independently from the nature of the input data, thus it permits the micro-level aggregation and the detection of behavioral patterns. This is a very promising result and should be motif for a wider experimentation of the stigmergy computing in other application domains.
The capability of stigmergy computing to adapt and recognize different patterns depends on the parameters of the SRF: the mark width and evaporation, and the clumping and activation (the input/output interfaces). The mark width tunes the local interaction of data, the evaporation manages the temporal dynamics, the clumping provides a granulation of the input data, and the activation refines the output. To manage the number of parameters of the system, an adaptation process based on Differential Evolution algorithm has been developed to help the user in the parameterization. As we discussed in Chapter 2, Differential Evolution has been recognized as one of the best evolutionary strategy compared with other algorithm such as particle swarm optimization and genetic algorithm. Moreover, it needs only a few parameters and we provided a comprehensive evaluation in Chapter 3 and 4.The results proved the capability of differential evolution in the optimization of the parameters of the SRF allowing an improvement of the performance. In general, the experiment demonstrated that differential evolution performs better with some combination of parameters, such as $0.4<CR<0.8$ and $0.2<F<0.9$; the "DE/best" strategy converges faster toward a good solution than "DE/rand" strategy; differential evolution benefits from the injection of a starting solution and limit constraint of the search space added by human expert of the domain. The adaptation is based on a supervised learning, which means that the evolutionary algorithm compares the output of the SRF with a training set to improve the parameters. However, it is worth noting that the training set can be made both of real data, annotated by human expert, and artificial data, automatically produced by a model of the behavioral pattern. While the former requires extensive human efforts, the latter is far lesser expensive. For this reason, we refer to the adaptation as a partially supervised learning.
Across the development of the presented applications, we refined the technique to perform macro-level analysis, finally developing a multilayer architecture, which is organized according to the connectionist paradigm. More in details, we organized the SRFs in a Stigmergic Perceptron, which decomposes a time series in a sequence of micro temporal pattern. Then, a SRF compares couple of sequences providing a similarity measure which we used to cluster the time series with a Self-Organizing Map. The result of the clustering is in line with human analysis and has better performance compared with other techniques such the dynamic time warping. In particular, the stigmergy computing proves to be highly robust when facing noise in the time series.
Finally, we presented an algorithm for the coordination of swarms of vehicles performing pattern analysis in an unexplored and unstructured environment (Chapter 3.5). The algorithm exploits two biologically inspired mechanisms: drones release a digital pheromone to attract other drones when finding a new target, such as insect foraging; drones mimic the flocking behavior of birds, flying in a dense formation. The results obtained by simulations on real world scenario supports the applicability of the approach as a mean to quick survey an area. More precisely, stigmergy leads the swarm to explore area with potentially undiscovered target and the flocking behavior improves the effect of the stigmergy. In fact, the flocking behavior keeps the drones in proximity of each other, thus the drone perceives the digital pheromone released by other drones.

In the rest of the section, we discuss the results obtained in each application we

presented in Chapter 3 and 4.

In Chapter 3.1, we presented an approach for supporting collaborative analytics in Open Collaborative Networks which is a challenging problem mainly due both to the complex interactions companies may have and the uncertainty such a dynamic environment rises. Business requirements of OCNs reveal characteristics of self-organization, distribution, transparency, and marketing concerns on data flow. Marker-based stigmergy allows protecting business privacy and enabling self-aggregation, thus supporting collaborative analytics when combined with workflows. The approach has been discussed and experimented with real-world data, through a pilot scenario of collaborative order planning.
In Chapter 3.2, we presented an approach for monitoring elderly behavior, by focusing on diseases events. In contrast with the literature in the field, the approach does not require explicit modeling of activities of daily living since it is based on the emergent paradigm. We have shown how localization systems with different error models can be used for this purpose. Moreover, we have discussed and analyzed a real-world case of application, discovering the most interesting properties of the approach.
Since the emergent character of stigmergy depends on biases and scale factors that can vary for different application contexts, an essential component of the design is the parametric adaptation. For this purpose, we designed a fitness function and adopted the differential evolution as an optimization strategy. Experimental results show the effectiveness of the approach and relevant improvements with respect to a human parameterization. In particular, in Chapter 3.3, we presented a design of swarm aggregation of vehicle positions applied to traffic congestion estimation. The design is based on marker-based stigmergy, properly interfaced with input-output activation mechanisms for a better interoperability with the sensing and the application layers. In Chapter 3.4, we designed and developed a software system for assessing unfolding trends in innovation indicators. Experimental results show the effectiveness of the approach and the relevant improvements with respect to a human parameterization.
In Chapter 3.5, we presented a swarm strategy for coordinating small drones performing target search, based on stigmergy and flocking behaviors. The approach uses stigmergy to attract drones in areas with potential targets and employs flocking to organize drones into swarms. Simulated results on synthetic and real-world scenarios proved the benefits of stigmergy and flocking. The overall mechanism can be better enabled if structural parameters are correctly tuned for the given scenario. Thus, an appropriate tuning via Differential Evolution has been presented to adapt parameters to the specific search area. We first evaluated several combinations of the structural parameters of the DE algorithm. Results show that a crossover rate of 0.5 and a differential weight of 0.7 produce better solutions. Then, to test the effectiveness and the reliability of the approach, we compared our algorithm with three search strategies over a number of real-world and synthetic scenarios. As a result, our approach resulted dominant in all scenarios.
Finally, in Chapter 4, we presented a novel approach to identify event-specific social discussion topics from a stream of posts in microblogging. The approach is based on deriving a scalar and temporal similarity measure between terms occurrences, and generating a relational term cloud, which is a cloud whose terms positions are related to the similarity measure. To derive the similarity measure from data, we have developed a novel multi-layer architecture, based on the concept of *Stigmergic Receptive Field*. The stigmergy allows the self-aggregation of samples in the time series, thus generating a

