A Semi-Supervised Learning-Aided Evolutionary Approach to Occupational Safety Improvement

Marco Cococcioni, Beatrice Lazzerini and Francesco Pistolesi
Department of Information Engineering, University of Pisa
Address: Largo L. Lazzarino 1, 56122 Pisa, IT
Email: {m.cococcioni, b.lazzerini, f.pistolesi}@iet.unipi.it

***Abstract*—Worldwide, four people die every minute as a consequence of illnesses and accidents at work. This considerable number makes occupational safety an important research area aimed at obtaining safer and safer workplaces. This paper presents a semi-supervised learning-aided evolutionary approach to improve occupational safety by classifying workers depending on their own risk perception for the task assigned. More in detail, a semi-supervised learning phase is carried out to initialize a good population of a non-dominated sorting genetic algorithm (NSGA-II). Each chromosome of the population represents a pair of classifiers: one determines a worker's risk perception with respect to a task, the other determines the level of caution of the same worker for the same task. Learning from constraints reinforces the initial training performance. The best Pareto-optimal solution to the problem is selected by means of the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). The proposed framework was tested on real-world data gathered through a website purposely developed. Results showed a good performance of the obtained classifiers, thus validating the effectiveness of the proposed approach in supporting the decision-maker in critical job assignment problems, where risks are a serious threat to the workers' health.**

I. Introduction

Every year, about 2.3 million people die worldwide as a result of occupational illnesses and accidents at work [1]. Also, there are 860,000 injury-causing occupational accidents every day.

Solutions to occupational safety problems are continuously being developed. However, with a more and more changing world of work, the workers' health and wellness remain a paramount concern. This requires new and integrated strategies for prevention to connect safety, health and welness of the individual.

One way to improve the safety and health of people at work could be to completely redesign the way the job assignment is carried out, i.e., the assignment process of workers to jobs. In fact, traditional job assignment lacks to consider people's risk perception, i.e., the way individuals evaluate characteristics and hazardousness of dangerous situations [2]–[6]. This is an important weakness particularly with reference to working environments characterized by risks that might cause serious consequences for workers' health.

With the foresight of achieving the awareness that every single task should be assigned to the person able to interact with the risks of that task in the safest way, this paper presents a way to model people's risk perception and their behaviour toward the risks of the tasks to be assigned. Such model should be reliable and consistent because it can significantly contribute, together with other typical aspects like the candidate's ability to perform a given task and the acquired expertise, to the final conclusions of the decision maker in the process of job assignment.

During a personnel recruitment (or assignment) process, the idea is to build two models (actually, classifiers) for each worker, representing the worker's risk perception for the risks of every single task, and the candidate's caution for that task, respectively. More in detail, the caution is expressed in terms of the preventive actions wokers would perform to prevent the risks of the task [7]. A preventive action can decrease the probability of risk occurrence and/or the risk impact on the worker's health.

Risk perception is influenced by criticality factors, i.e., age, education level, acquired expertise, number and type of accidents suffered in the workplace, etc [8]–[10]. Criticality factors are here split into *general* (like, e.g., age and education level), which are related to the worker no matter the task, and *task-related* (like, e.g., acquired expertise). The general criticality factors are used to compute the general perception level of risks by a worker. Then, the general perception level is used, together with the task-related criticality factors, to evaluate the workers' perception level of the risks associated with a given task. Further, the preventive actions a worker would perform with respect to the risks of a task are here used to assess the worker's caution level with respect to that task [11]. It would be reasonable to expect that when a person's perception level of the risks of a task is classified as high, then also the caution level of that person with respect to the same task should be classified as high, and vice-versa. The same would hold for the other levels.

Semi-supervised learning [12] within a stage-based learning scheme is used: for each classifier, a supervised learning stage is performed beforehand; the training process continues by using predictions on the unlabeled examples. For a given candidate, the outputs of the two classifiers represent two different (but related) aspects of that person (which is represented with different features in input to the two classifiers). Hence, given an unlabeled sample, the output of either classifier can be transformed into the corresponding label required by the other classifier, so as to build a supervised example for that classifier. This just stems from the fact that the two classifiers must produce consistent answers. For this reason, a consistency constraint between the two classifiers is established in terms of the coherence degree between their answers, given the same inputs.

A solution to this problem consists of a pair of systems that implement the first and the second classifier, respectively. The aim of the presented work is to generate a solution with the lowest possible classification error (i.e., the sum of the classification errors made by the two classifiers) and the lowest consistency constraint violation. The problem is here solved by means of a multi-objective optimization algorithm, namely the non-dominated sorting genetic algorithm (NSGA-II) [13], with two objectives to be minimized: the classification error and the consistency constraint violation extent.

