HYBRID NEURAL INTELLIGENT SYSTEM TO PREDICT BUSINESS FAILURE IN SMALL-TO-MEDIUM-SIZE ENTERPRISES

M. LOURDES BORRAJO
Department of Informática, University of Vigo
Edificio Politécnico, Campus Universitario As Lagoas s/n
Ourense, 32004, Spain
lborrajo@uvigo.es

BRUNO BARUQUE
Department of de Ingeniería Civil, University of Burgos
Esc. Politécnica Superior, Edificio C
C/Francisco de Vitoria, 09006, Burgos, Spain
bbaruque@ubu.es

EMILIO CORCHADO*, JAVIER BAJO† and JUAN M. CORCHADO‡
Departamento de Informática y Automática
University of Salamanca, Plaza de la Merced s/n
37008, Salamanca, Spain
*escorchado@usal.es
†jbajo@usal.es
‡corchado@usal.es

During the last years there has been a growing need of developing innovative tools that can help small to medium sized enterprises to predict business failure as well as financial crisis. In this study we present a novel hybrid intelligent system aimed at monitoring the *modus operandi* of the companies and predicting possible failures. This system is implemented by means of a neural-based multi-agent system that models the different actors of the companies as agents. The core of the multi-agent system is a type of agent that incorporates a case-based reasoning system and automates the business control process and failure prediction. The stages of the case-based reasoning system are implemented by means of web services: the retrieval stage uses an innovative weighted voting summarization of self-organizing maps ensembles-based method and the reuse stage is implemented by means of a radial basis function neural network. An initial prototype was developed and the results obtained related to small and medium enterprises in a real scenario are presented.

Keywords: Hybrid neural intelligent system; CBR; MAS; Business failure.

1. Introduction

During the last years there has been a growing need of developing innovative tools that can help SMEs (small to medium sized enterprises) to predict business failure as well as financial crisis.¹⁻⁵ Yet, SMEs require a business control mechanism in order to monitor their *modus operandi* and analyze whether they are achieving their goals.⁵⁻⁸ Such mechanisms are constructed around a series of organizational policies and specific procedures referred to as “controls”, and conform to the structure of the business control of the company. As a consequence of this, the need has arisen for periodic internal audits. However, evaluating and predicting the evolution of these
Types of business entities, which are characterized by their great dynamism, tend to be a complicated process. It is necessary to construct models that facilitate the analysis of the work carried out in changing environments such as finance.

The processes carried out inside a company are grouped into functional areas denominated “Functions”. A Function is a group of coordinated and related activities that are systematically and iteratively carried out during the process of reaching the company’s objectives. The functions that are usually carried out within a company, as studied within the framework of this research, are: Purchases, Cash Management, Sales, Information Technology, Fixed Assets Management. Compliance to Legal Norms and Human Resources. Each one of these functions is broken down into a series of activities. For example, the Information Technology function is divided into the following activities: Computer Plan Development, Study of Systems, Installation of Systems, Treatment of Information Flows, and Security Management.

Agents and multi-agent systems (MAS) have become increasingly relevant for developing distributed and dynamic intelligent environments. Agents are computational entities that can be characterized through their capacities in areas such as autonomy, reactivity, proactivity, social abilities, reasoning, learning and mobility. These capacities make the multi-agent systems very appropriate for constructing intelligent environments. An agent can act as an interface between the user and the rest of the elements of the intelligent environment. This paper presents an innovative solution to model distributed adaptive systems in business environments. It is based on a multi-agent architecture that can integrate Web services, and incorporates a novel planning mechanism that makes it possible to determine potential risky situations based on existing activities and previous results. The core of the multi-agent system is a hybrid intelligent mechanism aimed at detecting and predicting business failures. There are various artificial intelligence techniques such as artificial neural networks, Bayesian networks, and fuzzy logic which have been applied to business failure prediction. While these techniques can be applied to failure detection and prediction, the knowledge obtained cannot be incorporated into successive tests and included in subsequent analyses. This paper presents a hybrid neural intelligent system based on CBR (Case-Based Reasoning) which uses past experiences to solve new problems. CBR facilitates the creation of strategies similar to the processes followed in SMEs. The recovery of information from previous experiences simplifies the prediction process by detecting and eliminating relevant and irrelevant patterns detected in previous analyses. The retrieve phase of the hybrid neural intelligent system incorporates a novel version of the SOM (Self-Organizing Map) based on the application of ensembles techniques and called WeVoS-SOM. The reuse stage incorporates a radial basis function neural network. The revise and retain stages implement a decision support system for experts.

