Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 02, June 2022.

CONSENSUAL BASED CLASSIFICATION AS EMERGENT DECISIONS IN A COMPLEX SYSTEM

Rabah Mazouzi[1], Malanga Kennedy[2], Cyril De Runz[3] and Herman Akdag[4]

(Received: 9-Dec.-2021, Revised: 30-Jan.-2022, Accepted: 5-Feb.-2022)

ABSTRACT

In massive multi-agent systems that are used to model some complex systems, emergence is a key feature that allows to model high-level states of such systems. According to this perspective, the work we introduce in this paper entails the handling of emergence in massive multi-classifiers that we consider as complex systems. We aim to build a collaborative system for supervised data classification that we expect to provide better performance, compared to conventional classifiers. Modeled as a multi-agent system, the massive multi-classifier is composed of a high number of agents that are interconnected according to a given neighborhood. Each agent plays the role of a weak classifier. At the micro-level, the elementary interaction between agents consists of combining their respective classification results. Every agent, according to the majority vote rule, combines its result with those of its neighbors by taking into account their respective performances. This process is iterated continuously in a cyclic manner within the neighborhood of each agent. Therefore, a complex dynamic will be created within the system. After a certain time, this complex dynamic stabilizes, allowing the exhibition of an emergent structure that will be observed at the macro-level and is considered as a consensual class prediction for the data we want to classify. Obtained experimental results and the comparison with conventional classifiers show the potential of the approach to enhance classification and to be an alternative for classifier combination and aggregation.

KEYWORDS

Collaborative classification, Complex system, Emergence, Multi-agent system.

1. INTRODUCTION

In automatic data classification, precision is a crucial aspect. Enhancing precision still remains an open issue. Since the First classifiers were proposed —which were simple and acted individually—, their accuracy has continued to increase asymptotically. We can see that there are many research works proposing new approaches to achieve more powerful classifiers [1]-[3]. Moreover, since the volume and dimensionality of data to be processed are increasing, it has become necessary to design and use more complex and sophisticated classifiers to process the data [4]. The collaboration of classifiers was among the former approaches, but also the most used to overcome the complex data classification problem. Such approaches propose collaborative architectures by combining multiple classifiers using various schemes [5]. Parallel and series combinations are both trivial schemes that make collaborate classifiers. According to the state-of-the-art in the field of classifier combination, it has been demonstrated that the combination of classifiers according to different schemas can produce better results compared to those obtained by the classifiers considered separately [6]-[9].

Since automatic classification of data is often involved in data processing systems, looking for and reaching acceptable accuracy in such systems remain an important need. So, it remains always necessary to propose new architectures of classifier combinations. Indeed, the complexity of processed data and their huge volumes often hide complex patterns and relationships, where classical Multi-Classifier Systems (MCSs) based on simple combinations cannot be considered to elucidate such relationships and patterns. So, we believe that the use of diversity provided on one hand by the multiplicity of classification algorithms and on the other hand on the diversity within the sets of training data, allows us to propose new unconventional and more sophisticated MCSs to deal with complex data [10]-[11]. Furthermore, in order to take advantage of the large volumes of training data, it is not appropriate to use multi-classifiers with a low number of elementary classifiers. Indeed, if the volume of training data assigned to each classifier is big, then it results in some conventional classification flaws, such as over-fitting and lack of generalization [12]. Therefore, it is necessary to conceive massive multi-classifiers

1. R. Mazouzi is with the Department of Research and Innovation, Umanis SA, Levallois-Perret, France. Email: rmazouzi@umanis.com, ORCID: 0000-0002-5945-5152.
2. M. Kennedy is with Dedan Kimathi Uni. of Techno., Kenya. Email: malangalanga@dkut.ac.ke, ORCID: 0000-0001-8435-5084.
3. C. de Runz is with BDTLN Team of LIFAT Lab., Tours, France. Email: cyril.derunz@univ-tours.fr
4. H. Akdag is with Paragraphe Lab., University of Paris 8, France. Email: herman.akdag@univ-paris8.fr

that can handle large masses of training data and have good performance in terms of generalization and accuracy. Moreover, the volume, variety and velocity properties with which big data are often defined represent challenges in the field of data mining [13]. According to our point of view, these factors increase the complexity of the system and diversity of classifiers. These are actually the two key aspects in the proposed MCS.

However, with a massive multi-classifier that consists of hundreds or even thousands of elementary classifiers, we cannot adopt conventional combination schemes, such as parallel or serial. It is therefore necessary to propose new collaborative architectures of classifiers that allow the use of a large number of elementary classifiers ensuring a coherent and parsimonious integration of the involved classifiers. Indeed, classical combinations of elementary classifiers through serial, parallel or hybrid combinations have not resulted into efficient classifiers, particularly when they are used in the context of big data. This is also true with classifier aggregation, such as boosting and bagging. In such a context, the classical techniques for classifier combination do not carry the high diversity of data. They also do not allow enough interaction between the elementary classifiers in order to produce a consensual classification. Furthermore, classical combination of classifiers and their aggregation cannot be used with massive multi-classifiers that involve hundreds or thousands of elementary classifiers. This is because it would be hard to ensure an explicit interaction within such large sets of interacting elements.

Emergence is an inherent property of complex systems. Despite its wide scientific use, the concept of emergence is not defined unequivocally, since it is used differently depending on the discipline. However, the concept of emergence can be defined as the appearance of a new property of the system at a higher level of observation, called macro-level; this phenomenon results from a dynamic interaction among entities at a micro-level of a complex system. Several computational paradigms are used to model complex systems [14]. Such systems are characterized by a large number of entities that locally interact producing a complex dynamic environment that in turn leads to emergent properties within the system. In this work, we propose an architecture of massive multi-classifier system that simulates a complex system. Each element in such a system represents a classifier located in a neighborhood of other classifiers and with which it interacts, exchanging data and decisions (the labels of the data point). Thus, a dynamic environment is created within the system which consists of perpetual exchange of information among classifiers located in neighborhoods. After a certain time, this leads to spreading of emerging decisions in the entire system.

Within a given neighborhood, classifiers proceed with a classical parallel combination with weighted majority voting [15]. Then, the decision taken by the agent at the center of its neighborhood is adopted by all the classifiers that are in the same neighborhood. Each agent will do the same work considering the neighborhood to which it belongs. Such collaborative approach can be considered as a scale-up of the classical combination and aggregation of classifiers. This allows involvement of a high number of elementary classifiers ensuring a dynamic way of making them interact resulting into a consensual classification.