The proposed methodology was validated using a dataset of real-world data gathered thanks to shoe factories which participated to the experiments.

The paper is organized as follows: Section II contains the formal model of the worker's risk perception and caution; Section III presents the problem; Section IV describes the evolutionary optimization methodology and the semi-supervised learning-based initialization of the chromosomes; Section V discusses the results obtained during the experiments; Section VI draws the conclusions.

II. Modeling worker's risk perception and caution

A. Elements of the model and their correlation

Let us consider a work environment characterized by a set of tasks (or jobs) $\mathcal{J}$, wherein each task is assigned to one worker. Let $\mathcal{W} = \{w_1, \dots, w_{|\mathcal{W}|}\}$ be the set of the workers. Each task t_i, where $i \in \{1, \dots, |\mathcal{J}|\}$, exposes the worker performing that task to a set of risks $\mathcal{R}_i$. The set of all the risks of the work environment is $\mathcal{R} = \bigcup_{i=1}^{|\mathcal{J}|} \mathcal{R}_i$.

Each risk $r_k \in \mathcal{R}$, with $k \in \{1, \dots, |\mathcal{R}|\}$ can be prevented by a set $\mathcal{A}_k = \{a_{k,1}, \dots, a_{k,|\mathcal{A}_k|}\}$ of *preventive actions*, i.e., actions able to decrease the probability of the risk materializing and/or make the consequences of the risk on the worker's health (the so-called *risk impact*) less injurious. A preventive action is associated with a *prevention level* in $\mathcal{L} = \{1, \dots, L\}$. The more a preventive action reduces the risk probability and/or the risk impact, the higher its prevention level. Prevention levels are typically assigned by experts in risk assessment.

Consider a set of criticality factors $\mathcal{F} = \{f_1, \dots, f_{|\mathcal{F}|}\}$ affecting risk perception. Each criticality factor f_v takes values in a domain $\mathcal{D}_v$. The set $\mathcal{F}$ is composed, in the order, of two subsets of criticality factors: dependent and independent on the task performed. The former contains G *general* criticality factors, while the latter T *task-related* criticality factors. Given a worker w_j, where $j \in \{1, \dots, |\mathcal{W}|\}$, the *risk perception general level* gen_perc_j of w_j depends on the set $\mathcal{G}_j = \bigcup_{v=1}^{G} d_{v,j}$, where $d_{v,j} \in \mathcal{D}_v$ is the value in the domain $\mathcal{D}_v$ of the general criticality factor f_v for worker w_j. Obviously, it holds that $\mathcal{G}_j \in \mathcal{D}_1 \times \cdots \times \mathcal{D}_G$. Hence, there exists a function $\varphi_{GENERAL} : \mathcal{D}_1 \times \cdots \times \mathcal{D}_G \to [0,1]$ such that

$$\mathcal{G}_j \mapsto \varphi_{GENERAL}(\mathcal{G}_j) = gen\_perc_j. \tag{1}$$

On the other hand, still considering worker w_j, the perception level $task\_perc_{i,j}$ of w_j for the set of risks of task t_i depends on the set $\mathcal{T}_j = \bigcup_{v=G+1}^{G+T} d_{v,j}$. Here, $d_{v,j}$ is the value in the domain $\mathcal{D}_v$ of the task-related criticality factor f_v for worker w_j. Also, $task\_perc_{i,j}$ depends on the risk perception general level gen_perc_j of w_j. This means that there exists a function $\varphi_{TASK} : \mathcal{D}_{G+1} \times \cdots \times \mathcal{D}_{G+T} \times [0,1] \to [0,1]$ such that

$$(\mathcal{T}_j, gen\_perc_j) \mapsto \varphi_{TASK}(\mathcal{T}_j, gen\_perc_j) = task\_perc_{i,j}. \tag{2}$$

For each risk r_k and each worker w_j, the caution level of w_j with respect to r_k depends on the number of preventive actions w_j would perform to protect himself/herself from r_k, for every single prevention level. More rigorously, let $\#\mathcal{A}_{k,\ell=\bar{\ell},j}$ denote the count of $\bar{\ell}$-level actions performed by w_j to prevent r_k. There exists a set of functions $\rho_k : \{0, \dots, |\mathcal{A}_{k,1}|\} \times \cdots \times \{0, \dots, |\mathcal{A}_{k,L}|\} \to [0,1]$ such that

$$\begin{aligned}&(\#\mathcal{A}_{k,\ell=1,j}, \dots, \#\mathcal{A}_{k,\ell=L,j}) \mapsto \\ &\quad \rho_k(\#\mathcal{A}_{k,\ell=1,j}, \dots, \#\mathcal{A}_{k,\ell=L,j}) = risk\_caution_{k,j},\end{aligned} \tag{3}$$

for each $k = 1, \dots, |\mathcal{R}|$.