stigmergic trail, which represents a short-time scalar and temporal behavior of the series. In the stigmergic space, the similarity compares the current series with a reference series. The recognition of the combined behavior of multiple SRFs models is made by using a stigmergic perceptron. The similarity is used to guide a Self-Organizing Map, which carries out a clustering of the terms. Experimental studies completed for real-world data show that results are promising and consistent with human analysis, and that the Multi-layer architecture based on the Stigmergic Receptive Field similarity outperforms the dynamic time warping distance, the Euclidean distance and the K-Spectral Centroid algorithm.

5.2 Future Works

In this thesis, we presented six application of stigmergy computing with very promising results. However, it would be engaging to test the multi-layer architecture with other applications to assess the capability of the approach. In particular, interesting challenges are multi-dimensional input data, which would require using more Stigmergic Perceptrons and a consequent revision of the aggregation process of the output of the perceptrons. More precisely at a first stage the Stigmergic Perceptrons, one for each dimension of the input data, detect micro-behavioral single –dimensional patterns; at the second stage, the aggregation of the outputs of the Stigmergic Perceptrons could be adopted to perform macro-analysis, which is the detection of complex patterns occurring in the multi-dimensional data. The aggregation presented in this thesis exploits the ordered arrangement of the Stigmergic Receptive Fields. However, the aggregation could be performed with (i) a logic ruled based approach or (ii) maintaining the connectionist paradigm. The former could be faced with the development of rule learning technique or the definition of rules by an expert of the application domain. The latter could be tackled with a regression technique, such as linear square method, to fit an annotated training set of the multidimensional data.

Moreover, another appealing investigation would be an intensive testing of the architecture on benchmark dataset, which would permit deeper assessment of the properties of the proposed stigmergic computing.

The adaptation of the Stigmergic Receptive Field parameters and the swarm algorithm is supported by an evolutionary algorithm, which is differential evolution algorithm. Despite the very good results obtained in our experiments, it could be very interesting to test some variations of differential evolution such as adaptive strategy. Adaptive strategy consists in the automatic variation of the algorithm parameters during the optimization. Such approach would sensibly lower the efforts of the user in the tuning process. Still concerning the adaptation, it could be interesting to compare the performance of differential evolution algorithm with other evolutionary strategies, such as particle swarm optimization.

Finally, the proposed algorithm for the coordination of vehicles has been intensively tested on simulated scenario with small drones. The promising result assesses the effectiveness of the approach in terms of the localization of the targets in the unstructured and unexplored area. Moreover, the swarm performs the task in a very competitive time, which would be interesting to evaluate considering the actual energy capacity of small aerial drones. Eventually, a deeper analysis of the hardware requirements should lead toward real world experimentation.

Bibliography

[1] T.-C. Fu, «A review on time series data mining,» *Engineering Applications of Artificial Intelligence,* vol. 24, pp. 164-181, 2011.

[2] A. Ciaramella, M. G. Cimino, F. Marcelloni e U. Straccia, «Combining fuzzy logic and semantic web to enable situation-awarness in service reccomandation,» in *International Conference on Database and Expert Systems Applications*, 2010.

[3] A. Ciaramella, M. Cimino, B. Lazzerini e F. Marcelloni, «A situation-aware resource recommender based on fuzzy and semantic web rules,» *International Journal of Uncertainty, Fuzziness and Knowledge-Based System,* vol. 18, n. 4, pp. 411-430, 2010.

[4] D. Vernon, G. Metta e G. Sandini, «A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents,» *IEEE Transactions on Evolutionary Computation,* vol. 11, n. 2, p. 151, 2007.

[5] M. Avvenuti, D. Cesarini e M. G. C. A. Cimino, «MARS, a Multi-Agent System for Assessing Rowers' Coordination via Motion-Based Stigmergy,» *Sensors,* pp. 13, 12218-12243, 2013.

