Expert Systems with Applications 39 (2012) 11125–11134

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Classifying the risk of work related low back disorders due to manual material handling tasks

Jozef Zurada *

Department of Computer Information Systems, College of Business, University of Louisville, Louisville, KY 40292, United States

ARTICLE INFO

Keywords:
Manual lifting tasks
Manual material handling jobs
Low-back disorders
Classification accuracy rates
ROC charts
Computational intelligence methods

ABSTRACT

Work related low back disorders (LBDs) due to manual lifting tasks (MLTs) have long been recognized as one of the main occupational disabling injury that affects the quality of life of the industrial working population in the U.S. There have been a number of intensive research efforts devoted to understanding the phenomena of LBDs and building classification models that could effectively distinguish between high risk and low risk MLTs that contribute to LBDs. As of today, however, such models and the occupational exposure limits of different risk factors causing LBDs as well as the guidelines preventing them have not yet been fully proposed. One of the first efforts to comprehend the nature and phenomenon of LBDs was undertaken by Marras et al. (1993). They created a seminal data set and used it to build logistic regression (LR) models to identify significant variables and classify MLTs into high risk and low risk with respect to LBDs. Since then a number of studies have used the same data set to build and test various classifiers to detect the likelihood of LBDs due to manual material handling jobs. This paper summarizes and critiques the previous studies. It also employs this data set to build and test seven classification models, two of which have not been applied in this context yet. The parameters of the models have been calibrated for the best performance, and the models were constructed and validated on the full set and the reduced set of features. Though the performances of our best models are better than those reported in National Institute for Occupational Health and Safety (NIOHS) Guides and two of our previous studies, they are generally less optimistic than those reported in several other studies; this paper proposes a systematic and more reliable approach to creating and validating classifiers to distinguish between low and high risk MLTs that contribute to LBDs.

1. Introduction

Despite the widespread use of robots in industrial jobs such as assembling parts and MLTs, there are still many tasks in industry performed manually by humans. These tasks include lifting, holding, carrying, or moving heavy objects in the workplace. They often cause musculoskeletal injuries such as LBDs. Epidemiological studies (NIOSH, 1997, 1981; Waters, Putz-Anderson, & Garg, 1994; Waters, Putz-Anderson, Garg, & Fine, 1993) show that frequency rates and severity rates of LBDs increase significantly when objects lifted are bulky, objects are lifted from the floor, objects are frequently lifted, and loads are lifted asymmetrically (by one hand or at the side with the torso twisted). These tasks represent either cumulative exposure to handling objects over a long period of time or isolated incidents of overexertion when handling heavy objects.

Work related LBDs due to manual material handling (MMH) tasks have long been recognized as one of the main occupational disabling injury that affects the quality of life of the industrial working population in the U.S. (Ayoub, Karwowski, & Dempsey, 1996; Liberty, 2004). For example, in 1988, overexertion injuries across all industries accounted for 28.2% of all work injuries involving disability, while approximately 25% of all worker compensation claims were related to low back injuries (National Safety Council, 1990). According to Liberty Mutual Workplace Safety Index of Leading Occupational Injuries, the chief cause of workplace injuries in 2001 was an overexertion that accounted for 27.3% of all injuries. The top three injury causes (overexertion, falls on same level and bodily reaction) were the fastest growing of all injury causes, representing 50.1% of the total costs, i.e., about $23 billion a year or $450 million a week.

There have been a number of intensive research efforts devoted to understanding the nature and phenomena of LBDs as well as establishing risk factors that contribute to LBDs (Hou, Zurada, Karwowski, Marras, & Davis, 2007a, 2007b; Karwowski, Hancock, Zurada, & Ostaszewski, 1991; Karwowski, Ostaszewski, & Zurada, 1992; Karwowski, Zurada, Marras, & Gaddie, 1994; Karwowski et al., 2006; Marras, Fine, Ferguson, & Waters, 1999; Marras et al.,

* Tel.: +1 (502) 852 4681; fax: +1 (502) 852 4875.
E-mail address: jozef.zurada@louisville.edu

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.03.043

1993; Nelson & Hughes, 2009; Riihimaki, 1991; Spengler et al., 1986; Svensson & Andersson, 1989). Another stream of research concentrated on building classification models that could effectively distinguish between high risk and low risk MMH tasks that contribute to LBDs (Chen, Kaber, & Dempsey, 2000; Marras et al., 1993; Zurada, Karwowski, & Marras, 1997 Akay, 2011; Akay, Akcayol, & Kurt, 2008; Akay & Toksari, 2009; Asensio-Cuesta, Diego-Mas, & Alcaide-Marzal, 2010; Chandna, Deswal, & Pal, 2010; Chen, Kaber, & Dempsey, 2004; Zurada, 2012; Zurada, Karwowski, & Marras, 2004). Despite almost two dozen years of studies, however, classifiers which could effectively discriminate between low risk and high risk MMH jobs that cause LBDs have not been created yet. It also appears that the occupational exposure limits of different risk factors causing LBDs and the guidelines preventing them have not yet been fully established. Moreover, the nature and phenomena of LBDs is still relatively unknown.

One of the first attempts to understand the cause of LBDs was undertaken by Marras et al. (1993). They analyzed over 400 industrial lifting jobs from about 50 manufacturing companies. The authors selected 235 MMH tasks, defined several characteristics for them and created an experimental data set, which will be referred to as the Marras data set. The tasks were based on the combination of trunk motion and workplace factors and included 5 input variables: (1) LIFTR – lifting frequency (number of lifts per hour), (2) PTVAVG – peak twist velocity average, (3) PMOMENT – peak moment, (4) PSUP – peak sagittal angle and (5) PLVMAX – peak lateral velocity maximum. The PMOMENT and LIFTR variables are the workplace factors, whereas the remaining three variables are trunk motion factors. As the magnitude of each of these 5 variables increases, the risk of LBDs increases. Based on examination of the injury and medical records, all jobs were categorized into two groups: high risk of LBDs (111 cases – 47.2%) and low risk of LBDs (124 cases – 52.8%) to create the output variable, RISK. The authors developed a multiple LR model based on these five attributes. The predictive power of their model was more than three times greater than that of the NIOSH Work Practices Guide for Manual Lifting (1981); it allowed for discriminating between high and low risk jobs contributing to LBDs with the odds ratio of 1:10.7.

To create a better classifier than the one just described, Zurada et al. (1997) randomly split the Marras data set into 148 samples and 87 samples to build and test the models, respectively. Their best classification model was a feed-forward error-back propagation neural network (NN) with 10 neurons in the hidden layer which exhibited 74.7% overall correct classification accuracy rate on the test set. The correct classification accuracy estimates for high- and low-risk MLTs causing LBDs were 78.4% and 72.0%, respectively. Since then there have been a number of studies which used the same data set to build and test various classifiers. The performance of the best models from these studies was typically compared to the classification accuracy estimates reported in the original Zurada et al. (1997) study and the NIOSH guides. The results from all previous studies described below and this study are summarized in Table 1. In all of the studies, but two previous studies and this study, exactly the same two partitions of 148 samples and 87 samples were presumably used to construct and test the models. The review of the previous studies follow.

Chen et al. (2000) rightly argued that the NN with 10 neurons in the hidden layer trained and tested on the small data set could have caused overfitting, i.e., the network might have memorized training patterns and performed very well on the training data set, but not so well on the test set. They proposed a smaller NN with 3–5 neurons in a hidden layer and also used variable elimination techniques to arrive at a better model. Instead of using a standard error-back propagation gradient descent algorithm, they applied simulated annealing combined with conjugant gradient algorithms for models' building. The latter method improved a chance that a NN would converge to the optimal global solution and would not get trapped in local minimum. Their system showed a noticeable improvement, especially in classifying high risk jobs (83.8%). In the next study Chen et al. (2004) performed more extensive computer simulation testing the performance of LR, multiple discriminant analysis (MDA), and NN. They used the original data set with five input variables and also applied the variable reduction methods. One of their best models, which used only four variables on input to the NN with 4 neurons in a hidden layer, correctly classified 79.3% of test samples overall. The classification accuracy rates for low risk and high risk jobs were 76.0% and 83.8%, respectively. This was actually the same NN-based model that was reported in their previous study of 2000.