In Ref. 53 we presented a system composed of two case-based reasoning systems to detect the associate risk in the activities of SMEs in the textile sector and generate recommendations to improve the erroneous processes. In Ref. 54 we presented a decision support tool based on a case-based reasoning system that automates the internal control processes of a SME. The approach presented in this article is an evolution of our previous works that proposes an innovative perspective where a multi-agent system that integrates within web services is used to model the organizational structure of a SME. This perspective allows taking into account the advantages of virtual organizations to model the SME functions. Moreover, the core of the system is a CBR-BDI agent with the ability to distribute its internal functionalities in web services which notably improves the efficiency in the distribution of tasks. The new approach proposes a new method for the retrieval stage of the CBR system, the WeVoS-SOM, that notably improves the case’s recovery reducing the final quantity of cases stored and making it easier to recover the most similar cases to the problem introduced.

The article is structured as follows: the next section briefly introduces the problem that motivates this research and reviews the state of the art. Section 3 presents the multi-agent system for failure prediction in small and medium enterprises. Section 4 describes the novel strategies incorporated in the stages of the CBR cycle of the CBR-BDI evaluator agent. Section 5 describes a case study
Hybrid Neural Intelligent System to Predict Business Failure

2. The Crisis Prediction Problem

Many things in our business and auditing environment are changing at an increasing rate. Increased competition and the need for faster and better information for decisions mark today’s business environment. In addition, systems are complex and many times provide on-line access facilities. This complexity means that auditors have more and different kinds of work to do than they had earlier. Specifically developed to evaluate the hybrid neural intelligent system presented within this study.

Finally, Sec. 6 presents the results obtained after testing the model and Sec. 7 present the conclusions of this study.

3. Multi-Agent System for Preventing Crisis Failure in Medium-Sized Enterprises

Agent and multi-agent systems combine classical and modern functional architecture aspects. The integration and interoperability of agents and multi-agent systems with SOA and Web Services approaches has been recently explored. Bonino da Silva et al. in Ref. 34 propose merging multi-agent...
techniques with semantic web services to enable dynamic, context-aware service composition. Ricci et al. in Ref 35 have developed a java-based framework to create SOA and Web Services compliant applications, which are modeled as agents. Liu in Ref 36 proposes a multi-agent architecture to develop inter-enterprise cooperation systems using SOA and Web Services components and communication protocols. Our proposal not only provides communication and integration between distributed agents, services, and applications; it also proposes a new method to facilitate the development of distributed multi-agent systems by means of modeling the functionalities of the agents and the systems as services. More concretely, the paper presents a multi-agent enterprise control system aimed at developing a generic model useful in any type of SME. The enterprises that participated in the experiment were very interested in a tool such as the one developed within the framework of this investigation, where the constructed multi-agent system is able to determine the state of each of the activities and calculate the associated risk. It also detects any inefficient processes and predicts crisis situations.

Figure 1 shows the basic schema of the proposed architecture, where all requests and responses are handled by the agents in the platform. The system is modeled as a modular multi-agent architecture, where services and applications are managed and controlled by deliberative BDI (Belief, Desire, Intention) agents.

There are different kinds of agents in the architecture, each one with specific roles, capabilities and characteristics:

— Business Agent. This agent was assigned for each SME in order to collect new data and allow consultations. The SME can interact with the system by means of this agent, introducing information and receiving predictions.

— Evaluator Agent. It is responsible for the evaluation and predictions of potential risky situations. Every time that it is necessary to obtain a new estimate of the state of an activity, the agent evolves through several phases. On the one hand, this evolution allows the multi-agent system, to identify the latest situations most similar to the current situation in the retrieval stage, and to adapt the current knowledge in the reuse stage in order to generate an initial estimate of the state of the activity being analysed. The activity selected will then serve as a guide for establishing a risk level for the activity, its function, and
the company itself, to develop in a more positive way. The retention phase guarantees that the system evolves in parallel with the firm, basing the corrective actions on the calculation of the error previously made.

--- Expert Agent. This agent helps the auditors and enterprise control experts that collaborate in the project to provide information and feedback to the multi-agent system. These experts generate prototypical cases from their experience and they receive assistance in developing the Store agent case-base.

--- Store Agent. This agent has a memory that has been fed with cases constructed with information provided by the enterprise (through its agent) and with prototypical cases identified by 34 enterprises control experts, using personal agents who have collaborated and supervised the developed model.

The Evaluator agent is a CBR-BDI agent with advanced reasoning abilities that provided great adaptation and learning capacities. This agent type is the core of the multi-agent system and is presented in detail in Sec. 4.