[6] M. G. Cimino, A. Lazzeri e G. Vaglini, «Improving the Analysis of Context-Aware Information via Marker-Based Stigmergy and Differential Evolution,» in *Artificial Intelligence and Soft Computing (ICAISC)*, Zakopane, 2015.

[7] H. Van Dyke Parunak, «A Survey of Environments and Mechanisms for Human-Human Stigmergy,» 2005.

[8] P. P. Grassé, «La Reconstruction du nid et les Coordinations Inter-Individuelles chez,» *Insectes Sociaux,* pp. 6:41-84, 1959.

[9] E. Bonabeau, G. Theraulaz, J.-L. Deneubourg, S. Aron e S. Camazine, «Self-organization in social insects,» *Trends in Ecology & Evolution,* vol. 12, n. 5, pp. 188-193, 1997.

[10] F. Heylighen, «Stigmergy as a Universal Coordination Mechanism: components, varieties and applications,» *Human Stigmergy: Theoretical Developments and New Applications,* 2015.

[11] V. Kachitvichyanukul, «Comparison of Three Evolutionary Algorithms: GA, PSO, and DE,» *Industrial Engineering& Management Systems,* vol. Vol 11 , n. No 3, pp. 215-223, September 2012.

[12] R. Storn e K. Price, «Differential evolution-a,» International Computer Science Institute, Berkeley, CA, 1995.

[13] G. Theraulaz e E. Bonabeau, «A Brief History of Stigmergy,» *Artificial Life,* vol. 5, n. 2, pp. 97-116, 1999.

[14] M. Dorigo, E. Bonabeau e G. Theraulaz, «Ant algorithms and stigmergy,» *Future Generation Computer Systems,* vol. 16, n. 8, pp. 851-871, 2000.

[15] C. B. Mohan e R. Baskaran, «A survey: Ant Colony Optimization based recent research and implementation on several engineering domain,» *Expert Systems with Applications,* vol. 39, n. 4, pp. 4618-4627, 2012.

[16] R. Beckers, O. E. Holland e J.-L. Deneubourg, «From local actions to global tasks: Stigmergy and collective robots,» in *Artificial Life IV*, 1994.

[17] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain e L. Chretien, «The dynamics of collective sorting robot-like ants and ant-like robots,» in *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, 1991.

[18] O. Holland e C. Melhuish, «Stigmergy, self-organization, and sorting in collective robotics,» *Artificial Life,* vol. 5, n. 2, pp. 173-202, 1999.

[19] P. Valckenaers, Hadeli, B. S. Germain, P. Verstraete e H. Van Brussel, «MAS coordination and control based on stigmergy,» *Computers in Industry,* pp. 58, 621–629, 2007.

[20] T. Kuyucu, I. Tanev e K. Shimohara, «Superadditive effect of multi-robot coordination in the exploration of unknown environments via stigmergy,» *Neurocomputing,* vol. 148, pp. 83-90, 2015.

[21] P. Dasgupta, «A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles,» *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS,* pp. VOL. 38, NO. 3, 2008.

[22] Y. Tan e Y. Z. Zhong, «Research advance in swarm robotics,» *Defence Technology,* vol. 9, n. 1, pp. 18-39, 2013.

[23] J. A. Sauter, R. Matthews, H. Van Dyke Parunak e S. A. Brueckner, «Performance of digital pheromones for swarming vehicle control,» in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 2005.

[24] S. Kurihara, H. Tamaki, M. Numao, J. Yano, K. Kagawa e T. Morita, «Traffic congestion forecasting based on pheromone communication model for intelligent transport systems,» in *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, 2009.

[25] P. Bedi, N. Mediratta, S. Dhand, R. Sharma e A. Singhal, «Avoiding traffic jam using ant colony optimization-a novel approach,» in *International Conference onConference on Computational Intelligence and Multimedia Applications*, 2007.

[26] F. Caselli, A. Bonifietti e M. Milano, «Swarm-Based Controller for Traffic Lights Management,» in *Congress of the Italian Association for Artificial Intelligence*, 2015.

[27] T. Ito, R. Kanamori, J. Takahashi, I. Maestre e E. De La Hoz, «The comparison of stigmergy strategies for decentralized traffic congestion control: Preliminary results,» in *Pacific Rim International Conference on Artificial Intelligence*, 2012.

[28] W. Nartz, U. Wilflingseder, G. Pomberger, D. Kolb e H. Hortner, «Self-organising congestion evasion strategies using ant-based pheromones,» *IET Intelligent Transport Systems,* vol. 4, n. 1, pp. 93-112, 2010.

[29] V. Ramos e F. Almeida, «Artificial ant colonies in digital image habitats-a mass behaviour effect study on pattern recognition,» in *Proc. of ANTS 2000 - 2nd Int. Works. on Ant Algorithms*, Brussels, 2000.

[30] S. A. Brueckner e H. Van Dyke Parunak, «Swarming agents for distributed pattern detection and classification,» in *Workshop on Ubiquitous Computing AAMAS*, Bologna, 2002.