The complex massive multi-classifier system is modeled as a Multi-Agents System (MAS) [16], where each agent has its own different classifier and interacts with neighboring agents. The interaction consists of exchanging decisions within the same neighborhood. A given agent combines the decisions of its neighbors with its own decision, then spreads in its neighborhood the result of the combination. In such a system —which is complex considering the large number of its interacting elements and the heterogeneity of the data—, we expect that a complex dynamic environment will result and lead —in the manner of self- organized systems— to the emergence of a structure within the system. This structure consists of clusters of neighboring classifiers' agents which have reached a consensus of classification (labeling). The ideal case is when a single cluster of agents having a global consensus emerges such that the cluster is composed of all the agents of the system. According to such a massive classification approach, the high volume and complexity of big data are well dealt with. The resulting classification method can be easily parallelized and implemented in cloud and high-performance computing systems.

The remainder of the paper is organized as follows: In Section 2, we provide a review of the literature concerning both collaborative systems of multi-classifiers and some emergence-based systems. This allows us to discuss various aspects of the emergence in complex systems. Section 3 is devoted to the proposed system in which we start by presenting its architecture and its components. Next, we show how the complex dynamic environment is created in the system and how structures emerge, representing

the classification result observable in the system. In Section 4, we present the experimentation of our system by describing the experimental protocol, obtained results, analysis of the results and discussion. Finally, we conclude in Section 5 by summarizing our findings and highlighting some perspectives for this work.

2. RELATED WORK

As far as we know, few works have exploited emergence in the field of data mining. In fact, in such works, there are multiple challenges. For instance, it is a challenge to set local elements and patterns of interaction between elements so as to ensure the exhibition of an emerging phenomenon. On the other hand, it is also a challenge to detect and exploit an emerging phenomenon when it exists in a given outcome.

In the field of sociology, an agent-based modeling approach has been proposed by Y. Chen et al. in [17]. This modeling aims to study the concept of social capital; i.e., the benefits obtained by individuals through social interactions. These benefits can emerge in the form of social support, camaraderie, solidarity, influence …etc. This social capital should be measured in terms of the emerging structural properties generated by the links between homogeneous and like-minded individuals. In another work [18], the authors proposed a general framework based on social capital games for studying social structural pattern emergence.

A more general framework for the specification and simulation of emergence-based systems modeled as multi-agent systems was proposed by O. Paunovski et al. in [19] to control self-organized systems with emergence. They aimed at identifying in such systems some events specific to emergence and studying the causal relationships between the micro and the macro-levels in these systems. The proposed framework, which is software engineering-oriented, proceeds in two phases: In the first phase, the user proceeds by sequential and iterative refinement of the agent-based model, in order to ensure that the expected global behavior will be reached. This phase aims to detect some local elements that can influence the overall behavior of the system. In the second phase, a statistical correlation analysis allows to test if an emergence decision is observed within the system or not. Such observations allow the user to review some elements of the system in order to restart new modeling and simulation.

In a previous work, the same authors proposed a fuzzy approach for determining the herd forming in multi-agent system-based simulations [20]. The fuzzy reasoning is used to calculate mainly two values specific to individuals and groups within the system. The first expresses the membership of the individual to a given group and the second expresses the cohesion of this group.

In the tagging field, V. Robu et al. have studied the dynamic of tagging created in a collaborative system and how categorization patterns emerge from this activity [21]. In such systems, consensus on tags is reached and expressed as tag frequencies that follow a well-defined distribution law. From this, some structures emerge within the tagging system represented as graphs. The correlations within these graphs are used to extract tag vocabularies by partitioning them into sub-graphs formed by each of the correlated tags. The authors used the Kullback-Leibler distance for measuring the convergence of tag distributions. When this distance is close to zero, the convergence is assumed to have been reached. The time to be allocated to the system to reach a steady state, expressed by the number of tags produced for a given site, was also discussed. At the convergence of the tagging dynamic, the sub-graphs are constructed using a similarity criterion. Emerging vocabulary tags are then identified by using community detection algorithms.

To deal with problems related to data flow analysis in multimedia wireless sensor networks, Wang et al. proposed in [22] an agent-based model of a collaborative system for the classification of intruder targets, where audio information is collected by sensors and processed by statistical methods. Next, a step of classification of the characteristics of these data streams allows to provide the class of the observed target. To perform these treatments, multi-agent negotiation mechanisms preserving energy, which is an essential aspect in sensor networks, are specially designed to distribute the classification tasks among agents using the auction protocol. Individual decisions are combined in the manner of classical Multiple Classifier Systems (MCSs) in order to extend the life of the network and efficiently conduct the collaborative processing.

Among the possible schemes to implement a massive MCS, overcoming the problems associated with

the complexity of very large data, R. Mazouzi et al. proposed in [23] an architecture of a massive MCS modeled as a MAS and as an acquaintance network. In this system, the training data is distributed across a diversified set of classifier agents. Each agent is situated on a node of the network and is surrounded by a set of neighboring agents. The classification process begins with the arrival of the data to classify on a central node. The latter distributes the data across a set of agents designated according to their availability. Each agent labels the data to classify and combines the result with those of the agents in its neighborhood. In order to increase the accuracy of the system, the number of the classifiers involved in the combination can be increased by expanding the neighborhood (considering neighbors of neighbors).

Recently, Maystre et al. [24] have proposed a method to make several users interact in order to collaboratively classify a set of items assuming that the labels are corrupted by noise. Their method is based on a structured probabilistic model that relates the interaction user-item and the noisy labels to the items to classify. Such interaction allows collaboration within the set of users in order to infer the correct label of a given item. In order to classify data streams in the context of Internet of Things, Sun et al. [25] proposed a Misclassification-Aware Collaborative Classification Algorithm (MACCA) that makes two modules to collaborate: the misclassification judgment module and the decision one. Such a strategy is proposed as an alternative of the vote technique, which is considered as degrading results by the authors.

There are few systems that use MAS-based architectures for constructing multi-classifiers. The works proposed according to this paradigm aim at generating new schemes for combining classifiers, different from conventional schemes, based on the simple serial and/or parallel combination. However, no system to our knowledge has addressed the problem from the point of view of complex systems, where emergence can be exploited to differently represent the classification results.