4. CBR-BDI Agents for Crisis Classification and Prediction

The application of agents and multi-agent systems facilitates taking advantage of the inherent capabilities of the agents. Nevertheless, it is possible to increase the reasoning and learning capabilities by incorporating a case-based reasoning (CBR) mechanism into the agents. In the case at hand, a CBR classifier agent (CBR-BDI) is responsible for classifying the enterprise situation and predict possible crisis. In the BDI (Beliefs Desires Intentions) model, the internal structure of an agent and its capacity to choose is based on mental aptitudes: agent behaviour is composed of beliefs, desires, and intentions. A BDI architecture has the advantage of being intuitive and capable of rather simply identifying the process of decision-making and how to perform it.

Case-based Reasoning is a type of reasoning based on the use of past experiences. The fundamental concept when working with case-based reasoning is the concept of case. A case can be defined as a past experience, and is composed of three elements: A problem description which describes the initial problem, a solution which provides the sequence of actions carried out in order to solve the problem, and the final state which describes the state achieved once the solution was applied. The way in which cases are managed is known as the case-based reasoning cycle.

These systems manage cases (past experiences) to solve new problems. The way in which cases are managed is known as the CBR cycle. The CBR cycle shown in Fig. 2 consists of four sequential phases: retrieve, reuse, revise and retain. The retrieve phase starts when a new problem description is received. Similarity algorithms are applied in order to retrieve from the cases memory the cases with a problem description more similar to the current one. Once the most similar cases have been retrieved, the reuse phase begins. In this phase the solutions of the cases retrieved are adapted to obtain the best solution for the current case. The revise phase consists of an expert revision of the solution proposed. Finally, the retain phase allows the system to learn from the experiences obtained in the three previous phases and updates the memory case in consequence.

![Fig. 2. Diagram including a CBR-BDI agent reasoning cycle.](image)
The following sub-sections present the internal structure of the CBR-BDI Evaluator agents used to predict and prevent crisis in medium-sized enterprises. This agent cooperates with the Expert Agent to provide a decision support tool for the internal auditor of the SME, who is the main user of the proposed model. As can be seen in Fig. 3, the retrieval stage consists of a novel version of the SOM\textsuperscript{13,42} based on the application of ensembles techniques and called WeVoS-SOM.\textsuperscript{14} The reuse stage incorporates a radial basis function neural network.\textsuperscript{87–95} The revise and retain stages implement a decision support system for experts.

4.1. Retrieval stage

This stage is aimed at retrieving the most similar cases to the current problem. To do so, the approach used is to organize the case base in a way that cases with similar characteristics are stored “close” to each other, by means of topology preservation models. The learning process used in this kind of artificial neural networks make only the most similar unit of the map (called Best Matching Unit or BMU) react when a new data entry is presented. By keeping track of which units react to which cases, it is easy to keep also record of the similarity between cases, as it is determined by their proximity on the map obtained. That way, the algorithm can not only recover similar cases, but also order them according to their degree of similarity.\textsuperscript{65,66} This enables also the reduction of the case base, as it is easy to find if the case to be inserted is similar to a high degree to already existent cases and therefore can be discarded.\textsuperscript{59}

The specific technique used is the training of several Self-Organizing Maps using the classical bagging algorithm\textsuperscript{61,62} and a final step fusing all together to obtain a resultant map reflecting the best characteristics of each of its composing maps. The novelty of this approach lies in the algorithm used for the fusion of maps.
The SOM family of algorithms is one of the most common and well known among Prototype Learning Algorithms, included in the family of cluster based learning. The advantage that offers against other of the most commonly used models of that family (such as K-Nearest Neighbours) is its topological ordering. This allows the user to order the recovered cases according to the similarity to a given case, rather than obtaining an unordered set of examples. The authors of the paper have compared the WeVoS-SOM algorithm in other works with similar simpler model such as the SOM and ViSOM. As studies have proved, this algorithm is the best at ordering its composing units, and therefore it improves the results of the model including the simple SOM, regarding the reduction of the size of the case base. This can also be seen in the results section.

The bagging meta-algorithm consists on repeating several overlapping re-samplings over the dataset. That way, it simulates the obtaining of several different datasets from the same source as the original one. Each of the re-sampled datasets is used for the training of a single classifier. That way each is trained with very similar, but not exactly the same characteristics. Doing so, the errors that each of them will commit will not necessarily overlap, having therefore room for general improvement within the ensemble.

The idea behind the novel fusion method presented in this study, Weighted Voting Superposition-SOM (WeVoS-SOM), is to obtain the best position for each neuron and its neighbours, by choosing among several possibilities that were explored by generating different, but similar, composing maps. As a consequence, the final map is expected to be better than its composing maps, but keeping one of the most important features of this type of algorithms: its topological ordering.