[31] F. P. Appio, M. G. Cimino, A. Lazzeri, A. Martini e G. Vaglini, «Fostering distributed business logic in Open Collaborative Networks: an integrated approach based on semantic and swarm coordination,» *Information Systems Frontiers,* pp. 1-28, 2016.

[32] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, Ann Arbor,MI: University of Michigan Press, 1975.

[33] J. Kennedy e R. Eberhart, «Particle swarm optimization,» in *Proceedings of IEEE International Conference on Neural Networks*, Perth, WA, 1995.

[34] J. Vesterstrom e R. Thomsen, «A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems,» in *Congress on Evolutionary Computation, CEC2004*, 2004.

[35] E. Mezura-Montes, J. Velázquez-Reyes e C. A. Coello, «A comparative study of differential evolution variants for global optimization,» in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006.

[36] D. Zaharie, «A comparative analysis of crossover variants in differential evolution,» in *Proceedings of IMCSIT 2007*, 2007.

[37] R. Mallipeddi, P. N. Suganthan, Q. K. Pan e M. F. Tasgetiren, «Differential evolution algorithm with ensemble of parameters and mutation strategies,» *Applied Soft Computing,* vol. 11, n. 2, pp. 1679-1696, 2011.

[38] S. Das e P. N. Sunganthan, «Differential evolution: a survey of the state-of-the-art,» *IEEE Transactions on Evolutionary Computation,,* pp. 4-31, 2011.

[39] I. K. Nikolos e A. N. Brintaki, «Coordinated UAV path planning using differential evolution,» in *Proceedings of the 2005 IEEE International Symposium on Control and Automation Intelligent Control*, 2005.

[40] J. Chakraborty, A. Konar, L. Jain e U. Chakraborty, «Cooperative multi-robot path planning using differential evolution,» *Journal of Intelligent & Fuzzy Systems.*

[41] R. Cruz-Alvarez, F. Montes-Gonzales, E. Mezura-Montes e J. Santos, «Robotic behavior implementation using two different differential evolution variants,» in *Mexican International Conference on Artificial Intelligence*, 2012.

[42] J. Weng e B.-S. Lee, «Event Detection in Twitter,» *ICWSM,* vol. 11, pp. 401-408, 2011.

[43] W. Xie, F. Zhu, J. Jiang, E.-P. Lim e K. Wang, «Topicsketch: Real-time bursty topic detection from twitter,» in *2013 IEEE 13th International Conference on Data Mining (ICDM)*, 2013.

[44] J. Yang e J. Leskovec, «Patterns of temporal variation in online media,» in *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011.

[45] J. Lehmann, B. Goncalves, J. Ramasco e C. Cattuto, «Dynamical classes of collective attention in twitter,» in *Proceedings of the 21st international conference on World Wide Web*, 2012.

[46] G. Stilo e P. Velardi, «Efficient temporal mining of micro-blog texts and its application to event discovery,» *Data Mining and Knowledge Discovery,* pp. 1-31, 2015.

[47] P. Xu, Y. Wu, E. Wei, T.-Q. Peng, S. Liu, J. Zhu e H. Qu, «Visual analysis of topic competition on social media,» *IEEE Transactions on Visualization and Computer Graphics,* vol. 19, n. 12, pp. 2012-2021, 2013.

[48] S. Havre, E. Hetzler, P. Whitney e L. Nowell, «Themeriver: Visualizing thematic changes in large document collections,» *IEEE Transactions on Visualization and Computer Graphics,* vol. 8, n. 1, pp. 9-20, 2002.

[49] C.-H. Lee, «Mining spatio-temporal information on microblogging streams using a density-

based online clustering method,» *Expert Systems with Applications,* vol. 39, n. 10, pp. 9823-9641, 2012.

[50] S. Lohmann, J. Ziegler e L. Tetzlaff, «Comparison of tag cloud layouts: Task-related performance and visual exploration,» in *Human-Computer Interaction--INTERACT 2009*, vol. 5726, Springer, A cura di, Springer, 2009, pp. 392-404.

[51] W. Cui, S. Liu, L. Tan, C. Shi e Y. Song, «Textflow: Towards better understanding of evolving topics in text,» *IEEE Transactions on Visualization and Computer Graphics,* vol. 17, n. 12, pp. 2412-2421, 2011.

[52] D. Archambault, D. Greene, P. Cunningham e N. Hurley, «ThemeCrowds: Multiresolution summaries of twitter usage,» in *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, 2011.

[53] J. Tang, Z. Liu e M. Sun, «Measuring and visualizing interest similarity between microblog users,» in *Web-Age Information Management*, Springer, 2013, pp. 478-489.

[54] V. Raghavan, G. Ver Steeg, A. Galstyan e A. G. Tratakovsky, «Modeling temporal activity patterns in dynamic social networks,» *IEEE Transactions on Computational Social Systems,* vol. 1, n. 1, pp. 89-107, 2014.