3. A COMPLEX SYSTEM FOR DATA CLASSIFICATION

3.1 Overview

We propose in this paper an emergence-based approach for supervised classification. The approach involves using a complex system paradigm modeled as a multi-agent system with the aim of building a massive multi-classifier for classifying large and complex data. So, the proposed system can be used in the context of Big Data. The multi-agent system consists of a large number of agents organized in a neighborhood system, where each agent is located within several neighbors' agents. The neighbors' agents number varies within a given range. An agent has knowledge about its neighbors and is able to interact with them. It also has a classifier, trained by a dedicated training data subset.

After the training phase of all the classifiers —in order to have the most possible diversified classifiers by using different training subsets—, the system is ready to receive the data item to be classified. The latter, when available, is delivered to all agents in the system. For centralized implementation of the proposed system, this is done immediately by initializing the data to classify among all agents. For a physical distributed implementation, a delivery mechanism must be considered. It may consist in placing the data on any agent, then allowing agents to deliver to their neighbors. Thus, the data is progressively propagated in the system and finishes by reaching all the agents.

An agent begins with classifying a data point using its own classifier. Then, cycle after cycle, it reclassifies the data point, considering the results of classification provided by its neighboring agents. The reclassification consists of a combination of local results, taking into account the accuracy and the performance of every classifier involved in this combination. For this, the classification method adopted shall allow to re-inject results of the previous classification as prior knowledge to the new classification. Considering a given neighborhood, the classification is performed in this neighborhood according to the classical parallel combination with weighted majority vote [26]. The result of the combination may be considered by the agent in question or not, according to the certainty of classification and the performance of neighboring classifiers.

According to this pattern of interaction, classification results in a given neighborhood can spread in the system, because a neighbor agent is in turn a neighbor to other agents in other neighborhoods. The propagation evolves according to the quality of classification. Indeed, if the data point is well classified, then there is a tendency that the neighboring agents spread the result beyond the neighborhood.

However, if the classification quality is lower, there is a tendency that the agents beyond the neighborhood reject this result. The fact of repeating such interactions leads to creation of a complex dynamic environment and after a given time, the dynamic stabilizes on emerging patterns, representing the overall result of classification within the whole system. Ideally, a cluster of the majority of the agents producing the same result is formed. In other cases, separate clusters can be formed and a decision mechanism must be expected on how the result of classification must be retrieved. The worst situation happens when a cyclical dynamic environment —on the edge of chaos— remains in the system. In this case, no overall result can be observed and considered.

3.2 The Classifier Agent

It represents the basic element of the system the task of which is to classify the data presented at the input and interact with its neighbors in order to lead the system to a global consensus in terms of classification. The interaction consists of exchanging the classification results by sharing its own results and receiving the results of its neighbors, then combining them according to a given method of combination. Figure 1 shows the architecture of the used agent. The agent has two roles:

3.2.1 Training

The agent proceeds to train its classifier by using a training sub-set extracted by random sampling from a global training dataset. Let DS be the overall training dataset. By using statistical sampling, DS is partitioned into N sub-sets DS_i, i = 1..N.

The sampling method is based on some probability distribution selected according to the size of the overall training dataset, the nature of data it contains and the application field. In our case, DS is homogeneous; so, the uniform probability law may be used for its stochastic partitioning (Algorithm 1).

Algorithm 1 sampler

```
procedure Sampler(DS, N)                          ▷ dataset and agents' count
    k ← M/N
    for i ← 1,N do                                ▷ for each agent
        DS_i ← ϕ
        for i ← 1,N do                            ▷ for each sample
            Pr ← Random
            idx ← M · Pr
            DS_i ← DS_i ∪ DS[idx]
        end for
    end for
end procedure
```

We have considered "not disjunctive" sub-sets (sampling with replacement), so that the resulting sub-sets overlap with each other. Therefore, the different agents will be close in terms of classification results. Indeed, two agents which share some elements in their respective training subsets tend to produce similar results, thus promoting consensus while computing. This allows controlling the degree of diversity in the set of classifiers, which in the extreme diversity case hinders the convergence of the dynamic within the system.

3.2.2 Classification

This role is performed by the agent in two steps:

1) The calculation of the initial class, using its own classifier.
2) The re-calculation of the class using the classification results of neighbouring agents according to a combination rule. This calculation is iterated while the dynamic of the system has not been stabilized.

In order to be able to re-introduce the combination result in the subsequent calculation step, we opted for the Naive Bayesian classifier [27].

Let k classes C_1, ..., C_k and X be the data point to classify; the class to consider, i.e., Label, corresponds

to the Maximum A posteriori Probability (MAP), according to the Bayes probability law:

$$P(C/X) = \frac{P(X/C).P(C)}{P(X)} \quad (1)$$

In our case, P (X) is constant for the whole data, thereby maximizing P (C/X) is equivalent to maximizing P (X/C)·P (C) and in this case, the resulting class Label may be expressed as follows:

$$Label = argmax_{j,1..k}(P(C_j/X).P(C_j)) \quad (2)$$

This represents the initial calculation made by any agent in order to calculate the first probability vector according to the Bayes probability law. The agent, in subsequent cycles, should always keep a probability vector, which corresponds to the best classification, since the start of the calculation.

Combination Rule

The combination within a given agent uses the classification results from its neighbors' classifiers. These classification results are considered by the agent in question as a probability vectors, where each one corresponds to a neighboring agent. Let $Pr_j^{i(X)}$ be the probability corresponding to the class C_j, calculated by the classifier *i* and let b_i be the classifier weighting factor *i*, calculated from the error rate obtained at the training step using a test dataset. This weighting factor expresses the degree of importance of the classifier, compared to neighboring ones.

$$g_j(X) = \sum_{i \in Neighbors(A)} b_i \, . \, Pr_j^i(X) \quad (3)$$

By this expression, we calculate the elements of the probability vector corresponding to the combined weighted prediction according to the majority vote rule. After the calculation, the agent compares the quality of the new combination with that it keeps as the last better classification quality. The new classification combination is applied only if it is better than that stored at the agent.

When the agent adopts a new vector of probabilities, it will use it as a vector of classes prior probabilities ($P(C)$) in the next calculation cycles. In both cases, the agent starts a new calculation cycle after waiting a given time, required to insure that its neighbors have performed their own cycles.