In a continued attempt to improve the performance of the classifiers, Zurada et al. (2004) compared the classification performance of five models: NN, LR, decision tree (DT), memory-based reasoning (MBR), and Ensemble. The authors arbitrarily divided the data set into 3 parts. That is, 40%, 30%, and 30% of the samples in the data set were randomly allocated to the training set, validation set, and test set, respectively. The training set was used to create the models and the test set was used to test the classification effectiveness of the models. The validation set, though not used directly in training, helped to fine tune the models. Simply, the training stopped when the cumulative error on the validation set reached minimum. The authors pointed out that partitioning the small data set into 3 subsets might cause some loss of information as the number of samples used for building, validating, and testing the models was reduced. Also, extra sampling might introduce a new source of variability and decrease the stability of the results. This was countered, however, at least to some extent, by running computer simulation for 10 random generations of these three sets and averaging the classification accuracy rates to obtain somewhat unbiased and more reliable results. The MBR model (75.6%) performed the best (Table 1). The DT model was second and exhibited a 73.0% overall correct classification accuracy rate. It also correctly classified 67.6% and 79.1% of low risk and high risk jobs, respectively. Furthermore, the DT identified the PMOMENT variable as the most significant and placed it on the top of the tree. The DT model generated a rule *If PMOMENT* $\geqslant$ 22.45 *Then* 79.1% *of test samples have been correctly classified as high risk jobs*. Thus this variable was a major determinant in classifying jobs into a high risk category. The authors also introduced a ROC chart to find out if there is any distinction between the overall performances of the different models at probability cutoffs $\neq 0.5$.

Akay et al. (2008) proposed a neuro-fuzzy (NF) approach in which fuzzy sets represented by membership functions and fuzzy rules were generated from the input/output data pairs. Though they did not obtain a significant improvement in the classification performance, their NF system similarly to the crisp rule of the DT designed by Zurada et al. (2004), could generate simple and easy to understand membership functions and fuzzy rules which provided more insight into the decisions and allowed for more interpretability of the results. The NF model was consistent with studies by Chen, Kaber, and Dempsey (2000), Chen, Kaber, and Dempsey (2004) and Zurada et al. (2004) in identifying that the PMOMENT variable was the most significant in determining the outcome. Akay and Toksari (2009) proposed the ant colony optimization (ACO) approach, modified to handle classification problems. The ACO method exhibited the best overall classification accuracy (81.6%) and dramatic improvement in classification of high risk jobs (94.6%). The authors, however, give little details on how the model was designed, calibrated, and tested.

Chandna et al. (2010) used semi-supervised random forest (SSRF) achieving 78.2% in the overall classification accuracy rate, and 80.0% and 75.7% rate in correct classification of high- and low-risk jobs, respectively. Asensio-Cuesta et al. (2010)

Table 1
The correct classification accuracy rates [%] of the best models reported in the previous studies, 1981 and 1991 NIOSH guides, and this study.

Study	Best Model	Classification accuracy rates – 0.5 cut-off		
		Overall	Low risk	High risk
NIOSH guide (1981)	–	–	90.0	10.0
NIOSH Revised Lifting Equation of 1991 (Waters et al., 1993, 1994)	–	–	55.0	73.0
Zurada et al. (1997)	NN	74.7	72.0	78.4
Chen et al. (2000)	NN	79.3	76.0	83.8
Zurada et al. (2004)	MBR	75.6	73.2	78.2
Chen et al. (2004)	NN	79.3	76.0	83.8
Akay et al. (2008)	NF	77.0	82.0	72.2
Akay and Toksari (2009)	ACO	81.6	72.0	94.6
Chandna et al. (2010)	SSRF	78.2	80.0	75.7
Asensio-Cuesta et al. (2010)	NN	81.6	78.0	86.5
Akay (2011)	Grey technique	91.7	97.6	85.1
This Study:				
All 5 input attributes	SVM	74.2	**83.0**	64.0
4 input attributes	LR	73.3	81.0	64.7
3 input attributes	NN	**75.9**	72.9	**79.1**

implemented several multi-layered NNs with different number of neurons in the hidden layers. They split the original data set into training, validation and test sets. The test set was exactly the same as in Zurada et al. (1997) study. The training stopped when the cumulative error on the validation set reached minimum. The authors reported that their best NN model accurately classified 78.0% and 86.5% of low risk and high risk jobs, respectively, whereas the overall correct classification accuracy rate measured on the test data was 81.6%. The model outperformed the models created by Zurada et al. (1997) and Chen et al. (2000, 2004).

Akay (2011) used the grey relational analysis based on the instance based learning approach for predicting of LBDs. This technique is borrowed from MBR and *k*-nearest neighbor (*k*NN). Presumably, computer simulation was performed on multiple folds of data as the author reported the best, the worst, and the overall classification accuracy rates averaged over the number of folds as well as the estimates for high- and low-risk jobs contributing to LBDs. Though the grey technique is very sensitive to the distance measure used to compute the similarity between the samples and the number of nearest neighbors from the training set used to classify the test sample, the author neither reported on the latter two nor gave any indications on the count of folds used. The average classification accuracy rates were extremely high and amounted to 91.7% (overall), high risk jobs (85.1%), and low risk jobs (97.6%). Finally, Zurada (2012) presented some preliminary results from computer simulations which were consistent with this study.

Table 1 summarizes the test set classification accuracy rates at 0.5 probability cut-off generated by the best models from the mentioned studies, NIOSH guides, and this study. The results from this study are averaged over 10 folds and 10 runs. The results from the Zurada et al. (2004) are averaged over 10 random generations of test sets, whereas the results from the remaining studies represent a single classification accuracy rate obtained on the test set containing 87 samples. The number of folds used in the Akay (2011) study was not reported.

The major drawback of the mentioned studies, however, was that in all of them, but Zurada et al. (2004), presumably Akay (2011), and this study the models were built and tested on the same single partitioning of the data set into training and test sets, containing 148 and 87 samples, respectively. The reported overall, high-risk, and low-risk classification accuracy rates on the test set for the best models were represented by a single number. Since a classifier may give results that differ by several percent on slightly different data partitions, single numbers do not mean much. In other words, building and testing the models on a single split of the data set could make the classification accuracy rates sample specific and somewhat unrealistic, and would not allow one to generalize the obtained results. Furthermore, the exact purpose of using the validation data set is not entirely clear from the Chen, Kaber, and Dempsey (2000, 2004) studies. If the validation set was used in training to fine tune the models and then test the models, the reported classification accuracy rates were too optimistic. Also, a single 0.5 probability cutoff was used to determine whether a job is classified as high risk or low risk, i.e., if the target event is detecting high risk jobs and the model produces a value $\geqslant 0.5$ the job is classified as high risk; otherwise it is classified a low risk job. If the cost of misclassifying a high-risk task and a low risk task is, for example, 2.3 times larger than the cost of misclassifying a low risk task as a high risk, a 0.3 threshold should be used. Consequently, receiver operating characteristics (ROC) curves and the areas under them which testify to the global performance of the models at different cut-offs should be employed.