WeVoS is an improved version of an algorithm presented in several previous works on superposition. In this study, it is applied to the recovery of cases with a description similar to the current situation of a given enterprise.

The final map is obtained again on a neuron-by-neuron basis. First, the neurons of the final map are initialized by calculating the centroids of the neurons in the same position of the map grid in each of the trained maps. This is calculated as in Eq. (1).

$$w_{\text{init},\text{VG}} = \frac{1}{|W_k|} \sum_{w_i \in W_k} w_i$$  (1)

where $w_i$ are the weights of an individual neuron and $|W_k|$ denotes the number of neurons that are participating in the calculation of the centroid. In this case, it coincides with the number of maps trained in the ensemble.

Then, a recalculation of the final position of that neuron uses the information associated with the neurons in that same position in each map of the ensemble. For each neuron, a sort of voting process is performed, as in Eq. (2):

$$V_{p,m} = \sum_{i=1}^{N} b_{p,m} \cdot \sum_{i=1}^{N} q_{p,m}$$  (2)

where $V_{p,m}$ is the weight of the vote for the neuron included in map $m$ of the ensemble, in position $p$. $N$ is the total number of maps in the ensemble, $b_{p,m}$ is the binary vector used for marking the data set entries recognized by neuron in position $p$ of map $m$, and $q_{p,m}$ is the value of the desired quality measure for a neuron in position $p$ of map $m$.

Associated with each map the algorithm has binary matrix ($b$) with as many rows as neurons in the map and as many columns as number of entries. Initially all values are initialized to 0 and when a neuron is selected as the BMU for an entry, the bit for that neuron and that entry is switched to 1.

The algorithm can make use of any of the quality/error measures in topology preservation literature as long as it can be calculated for each neuron individually, as opposed to a general map measure. In the case of the experiments performed in the paper.

Fig. 4. Adaptation of the units of the final map according to the composing units of different maps of the ensemble.
the Goodness of Adaptation proposed in Ref. 63 and expressed in Eq. (3):

\[ d(x_i) = \|x_i - e_i\| + \min_{k=0}^{p-1} \sum_{j=0}^{k} \|w_{j+1}(k) - w_{j+1}(k+1)\| \] (3)

where \(e_i\) and \(e_i'\) represent the weights of the first BMU and the second BMU respectively, corresponding to data entry \(x_i\). \(I_{j}(k)\) and \(I_{j}(k+1)\) represent indexes of the \(k\)th and the \((k + 1)\)th neurons in the minimum path from \(x_i\) to \(x_i'\) (both neurons being direct neighbours in the map grid). According to that definition, that is, the first neuron in the path is the first BMU for data entry \(x_i\) and, that is, the last neuron in the path, corresponds to the second BMU for data entry \(x_i\).

In summary, the algorithm takes into account both the recognition of units in a certain position, as expressed in Eq. (4); and the quality of adaptation of those units, as expressed in Eq. (5).

\[ B(p) = \sum_{i=1}^{N} \sum_{j=1}^{E} BMU_j(x_i) \] (4)

\[ Q(p) = \sum_{i=1}^{N} q_{i,p} \] (5)

being \(q_{i,p}\), the quality measure calculated for position \(p\) in map \(i\).

The weights of the neurons are fed into the final network as with the data inputs during the training phase of a SOM, considering the “homologous” neuron in the final map as the BMU (Best Matching Unit).

The weight of the vote (\(V_{w,k}\)) is used as the learning rate (\(\alpha\)) for the update of the final neuron of the fused map in the characteristic weight update of the SOM (Eq. (6)):

\[ w_k(t + 1) = w_k(t) + \alpha(t)\eta(v, k, t) \times (x(t) - w_k(t)) \] (6)

where \(x\) is the input to the network, \(w_k\) is the weight vector associated with neuron \(k\), while \(w_k\) is the weight vector associated to the winning unit in the lattice, also called best matching unit (BMU). \(\alpha(t)\) is the learning rate of the algorithm, \(\eta(v, k, t)\) is the neighbourhood function (usually a gaussian function) where \(v\) represents the position of the BMU for the particular \(x\) of time \(t\) and \(k\) the positions of the units in the neighbourhood of this one.