Finally, for each task t_i and each worker w_j, the caution level of w_j with respect to t_i depends on every single $risk\_caution_{k,j}$ of w_j, each with respect to a risk r_k of task t_i. A set of functions $\tau_i : [0,1]^{|\mathcal{R}|} \to [0,1]$ such that

$$\begin{aligned}&\textstyle\bigcup_{r_k \in \mathcal{R}_i} risk\_caution_{k,j} \mapsto \\ &\quad \tau_i\left(\textstyle\bigcup_{r_k \in \mathcal{R}_i} risk\_caution_{k,j}\right) = task\_caution_{i,j}\end{aligned} \tag{4}$$

maps a configuration of risk cautions into the task caution of w_j for each task t_i of the workplace.

Therefore, in this model, a worker w_j is represented by the tuple

$$\theta_j = \left\{ \textstyle\bigcup_{v=1}^{G+T} d_{v,j}, \bigcup_{k=1}^{|\mathcal{R}|} \bigcup_{\lambda=1}^{L} \#\mathcal{A}_{k,l=\lambda,j} \right\}, \tag{5}$$

where $\bigcup_{v=1}^{G+T} d_{v,j}$ are the values of each criticality factor and $\bigcup_{k=1}^{|\mathcal{R}|} \bigcup_{\lambda=1}^{L} \#\mathcal{A}_{k,l=\lambda,j}$ are the counts of preventive actions for each prevention level toward each risk. Note that $v \in \{1, \dots, G\}$ denotes general criticality factors, while task-related criticality factors are indexed with $v \in \{G+1, \dots, G+T\}$.

Fig. 1 summarizes the relationships between all the components described above.

III. Problem

A. Outline

The problem dealt with in this paper is developing the two classifiers C1 and C2 of Fig. 1, which compute, respectively, $task\_perc_{i,j}$ and $task\_caution_{i,j}$, coherently with each other. In order to intuitively explain the coherence concept, three

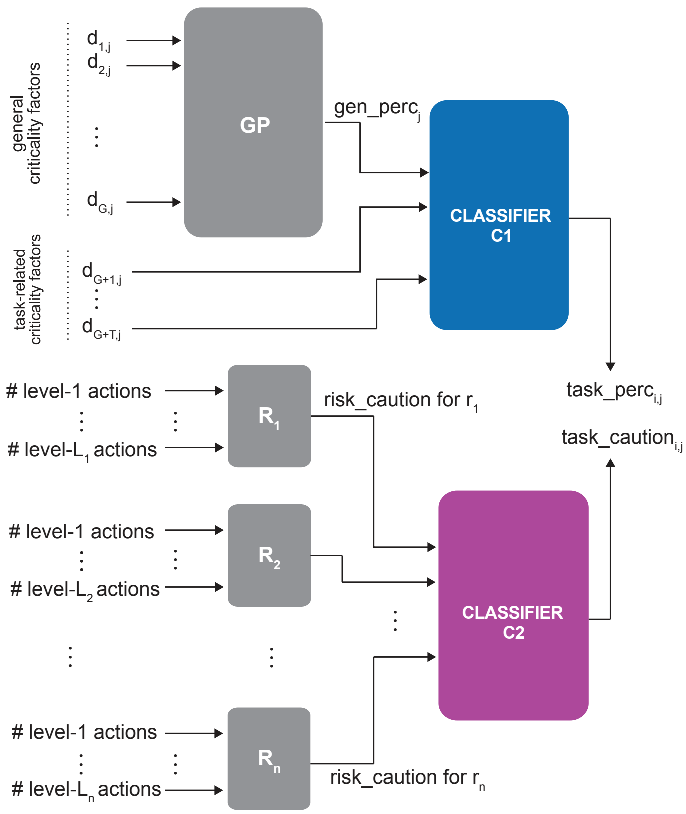

Fig. 1. Block diagram of the system. In the figure, it is supposed $|\mathcal{R}_i|= n$.

perception levels for every single risk of a task are here considered: `low`, `medium`, `high`. The same levels are used for the worker's caution with respect to a task.

The coherence of the outputs of the two classifiers (C1 and C2) establishes that when C1 classifies a worker as having a high perception level for the risks of a given task, then C2 classifies the same worker as having a high caution level with respect to that task, and vice-versa. The same holds for the other two levels, no matter the sets of definition of the labels.