This paper represents a substantial extension of the previous studies. In this study the same data set is used to build and test seven classification models on the full data set with 5 attributes (the original data set) and two data sets with reduced number of attributes, i.e., four and three variables. The models include: LR, NNs, radial basis function neural network (RBFNN), support vector machines (SVM), *k*NN, DT, and random forest (RF). To our best knowledge, SVM and RBFNN have not been used in this context yet. To obtain true and unbiased classification accuracy estimates a 10-fold cross-validation experiment is applied and run 10 times to obtain 100 rates which are averaged over the number of folds and runs. The parameters of the models have been tuned for the best performance. A two-tailed paired *t*-test is also applied to find out if the differences between the performances of the models for each data set and the performance of data sets for each model are statistically significant at $\alpha = 0.05$ and $\alpha = 0.01$ (Witten & Frank, 2005). Finally, the ROC charts and the areas under them are used to determine the global predictive power of the created models for a continuum of probability cut-offs from within the range [0,1]. Though the obtained classification accuracy rates for our best models are better than those reported in the NIOHS Guides and two of our previous studies, they are generally less optimistic than those reported in other studies; this paper proposes a reliable and systematic approach to building and testing classifiers to discriminate between low and high risk MLTs that contribute to LBDs. This study basically attempts to answer the following question: How "good" is the Marras data set for building models in classifying high- and low-risk MLTs that contribute to LBDs? In other words,

what are the realistic classification performances of the various models constructed and tested on this data set?

The paper is organized as follows. A brief description of the methods used is covered in Section 2. The results and computer simulations are described in Section 3. Finally, Section 4 concludes the paper.

2. Brief description of the selected methods

To make the paper self-contained and coherent this section briefly describes the fundamental properties of the seven methods used in this study. These are LR, NN, RBFNN, SVM, *k*NN, DT, and RF.

2.1. Logistic regression

The purpose of the LR model is to obtain a regression equation that could predict in which of two groups an object could be placed (e.g. a high risk job category or a low risk job category). The LR model also attempts to predict the probability that a binary or ordinal target will acquire the event of interest (e.g. high risk or low risk) as a function of one or more independent variables based on the combination of trunk motion and workplace factors for the problem at hand. The logistic model is represented by the response function $P(y)$ of the form:

$$P(y) = \frac{1}{1+e^{-z}}, \quad \text{where} \quad z = b_0 + \sum_{i=1}^{m} b_i x_i \tag{1}$$

The function $P(y)$ describes a dependent variable y containing in our case two qualitative outcomes, z is the function of m independent variables x called predictors, and b represents the parameters. The x variables can be categorical or continuous variables of any distribution. The value of $P(y)$ that varies from 0 to 1 denotes the probability that a dependent variable y belongs to one of two groups. The principal of maximum likelihood can commonly be used to compute estimates of the b parameters. This means that the calculations involve an iterative process of improving approximations for the estimates until no further changes can be made (Agresti, 1992; Christensen, 1997). Unlike NNs, LR models are designed to predict one dependent variable at a time. On the positive side, one can note that LR output provides statistics on each variable included in the model. Based on these statistics one can evaluate contribution of each variable to the predictive capability of the model.

2.2. Neural networks

Neural networks (NNs) are nonlinear mathematical models that mimic the architecture of the human brain. They try to emulate the way the human brain functions and processes information. Neural systems are built of highly interconnected neurons. The most attractive features of these networks are their ability to learn from training patterns, adapt to changing conditions, and generalize. NN models are characterized by their three properties: the computational property, the network architecture, and the learning method. A typical neuron contains a summation node and a nonlinear sigmoid activation function. A neuron accepts vectors on input called training patterns. Neurons are organized in layers and are connected by weights represented by small numerical values. In this study a common two-layer feed-forward NN with error back-propagation is used. The network has two layers: a hidden layer and an output layer. The neurons at the hidden layer receive the values of input vectors and propagate them concurrently to the output layer.

NNs' learning is a process in which a set of input vectors is presented sequentially and repeatedly to the input of the network in order to adjust its weights in such a way that similar inputs give the same output. In supervised learning the training set consists of the training patterns that appear on input to the NN and the corresponding desired responses provided by a teacher. The differences between the desired response and the network's actual response for each single training pattern modify the weights of the network in all layers. The training continues until the performance function measured as the mean sum of squares of the network errors $mse = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{t}_i - \mathbf{a}_i)^2$ for the entire training set containing N training vectors is reduced to a sufficiently small value close to zero. The **t** vector and **a** vector represent network's desired response and actual response, respectively.

There are many variations of the back-propagation algorithm used for training feed-forward networks. The simplest one updates the weights in the steepest descent direction – the negative of the gradient of the performance function. A single iteration k of this algorithm can be written as $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{g}_k$, where $\mathbf{x}_k$ is a vector of current weights, $\mathbf{g}_k$ is the current gradient, and α_k is the learning rate. To improve the speed of convergence of the algorithm, the Newton or quasi-Newton modification to the gradient method can be introduced as $\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1}\mathbf{g}_k$, where $\mathbf{A}_k$ is the Hessian matrix (second derivatives) of the performance index at the current value of weights (Battiti, 1992; Moller, 1993).

NNs are robust classifiers which can detect complex relationships between input attributes and an output attribute by building nonlinear and complex partitions around data. However, multiple experimentations are needed to determine the right number of neurons in a hidden layer and finding global optimum for a set of weights is not guaranteed. For a long time, NNs were considered black boxes whose decisions may be difficult to explain. However, recent research has shown that NNs can generate easy to explain if-then rules that can make sense of the relationships between input-output. These rules may be irreproducible, i.e., when weights of a NN are initialized with another set of random values it is likely to generate different rules.

2.3. Radial basis function neural network

A radial basis function neural network (RBFNN) is simply an SVM with the radial basis Gaussian function kernel used in each neuron of the hidden layer (Mitchell, 1997; Park & Sandberg, 1991; Poggio and Girosi, 1990). In RBFNN each neuron represents a point in input space, and its output for a given training pattern depends on the distance between its point and the pattern. The closer these two points are, the stronger the activation. The RFBNN uses Gaussian activation functions u_j whose width may be different for each neuron. The output u_j of the jth hidden neuron is given by

$u_j = \exp\left[-\frac{(\mathbf{x}-\mu_j)^T(\mathbf{x}-\mu_j)}{2\sigma_j^2}\right]$, where $j = 1,2,\ldots,m$, and m is the number of hidden neurons, $\mathbf{x}$ is the input pattern vector, $\boldsymbol{\mu_j}$ is its input weight vector (the center of the Gaussian for node j), and σ_j^2 is the normalization parameter, such that $0 \leqslant u_j \leqslant 1$ (the closer the input to the center of the Gaussian, the larger the response of the neuron).

The output layer forms a linear combination from the outputs of neurons in the hidden layer of the form $y_j = \mathbf{w}_j^T\mathbf{u}$, $j = 1,2,\ldots,l$, where l is the number of neurons in the output layer, y_j is the output from the jth neuron in the output layer, $\mathbf{w}_j$ is the weight vector for this layer, and $\mathbf{u}$ is the vector of outputs from the hidden layer.

A network learns two sets of parameters. First, it learns the centers and width of the Gaussian functions by employing the c-means clustering algorithm and then it uses the least mean square error algorithm to learn the weights used to form the linear combination of the outputs obtained from the hidden layer. As the first set of parameters can be obtained independently of the second set,

RFBNN learns almost instantly if the number of hidden units is much smaller than the number of training patterns.