The pseudo-code in Algorithm 2 represents the cycle (combining process) of an agent A:
Let *Pr(X)* be the set of the vectors of probabilities corresponding to the agents neighboring the agent *A* and *A.Prediction* be its own vector. The following pseudo-code calculates the new probability vector as a result of combination with the neighboring vectors:

Algorithm 2 Agent cycle

```
procedure A.COMBINE(Pr(X))
    for j ← 1,Prediction.size do
        g_j(X) ← 0
        for i ← 1,A.Neighbors do
            b_i . Pr_j^i(X)
        end for
    end for
    PrMax ← argmax(g_j(X))
    if PrMax > argmax(A.Prediction) then
        A.Prediction = g(X)
    end if
end procedure
```

The class kept by the agent corresponds to the maximum of probabilities and its probability represents the quality of the classification of the data point X by the agent. Indeed, the latter updates its classification result according to the combination rule if and only if the new quality (MAP in the vector *PrMax*) is greater than the quality it keeps; namely, *max(A.Prediction)*.

Synchronization

When an agent is about to carry out a re-calculation of combination, it invokes its neighbors to get their

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 02, June 2022.

classification results. It is possible that some of these agents may still have not achieved their current cycle since the last iteration. Hence, their results are the same as used by the agent in question during the previous cycle. But, even if that's the case, this does not adversely affect the calculation. For the next iteration, the calculation will be carried out with the new results of neighboring agents when they are ready.

3.3 The Multi-agent System

As shown in Figure 2, the complex system is built as a massive multi-agent system. First, each agent within the system is trained by a randomly selected sub-set of the whole training data. At this stage, agents are independent, since they do not interact with each other. After agents are created and trained, the neighboring relations are randomly created, where each agent is linked to a set of other agents. This ensemble forms the local neighborhood, where decisions are shared and adjusted. After the dynamic environment of the system is stabilized, the emergent decision (final classification) is provided as an output of the whole system. In the next sub-sections, we provide further details of how such stages are initialized and executed in the multi-agent system.

3.3.1 Creation

A set of N agents are created to form the multi-agent system. Each agent; let it be A, at its initialization, selects a set of NV_A agents that form its neighborhood. This number is arbitrarily chosen by the agent in question belonging to the interval $[Min_V, Max_V]$ (Algorithm 3). The number of neighboring agents defines the density of acquaintances in the system, where the arbitrary choice of this number generates an asymmetrical system. The asymmetrical property allows the multi-classifier system to acquire the aspect of heterogeneity necessary to explore vast spaces of states in search of the best consensual classification.

Algorithm 3 Create Neighborhood

procedure A.CreateNeighberhood(Agents) ▷ agents set
  $A.Neighbors \leftarrow \Phi$
  $NV_A \leftarrow MinV + random(MaxV - MinV)$
  for $i \leftarrow 1, NA_A$ **do**
    $idV \leftarrow 1 + random(Agents.size)$ ▷ agent id≠A.id
    $A.Neighbors \leftarrow A.Neighbors \cup \{Agents(idV)\}$
  end for
end procedure

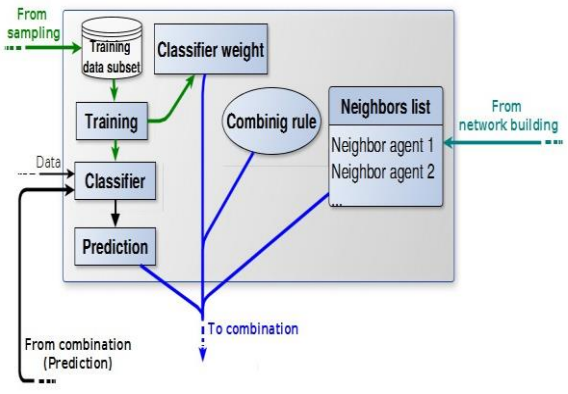


Figure 1. Agent structure.

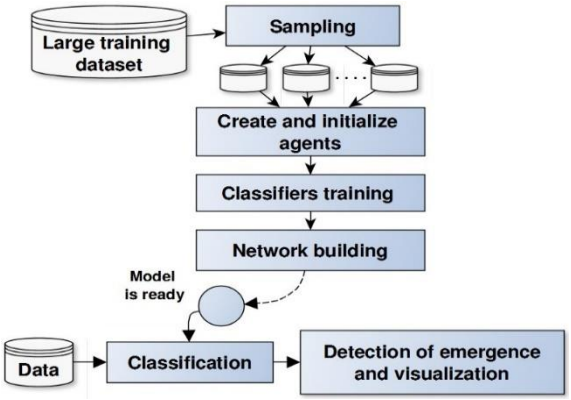


Figure 2. System building.

3.3.2 Execution

After the system is created, the data point to classify is submitted to all agents in the system. In the First step, each agent performs the initial classification of the data point using the Bayesian classifier, trained by its own training sub-set. Then, agents initiate a long interaction phase, where in each cycle, an agent carries out the re-classification of the data point by the combination of its result with those of its neighboring agents. This is repeated by all the agents of the system until a convergence of the system

dynamic occurs. This convergence is indicated by the fact that any agent maintains a steady state, expressed by the class that it holds and becomes constant over time.

Figure 3 shows an example of an agent surrounded by its neighbors. The central agent *A* retrieves the classification results of a given data point *X* from the neighboring agents, respectively *B*, *C*, *D*, *E* and *F*. The calculated combination is kept within the agent A. The neighboring agents *B* . . . *F*, when they perform their cycles, retrieve in their turn the classification result of the agent *A*.

3.4 Collaborative Classification

The classification according to the presented scheme can be considered as consensual classification. Each agent is under the influence of two trends:

- Its own classification (initial one), obtained from its local classifier and
- Its neighborhood, where the classification results are provided from other different classifiers. Therefore, over time, some dominance in the system —in term of classification results— tends to spread to the whole system.

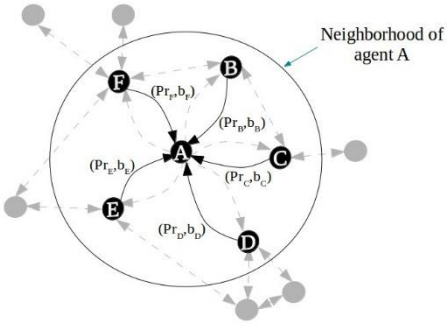


Figure 3. Local interaction of an agent.