The characteristics vector of the final neuron will be updated towards the characteristics vector of the composing maps correspondent neurons. The difference of the updating performed for each “homologous” neuron in the composing maps depends on the quality measure calculated for each neuron. The higher the quality (or the lower the error) of the neuron of the composing map; the stronger the neuron of the fused map updated towards the weights of that neuron. Either a single or a linear combination of several quality measures can be used to determine the quality of a unit. The number of data inputs recognized by each neuron is also taken into account in this quantization of the “best suitability” of one neuron or another for the same position in the final map. In short, the fusion algorithm will consider “more suitable” weights of a composing neuron to be the weights of the neuron in the final map according to both the number of inputs recognized and the adaptation quality of the neuron. The model, called WeVoS is described in detail in Table 1.

This approach takes into account not only the characteristics of a single neuron, but also the topographic ordering of its neighbourhood. This new approach obtains more maps that are more faithful to the inner structure of the data set from a visualization point of view.

### 4.2. Reuse stage

This phase aims to obtain an initial estimation of the state of the activity analysed. In order to obtain this estimation, RBF networks are used. As in the previous phase, the number of attributes of the problem case depends on the activity analysed. Therefore, it is necessary to establish a RBF network system, one for each of the activities to be analysed.

The \(k\)-cases retrieved in the previous phase are used by the RBF network as a training group that allows it to adapt its configuration to the new problem encountered before generating the initial estimation. The system presented in this article has
Hybrid Neural Intelligent System to Predict Business Failure

Table 1. Algorithm 1. Weighted voting summarization algorithm.

| Input: | Set of trained topology-preserving maps: $M = (M_1 \ldots M_N)$, training data set: $S$ |
| Output: | A final fused map: $M_{\text{fus}}$ |

1. Select a training set $S = ((x_1, y_1) \ldots (x_E, y_E))$
2. train several networks by using the bagging meta-algorithm: $M_1 \ldots M_N$
3. For each map $M_i \in M$
4. calculate the quality/error measure chosen for ALL neurons in the map. Calculate which is the neuron recognizing each entry of the dataset ($x_j$) in the map ($BMU(x_j)$).
5. These two values are used in Eq. (2).
6. calculate recognition rate for each position (Eq. (4)).
7. calculate an accumulated total of the quality/error for each position (Eq. (5)).
8. For each unit position $p$ in $M_i$
9. initialize the fused map ($M_{\text{fus}}$) by calculating the centroid ($w_{c}$) of the neurons of all maps in that position ($p$) as in Eq. (1).
10. For each map $M_i \in M$
11. For each unit position $p$ in $M_i$
12. calculate the vote weight ($V_{p,M_i}$) using Eq. (2).
13. Feed the weights vector of neuron $w_p$ into the fused map ($M_{\text{fus}}$) as if it was an input to the network. The weight of the vote ($V_{p,M_i}$) is used as the learning rate ($\alpha$) for the update of the final neuron of the fused map in the characteristic weight update of the SOM. Eq. (6).
14. The position of that neuron ($p$) is considered as the position of the $BMU (v)$. This causes the neuron of the fused map ($w_{p}^{*}$) to approximate the neuron of the composing ensemble ($w_{p,m}$) according to the quality of its adaptation.
15. end
16. end
17. end

- A RBF network for each of the activities that will be evaluated for a SME. Each of the RBF networks has as inputs as tasks are evaluated for the activity.

The topology of each of the RBF networks used in this task consists of an input layer with as many neurons as attributes possessed by the input vector which constitutes the problem descriptor, a hidden layer with 14 centres, and an output layer with a single neuron corresponding to the variable to be estimated (correction level or state of activity analyzed in percentage). The number of centers (14) was empirically obtained after various tests. The centers are the neurons in the hidden layer of the RBF neural network.

- The RBF network is characterized by its ability to adapt, to learn rapidly, and to generalize. Specifically, within this system the network acts as a mechanism capable of absorbing knowledge about a certain number of cases and generalizing from them. During this process, the RBF network, interpolates and carries out estimations without forgetting part of those already carried out. The Store agent acts as a permanent memory capable of maintaining many cases or experiences by consulting the WeVoS-SOM model of the case base; while the RBF network, in the Evaluator agent, acts as a short term memory, able to recognize recently learnt patterns and to generalize from them.

4.3. Revise stage

- The objective of this phase is to guarantee or refute the initial solution proposed by the RBF network,
obtaining a final solution and working out, apart from this, the control risk.

Faced with the initial estimation generated by the RBF network, the internal auditor will be responsible of deciding whether to accept the aforementioned solution. It will be based on the specific knowledge he possesses about the company in which he is working. If the estimation provided by the system is considered to be valid, the system will take this solution as the final solution and, in the next phase of the CBR cycle, will store a new case in the case base, formed from the problem case and the final solution.