2.4. Support vector machines

Support vector machines (SVM) is a system that represents a blend of linear modeling and instance-based learning to implement nonlinear class boundaries (Vapnik, 1998). This system chooses several critical boundary patterns called support vectors for each class (high risk and low risk of the output variable) and create a linear discriminant function that separates them as widely as possible by applying a linear, quadratic, cubic or higher-order polynomial term decision boundaries. A hyperplane that gives the greatest separation between the classes is called the maximum margin hyperplane in the form of

$$x = b + \sum \alpha_i y_i (\mathbf{a}(i) \cdot \mathbf{a})^n \tag{2}$$

where i is support vector, y_i is the class value of training pattern $\mathbf{a}(i)$, while b and α_i are parameters determined by the learning algorithm. The vectors $\mathbf{a}$ and $\mathbf{a}(i)$ represent a test pattern and support vectors, respectively, while an expression $(\mathbf{a}(i) \cdot \mathbf{a})^n$, which computes the dot product of the test pattern with one of the support vectors and raises the result to the power n, is called a polynomial kernel. Other kernel functions such as radial basis functions could also be used to implement a different nonlinear mapping. Constrained quadratic optimization is applied to find support vectors for the pattern sets as well as parameters b and α_i.

2.5. k-nearest neighbor

*k*NN is the process of solving new problems based on the solutions of similar past cases. The method requires no model to be fitted, or function to be estimated. Instead it requires all cases with their known solutions to be maintained in memory, and when a prediction is required, the method recalls items from memory and predicts the value of the dependent variable. In solving a new case, the *k*NN approach retrieves cases it deems sufficiently similar and uses them a basis for solving the new case.

The *k*-NN method requires no model to be fitted, or function to be estimated. The *k*-nearest neighbor algorithm takes a data set of existing cases and a new case to be classified, where each existing case in the data set is composed of a set of variables and the new case has one value for each variable. The normalized Euclidean distance or Hamming distance between each existing case and the new case (to be classified) is computed. The k existing cases that have the smallest distances to the new case are the *k*-nearest neighbors to that case. Based on the target values of the *k*-nearest neighbors, each of the *k*-nearest neighbors votes on the target value for a new case. The votes are the posterior probabilities for the class dependent variable (RISK).

More formally, a high-level summary algorithm that computes the distance between each new case $z = (\mathbf{x}', y')$ and all the training patterns $(\mathbf{x}, y) \in D$ to calculate its nearest-neighbor list, D_z, can be outlined as follows (Tan, Steinbach, & Kumar, 2006).

1. Let k be the number of nearest neighbors and D be the set of training patterns.
2. **for** each new case $z = (\mathbf{x}', y')$ **do**
3. Compute $d(\mathbf{x}', \mathbf{x})$, the distance between z and every pattern, $(\mathbf{x}, y) \in D$.
4. Select $D_z \subseteq D$, the set of k closest training patterns to z.
5. $y' = \underset{v}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_z} I(v = y_i)$

6. end for

Once the nearest neighbors list is obtained, the new case is classified based on the majority class of its nearest neighbors: Majority voting: $y' = \operatorname{argmax} \sum_{(x_i, y_i) \in D_z} I(v = y_i)$, where v is a class label, y_i is the class label for one of the nearest neighbors, and $I(\cdot)$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

In the majority voting approach, every neighbor has the same impact on the classification. This makes the algorithm more sensitive to the choice of k. To reduce the influence of k, one can weigh the impact of each nearest neighbor $\mathbf{x}_i$ according to its distance: $w_i = 1/d(\mathbf{x}', \mathbf{x}_i)^2$. As a result, training patterns that are located far away from z will have a smaller influence on the classification compared to those that are located closer to z. Using the distance-weighted voting scheme, the class label of the new case can be determined as follows: Distance-Weighted Voting:

$$y' = \underset{v}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_z} w_i \times I(v = y_i) \tag{3}$$

There are two critical choices in the *k*-NN method, namely, the distance function and the cardinality k of the neighborhood. We performed several experiments for different values of k and used the normalized Euclidean distance for the variables to calculate the similarity between cases. Normalization was required to ensure that features with larger values do not overweight features with lower values. Furthermore, to minimize the influence of k, we used the voting approach with weighted-distance in computer simulation. For more details on the *k*-NN method, the reader is encouraged to refer to (Guidici, 2003; Han & Kamber, 2001; Mitchell, 1997; SAS Enterprise Miner at www.sas.com; Tan et al., 2006).

2.6. Decision trees

DTs are fairly simple and widely applied tools for classification. A tree is built of nodes and branches. It has 3 types of nodes: a root node (a top node), internal nodes, and leaf nodes. In a binary tree, a top node has no incoming branches and two outgoing branches. Each internal node has exactly one incoming branch and two outgoing branches. Finally, each leaf node has exactly one incoming branch and no outgoing branches. Each leaf node is assigned a class label. Branches coming of the root and other internal nodes contain attribute test conditions to separate cases that have different characteristics. One of the greatest advantages of DTs is the fact that knowledge can be extracted and represented in the form of classification if-then rules between the input attributes and the target attribute.

The operation of DTs are based on the ID3 or C4.5 divide-and-conquer algorithms (Quinlan, 1987) and search heuristics which make the clusters at the node gradually purer by progressively reducing impurity in the original data set. The algorithms place the attribute that has the most predictive power at the top node of the tree and they have to find the optimum number of splits and determine where to partition the data to maximize the information gain. The fewer the splits, the more explainable the output is (there are less rules to understand). Selecting the best split is based on the degree of impurity of the child nodes. For example, a node which contains only cases of class *high-risk* or class *low-risk* has the smallest impurity = 0. Similarly, a node that contains an equal number of cases of class *high-risk* and class *low-risk* has the highest impurity = 1. Impurity can be measured by the well-established concept of entropy and information gain. Other popular measures include Gini reduction, classification error, and chi square. We formally introduce the entropy method measure below.

Given a collection S, containing the positive (*high-risk*) and negative examples (*low-risk*) of some target concept, the entropy of S relative to this Boolean classification is

$$Entropy(S) \equiv -p_{high-risk}\log_2 p_{high-risk} - p_{low-risk}\log_2 p_{low-risk} \quad (4)$$

where $p_{high\text{-}risk}$ is the proportion of positive examples in S and $p_{low\text{-}risk}$ is the proportion of negative examples in S. If the output variable takes on k different values, then the entropy of S relative to this k-wise classification is defined as

$$Entropy(S) = -\sum_{i=1}^{k} p_i \log_2 p_i \quad (5)$$

For example, if impurity is measured by entropy, the information gain, $Gain(S,A)$ of an attribute A, relative to a collection of examples S, can be computed as

$$Gain(S,A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{S_v}{S} Entropy(S_v) \quad (6)$$

where $Values(A)$ is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A has the value v (i.e., $S_v = \{s \in S | A(s) = v\}$.

2.7. Random forest

A random forest (RF) model comprises of many decision tree classifiers where each classifier is created using examples/patterns chosen randomly from the data set. RF outputs the class that is the mode of the class's output by individual trees (Breiman, 2001). Random forest classifier consists of using randomly selected features or a combination of features at each node to grow a tree. It uses the Gini index as an attribute selection measure, which measures the impurity of an attribute with respect to the classes. A random tree is grown to the maximum depth on new training data using a combination of features. These full-grown trees are not pruned. An algorithm for constructing each tree can be outlined as follows.

1. Let the number of training patterns be N, and the number of attributes in the classifier be M.
2. Choose the number m of input variables to be used to determine the decision at a node of the tree; $m << M$.
3. Choose a training set for this tree by choosing N times with replacement from all N available training patterns. Use the rest of the cases to estimate the error of the tree, by predicting their classes.
4. For each node of the tree, randomly choose m variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.
5. Each tree is fully grown and not pruned (as may be done in building a normal tree classifier).