B. Formulation and objectives

The problem is here conceived as a multi-objective optimization problem:

$$\min_{\mathbf{p}} \ \mathbf{E}(\mathbf{p}) = [L(\mathbf{p}), C(\mathbf{p})], \tag{6}$$

where $\mathbf{E}(\mathbf{p})$ is a vector-valued function representing the global error expressed by means of two terms:

- empirical risk (or loss) $L(\mathbf{p})$;
- constraint violation $C(\mathbf{p})$.

Objectives $L(\mathbf{p})$ and $C(\mathbf{p})$ depend on the parameters $\mathbf{p}$ used to implement the classifiers.

Loss is expressed as the total mean-squared error as:

$$L(\mathbf{p}) = MSE_{C1} + MSE_{C2} = \frac{1}{n}\left(\sum_{i=1}^{n}(\hat{y}_i^{C1} - y_i^{C1})^2 + \sum_{i=1}^{n}(\hat{y}_i^{C2} - y_i^{C2})^2\right), \tag{7}$$

where MSE_{C1} and MSE_{C2} are the mean-squared errors of classifier C1 and C2, respectively, n is the number of examples, and $\hat{y}_i^{C1}$, y_i^{C1} and $\hat{y}_i^{C2}$, y_i^{C2} are the obtained and desired output for classifier C1 and C2, respectively.

Constraint violation is measured with the opposite of the Pearson product-moment correlation coefficient:

$$C(\mathbf{p}) = -\frac{\mathbb{E}[(\hat{Y}^{C1} - \mu_{\hat{Y}^{C1}})(\hat{Y}^{C2} - \mu_{\hat{Y}^{C2}})]}{\sigma_{\hat{Y}^{C1}}\sigma_{\hat{Y}^{C2}}} \tag{8}$$

where $\mathbb{E}$ is the expectation, $\hat{Y}^{C1}$ and $\hat{Y}^{C2}$ are the output vectors of classifier C1 and C2, respectively, $\mu_{\hat{Y}^{C1}}$ and $\mu_{\hat{Y}^{C2}}$ are the means of $\hat{Y}^{C1}$ and $\hat{Y}^{C2}$, respectively, and $\sigma_{\hat{Y}^{C1}}$ and $\sigma_{\hat{Y}^{C2}}$ are the standard deviations of $\hat{Y}^{C1}$ and $\hat{Y}^{C2}$, respectively. $C(\mathbf{p})$ measures the extent of consistency violation on the outputs of the two classifiers: the greater the violation, the higher $C(\mathbf{p})$.

IV. Optimization Methodology

A. Overview

A semi-supervised learning-aided evolutionary methodology is here proposed to solve the problem presented in Section III. The methodology is aimed at finding the best implementation of C1 and C2 in order to minimize $\mathbf{E}(\mathbf{p})$.

Semi-supervised learning is used beforehand to initialize the population of a non-dominated sorting genetic algorithm (NSGA-II). In particular, each chromosome contains an MLP-based implementation of both C1 and C2. Implementations evolve till the genetic algorithm ends. The solution to the problem is represented by the chromosome containing the best implementation of classifiers C1 and C2. This solution is selected by means of the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [14]. The optimization methodology is described in detail in the next sections.

B. Representation of the chromosome

In Fig. 2, the representation of the chromosome is shown. In detail, each chromosome consists of two genes, related to classifier C1 and C2, respectively. These genes contain data structures representing the corresponding neural system implementation (in Fig. 2, `MLP1` for classifier C1 and `MLP2` for classifier C2).

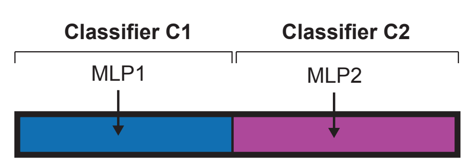

Fig. 2. Representation of the chromosome.

C. Risk perception general level and risk caution

Considered a worker w_j and a task t_i, this first phase is aimed at developing:

- an MLP neural network to perform the fitting of $\varphi_{GENERAL}$ in order to compute gen_perc_j, given $\mathcal{G}_j$;

- an MLP neural network for each risk of t_i, to fit ρ_k so as to compute each $risk\_caution_{k,j}$ (one for each $r_k \in \mathcal{R}_i$) from the counts of actions $\{\bigcup_{\lambda=1}^{L} \#\mathcal{A}_{k,l=\lambda,j}\}$ that w_j would perform for each prevention level.

D. Initialization and training technique for C1 and C2

In the second phase, classifiers C1 and C2 are developed so as they make coherent decisions (as specified in Section III) acting on a different representation of the same pattern (worker) expressed by a subset of the tuple in Eq. (5).