The system will assign this new case with an initial reliability of 100%. If, on the contrary, the internal auditor considers that the solution provided by the system is not valid, they will provide their own solution that the system will take as the final solution. This, along with the problem case, will be stored as a new case in the case memory in the following phase. This new case will be assigned a reliability of 30%. This value has been determined bearing in mind the opinion of various auditors as to the importance that should be assigned to the personal opinion of the internal auditor.

As of the final solution: the state of the activity, the system must work out the control risk associated with the activity. All developed activity in the enterprise sector has an associated risk that will indicate the negative influence that it exerts over the proper running of the enterprise. That is to say, the control risk of an activity measures the impact, on the whole on all of the managerial processes that the current state of the aforementioned activity represents. In this investigation it is considered that the risk level can assume one of these three possible values: low, medium and high. The calculation of the control risk level associated with an activity is based on the current state of the activity and on its level of importance. This last value is obtained after analysing the information obtained from a series of surveys (98 in total) answered by auditors all over Spain. In the previously mentioned surveys the auditors were asked to rank from 1 to 10 the importance that they gave the above-mentioned activity together with the function to which it belongs. The greater the importance is, the greater weighting the activity will have within the internal control system.

Then, as of the level of importance of the activity, and bearing in mind the final solution obtained after the revision phase, the control risk level can be worked out. The calculation of the control risk level is carried out using if-then rules in which the importance the auditors have assigned to activity is compared to the final solution or state of the activity.

4.4. Retain stage

The retain phase is aimed at learning from the new past experience. As stated in the previous subsection, a case together with its corresponding solution is stored in the memory of cases if it is considered as reliable, that is, a confidence level of 100%. Moreover, the system also stores those past experiences that are considered as not reliable by the expert. These past experiences are considered of interest by the CBR because they can be used as an indicator for invalid solutions, although this learning ability is not taken into consideration in this study. In this phase, the risk level associated with the case is also stored in the memory of cases, that is, the if-then rules that contain the know-how of the auditors.

5. Case Study: Castilla and León

Business Scenario

The case study was aimed to detect and predict possible failures in SMEs in Castilla y León. The experiment consisted on the construction of the initial prototype of memory of cases and then in predicting potential risky situations for the enterprises taken into considerations.

The data used to construct the model were obtained by surveys conducted with enterprise experts in the different functional areas of various enterprises, using the Expert agents. This type of survey attempts to reflect the experience of the experts in their different fields. For each activity, the survey presents two possible situations: the first one tries to reflect the situation of an activity with an incorrect activity state, and the second one tries to reflect the situation of an activity with a satisfactory activity state. Both situations will be valued by a human expert using a percentage. Figure 5 shows a generic survey relative to any activity.
Fig. 5. Generic experts’ survey.

Table 2. Case structure.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case number</td>
<td>Function vector</td>
</tr>
</tbody>
</table>

Table 3. Assessment of the different levels of importance of the tasks.

<table>
<thead>
<tr>
<th>Importance rate</th>
<th>Numeric value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VHI (Very High Importance)</td>
<td>5</td>
</tr>
<tr>
<td>HI (High Importance)</td>
<td>4</td>
</tr>
<tr>
<td>AI (Average Importance)</td>
<td>3</td>
</tr>
<tr>
<td>LI (Low Importance)</td>
<td>2</td>
</tr>
<tr>
<td>VLI (Very Low Importance)</td>
<td>1</td>
</tr>
</tbody>
</table>

Case structure. The data acquired by means of surveys were used to build the prototype cases for the initial Store agent case base.

Table 2 shows the case structure that constitutes the case base. Each case is composed of the following attributes, that can be observed in Fig. 5:

- Case number: Unique identification: positive integer number.
- Input vector: Information about the tasks (n sub-vectors) that constitute an industrial activity: ((IR1, V1), (IR2, V2), …, (IRn, Vn)) for n tasks. Each task sub-vector has the following structure (IRi, Vi):
  - IRi: importance rate for this task within the activity. It can only take one of the following values: VHI (Very high importance), HI (High Importance), AI (Average Importance), LI (Low Importance), VLI (Very low importance).
  - Vi: Value of the realization state of a given task: a positive integer number (between 1 and 10).
- Function number: Unique identification number for each function
- Activity number: Unique identification number for each activity
- Reliability: Percentage of probability of success. It represents the percentage of success obtained using the case as a reference to generate predictions.
• Activity State: degree of perfection for the development of the activity, expressed by percentage. This is the solution of a problem case.