3. Results and discussion

The performances of seven models: LR, NN, RBFNN, SVM, k-NN, DT, and RF on three versions of the Marras data set are investigated. These versions are: the original data set with five input features and two data sets with the reduced number of features, i.e., four and three features. Several feature reduction techniques such as chi squared and entropy reduction consistently showed that the LIFTR and PSUP attributes (in this order) had the least predictive power. Consequently, the data set with four attributes does not contain the LIFTR feature, whereas the data set with three variables does not include both the LIFTR and PSUP features.

A 10-fold cross-validation is employed to each of the seven methods and three data set pairings, and for reliable and unbiased classification rates each experiment is repeated 10 times. The performance measures of the methods and data sets are then averaged across these 10 folds and 10 runs. This approach ensures that data subsets used to train the models are completely independent from data subsets used to test the models. A two-tailed paired t-test (at $\alpha = 0.05$ and $\alpha = 0.01$) is used to verify whether the classification performances across the models and data sets are significantly different from the baseline (LR) method and the baseline (original) data set with five features, respectively (Witten & Frank, 2005). The LR method is used as the baseline because this traditional technique has been successfully applied to many classification problems and it was also used in early study by Marras et al. (1993). The original data set with five variables was selected as the baseline as it was used in most of the previous studies for constructing and testing the models. Most of these studies have not used any attribute reduction techniques. The parameters for the seven models for each of the three data sets were optimized for the best performance. Finally, the ROC charts and the areas under them are used to determine the global predictive power of the created models for a continuum of probability cut-offs from within the range [0,1].

The models were implemented using Weka, an open source data mining software written in Java which contains a collection of machine learning algorithms (http//www.cs.waikato.ac.nz/ml/weka/). Standard Weka settings were used for LR, DT, and RF. However, the parameters for the four remaining models: NN, RBFNN, SVM, and kNN were tuned using the Grid and CVParameterSelection functions which perform grid search (Witten & Frank, 2005). The NN model with 4 neurons in the hidden layer seemed to perform the best. The kNN method used 18 nearest neighbors, the Euclidean distance measure, and no distance weighting. Our best RBFNN had 5 clusters with the 0.5 standard deviation each. A polynomial kernel of power 2 and complexity parameter of 10 was used for SVM.

Tables 2–4 depict the overall, high risk, and low risk correct classification accuracy rates with their respective standard deviations at a single 0.5 probability cut-off for seven models and three data sets. The 0.5 cut-off means that if the event of interest is detecting a high risk task and a classifier generates the probability $\geqslant 0.5$, the sample is classified as high risk; otherwise it is a low risk task. The 0.5 cut-off also enables one to compare the classification accuracy rates from the best models in this paper to the results from the studies described earlier (Table 1). Table 5 shows the areas under the ROC curves and standard deviations. With LR as the baseline method the rates of seven methods on each of the three data sets are compared across that table rows. The performance rate is suffixed with the superscripts ^{b,bb} and ^{w,ww} to indicate whether compared to the LR method (i.e. baseline) each one of the six other methods performs significantly better or worse (at $\alpha = 0.05$ and $\alpha = 0.01$ respectively) than the baseline. With the original data set with five features as the baseline the rates of the three data sets for each of the seven methods are compared down the table columns. The rate is prefixed with subscripts $_{h,hh}$ and $_{l,ll}$ to indicate whether each remaining data set is significantly better or worse (at $\alpha = 0.05$ and $\alpha = 0.01$, respectively) than the original (i.e. baseline) data set in terms of classification performance. The ROC curves in Figs. 1–4 compare the global performance of the five selected methods on the original data set with five variables and the data set with three variables. The ROC charts for the data set with four variables are not shown as they exhibit similar patterns to the ROC charts in Figs. 1–4. We do not include ROC charts which would allow one to compare the performance of three data sets for each of the selected methods either.

Table 2 shows the average overall correct classification accuracy results for seven models and three data sets. Across the table rows we compare each of the six models to LR (the baseline) for each of the three data sets. Down the table columns we compare each of the two data sets to the original data set with 5 input variables (the baseline) for each of the seven models. In terms of the overall rates for the original data set, one can see that there are no

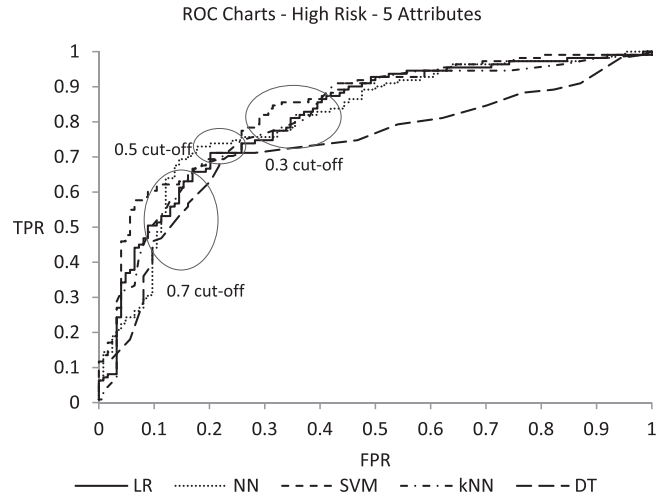

Fig. 1. The ROC charts for the high risk tasks for the 5 selected models built on 5 input variables.

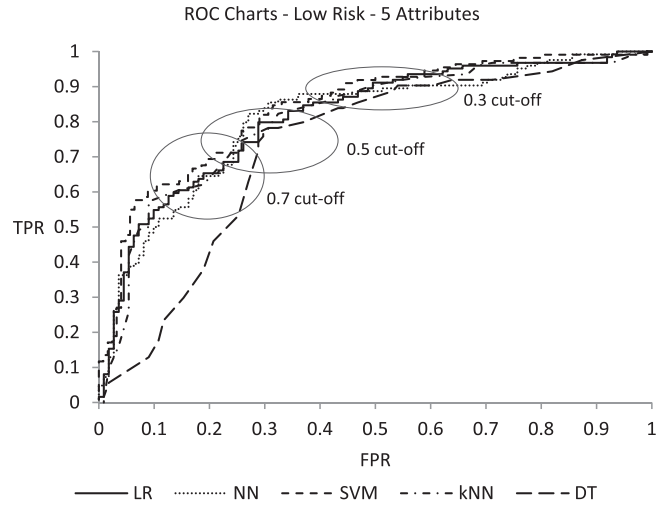

Fig. 2. The ROC charts for the low risk tasks for the 5 selected models built on 5 input variables.

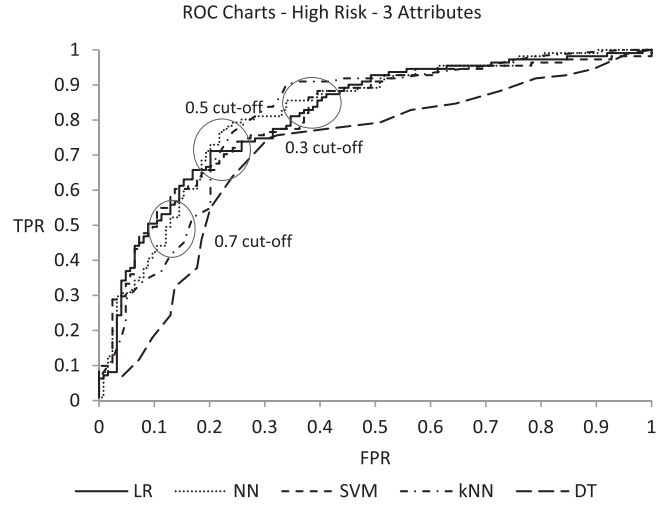

Fig. 3. The ROC charts for the high risk tasks for the 5 selected models built on 3 input variables.