The complex dynamic environment that is created and maintained over time in the system tends to converge to a stable state. This state represents a consensus of classification within the population of agents. The ideal case is that all agents are stabilizing on the same result. An alternative case is to have clusters of agents that led each one to the same classification result. Each cluster is naturally contiguous due to the local interaction of agents within their respective neighbors. Otherwise, the system remains in an unstable state, on the edge of chaos, where no even partial consensual results of classification can be observed.

The classification thus made by consensus of classifiers, observed by the emergence of stable structures within the system, represents a compromise decision between the different agents. Indeed, it is certain that if a given data point that can be conveniently classified by some classifier does not necessarily get the same result with other classifiers, since they have different training sub-sets. The aim of our approach is to achieve the best compromise between the different agents of the system, so that agents which have contributed significantly through their classification quality dominate the final decision. In fact, there is a steady state in the space of states characterized by a low dynamic environment exhibiting the consensual classification. This is considered as the objective function to be optimized.

3.5 Convergence of the Complex Dynamic and Emergence of Consensus

The complex dynamic environment, in the sense of our approach, consists of the degree of variation of

the states (calculated classes) within the population of agents over time. Initially, the dynamic should be high, since the state of each agent is the result of its own decision, considering its own knowledge (obtained from its local initial classification model). Then, some agents begin to change their states under the effect of the decision combination rule. If there is an optimum of the dynamic, expressed by a classification consensus, this dynamic tends to be reduced and stabilizes over time.

To detect the convergence of the dynamic interaction of agents, we observe two phenomena:

1) It is considered that there is convergence if all agents (or the majority of them) keep their states unchanged for several execution cycles. When the dynamic environment stabilizes, the system stops and classification result can be retrieved.

2) By using some global entropy quantification to measure the degree of variation of the overall system state. This global state is the superposition of all states of the agents within the system. In the literature, several studies have used entropy to measure activity in dynamic systems. Indeed, when the system becomes homogeneous, its entropy tends to be minimal [28]. The homogeneity corresponding to the absence of any structure within the system is called in this case isotropic. In digital systems, such as ours, we specifically use the Shannon entropy [29]. If the calculated entropy is low enough, we assume that there is a global consensus within the population of agents and the authenticity of the obtained results, depends on the value of the entropy.

In complex systems, the dynamic environment is something that is observed by an external observer. The convergence of the dynamic that can lead to the emergence of certain stable structures within the system can be detected only by this observer. In our case, we must be able to perform a number of steps to determine whether there is convergence or divergence in the sense of emergent stable structures within the population of agents.

To do this, we arbitrarily choose a small number of agents in the system and follow the evolution of their states with the aim of detecting convergence of the dynamic within this sub-set of agents. This idea is inspired by the Monte-Carlo method [30], which involves estimating properties of a system by measuring properties related to a sub-set of its elements drawn randomly. In our case, each agent in the considered subset is observed in order to see whether its state becomes invariant over time. This state corresponds to a class that the agent has by calculating the maximum *a posteriori* probability according to Bayes law. The system continues evolving until the states of the various agents in the observed sample become invariant. In this case, the dynamic environment is considered to have stabilized, then we proceed to the entropy measure to decide whether there is consensus within the population of agents or not.

3.5.1 Sample Selection and Test of Convergence

We consider 10 % of the population of agents as a sample for measuring the convergence of the dynamic. This number is assumed sufficient to estimate the overall state of the system. Let E be the set of agents representing the observed sample. To test the stability of the state of each agent A within the sample, we need to perform certain calculations. For each agent A, an attribute, called Invariant, is initialized to False, given that all agents are not invariant at the beginning of the dynamic.

Within an agent, at each execution cycle, the pseudo-code in Algorithm 4 is executed; so, after several cycles, the value of Invariant is set to True when the corresponding agent executes several cycles (CyclesThreshold).

Algorithm 4 Test of agent's state invariance

```
if A.NewState=A.CurrentState then
    Inc A.CyclesCount
else
    A.CyclesCount ← 0
end if
if A.CyclesCount>CyclesThreshold then
    A.Invariant← True
end if
A.NewState=A.CurrentState
```

CyclesThreshold, which represents the number of cycles to consider before deciding whether the dynamic stability is reached or not, is a system parameter defined experimentally using a testing dataset.

After a certain amount of time to allow the system to move towards a convergence state, tests are performed periodically by an observer entity of the system. This involves testing whether all the states' agents in the considered sample are invariant or not (all set to *True*). The observer pseudo-code to test the overall convergence is introduced in algorithm 5. If the time allowed for the system to stabilize; i.e., *MaxTime*, has elapsed without convergence, we can deduce that the system remains in permanent instability and a convergence state is not possible for the data point to be classified. The maximum period of calculation will also be set experimentally.

3.5.2 Entropy Calculation

Equation 4 expresses the entropy, corresponding to a system with M states, where P_i represents the probability that the system is in the state i. We use the value of entropy to identify the presence or absence of structures in the population of agents. Typically, the presence of clusters of homogeneous agents (leading to the same class), corresponds to a low entropy. In contrast, the divergence of all the agents in terms of classes corresponds to high values of entropy. Ideally, if all agents have concluded to the same class, we will have the lowest value of entropy.

Algorithm 5 System overall convergence

```
Convergence ← False
while ComputingTime < MaxTime and Convergence = False do
    if ∀i, E(i).Invariant = Ture then
        Convergence ← True;
    end if
    wait(Delay)
    ComputingTime ← ComputingTime + Delay
end while
if Convergence = True then
    Computing the entropy
else
    Non stability ⇒ Consensus failed
end if
```

$$H = -\sum_{i} P_i \,.\, Log P_i \tag{4}$$

Let A be a given agent, with NV_A neighbors. The entropy in the neighborhood of this agent is expressed as follows:

$$H_A = -\sum_{i=1}^{C} P_A^l \,.\, Log P_A^l \tag{5}$$

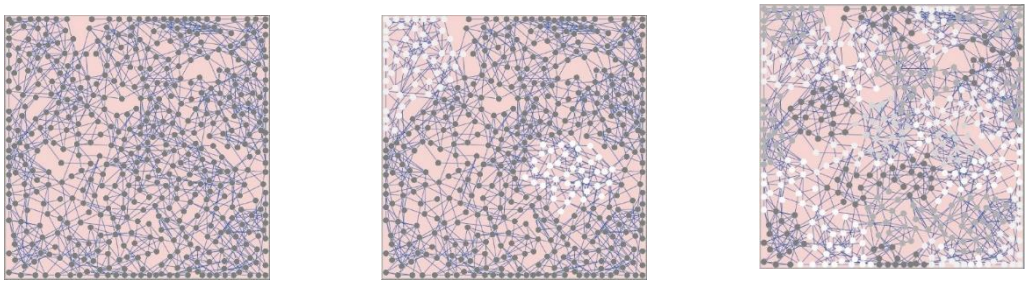
where C is the set of classes and P_A^l the probability that the agent A concludes to the class l. We estimate P_A^l by calculating the relative frequency of class l in the neighborhood of the agent A.