[55] G. Liang, W. He, C. Xu, L. Chen e J. Zeng, «Rumor Identification in Microblogging Systems Based on Users' Behavior,» *IEEE Transactions on Computational Social Systems,* vol. 2, n. 3, pp. 99-108, 2015.

[56] P. Barsocchi, M. G. Cimino, E. Ferro, A. Lazzeri, F. Palumbo e G. Vaglini, «Monitoring elderly behavior via indoor position-based stigmergy,» *Pervasive and Mobile Computing,* vol. 23, pp. 26-42, 2015.

[57] A. Tabibiazar e B. Otman, «Kernel-based modeling and optimization for density estimation in transportation systems using floating car data,» in *Intelligent Transportation Systems (ITSC), 14th International IEEE Conference*, 2011 .

[58] Q.-J. Kong, Y. Chen e Y. Liu, «A fusion-based system for road-network traffic state surveillance: a case study of Shanghai,» *Intelligent Transportation Systems Magazine,* vol. 1, n. 1, pp. 37-42, 2009.

[59] Q.-J. Kong, Q. Zhao, C. Wei e Y. Liu, «Efficient traffic state estimation for large-scale urban road networks,» *Intelligent Transportation Systems, IEEE Transactions,* vol. 14, n. 1, pp. 398-407, 2013.

[60] Y. Chen, L. Gao, Z. Li e Y. Liu, «A new method for urban traffic state estimation based on vehicle tracking algorithm,» in *Intelligent Transportation Systems Conference IEEE*, 2007.

[61] Q. Zhao, Q. J. Kong e Y. Liu, «Sample size analysis of GPS probe vehicles for urban traffic state estimation,» in *Intelligent Transportation Systems (ITSC), 14th International IEEE Conference*, 2011.

[62] J. Yoon, B. Noble e M. Liu, «Surface street traffic estimation,» in *Proceedings of the 5th international conference on Mobile systems, applications and services*, 2007.

[63] E. J. Keogh e M. J. Pazzani, «An Enhanced Representation of Time Series which allows fast and Accurate Classification, Clustering and Relevance Feedback,» 1998.

[64] P. Esling e C. Agon, «Time-series data mining,» *ACM Computing Surveys (CSUR),* vol. 45, n. 1, p. 12, 2012.

[65] E. Keogh e S. Kasetty, «On the need for time series data mining benchmarks: a survey and empirical demonstration,» *Data mining and knowledge management,* vol. 7, n. 4, pp. 349-371, 2003.

[66] D. J. Berndt e J. Clifford, «Using dynamic time warping to find patterns in time series,» in

KDD workshop, Seattle, WA, 1994.

[67] G. Das, D. Gunopulos e H. Mannila, «Finding similar time series,» in *European Symposium on Principles of Data Mining and Knowledge Discovery*, 1997.

[68] L. Chen, T. M. Ozsu e V. Oria, «Robust and fast similarity search for moving object trajectories,» in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005.

[69] L. Chen e R. Ng, «On the marriage of Lp-norms and edit distance,» in *Proceedings of the Thirtieth international conference on Very large data bases*, 2004.

[70] H. Shatkay e S. B. Zdonik, «Approximate queries and representations for large data sequences,» in *Proceedings of the Twelfth International Conference on Data Engineering*, 1996.

[71] K.-P. Chan e A. W.-C. Fu, «Efficient time series matching by wavelets,» in *Proceedings of the 15th International Conference on Data Engineering*, 1999.

[72] G. J. Janacek, A. J. Bagnall e M. Powell, «A likelihood ratio distance measure for the similarity between the fourier transform of time series,» in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2005.

[73] M. Vlachos, S. Y. Philip e V. Castelli, «On periodicity detection and structural periodic similarity,» in *SDM*, 2005.

[74] S. Papadimitriou, J. Sun e S. Y. Philip, «Local correlation tracking in time series,» in *Sixth International Conference on Data Mining (ICDM'06)*, 2006.

[75] H. Mannila e J. Seppnen, «Recognizing similar situations from event sequences,» in *Proceedings of the 9th ACM*, 2001.

[76] J. Flanagan, «A non-parametric approach to unsupervised learning and clustering of symbol strings,» in *Proceedings of the 4th Workshop on Self-Organizing Maps (WSOM03)*, 2003.

[77] J. Abfalg, H. Kriegel, P. Kroger, P. Kunath, A. Pryyakhin e M. Renz, «Similarity search on time series based on threshold queries,» in *Proceedings of the 10th International Conference on Extending Database Technology*, 2006.

[78] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang e E. Keogh, «Querying and mining of time series data: experimental comparison of representations and distance measures,» in *Proceedings of the VLDB Endowment*, 2008.