For each classifier, a neural system based on a multi-layer perceptron neural network (MLP) is implemented. Each neural system is trained in two steps.

1) Step 1: supervised learning: Supervised data are used here to train each system beforehand. At the end of this step, a trained system for each classifier is obtained. In the next sections, these systems are referred to as MLP1 for classifier C1, and MLP2 for classifier C2, respectively.

2) Step 2: semi-supervised refining: In this second step, the two systems resulting from the supervised learning of the previous step (i.e., MLP1 for C1, and MLP2 for C2) are refined through a learning phase, by means of newly created data, starting from the values the system parameters assume at the end of the first step. In this way, the classification performance achieved at the end of the previous step is expected to improve, because each classifier improves itself by taking advantage of what the other learned in the previous step.

More in detail, let us consider, e.g., C1. For each sample, the output of C1 is taken into account. This output is transformed into the corresponding "consistent" output of C2. Therefore, the supervised pair consisting of the input (represented as required by C2) and the output of C1 (transformed as required by C2) is built. The same procedure is carried out for C2. Now, C1 and C2 are trained with the new supervised data obtained as described above, assuring that the training process starts from the values assumed by the system parameters at the end of the first step. In practice, unsupervised data are used to generate, through each of the two classifiers, the desired outputs from the other. So-generated training sets are used to improve the performance of C1 and C2.

E. Generating the initial population

Individuals of the initial population are generated as follows. For each classifier, m MLP-based neural systems are initialized as follows. In particular, m random integer numbers are generated, with uniform probability, between a minimum and a maximum heuristically chosen. So-generated numbers represent the number of hidden neurons of the single hidden layer of as many MLPs. As demonstrated by the universal approximation theorem, under mild conditions, a single hidden layer is enough to represent any function of the inputs [15]. The resulting networks are created and trained as explained in Section IV-D. In this way, the obtained population is made of pairs of classifiers having a good performance.

The m neural systems are eventually distributed over m chromosomes.

F. Evolution

The evolution of the population is here performed with a modified version of the NSGA-II algorithm, based on the mutation operator only. Anyway, this work just represents a first step to validate the methodology, and it can obviously be enhanced (as future work) with purposely designed recombination operators and other types of system to implement the two classifiers in order to investigate the achievable improvement extent.

Here, for each part of the chromosome, the corresponding neural system (MLP1 or MLP2) is mutated with probability P_{MLP}. More in detail, each parameter ω of the network (i.e., neural weights) is transformed into $\xi\omega$, with $\xi \in [\xi_{min}, \xi_{max}]$. Also, with probability P^-_{MLP}, the sign changes.

Limit cases (e.g., boundaries) are managed. Also, sign change is applied just when it really makes sense.

V. Experiments and Discussion

A. Dataset

This section presents the validation results of the proposed methodology applied to real-world data gathered within small manufacturing enterprises producing shoes.

The methodology proposed in this paper was validated in MATLAB. A web application containing a questionnaire (i.e., a multi choice test) was implemented in Java EE and MySQL. Data gathering was carried out thanks to the workers of the aforementioned shoe factories, which filled out the questionnaire anonymously. The questionnaire is composed of two parts, each aimed at:

1) collecting data related to the worker's general and task-related criticality factors (the elements $d_{v,j}$ in Eq. (5));
2) collecting data related to the worker's behaviour in dealing with risk (the elements $\#\mathcal{A}_{k,l=\lambda,j}$ in Eq. (5)).

As an example, considering the cut risk, the preventive actions here used to characterize the worker's behaviour are:

- activation of the machinery safety elements;
- verification of the safety elements efficiency;
- put the gauntlet on;
- keep hands away from the cutting elements;
- switch off the cutting machine to fix a fault;
- periodically check and sharpen the cutting utensils;
- no particular action.

Each preventive action above has a prevention level.

Data collection was very difficult because of privacy laws and the fact that workers had to spend part of their working day in filling the questionnaire. The dataset consists therefore of 140 interviews. The footwear industry was chosen since it is characterized by serious risks, e.g., intoxication, crushing, fall, burn and amputation. The experiments were carried out by considering 20 typical tasks of a shoe factories, with their own risks.

B. Setup and parametrization

The values of the most important parameters used in the experiments are summarized in Table I and are justified in the following.

NSGA-II was set up with a population of 20 individuals, each consisting of a variable-length chromosome as explained in Section IV-B. This number was chosen to speed up the simulations. Even though this is a low number of individuals, the obtained Pareto front approximation was good. The NSGA-II parametrization was validated by using the Student's t-test on 27 different parameter configurations. Configurations were obtained by combining a population size in $\{20, 50, 100\}$, a mutation probability and a sign inversion probability in $\{0.05, 0.1, 0.15\}$. Therefore, 27 configurations were considered. The maximum number of generations was kept to 2000.