$$P_A^l = \frac{Number\ of\ agents\ concluded\ in\ class\ l}{NV_A + 1} \tag{6}$$

H_A is null if all agents have the same state, $\exists P_A^c = 1, \forall k, k \neq c, P_A^l = 0$; in this case:
$-\sum P_A^l \, log P_A^l = -P_A^c \, log P_A^c = 0$, Which corresponds to the minimal value of H_A.

The entropy at an agent is high if there is a divergence in the different classes concluded in its neighborhood. It is low in the opposite case and null if all the agents in the neighborhood conclude to the same class.

H_A is high if $P_A^l, l = 1...C$ are uniformly distributed.



(a) (b) (c)

Figure 4. Clusters of agents according to H: (a) The ideal case where H is minimal: presence of one cluster (representing the emergent consensual decision); (b) H is low: presence of clusters, but we can retrieve a dominant class from the biggest cluster; (c) H is high: no emergent structure.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 02, June 2022.

Global Entropy

Considering the entropy in each agent of the system, the overall entropy of the system is the sum of the entropies of the different agents: $H = \sum_A H_A$

The use of the thresholds for the global entropy H allows determining in which state the classification system has been stabilized. Figures 4 (a), (b) and (c) illustrate some examples of convergence with different system entropies. Various colors represent the difference in terms of classes concluded by each respective agent, corresponding to the data point to be classified.

4. EXPERIMENTATION

We have performed experimentation to demonstrate the significance of the proposed classification approach. In the following experimental analysis, we will show that results are obtained according to emergent consensus within the population of agents and that leveraging emergence in such complex system results in improvement of the classification results.

First, we consider various training datasets. From each training set, we will extract N non-disjoint training sub-sets and link each one to an agent within the multi-agent system. Such a way of selecting training data ensuring the desired diversity is a key aspect for Big Data classification. To do this, we consider several agents selected randomly within their population of agents. Each selected agent with its neighboring agents are considered as a classic parallel multi-classifier. We first calculate the accuracy of each classifier separately; then, we determine accuracy of results from the combination according to a weighted majority vote of each agent with its neighbors. Finally, we compare the obtained accuracy with that obtained by detection of emergent classes.

At the current state of our work, we have proceeded for experimenting our approach by an endogenous comparison, where results obtained according to the proposed method are compared to those obtained with conventional classifiers. We used the Naive Bayesian classifier, because it provides the quality of labeling (as the posterior probability) of a given data item. This is a key attribute necessary to make interacting agents and adjusting their decisions.

In a first experiment, we put 500 agents in interaction to classify a test set of 10,000 instances from the well-known KDDCup99 dataset on intrusion detection, which contains about 5 million instances, described by 43 attributes [31]. The comparison of: First, the five best average accuracy rates obtained by classifiers, taken separately; second, the five best sets of classifiers combined with their direct neighbors; and third, the accuracy rate achieved by the system of emergent classes, is as follows:

- 84.11% of accuracy rate on average for the 5 best classifiers.
- 84.88% of accuracy rate on average for local sets of classifiers (neighborhood).
- 86.85% achieved by the overall system (with emergence).

This result confirms our hypothesis that with a complex system with "relevant" interactions at the local level, the labels that emerge at the global level within this system are overally better than all the results of individual classifiers or local sets of classifiers. On the other hand, the weak point of the complex system-based method is mainly related to the computation time.

So, in terms of execution time, the classification time for the entire test sub-set is 7920 seconds, which is almost 4.5 times longer than classifying with a three-neighborhood level (considering neighbors of neighbors) and almost 8 times longer than using one level of neighborhood. The deployment of the experience to a real Cloud architecture could reduce the time of classification, but this does not seem fully compatible with big data labeling, because:

- The execution time is longer,
- The entire infrastructure is used to classify each data point.

Additionally, we should note that significantly better accuracy rates can be found in other work dealing with KDDCup99. This can be mainly the consequence of the choice of the type of classifier (Naive Bayes) which has relatively low individual performances on the considered dataset. Nevertheless, it is the most suitable algorithm for our approach, because it provides some parameters necessary for agent interaction. However, the goal of our experiments was to confirm the improvement brought by the approach based on a set of classifiers compared to classifiers taken separately and the improvement

brought by our approach based on the emergence within the global system, compared to that based on small sets of classifiers.

Table 1. Accuracy rates (Agents, Neighborhoods, Global) - Connect-4.

	Top three accuracy rates (%)	
	Agents	**Neighborhoods**
#1	55	75,4
#2	53,6	73,9
#3	52,1	72,5
Average	53,5	73,9
Global	**78,3**	

Table 2. Accuracy rates (Agents, Neighborhoods, Global) – Covertype.

	Top three accuracy rates (%)	
	Agents	**Neighborhoods**
#1	63,1	75,9
#2	62,8	75,3
#3	62,1	75,1
Average	62,8	75,4
Global	**76,7**	

In order to consolidate the results achieved with the KDDCup99 dataset, we have used less known other datasets: Connect−4 [32] and Covertype [33]. The following is a brief description of each one:

- *Connect−4* is a game, where each of the two players tries to line up four of their pieces in a grid of 6 rows and 7 columns. The strategy of the game is to try to line up four of one's own pieces, but at the same time prevent the opposing player from having a row of four of his/her pieces. Each space in the game grid represents an attribute of the dataset and each instance in the dataset represents the legal positions corresponding to the first 8 moves of a game for which no player has yet won. The class represents the result of the game *vis-à-vis* the first player (won, lost or drawn); this result will be obtained if both players continue to play "perfectly". The dataset contains 67557 instances (games), 42 attributes and 3 classes.
- *Covertype* dataset is composed of geospatial data created by the USFS Forest Inventory and Analysis (FIA) and the Remote Sensing Applications Center (RSAC) to show the extent, distribution and composition of forest cover in the United States. It contains 581,012 instances, 54 attributes and 7 classes. Each instance contains data for a 30 x 30 meter cell and the class represents the type of forest cover.