[79] I. Bartolini, P. Ciaccia e M. Patella, «ccurate retrieval of shapes using phase of fourier descriptors and time warping distance,» *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 27, n. 1, p. 142–147, 2005.

[80] Y. Xiong e D. Yeung, «Time series clustering with ARMA mixtures,» *Pattern Recognition,* vol. 37, n. 8, pp. 1675-1689, 2004.

[81] X. Ge e P. Smyth, «Deformable Markov model templates for time-series pattern matching,» in *Proceedings of the 6th ACM International Conference on Knowledge Discovery and Data Mining*, 2000.

[82] M. Bicego, V. Murino e M. Figueiredo, «Similarity-Based clustering of sequences using hidden Markov models,» in *Lecture Notes in Computer Science, Springer*, 2003.

[83] E. Keogh, S. Lonardi e C. Ratanamahatana, «Towards parameter-free data mining,» in *Proc. of 10th ACM International Conference on Knowledge Discovery and Data Mining*.

[84] C. Costa Santos, J. Bernardes, P. Vitanyi e L. Antunes, «Clustering fetal heart rate tracings by compression,» in *Proceedings of the 19th International Symposium on Computer-Based Medical Systems*, 2006.

[85] M. Degli Esposti, C. Farinelli e G. Menconi, «Sequence distance via parsing complexity: Heartbeat signals,» *Chaos, Sol. Fractals,* vol. 39, n. 3, pp. 991-999, 2009.

[86] L. M. Camarinha-Matos e H. Afsarmanesh, «A framework for virtual organization creation in a breeding environment,» *Annual Reviews in Control,* vol. 31, n. 1, pp. 119-135, 2007.

[87] L. M. Camarinha-Matos e H. Ajsarmanesh, «Collaborative networked organizations – Concepts and practice in manufacturing enterprises,» *Computers & Industrial Engineering,* vol. 57, n. 1, pp. 46-60, 2009.

[88] D. Romero e A. Molina, «Virtual organisation breeding environments toolkit: reference model, management framework and instantiation methodology,» *Production Planning & Control,* vol. 21, n. 2, pp. 181-217, 2010.

[89] H. W. Chesbrough e A. K. Crowther, Open Innovation: The New Imperative for Creating and Profiting from Technology, Boston: Harvard Business School Press, 2003.

[90] T. H. Davenport e J. G. Harris, Analytics at Work: Smarter Decisions, Better Results, New York: Harvard Business Review Press, 2010.

[91] C. K. Prahalad e M. S. Krishnan, The new age of innovation: driving co-created value through global networks, New York: McGraw-Hill, 2008.

[92] H. Chen, R. H. Chang e V. C. Storey, «Business Intelligence and Analytics: From Big Data to Big Impact,» *MIS Quarterly,* vol. 36, n. 4, pp. 1165-1188, 2012.

[93] D. Well, «Collaborative Analytics – An Emerging Practice».

[94] P. Fiala, «Information Sharing in Supply Chains,» *Omega,* vol. 33, n. 5, pp. 419-423, 2005.

[95] M. Barut, W. Faisst e J. J. Kanet, «Measuring Supply Chain Coupling: An Information System Perspective,» *European Journal of Purchasing and Supply Management,* vol. 8, n. 3, pp. 161-1771, 2002.

[96] A. B. Palley e M. Kremer, «Sequential Search and Learning from Rank Feedback: Theory and Experimental Evidence,» *Management Science,* vol. 60, n. 10, pp. 2525-2542, 2014.

[97] B. J. Bates, «Information as an economic good: A reevaluation of theoretical approaches,» in *Mediation, Information, and Communication*, New Brunswick, NJ: Transaction Publishers, 1989.

[98] P. Macedo e L. M. Camarinha-Matos, «A qualitative approach to assess the alignment of Value Systems in collaborative enterprises networks,» *Computers & Industrial Engineering,* vol. 64, n. 1, pp. 412-424, 2013.

[99] P. Macedo, T. Cardoso e C.-M. L. M, «Value Systems Alignment in Product Servicing Networks,» *Collaborative Systems for Reindustrialization, IFIP Series,* vol. 408, pp. 71-80, 2013.

[100] Verbeek, M, A guide to modern econometrics, John Wiley & Sons, 2008.

[101] U.Nations, «World Population Ageing,» United Nations Pub., 2013.

[102] T. Kleinberger, M. Becker, A. Holzinger e P. Muller, «Ambient intelligence in assisted living: enable elderly people to handle future interfaces,» *Universal Access in Human–Computer Interaction. Ambient Interaction,* pp. 103-112, 2007.

[103] F. Furfari, J. Ullberg, A. Stimec, F. Furfari, L. Karsson e S. Coradeschi, «Sensor network infrastructure for a home care monitoring system,» *Sensors,* vol. 14, n. 3, pp. 3833-3860, 2014.