Each chromosome was built with two genes containing the two MATLAB data structures representing the single-hidden layer MLPs that implement the first and the second classifier (i.e., C1 and C2), respectively.

TABLE I
BEST PARAMETERS FOUND IN THE EXPERIMENTS

Parameter	**Value**
Hidden neurons range	{3, 15}
Neural learning algorithm	Backpropagation
Number of individuals	20
Max epochs for evolution	2000
Mutation probability (P_{MLP})	0.05
Mutation extent (ξ) range	[0.8, 1.2]
Sign inversion probability (P^{-}_{MLP})	0.05

The initial population was created as follows. For each chromosome, two random integer numbers were generated between 3 and 15 to represent the number of hidden neurons of the single hidden layer of the two MLPs (i.e., C1 and C2). Minimum and maximum values were chosen heuristically. So-structured networks were created and trained through the two-stage learning process explained in Section IV-D.

Individuals evolved through the mutation operator as follows. Upon selection, for every single gene of the selected chromosome, each parameter of the corresponding MLP (i.e., the neural weights) is perturbed with a probability equal to 0.05. Greater values showed too much of randomness. Lower values made the search process dramatically slow and generations ran out producing a poor approximation of the Pareto front.

From an operation point of view, a neural weight ω to be perturbed was transformed into $\xi\omega$, with $\xi \in [0.8, 1.2]$. This range produced the best results and was determined by means of a trial-and-error procedure. Finally, with a probability of 0.05, the sign of ω is reversed. For the sake of simplicity, this probability was chosen to be equal to the weight perturbation probability.

C. Discussion

The best Pareto front approximation obtained in the experiments is shown in Fig. 4.

Within Fig. 4, the solution with the lowest classification error (i.e., $L(\mathbf{p})$) is solution 19, while the one with the lowest consistency constraint violation (i.e., $C(\mathbf{p})$) is solution 20. The best compromise, i.e., solution 11, was automatically selected by the TOPSIS algorithm, with weights 0.5 and 0.5, for classification error and consistency constraint violation, respectively. With these weights we just gave the same importance to the objectives.

Now, let us examine solutions 11, 19 and 20 more in detail. As stated above, solution 19 is the one of the front having the lowest classification error. However, it cannot be used in practical applications since the consistency constraint violation is high.

This consistency constraint violation is evident from the scatter plot in Fig. 3, since the associated cloud is very sparse. The cloud dispersion stems from the poor Pearson correlation coefficient, with a value of ∼0.59. This suggests a poor consistency of the two classifiers, even though they are very accurate if considered singularly (the classification error is ∼0.017).

On the other side, solution 20 can be neither used in real-world situations, since the error of the two classifiers is high, leading to inaccurate results, even though the two classifiers exhibit high consistency. We verified that the high consistency is due to correlated mistakes (i.e., C1 and C2 tend to misclassify the same patterns). This is why solution 20 deserves to be discarded definitely.

TABLE II
OBJECTIVE FUNCTIONS VALUES OF THE INVESTIGATED MORE RELEVANT SOLUTIONS OF THE PARETO FRONT

Solution	$\mathbf{L}(\mathbf{p})$	$\mathbf{E}(\mathbf{p})$
19	0.59	0.017
11	0.682	0.0188
20	0.71	0.0265
4	0.681	0.02
9	0.68	0.0218
15	0.681	0.0225
8	0.688	0.023
16	0.69	0.0235
17	0.705	0.026

Solution 11, surrounded by a square in Fig. 4, is a good trade-off instead. Its classification error is lower than the one obtained by solution 20 and the coherence is higher than the one showed by solution 19, and it is good enough, in absolute. Fig. 3 shows a much better scatter plot if compared to the one of solution 19: in this case, the Pearson correlation coefficient is ∼0.682. In addition, the scatterplot of solution 11 is the unique wherein boundary lines separate the three classes (low, medium, high) pretty well, leading to just 10 misclassified workers out of 140. Considering the inherent difficulty of the problem at hand, this result is judged as good.

Even though TOPSIS selected solution 11 as the best compromise between classification error and constraint violation, during the experiments, we also investigated the performance of all the solutions in the region of the Pareto front on the right of solution 11.