The case study presented in this work has been implemented under the framework of the Crisis project. This project was oriented to detect possible risky situations in SMEs, taken into account the crisis that affects the market. A multi-agent system was implemented and 18 SMEs participated in the experiment and were assigned a personal business agent. The enterprises were situated in different sectors and located in the Spanish region of Castilla y León. The economic context is the same for all the SMEs. The system was tested during 24 months, from January 2008 to December 2009, tuned and improved taking into account the experience acquired using a total of 196 cases. Next section presents the results obtained after testing the novel hybrid neural intelligent system in the real scenario.

6. Results

The approach has been applied to a real case scenario in the Spanish region of Castilla y León, for enterprises from different sectors. The human experts (the auditors) have remarked on the advantages of using the presented multi-agent system approach as a decision support system for detecting potential risky situations in SMEs, and have especially noted the facility in acquiring knowledge and explanations.

The data obtained demonstrates that the application of this hybrid neural intelligent system caused a positive evolution in all enterprises. This evolution was reflected in the reduction of inefficient processes. The indicator used to determine the positive evolution of the companies was the state of each of the activities analysed. After analysing one of the company’s activities, it was necessary to prove that the state of the activity (valued between 1 and 100) had increased beyond the state obtained in the previous three month period. The system considers small changes in the tasks performed in the SMEs, and all the experts that participated in the experiments considered 3 months as a significant time to evaluate the evolution of a SME related to these changes. Thus, it was possible to conclude that the inefficient processes had been reduced within the same activity. When there was measurable improvement in the majority of activities (above all those most relevant to the enterprise), it could be affirmed that the enterprise had improved its situation. Figure 6 shows two screenshots of examples of activities for an expert agent corresponding to activities for the same enterprise, Information Technologies and Imobilized. Figure 6 shows a positive evolution of the

![Fig. 6. Example of monitoring and prediction for two activities: Information technology and capital assets.](image-url)
activities. Taking into account the global results, in 86% of these companies, the number of inconsistent processes was reduced improving the state of activities by an average of 18%. In general, it could be said that these results demonstrate the suitability of the techniques used for the integration of the developed hybrid neural intelligent control system. The best results occurred in the smaller sized enterprises. This is due to the fact that these businesses have a greater facility to adopt and adapt to the changes suggested by the system’s predictions.

In order to have an objective evaluation of the prediction abilities of the hybrid neural intelligent system presented in this study, we have evaluated the system with different configurations and compared the results obtained: (i) Using a RBF network; (ii) Using a CBR mechanism with a RBF network in the Reuse phase; (iii) Using a CBR with a SOM in the Retrieve phase and a RBF in the Reuse phase; (iv) Using a CBR with WeVoS-SOM in the Retrieve phase and a RBF in the Reuse phase. Table 4 shows a summary of the results obtained. The four different techniques are compared, using an incremental case base containing from 10 to 190 cases. There is a new prediction for each of the new cases.

Table 4 shows the evolution of the results along with the increase in the number of cases stored in the memory of cases. The numerical results shown the average result of a series of tests performed with the available information. The number of tests carried out in each iteration (every time the case base grows) is ten percent of the size of the case base (for instance, when the case base contained 100 elements, 10 tests were performed, and so on). This percentage of the data used to test the correction of the system was not previously used in its training. Those cases are also part of the historical information used, so they can be used to check the correction of the predictions generated by the system. The results for each of the techniques being analyzed improved when the number of cases stored was increased. The incorporation of the WeVoS-SOM system in the Retrieve phase provides more accurate predictions. The size of the maps was calculated as suggested in Ref. 64, while the size of the ensemble was also increased from 2 maps up to 5, depending on the data managed. The improvement over the CBR-SOM consists of using the WeVoS-SOM algorithm to structure the case base, enhancing the organizational characteristics of the case base. More visual results of these experiments can be found on Fig. 7(a). In Fig. 7(a) the X axis represent the number of cases used in each of the tests and the Y axis represents the percentage of successful predictions.

Table 4. Percentage of good predictions obtained with four different techniques and for a different amount of cases.

<table>
<thead>
<tr>
<th>Number of cases</th>
<th>RBF (%)</th>
<th>CBR + RBF (%)</th>
<th>CBR + SOM (%)</th>
<th>CBR + WeVoS-SOM (%)</th>
<th>WeVoS-SOM + RBF (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>46</td>
<td>38</td>
<td>42</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>48</td>
<td>43</td>
<td>47</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>53</td>
<td>47</td>
<td>58</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>56</td>
<td>58</td>
<td>64</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>61</td>
<td>64</td>
<td>72</td>
<td>87</td>
<td></td>
</tr>
</tbody>
</table>
Fig. 7. (a) % Prediction results. (b) Size of the memory of cases. (c) Efficiency of the CBR system.