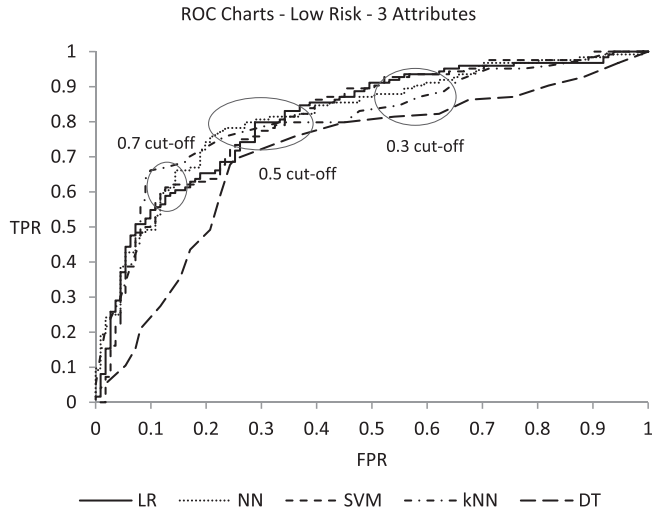

Fig. 4. The ROC charts for the low risk tasks for the 5 selected models built on 3 input variables.

significant differences between the performances of LR (73.8%), NN (72.6%), SVM (74.1%), and *k*NN (74.2%); while RBFNN (71.0%), DT (71.7%), and RF (70.3%) are classifying significantly worse than LR. For the models constructed on the data set with 4 variables, there is no significant difference between the performance of LR (73.3%), NN (72.9%), and SVM (72.4%), whereas three remaining models classify significantly worse than LR. Similar patterns are shown in the last row for the data set with 3 variables. Both the NN model (75.9%) and *k*NN (75.0%) significantly outperform LR (72.9%) in the overall rates. The rate analysis down the table columns reveals that only the NN model significantly benefited from the variable reduction. Its overall rate for the data set with 3 variables is 75.9%. The variable reduction, however, had a profound negative effect on the performance of SVM whose performance decreased for the data sets with 4 and 3 variables, and partially negative effect on the classification accuracy of LR and *k*NN. The variable reduction did not affect the DT and RF models. The standard deviations calculated over 100 rates are between 7.9% and 9.4%. Theoretically about two thirds of the 100 measures should be within a single standard deviation from the average, and 95% of the results should be within two standard deviations, and one should see very rarely results that are better or worse than 2 standard deviations from the average.

Table 2
The means and standard deviations of the overall correct classification accuracy rates [%].

Data set with	Models						
	LR	NN	RBF NN	SVM	*k*NN	DT	RF
5 Attributes	73.8	72.6	71.0^{ww}	74.1	74.2	71.7^{w}	70.3^{ww}
	8.9	9.0	9.3	9.4	9.0	8.4	8.9
4 Attributes	73.3	72.9	70.3^{ww}	$_{II}72.4$	$_{II}71.6^{ww}$	71.4^{w}	69.2^{ww}
	8.5	9.4	7.9	8.2	9.0	8.8	9.3
3 Attributes	$_{I}72.9$	$_{hh}75.9^{bb}$	70.9^{ww}	$_{I}72.8$	75.0^{bb}	72.2	68.9^{ww}
	8.6	8.4	8.5	8.7	8.2	9.1	8.5

^{b,bb} – significantly better than LR at $\alpha = 0.05$ and $\alpha = 0.01$, respectively.
^{w,ww} – significantly worse than LR at $\alpha = 0.05$ and $\alpha = 0.01$, respectively.
$_{h,hh}$ – significantly better than the full data set with 5 attributes at $\alpha = 0.05$ and $\alpha = 0.01$, respectively.
$_{l,ll}$ – significantly worse than the full data set with 5 attributes at $\alpha = 0.05$ and $\alpha = 0.01$, respectively.

One can note that a formula for computing t-value $t = \bar{d}/\sqrt{\bar{\sigma}_d^2/k}$ for a paired comparison includes the difference $\bar{d}$ of the mean rates generated by two models divided by square root of the variance of the differences normalized by the sample size k = 100. Thus, even though the difference between the mean rates for the two models (73.8–72.9 = 0.9, LR – the second column in Table 2) is small, t-test may still show the statistically significant difference in the classification performance of the 2 models. In other words, if one model consistently generates a smaller rate than another model, even by a small amount, it is very likely that t-test will show that the difference between the performance of the two models is statistically significant, especially at α = 0.05.

Table 3 presents the average correct classification accuracy rates for high risk jobs. For the original data set RF (71.6%) and NN (69.6%) significantly outperform LR (66.0%), whereas for the data set with 4 variables, NN (72.4%), RF (71.1%), and DT (70.1%) appear to be significantly better than LR (64.7%). However, for the data set with 3 variables, NN exhibits significantly highest rate of 79.1% followed by *k*NN (75.2%), DT (73.7%), and RF (71.2) than LR (64.1%). Feature reduction has a large positive effect on the classification performance of NN (79.1%), DT, and *k*NN and a very large negative effect on the performance of SVM. One can see that the average standard deviations for the high risk rates are high and more volatile, with the lowest and the largest being 12.0% and 17.4%, respectively. This pattern may be attributed to the smaller number of high-risk jobs than low risk jobs in each of the 100 training sets. This large spread in standard deviations testifies that building and testing models on a few selected partitions of the data set may indeed yield very high classification accuracy rates that exceed 95%.

Table 4 depicts the average correct classification accuracy rates for low risk tasks. For the original data set and the data sets with 4 and 3 variables, SVM stands out, exhibiting the rates of 83.0%, 84.2%, and 85.4%, respectively, significantly higher than the LR baseline (80.8%, 81.0%, and 80.7%). Down the table columns the data sets are compared. The variable reduction does not have much effect on LR. However, the performance of SVM significantly improves with feature reduction, while are methods are doing much worse. The standard deviations vary between 10.7% and 15.8% with SVM being the most stable.

Table 5 describes the average areas under ROC curves and their respective standard deviations. Analyzing the table across the rows one can see that for the data sets with 5 and 4 variables, SVM (83.6%, 82.0%) performs significantly better than LR (81.5%, 80.8%). For the data set with 3 variables, there is no clear superiority pattern and it appears that LR (81.1%), NN (82.0%), SVM (81.4%), and *k*NN (81.6%) perform equally well. If one looks at the rates down the columns there are 2 methods (NN and RBFNN) which somewhat benefited from feature reduction. The performance of 3 methods (LR, *k*NN, and DT) is statistically the same, whereas RF and SVM do significantly worse. The standard deviations across the methods and data sets are pretty stable and oscillate around 9.5%.

In a binary classification problem, where the response attribute has only two classes: true (event – high risk) and false (non-event – low risk), two different related error rates are of interest. These are false negative error (type I error) and false positive error (type II error). In the former, the actual event (high risk) is classified as non-event (low risk) and in the latter the actual non-event (low risk) is classified as event (high risk). The chosen 0.5 probability cutoff is correct when the cost of making type I error is equal to the cost of type II error. When making one type of error is more costly than another type of error, a different probability threshold is desired. For example, if committing the type I error (predicting a high risk task as a low risk task) is 2.3 times more costly than committing type II error (predicting a low risk task as a high risk task), a cutoff of 0.3 should be used. Thus, if a classifier produces a value $\geqslant$ 0.3, a task is classified as a high risk task; otherwise it is classified as a low risk task. Similarly, if committing the type II error (predicting a low risk task as a high risk task) is 2.3 times more costly than committing type I error (predicting a high risk task as a low risk task), a cutoff of 0.7 should be applied. In MLT jobs, making the type I error is of special significance as it is more serious and costly than the type II error. Consequently, committing this error may cause a serious low back injury which may require expensive treatment and/or paying subsequent disability benefits. The ROC charts allow one to look at the performance of the classifiers at specific cut-offs as well as the continuum of probability cut-offs from within the range [0,1] (Figs. 1–4).