Tables 1 and 2 represent results recorded for the *Connect−4* and *Covertype* datasets, respectively. We have noted the three best accuracy rates considering the results obtained at the agent level and at the neighborhood level. The overall accuracy rate achieved by all the agents with a consensus on the classification is also shown.

The obtained results show the improvement in accuracy rates from the agent and its neighborhood to the overall system result (Figure 5). The improvement recorded by agent sets in the neighborhood is explained by the principle of the classical combination of classifiers (on average 73,9% for *Connect–4* and 75,4% for *Covertype*). We also note the improvement of accuracy rate at the level of the overall system (78,3% for *Connect-4* and 76,7% for *Covertype*), achieved by the combination of results within the global population of agents having reached consensus for labeling (emerging classification).

These results confirm our hypothesis about the potential of the interaction of agents (classifiers) in a massive system for improving data classification accuracy. However, certainly there is still some more work to be done in studying the dynamic environment within such systems so as to understand how the parameters inherent to these systems influence the quality of the results.

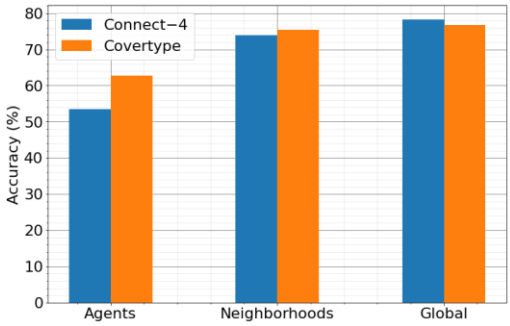


Figure 5. Accuracy rates improvement for *Connect-4* and *Covertype*.

5. CONCLUSION

In this work, we proposed a consensual approach for data classification. The approach is intended to be adapted to complex data. Such data which can be of very high volume, of high dimensionality and possibly distributed can be handled efficiently using our approach with emphasis on both the quality of classification and the scalability of the solution. We used the paradigm of complex systems modeled as multi-agent systems to design a massive multi-classifier system. We have qualified it as massive, because it is assumed to contain a very large number of elementary classifiers put in interaction. The interaction generates phenomena relating to complex systems, such as the emergence of structure at the macro-level of the system. The emergent structure we desire is the consensual decision that the population of a majority of agents may have. The timeline is arrived at when agents that belong to local neighborhoods communicate their classification results. Then, over time, a complex dynamic environment sets within the system, which tends to converge on an emerging structure that represents a classification consensus within the population of agents. Finally, we retrieve this consensual decision as the label of the treated data point. Our approach is not suitable for a system that needs real-time processing. This is because in such a system, the time needed for the complex dynamic environment to stabilize so that the phenomenon of emergence appears may be too long. This is not desirable for a real-time processing system.

In the future perspective of this work, it is essential to push the experiments further in order to improve the proposed approach. This will enable better understanding of how to refine the parameters of the system and control the created dynamic environment in order to achieve better classification results.

REFERENCES

[1] S. Umadevi and K. S. J. Marseline, "A Survey on Data Mining Classification Algorithms," Proc. of the IEEE Int. Conf. on Signal Process. Comm. (ICSPC), pp. 264-268, Coimbatore, India, 2017.

[2] S. Minaee, N. Kalchbrenner, E. Cambria et al., "Deep Learning-based Text Classification: A Comprehensive Review," ACM Computing Surveys, vol. 54, pp. 1-40, 2021.

[3] A. H. Mohd Aman, E. Yadegaridehkordi, Z. S. Attarbashi, R. Hassan and Y. Park, "A Survey on Trend and Classification of Internet of Things Reviews," IEEE Access, vol. 8, pp. 111763- 111782, 2020.

[4] L. F. Cordeiro Robson, C. Faloutsos and C. Traina Jnior, Data Mining in Large Sets of Complex Data, ISBN-13: 978-1447148890, Springer Publishing Company, Incorporated, 2013.

[5] L. I. Kuncheva, "Combining Pattern Classifiers: Methods and Algorithms", Wiley-Interscience, 2004.

[6] X. Dong, Z. Yu, W. Cao, Y. Shi and Q. Ma, "A Survey on Ensemble Learning," Frontiers of Computer Science, vol. 14, no. 2, pp. 241-258, 2020.

[7] O. Sagi and L. Rokach, "Ensemble Learning: A Survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8. No. 4, DOI: 10.1002/widm.1249, 2018.

[8] F. Yucalar, A. Ozcift, E. Borandağ and D. Kilinç, "Multiple-classifiers in Software Quality Engineering: Combining Predictors to Improve Software Fault Prediction Ability," Engineering Science and Technology: An International Journal, vol. 23, no. 4, pp. 938-950, 2020.

[9] S. Tulyakov, S. Jaeger, V. Govindaraju and D. Doermann, "Review of Classifier Combination Methods," Proc. of Machine Learning in Document Analysis and Recognition, Part of the Studies in Computational Intelligence Book Series, vol. 90, pp. 361-386, 2008.

[10] L. I. Kuncheva and C. J. Whitaker, "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy," Machine Learning, vol. 51, no. 2, pp.181-207, 2003.

[11] D. Gacquer, On the Usability of Diversity When Training Multiple Classifier Systems. Application to the Detection of Hazardous Smokes on Industrial Facilities, Thesis at University of Valenciennes and the Hainaut-Cambresis, France, 2008.

[12] R. Roelofs et al., "A Meta-analysis of Overfitting in Machine Learning," NeurIPS Proceedings, [Online], Available: https://papers.nips.cc/paper/9117-a-meta-analysis-of-overfitting-in-machine-learning, 2019.

[13] D. Che, M. S. Safran and Z. Peng, "From Big Data to Big Data Mining: Challenges, Issues and Opportunities," Proc. of the 18th Int. Conf. DASFAA, vol. 7827, pp. 1-15, Springer, 2013.

[14] M. A. Niazi, "Introduction to the Modeling and Analysis of Complex Systems: A Review," Complex Adaptive Systems Modeling, vol. 4, Article no. 3, DOI: 10.1186/s40294-016-0015-x, 2016.