As it can be seen from Fig. 4, all these solutions are characterized by similar classification errors. Considering the solution chosen by TOPSIS as reference, i.e., solution 11, the

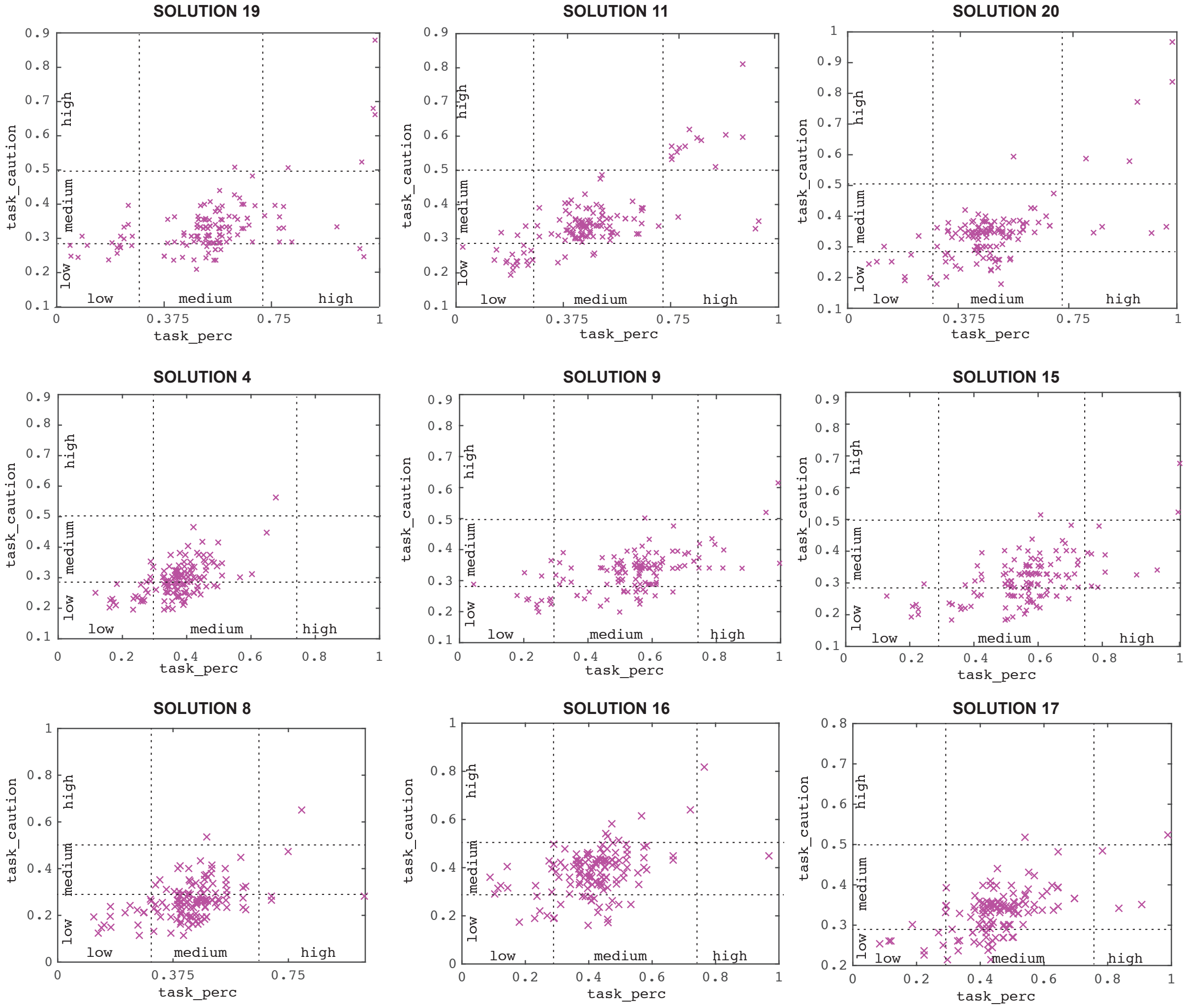

Fig. 3. Scatterplots of the most relevant Pareto-optimal solutions, each denoting a pair of classifiers (i.e., C1 and C2, whose outputs are `task_perc` and `task_caution`, respectively), characterized by a correlation higher than 0.68.

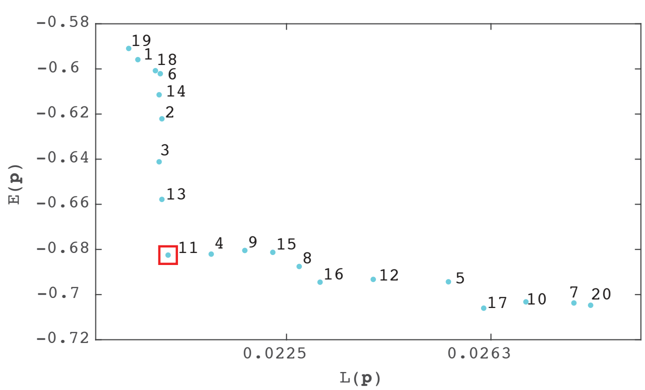

Fig. 4. Approximation of the Pareto front obtained at the end of the experiments. Each circle represents a Pareto-optimal solution identified by the closest number.