[104] S. Chernbumroong, S. Cang, A. Atkins e H. Yu, «Elderly activities recognition and classification for applications in assisted living,,» *Expert Systems with Applications,* vol. 40, n. 5, pp. 1662-1674, 2013.

[105] J. A. Botia, A. Villa e J. Palma, «Ambient assisted living system for in-home monitoring of

healthy independent elders,» *Expert Systems with Applications,* vol. 39, n. 9, pp. 8136-8148, 2012.

[106] H. Storf, T. Kleinberger, M. Becker, M. Schmitt, F. Bomarius e S. Prueckner, An Event-Driven Approach to Activity Recognition in Ambient Assisted Living, Springer, 2009.

[107] C. Anagnostopoulos e S. Hadjiefthymiades, «Advanced fuzzy inference engines in situation aware computing,» *Fuzzy Sets and Systems,* vol. 161, n. 4, pp. 498-521, 2010.

[108] M. G. Cimino, B. Lazzerini, F. Marcelloni e A. Ciaramella, «An adaptive rule-based approach for managing situation-awareness,» *Expert Systems with Applications,* vol. 39, n. 12, pp. 10796-10811, 2012.

[109] P. Rashidi e A. Mihailidis, «A survey on ambient-assisted living tools for older adults,» *IEEE J. Biomed. Health Inform,* vol. 17, n. 3, pp. 579-590, 2013.

[110] H. Storf, M. Becker e M. Riedi, «Rule-based activity recognition framework: Challenges, technique and learning,» in *The 3rd International Conference on Pervasive Computing Technologies (IEEE)*, 2009.

[111] L. Chen e C. Nugent, «Ontology-based activity recognition in intelligent pervasive environments,» *Int. J. Inf. Syst,* vol. 5, n. 4, pp. 410-430, 2009.

[112] L. Chen e I. Khalil, «Activity recognition: Approaches, practices and trends,» *Activity Recognition in Pervasive Intelligent Environments,* pp. 1-31.

[113] H. Liu, H. Darabi, P. Banerjee e J. Liu, «Survey of wireless indoor positioning techniques and systems,» *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.,* vol. 37, n. 6, pp. 1067-1080, 2007.

[114] F. Palumbo e P. Barsocchi, «SALT: Source-agnostic localization technique based on context data from binary sensor networks,» in *Lecture Notes in*, 17-32, 2014.

[115] S. Y. Sato, T. Nakamura e Y. Sato, «Behavior-based stigmergic navigation,» in *Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing-Adjunct, ACM*, 2010.

[116] P. Barsocchi, S. Chessa, F. Furfari e F. Potorti, «Evaluating Ambient Assisted Living Solutions: The Localization Competition,» *Pervasive Computing,* vol. 12, n. 4, pp. 72-79, 2013.

[117] M. Bocca, O. Kaltiokallio e N. Patwari, Radio tomographic imaging for ambient assisted living, Springer, 2013.

[118] D. Tapia, O. Garcia, R. S. Alonso, F. Guevara, J. Catalina, R. Bravo e J. M. Corchado, «The n-Core Polaris Real-Time Locating System at the EvAAL Competition,» in *Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking*, Springer, 2012, pp. 92-106.

[119] A. Moschevikin, A. Galov, A. Soloviev, A. Mikov e S. Reginya, «RealTrac Technology Overview,» in *Evaluating AAL Systems Through Competitive Benchmarking*, Springer, 2013, pp. 60-71.

[120] M. Tan, H.-B. Fang, G.-L. Tian e G. Wei, «Testing multivariate normality in incomplete data of small sample size,» *Journal of Multivariate Analysis,* vol. 93, n. 1, pp. 164-179, 2005.

[121] K. V. Mardia, «Applications of some measures of multivariate skewness and kurtosis in testing normality and robustness studies,» *The Indian Journal of Statistics, Series B,* pp. 114-128, 1974.

[122] g. Fasano e A. Franceschini, «A multidimensional version of the Kolmogorov--Smirnov test,» *Monthly Notices of the Royal Astronomical Society,* vol. 225, n. 1, pp. 155-170, 1987.

[123] J. A. Doornik e H. Hansen, «An omnibus test for univariate and multivariate normality*,»

Oxford Bulletin of Economics and Statistics, vol. 70, n. s1, pp. 927-939, 2008.

[124] M. G. Cimino, W. Pedrycz, B. Lazzerini e F. Marcelloni, «Using Multilayer Perceptrons as Receptive Fields in the Design of Neural Networks,» *Neurocomputing,* vol. 72, n. 10, pp. 2536-2548, 2009.

[125] S. Pellicer, G. Santa, A. Bleda, R. Maestre, A. J. Jara e A. Gomez Skarmeta, «A global perspective of smart cities: a survey,» in *IEEE 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2013.

[126] A. Ciaramella, M. G. Cimino, B. Lazzerini e F. Marcelloni, «Using context history to personalize a resource reccommender via a genetic algorithm,» in *Proceeding of the International Conference on Intelligent Systems Design and Applications, ISDA*, 2010.