Table 5. Multiple comparison procedure among different techniques.

<table>
<thead>
<tr>
<th></th>
<th>RBF</th>
<th>CBR + RBF</th>
<th>CBR + SOM + WeVoS + RBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBR + RBF</td>
<td></td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>CBR + SOM +</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RBF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBR + WeVoS +</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

Table 5 shows a multiple comparison procedure (Mann-Whitney test) used to determine which models are significantly different from the others. The asterisk indicates that these pairs show statistically significant differences at the 99.0% confidence level. Table 5 shows that the CBR + WeVoS + RBF system presents statistically significant differences compared to the other models.

We had certain problems implementing the system, partly because the management and experts were not familiar with the use of computational resources.
devices and multi-agent systems, so some courses were given to introduce them to these technologies and teach them how to use the system interface. The proposed novel hybrid neural intelligent system is a unique system useful for dynamic environments, and open enough to be used in other enterprise environments. The experts noted that the behavior of the system improves as the number of cases in the memory of cases grows. Figure 8 shows a representation of the average satisfaction degree of the experts related to the growth of the number of cases. The expert satisfaction is obtained by means of surveys that were completed for different sizes of the memory of cases and taking into account the effectiveness of the system. 24 experts participated in the experiments and provided a satisfaction degree that takes values in the interval \([0, 100]\). We found some difficulties with this system, because some of the SMEs were reticent to complete surveys.

Management are the most reticent about trusting the system for several reasons: (i) they do not trust the partiality of the system, and are reluctant to facilitate their internal data (ii) updating the information about the enterprise requires specialized human resources and time. However, the auditors and experts believe that the CBR-BDI agents may favour their work and provide a highly appreciated decision support tool. They believe that this hybrid neural intelligent architecture has more advantages than disadvantages and that the system helped them to detect inconsistent processes in the enterprises. They tend to argue that the hybrid neural intelligent architecture should incorporate a shared memory of cases to compare data from different firms, but with the guarantee of data privacy.

### 7. Conclusions

This study has presented a hybrid neural intelligent system to predict enterprise failure in SMEs. It applies a hybrid reasoning system specifically designed to analyze data from enterprises and predict potential failures, in order to provide an innovative hybrid neural method for exploring the failure prediction process and extracting knowledge in the form of rules which help the human experts to understand the prediction process and to obtain conclusions about the relevance of the situation of the enterprise. If the system does not use information from banks or policy makers, is because the model proposed in this article is based on a previous study that does not take these considerations into account. However, we are working in an improvement of the system that takes into consideration information about financial crisis and external factors.

Nowadays, it is highly recommended for the SMEs to have available tools for predicting potential risky situations and have a better understanding of their internal functioning. The approach presented in this article provides the SMEs with a decision support tool that can contribute to predict failures and identify malfunctioning, as demonstrated in the experimental results. Management staff are, in general, reticent about trusting Artificial Intelligence-based systems, but the incorporation of the CBR paradigm helps to solve this situation by using past experiences. The proposed approach was applied to different SMEs (taking into account the particular problem/casusitic of each SME), and we have shown a high percentage of success to improve the behavior of the SMEs that participated in the experiments. However, it is not possible to extract general conclusions (i.e. All the SMEs should revise task X in Activity X), because this is not the aim of the proposed model. Although the defined model has not been tested in big enterprises, we believe that it could work adequately, although changes would take place more slowly than in SMEs. We are moving in this direction and expect that an evaluation of the system will soon be possible in a major international company from the textile sector. Moreover, additional
work is still required to improve the adaptation of the CBR-BDI agents when the size of the memory of cases is very high and to learn from the cases with invalid solutions. Besides, it is necessary to increase the sample size by incorporating more SMEs to the experiments as well as explore different sets of samples. These are our next challenges.

Acknowledgments

This research has been partially supported through the MICINN under project IDI-20110343; project of the Spanish Ministry of Science and Innovation TIN2010-21272-C02-01 (funded by the European Regional Development Fund). The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A., under project MAGNO2008 — 1028. — CENIT Project funded by the Spanish Government.

References

39. A. Polakar, L. Bruschel and W. Lamsdorff, Jaede: Implementing a BDI-Infrastructure for JADE Agents, in EXP — In Search of Innovation (Special Issue on JAIDE), Department of Informatics, University of Hamburg, Germany (2003), 76–85.
54. M. L. Borrojo, J. M. Corchado, C. Yanez, C. Fdez-Riverola and F. Diaz, Autonomous internal control system for small to medium firms, in Proc.