Pearson coefficient can be increased from 0.682 to 0.705. This is why we considered it important to evaluate this increase of consistency compared to the low increase of the classification error. Fig. 3 shows the scatterplots of the most significant solutions among those having a Pearson coefficient higher than solution 11. The objective functions values of these solutions are summarized in Table II. By inspecting the figure, it is possible to notice that all the other solutions are characterized by a poor performance. This is evident by looking at the thresholds used to define the `low`, `medium` and `high` classes, i.e., the dotted lines within the scatterplots.

VI. Conclusion

This paper presented a semi-supervised learning-aided evolutionary approach to occupational safety improvement. The goal was the automatic design of a pair of classifiers able to meet a consistency constraint between each other. The first classifier determines a worker's risk perception with respect to a task, while the second determines the level of caution of the same worker for the same task.

A semi-supervised learning approach is used within a two-stage learning scheme. During the first stage, the two classifiers are trained independently on their own training data. In the second stage, the training is continued using the predictions of the classifiers on an unlabeled dataset. In particular, the output provided by either classifier is transformed into the corresponding label required by the other classifier so as to build a new supervised example for that classifier. This strategy was used to create an initial population of multi-layer perceptron neural network-based classifiers of an evolutionary multi-objective algorithm (NSGA-II). The obtained Pareto front approximation contains trade-offs between the classification error and the associated coherence constraint violation of each pair of classifiers. By using TOPIS, we picked a single solution from the Pareto front, characterized by an acceptable classification error and an acceptable coherence constraint violation. This solution was able to predict both an accurate risk perception and an accurate (and consistent) level of caution, for 130 out of the 140 considered workers.

Therefore, the proposed technique represents a promising approach in supporting the decision-maker in critical job assignment problems, where risks are a serious threat to the workers' health.

As a future work, we are planning to strengthen each sub-module of our system (more powerful neural network-based classifiers), purposely developed recombination operators, and additional constraints/objectives to improve the quality of the final solution.

References

[1] "International Labour Organization Statistics Database (ILOSTAT)," http://www.ilo.org.

[2] M. Bouyer, S. Bagdassarian, S. Chaabanne, and E. Mullet, "Personality correlates of risk perception," *Risk Analysis*, vol. 21, no. 3, pp. 457–465, 2001.

[3] B. Chauvin, D. Hermand, and E. Mullet, "Risk perception and personality facets," *Risk Analysis*, vol. 27, no. 1, pp. 171–185, 2007.

[4] E. Peters and P. Slovic, "The role of affect and worldviews as orienting dispositions in the perception and acceptance of nuclear power," *Journal of applied social psychology*, vol. 26, pp. 1427–1428, 1996.

[5] L. Sjöberg, "Factors in risk perception," *Risk Analysis*, vol. 20, no. 1, 2000.

[6] P. Slovic, "Perception of risk," *Science*, vol. 236, no. 4799, pp. 280–285, 1987.

[7] B. Lazzerini and F. Pistolesi, "Profiling risk sensibility through association rules," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1484–1490, 2013.

[8] M. S. Horswill and F. P. McKenna, "The effect of perceived control on risk taking," *Journal of Applied Social Psychology*, vol. 29, no. 2, pp. 377–391, 1999.

[9] C. T. F. Klein and M. Helweg-Larsen, "Perceived control and the optimistic bias: a meta-analytic review," *Psychology and health*, vol. 17, no. 4, pp. 437–446, 2002.

[10] J. Costa-Font, E. Mossialos, and C. Rudisill, "Optimism and the perception of new risks," *Journal of risk research*, vol. 12, no. 1, pp. 27–41, 2009.

[11] B. Lazzerini and F. Pistolesi, "Classifying workers into risk sensibility profiles: a neural network approach," in *Proc. 8th European Modelling Symposium (EMS 2014)*, Pisa, IT, 2014, pp. 33–38.

[12] X. Zhu, A. B. Goldberg, R. Brachman, and T. Dietterich, *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comp*, vol. 6, no. 2, pp. 182–197, 2002.

[14] K. Y. Ching-Lai Hwang, *Multiple attribute decision making*. Springer-Verlag, 1981.

[15] G. Dreyfus, *Neural networks methodology and applications*. Springer, 2005.