[15] A. Dogan and D. Birant, "A Weighted Majority Voting Ensemble Approach for Classification," Proc. of the 4th IEEE Int. Conf. on Computer Science and Eng. (UBMK), pp. 1-6, Samsun, Turkey, 2019.

[16] R. Olfati-Saber, J. A. Fax and R. M. Murray, "Consensus and Cooperation in Networked Multi-agent Systems," Proceedings of the IEEE, vol. 95, no. 1, pp. 215-233, 2007.

[17] Y. Chen, J. Liu, H. Zhao and H. Su, "Social Structure Emergence: A Multi-agent Reinforcement

Learning Framework for Relationship Building," Proc. of the 19th International Conference on Autonomous Agents and Multi Agent Systems (AAMAS '20), pp. 1807-1809, 2020.
[18] C. Yang and L. Jiamou, "Social Capital Games As a Framework for Social Structural Pattern Emergence," Proc. of the 12th IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 309-316, The Hague, Netherlands, 2020.
[19] O. Paunovski, G. Eleftherakis and T. Cowling, "Disciplined Exploration of Emergence Using Multi-agent Simulation Framework," Computing and Informatics, vol. 28, pp. 369-391, 2009.
[20] O. Paunovski, G. Eleftherakis and A. J. Cowling, "A Fuzzy Reasoning Approach to Automated Detection of Emergent Herd Formations in Computer Aided Simulation," Proc. of the 3rd South East European Doctoral Student Conference, vol. 2, pp. 150-165, Thessaloniki, Greece, 2008.
[21] V. Robu, H. Halpin and H. Shepherd, "Emergence of Consensus and Shared Vocabularies in Collaborative Tagging Systems," ACM Transactions on the Web, vol. 3, no. 4, pp. 1-14, 2009.
[22] X. Wang, D.-W. Bi, L. Ding and S. Wang, "Agent Collaborative Target Localization and Classification in Wireless Sensor Networks," Sensors, vol. 8, pp. 1359-1386, 2007.
[23] R. Mazouzi, L. Seddiki, C. de Runz and H. Akdag, "Towards a Collective and Distributed System for Consensual Data Classification," HAL-EGC 2014, hal-03251347, version 1, 2014.
[24] L. Maystre, N. Kumarappan, J. Bütepage and M. Lalmas, "Collaborative Classification from Noisy Labels," Proc. of the 24th International Conference on Artificial Intelligence and Statistics, Proc. of Machine Learning Research, vol. 130, pp. 1639-1647, 2021.
[25] W. Sun, Z. Wang, G. Zhang and B. Dong, "MACCA: A SDN Based Collaborative Classification Algorithm for QoS Guaranteed Transmission on IoT," Proc. of the 15th International Conference on Advanced Data Mining and Applications (ADMA 2019), pp. 815-827, Springer, 2019.
[26] C. D. Stefano, A. D. Cioppa and A. Marcelli, "An Adaptive Weighted Majority Vote Rule for Combining Multiple Classifiers," Proc. of the IEEE International Conference on Pattern Recognition (ICPR), pp. 192-195, Quebec City, Canada, 2002.
[27] K. P. Murphy, "Naive Bayes Classifiers," University of British Columbia, vol. 18, no. 60, pp. 1-8, 2006.
[28] R. Hickey, "A Note on the Measurement of Randomness," Journal of Applied Probability, vol. 19, no. 1, pp. 229-232, 1982.
[29] J. Lin, "Divergence Measures Based on the Shannon Entropy," IEEE Transactions on Information Theory, vol. 37, no. 1, pp. 145-151, 1991.
[30] J. E. Hurtado and A. H. Barbat, "Monte Carlo Techniques in Computational Stochastic Mechanics," Archives of Computational Methods in Engineering, vol. 5, no. 1, pp. 3-29, 1998.
[31] M. Tavallaee. E. Bagheri, W. Lu and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," Proc. of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2009), pp. 1-6, Ottawa, Canada, 2009.
[32] M. Lichman, "Connect-4", UCI Machine Learning Repository, 2013.
[33] J. A. Blackard, "Comparison of Neural Networks and Discriminant Analysis in Predicting Forest Cover Types," PhD Thesis, Computer Science Dept. Fort Collins, Colorado State University, USA, 1998.

ملخص البحث:

يتضمّن العمل الذي نعرضه في هذه الورقة التعامل مع الانبثاق في المصنِّفات المتعدّدة الضّخمة التي نعدُّها نُظماً معقّدة. فنحن نهدف الى بناء نظامٍ تعاوني لتصنيف البيانات على نحوٍ مُراقَب نتوقّع منهُ أن يقدّم أداءً أفضل مقارنةً بالمصنِّفات التّقليدية. وبنمذجة المصنِّف المتعدّد الضّخم كنظام مُتعدّد العوامل، فإنّه يتكون من عددٍ كبيرٍ من العوامل المترابطة فيما بينها وفْق جِوارٍ معيّن؛ إذ يلعب كلّ عاملٍ دوْر مصنِّفٍ ضعيف. وعلى المستوى الميكروي، فإنّ التّفاعل الابتدائي بين العوامل يتمثّل في جمْع نتائج تصنيف كلٍّ منها. ويقوم كلّ عاملٍ وفقاً لقاعدة تصويت الأغلبية بجمع نتائجه مع نتائج جِوارِهِ من العوامل، آخذاً أداء كلٍّ منها بعيْن الاعتبار. وتتكرّر هذه العملية بشكلٍ متواصِل وبصورة دوريةٍ داخل جِوار كلّ عامل. وبذلك تنشأ ديناميّة معقّدة داخل النّظام. وبعد زمنٍ معيّن، تستقر هذه الديناميّة المعقّدة، سامحةً بعرْض بنْيةٍ انبثاقية ستتمّ ملاحظتها على المستوى الماكروي وتُعدّ توقُّعاً رِضائياً للأصناف بالنسبة للبيانات التي نرغبُ في تصنيفها. والجدير بالذكر أنّ النتائج التّجريبية التي حصلنا عليها والمقارنة مع نتائج المصنِّفات التّقليدية تُظهر إمكانيّة الطّريقة المقترحة في تحسين التّصنيف وأن تكون بديلاً لجمْع المصنِّفات ومراكمتها.