The ROC curve represents the true positive rate (TPR) on the y axis and the false positive rate (FPR) on the x axis for any fixed cut-off value. Each point on the curve corresponds to a particular cut-off. To trade off TPR and FPR, one can select cutoff points on the curve. For example, one can tune a model by choosing a desired/appropriate value for the FPR on the x axis. If one tries to decrease/increase the FPR parameter of the model, it would increase/decrease the false negative rate and vice versa. In terms of model comparison, the ideal curve coincides with the vertical axis, i.e., the best curve is the one which pushes up and to the left. The extreme points (1,1) and (0,0) on a curve represent an ideal model which classifies all high risk or low risk samples correctly on the test set. The curve permits one to assess the performance of the model at various operating points (thresholds in a decision process using the available model) and the performance of the model as a whole (using as a parameter the area below the ROC curve). The area between the curve and the 45° line, which represents the worthless model not shown on the charts, gives so called the Gini index of performance: a number between 50% and 100%. The higher the area/number is, the better the overall model. For more details, see (Han & Kamber, 2001; Kantardzic, 2011; Witten & Frank, 2005).

The ROC charts are depicted in Figs. 1–4. The target events were detecting high risk or low risk tasks. The models were built on five, four, and three input attributes. To avoid clutter on the charts four best models (NN, SVM, *k*NN, and LR) and the worst one (DT) are presented. The approximate locations of the 0.3, 0.5, and 0.7 cut-points are shown on the charts. There exact locations vary and depend on the models. Lower cut-off points tend to be in the upper right areas of the curves, whereas higher cut-offs are in lower left areas. One can see that the global classification accuracy of DT is

Table 3
The means and standard deviations of the high risk correct classification accuracy rates [%].

Data set with	Models						
	LR	NN	RBF MN	SVM	*k*NN	DT	RF
5 Attributes	66.0	69.6^{bb}	61.0^{ww}	64.0^{ww}	66.9	62.9^{w}	71.6^{bb}
	15.1	17.4	15.1	15.3	13.5	14.8	14.5
4 Attributes	64.7	$_{h}72.4^{bb}$	60.8^{ww}	$_{ll}59.3^{ww}$	65.6	$_{hh}70.1^{bb}$	71.1^{bb}
	13.9	17.4	13.5	15.4	14.2	16.2	14.1
3 Attributes	$_{l}64.1$	$_{hh}79.1^{bb}$	$_{h}63.9$	$_{ll}58.6^{ww}$	$_{hh}75.2^{bb}$	$_{hh}73.7^{bb}$	71.2^{bb}
	14.3	12.9	14.4	15.0	12.0	14.4	13.4

Table 4
The means and standard deviations of the low risk correct classification accuracy rates [%].

Data set with	Models						
	LR	NN	RBF MN	SVM	*k*NN	DT	RF
5 Attributes	80.8	75.2^{ww}	79.9	83.0^{bb}	80.7	79.5	69.3^{ww}
	13.1	15.1	13.3	11.9	12.7	15.8	14.3
4 Attributes	81.0	$_{l}73.5^{ww}$	78.9^{w}	$_{h}84.2^{bb}$	$_{ll}77.1^{ww}$	$_{ll}72.6^{ww}$	67.6^{ww}
	12.5	15.1	12.6	10.9	13.2	14.9	14.5
3 Attributes	80.7	$_{l}72.9^{ww}$	$_{ll}77.2^{ww}$	$_{hh}85.4^{bb}$	$_{ll}74.9^{ww}$	$_{ll}70.9^{ww}$	$_{l}66.9^{ww}$
	12.4	13.7	12.8	10.7	13.0	15.2	14.0

Table 5
The means and standard deviations of the areas under ROC curves [%].

Data set with	Models						
	LR	NN	RBF NN	SVM	*k*NN	DT	RF
5 Attributes	81.5	80.6	77.3^{ww}	83.6^{bb}	81.1	71.6^{ww}	78.8^{ww}
	9.6	9.5	10.3	9.0	9.0	10.6	9.3
4 Attributes	80.8	80.9	77.1^{ww}	$_{ll}82.0^{bb}$	80.6	72.0^{ww}	$_{ll}76.4^{ww}$
	9.8	8.9	9.3	9.1	9.0	9.9	9.3
3 Attributes	81.1	$_{h}82.0$	$_{hh}79.2^{ww}$	$_{ll}81.4$	81.6	72.8^{ww}	$_{ll}74.5^{ww}$
	9.5	8.9	9.5	9.7	8.8	9.1	9.4

significantly worse than the four remaining models at all cut-off points, whereas the differences between the performances of the four best models can be profound at different cut-offs. The consistent poor performance of DTs is somewhat surprising as they are tools that can be easily interpreted; DTs encode knowledge they learn in simple to understand if-then rules. The areas under ROC charts are reported in Table 5.

4. Conclusion

Building classification models that could effectively discriminate between high risk and low risk MLTs that contribute to LBDs is a very challenging topic which has received a great deal of attention in the last two decades. This study discussed several recent papers that built and validated the classification models using the experimental Marras data set. Seven models: LR, NN, RBFNN, SVM, *k*NN, DT, and RF were constructed and tested, two of which (SVM and RBFNN) have not been used in this context yet. Computer simulation was first performed on the full set of five variables. Then variable reduction techniques were performed to eliminate attributes with the least predictive power. Consequently, we built and tested the models on four and three input variables as well. To obtain true and reliable classification accuracy estimates for the models we used the 10-fold cross-validation experiment and ran it 10 times. We than averaged these estimates over 10 folds and 10 runs. We employed a 2-tailed paired *t*-test to see if the performances of the models across the data sets and the data sets across the models were significantly different from LR and the data sets with 5 variables which were used as the baselines. The parameters for the models have been calibrated for the best performance using the optimization functions built-in in Weka.

Though the classification accuracy rates presented in this paper are better than those in the NIOSH guides, (1981) and NIOSH guides (1991) as well as those reported by Zurada, Karwowski, and Marras (1997, 2004), especially in detecting high risk tasks, they are generally much less optimistic than those presented in several mentioned studies (Akay, 2011; Akay & Toksari, 2009; Asensio-Cuesta et al., 2010; Chen et al., 2000, 2004). The approach to building and testing classifiers for detecting LBDs presented in this study is sound and systematic. As a result, the reported classification accuracy estimates for the models are more realistic as they are not partition or sample dependent. The SVM, NN, and *k*NN models exhibited the best classification ability and they appeared to be significantly better than the other four models. In terms of the overall and high risk classification accuracy rates, the NN model based on three variables turned out to be the best (75.9% and 79.1%), whereas SVM outperformed all other models in classification of low risk jobs (85.4%). The poor overall performance of DT and RF was somewhat surprising as these tools offer the best interpretability; the if-then rules they produce are easy to understand and explain. This study also analyzed ROC charts and the areas under them which allow one to evaluate the global performance of the models at a continuum of probability cut-offs from within the range [0,1]. This study shows that the Marras data set exhibits limited performance in correctly classifying low and high risk MLTs that contribute to LBDs.

The findings of this study indicate that more elaborate and dynamic set of experiments and solutions needs to be implemented to understand the nature and phenomenon of LBDs better and build more effective classifiers that could distinguish between high risk and low risk jobs. Such studies have already been undertaken by Karwowski et al. (2006) and Hou, Zurada, Karwowski, Marras, and Davis (2007a), Hou, Zurada, Karwowski, Marras, and Davis (2007b). The authors implemented soft computing methodologies such as recurrent fuzzy neural networks, fuzzy relational rule network, and fuzzy average with fuzzy cluster distribution to identify key variables, estimate dynamic spinal forces, and model electromyographical activity of trunk muscles in MLTs. Hopefully, the continuation of these studies will lead to more effective detection of the risk of LBDs due to MLTs.