[127] F. Dobslaw, «Recent development in automatic parameter tuning for metaheuristics,» in *Proceedings of the 19th Annual Conference of Doctoral Students-WDS*, 2010.

[128] P. Pellegrini, T. Stutzle e M. Birattari, «A critical analysis of parameter adaptation in ant colony optimization,» *Swarm Intelligence,* vol. 6, n. 1, pp. 23-48, 2012.

[129] D. Foray, «The Econominc Fundamentals of Smart Specialization,» *Ekonomiaz,* vol. 83, pp. 55-82, 2013.

[130] P. McCann e R. Ortega-Argiles, «Smart specialization, regional growth and applications to European union cohesion policy,» *Regional Studies,* vol. 49, pp. 1291-1302, 2013.

[131] O. f. E. C.-o. a. Development, «Innovation-driven growth in regions: the role of Smart Specialization,» OECD, Paris, 2013.

[132] K. Whitehead e C. H. Hugenholtz, «Remote sensing of the environment with small unmanned aircraft systems (UASs), part 1: A review of progress and challenges 1,» *Journal of Unmanned Vehicle Systems,* vol. 2, n. 3, pp. 69-85, 2014.

[133] R. McCune, R. Purta, M. Dobski, A. Jaworski, G. Madey, Y. Wei, A. Madey e B. M. Blake, «Investigations of DDDAS for command and control of uav swarms with agent-based modeling,» in *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, 2013.

[134] M. Brambilla, E. Ferrante, M. Birattari e M. Dorigo, «Swarm robotics: a review from the swarm engineering perspective,» *Swarm Intelligence,* vol. 7, pp. 1-41, 2013.

[135] M. G. Cimino, A. Lazzeri e G. Vaglini, «Combining stigmergic and flocking behaviors to coordinate swarms of drones performing target search,» in *Proceedings of the 6th International Conference on Inforamtion, Intelligence, Systems and Application (IISA2015)*, 2015.

[136] NetLogo, 2016. [Online]. Available: ccl.northwestern.edu/netlogo/.

[137] MatLab, 2016. [Online]. Available: www.mathworks.com/products/matlab.

[138] S. Lohmann, M. Burch, H. Schmauder e D. Weiskopf, «Visual analysis of microblog content using time-varying co-occurrence highlighting in tag clouds,» in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2012.

[139] O. Kaser e D. Lemire, «Tag-cloud drawing: Algorithms for cloud visualization,» in *Proceedings of Tagging and Metadata for Social Information Organization (WWW 2007)*, 2007.

[140] X. Zhang, J. Liu, Y. Du e T. Lv, «A novel clustering method on time series clustering,» *Expert Systems with Applications,* vol. 38, n. 9, pp. 11891-11900, 2011.

[141] A. L. Alfeo, F. P. Appio, M. G. Cimino, A. Lazzeri, A. Martini e G. Vaglini, «An adaptive stigmergy-based system for evaluating technological indicator dynamics in the context of smart specialization,» in *Proc. INSTICC of the 5th International Conference on Pattern*

Recognition, Applications an Methods (ICPRAM), 497-503, 2016.

[142] M. G. Cimino, A. Lazzeri e G. Vaglini, «Enabling swarm aggregation of position data via adaptive stigmergy: a case study in urban traffic flows,» in *Proc. IEEE of the 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*, Corfù - Greece, 2015.

[143] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy e A. Bouchachia, «A survey on concept drift adaptation,» *ACM COmput. Surv.,* vol. 46, n. 4, p. 44, 2014.

[144] P. N. Tan, M. Steinbach e V. Kumar, Introduction to data mining, Pearson Education India, 2006.

[145] T. Kohonen, «The self-organizing map,» *Proceedings of the IEEE,* vol. 78, n. 9, pp. 1464-1480.

[146] J. C. Principe e R. Mikkulainen, «Advances in Slef-Organizing Maps,» in *Proc. of the 7th International Workshop (WSOM)*, St. Augustine, 2009.

[147] T. M. Fruchterman e E. M. Reingold, «Graph drawing by force-directed placement,» *Software: Practice and experience,* vol. 21, n. 11, pp. 1129-1164, 1991.

[148] R. Tamassia, Handbook of Graph Drawing and Visualization, CRC Press, 2013.

[149] C. Chen, J. Zhang, Y. Xie, Y. Xiang, W. Zhou, M. M. Hassan, A. Alelaiwi e M. Alrubaian, «A performance evaluation of machine learning-based streaming spam tweets detection,» *IEEE Transactions on Computational Social Systems,* vol. 2, n. 3, pp. 65-76, 2016.

[150] E. Amigo, J. Gonzalo, J. Artiles e F. Verdejo, «A comparison of extrinsic clustering evaluation metrics based on formal constraints,» *Information retrieval,* vol. 12, n. 4, pp. 461-486, 2009.