References

Agresti, A. (1992). *Categorical data analysis*. Hoboken, NJ: John Wiley & Sons.

Akay, D., Akcayol, M. A., & Kurt, M. (2008). NEFCLASS based extraction of fuzzy rules and classification of risks of low back disorders. *Expert Systems with Applications, 35*. 2107-2012.

Akay, D., & Toksari, M. D. (2009). Ant colony optimization approach for classification of occupational low back disorder risks. *Human Factors and Ergonomic in Manufacturing, 19*(1), 1–14.

Akay, D. (2011). Grey relational analysis based on instance based learning approach for classification of risk of occupational low back disorders. *Safety Science, 49*, 1277–1282.

Asensio-Cuesta, S., Diego-Mas, J. A., & Alcaide-Marzal, J. (2010). Applying generalized feed-forward neural networks to classifying industrial jobs in terms of risk of low back disorders. *International Journal of Industrial Ergonomics, 40*, 629–635.

Ayoub, M. M., Karwowski, W., & Dempsey, P. G. (1996). Manual materials handling. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (2nd Ed. New York: John Wiley.

Battiti, R. (1992). First- and second-order methods for learning: Between steepest descent and Newton's method. *Neural Computation, 4*(2), 141–166. 10.1162/neco.1992.4.2.141.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Chandna, P., Deswal, S., & Pal, M. (2010). Semi-supervised learning based prediction of musculoskeletal disorder risk. *Journal of Industrial and Systems Engineering, 3*(4), 291–295.

Chen, C.-L., Kaber, D. B., & Dempsey, P. G. (2000). A new approach to applying feed-forward neural networks to the prediction of musculoskeletal disorder risk. *Applied Ergonomics, 31*, 269–282.

Chen, C.-L., Kaber, D. B., & Dempsey, P. G. (2004). Using feed-forward neural networks and forward selection of input variables for an ergonomics data

classification problem. *Human Factors and Ergonomics in Manufacturing, 14*(1), 31–49.
Christensen, R. (1997). *Log-linear models and logistic regression*. New York: Springer-Verlag.
Guidici, P. (2003). *Applied data mining: Statistical methods for business and industry*. Wiley.
Han, J., & Kamber, M. (2001). *Data mining: Concepts and techniques*. San Francisco, CA: Morgan Kaufmann Publishers.
Hou, Y., Zurada, J. M., Karwowski, W., Marras, W. S., & Davis, K. (2007a). Identification of key variables using fuzzy average with fuzzy cluster distribution. *IEEE Transactions on Fuzzy Systems, 15*(4), 673–685.
Hou, Y., Zurada, J. M., Karwowski, W., Marras, W. S., & Davis, K. (2007b). Estimation of the dynamic spinal forces using a recurrent fuzzy neural network. *IEEE Transactions on Systems, Man, and Cybernetics: Part B, 37*(1), 100–109.
Kantardzic, M. (2011). *Data mining: Concepts, models, methods, and algorithms*. IEEE Press/Wiley.
Karwowski, W., Hancock, P., Zurada, J., & Ostaszewski, K. (1991). Risk of low back overexertion injury due to manual load lifting in view of the catastrophe theory. In Y. Queinnec & F. Daniellou (Eds.), *Designing for Everyone* (pp. 66–68). London: Taylor and Francis.
Karwowski, W., Ostaszewski, K., & Zurada, J. (1992). Applications of catastrophe theory in modeling the risk of low back injury in manual lifting tasks. *Le Travail Humain, 55*, 259–275.
Karwowski, W., Zurada, J., Marras, W. S., & Gaddie, P. (1994). A prototype of the artificial neural network-based system for classification of industrial jobs with respect to risk of low back disorders. In F. Aghazadeh (Ed.), *Proceedings of the Industrial Ergonomics & Safety Conference* (pp. 19–22). London: Taylor & Francis.
Karwowski, W., Gaweda, A., Marras, W. S., Davis, K., Zurada, J. M., & Rodrick, D. (2006). A fuzzy relational rule network modeling of electromyographical activity of trunk muscles in manual lifting based on trunk angles, moments, pelvic tilt, and rotation angles. *International Journal of Industrial Ergonomics, 35*, 847–859.
Liberty Mutual Workplace safety index of leading occupational injuries (2004).
Marras, W. S., Lavender, S. A., Leurgans, S., Sudhakar, L. R., Allread, W. G., Fathallah, F., et al. (1993). The role of dynamic three-dimensional trunk motion in occupationally-related low back disorders. *Spine, 18*, 617–628.
Marras, W. S., Fine, L., Ferguson, S., & Waters, T. (1999). The effectiveness of two types of common lifting measures for the control of low back disorders in industries. *Ergonomics, 42*(1), 229–245.
Mitchell, T. M. (1997). Machine Learning. Boston, MA: WCB/McGraw-Hill.
Moller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks, 6*, 523–533.
National Safety Council (1990). Accident Statistics. Chicago, IL.
Nelson, N. A., & Hughes, R. E. (2009). Quantifying relationships between selected work-related risk factors and back pain: a systematic review of objective biomechanical measures and cost-related health outcomes. *International Journal of Industrial Ergonomics, 39*(1), 202–210.
National Institute for Occupational Safety and Health (NIOSH), (1981). Work practices guide for manual lifting, DHHS (NIOSH) Pub. No. 81-122, Cincinnati, OH, US Department of Health and Human Services.
NIOSH musculoskeletal disorders and workplace factors (1997), Cincinnati, Ohio: U.S. Department of Health and Human Services, DHHS Publication No., 97-141.
Park, J., & Sandberg, J. W. (1991). Universal approximation using radial basis functions network. *Neural Computation, 3*, 246–257.
Poggio, T., & Girosi, F. (1990). Networks for approximation and learning. *Proceedings IEEE, 78*(9), 1481–1497.
Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man–Machine Studies, 27*, 221–234.
Riihimaki, H. (1991). Low-back pain, its origin and risk indicators. *Scandinavian Journal of Work, Environment & Health, 11*, 81–90.
Spengler, D. M. J., Bigos, S. J., Martin, N. A., Zeh, J., Fisher, L., & Nachemson, A. (1986). Back injuries in industry: a retrospective study. *Spine, 11*, 241–256.
Svensson, H.-O., & Andersson, G. B. J. (1989). The relationship of low-back pain, work history, work environment and stress: a retrospective cross-sectional study of 38–64-year old women. *Spine, 14*, 517–522.
Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining*. Pearson Addison Wesley.
Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.
Waters, T. R., Putz-Anderson, V., Garg, A., & Fine, L. J. (1993). Revised NIOSH equation for the design and evaluation of manual lifting tasks. *Ergonomics, 36*, 749–776.
Waters, T. R., Putz-Anderson, V., & Garg, A. (1994). Application Manual for the Revised NIOSH Lifting Equation, US Department of Health and Human Services, Cincinnati, OH.
Witten, I. H., & Frank, E. (2005). *Data mining: Practical learning tools and techniques*. Morgan Kaufman.
Zurada, J., Karwowski, W., & Marras, W. S. (1997). A neural network-based system for classification of industrial jobs with respect to low back disorders due to workplace design. *Applied Ergonomics, 28*(1), 49–58.
Zurada, J., Karwowski, W., & Marras, W. S. (2004). Classification of jobs with risk of low back disorders by applying data mining techniques. *Occupational Ergonomics, 4*(4), 291–305.
Zurada, J. (2012). Predicting Low Back Disorders Due to Manual Handling Tasks. In R. Sprague (Ed.), *Proceedings of the 45th Hawaii international conference on system sciences*, IEEE Computer Society Press.