Information Sciences 516 (2020) 401–418

Contents lists available at ScienceDirect

Information Sciences

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

DGHNL: A new deep genetic hierarchical network of learners for prediction of credit scoring

Paweł Pławiak[a,b,*], Moloud Abdar[c], Joanna Pławiak[b,d], Vladimir Makarenkov[c], U Rajendra Acharya[e,f,g]

[a] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland
[b] Department of Information and Communications Technology, Faculty of Computer Science and Telecommunications, Cracow University of Technology, Warsaw 24 st., F-3, Krakow 31-155, Poland
[c] Department of Computer Science, University of Quebec in Montreal, Montreal, QC H2X 3Y7, Canada
[d] Department of Biocybernetics and Biomedical Engineering, Faculty of Electrical Engineering, Automatics, Computer Science, and Biomedical Engineering, AGH University of Science and Technology, Krakow 30-059, Poland
[e] Department of Electronics and Computer Engineering, Ngee Ann Polytechnic, Clementi 599489, Clementi 599491, Singapore
[f] Department of Biomedical Engineering, School of Science and Technology, SUSS University, Clementi 599491, Singapore
[g] International Research Organization for Advanced Science and Technology (IROAST), Kumamoto University, Kumamoto, Japan

ARTICLE INFO

Article history:
Received 29 May 2019
Revised 20 November 2019
Accepted 22 December 2019
Available online 27 December 2019

Keywords:
Credit scoring
Machine learning
Data mining
Ensemble learning
Deep learning
Genetic algorithm
Feature extraction and selection

ABSTRACT

Credit scoring (CS) is an effective and crucial approach used for risk management in banks and other financial institutions. It provides appropriate guidance on granting loans and reduces risks in the financial area. Hence, companies and banks are trying to use novel automated solutions to deal with CS challenge to protect their own finances and customers. Nowadays, different machine learning (ML) and data mining (DM) algorithms have been used to improve various aspects of CS prediction. In this paper, we introduce a novel methodology, named Deep Genetic Hierarchical Network of Learners (DGHNL). The proposed methodology comprises different types of learners, including Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Probabilistic Neural Networks (PNN), and fuzzy systems. The Statlog German (1000 instances) credit approval dataset available in the UCI machine learning repository is used to test the effectiveness of our model in the CS domain. Our DGHNL model encompasses five kinds of learners, two kinds of data normalization procedures, two extraction of features methods, three kinds of kernel functions, and three kinds of parameter optimizations. Furthermore, the model applies deep learning, ensemble learning, supervised training, layered learning, genetic selection of features (attributes), genetic optimization of learners parameters, and novel genetic layered training (selection of learners) approaches used along with the cross-validation (CV) training-testing method (stratified 10-fold). The novelty of our approach relies on a proper flow and fusion of information (DGHNL structure and its optimization). We show that the proposed DGHNL model with a 29-layer structure is capable to achieve the prediction accuracy of 94.60% (54 errors per 1000 classifications) for the Statlog German credit approval data. It

* Corresponding author.
E-mail addresses: pplawiak@iitis.pl, plawiak@pk.edu.pl (P. Pławiak), m.abdar1987@gmail.com (M. Abdar), joanna.plawiak@gmail.com (J. Pławiak), makarenkov.vladimir@uqam.ca (V. Makarenkov), aru@np.edu.sg (U.R. Acharya).

https://doi.org/10.1016/j.ins.2019.12.045

is the best prediction performance for this well-known credit scoring dataset, compared to the existing work in the field.

1. Introduction

Financial risk management is one of the most sensitive subjects that should be investigated taking into account several important factors. Bank industry includes many types of risk affecting both banks and their customers. Credit scoring (CS) has a close relationship to banks being an effective and crucial approach for analyzing the borrowing and lending of money [8]. It is important to collect information from bank customers and other financial institutions to manage the financial risks, and at the same time, to reach an important decision to lend some money to their clients or not. In other words, this process can help to separate good borrowers from bad ones. This means that some borrowers have clean and good records; therefore, banks can classify them as "good borrowers". A few others, not having such good records, can be considered as "bad borrowers". It is worth noting that such simple selection process may not guarantee a correct classification. Hence, new accurate automated systems reducing the prediction errors are urgently needed in order to handle large and complex CS datasets.

Nowadays, computer-based systems have become extensively popular among researchers and organizations to deal with this challenge. In other words, these machine-based systems represent faster and cheaper prediction tools which become more and more accurate. The credit scoring is not a single-step process. It is a periodical process. It should be noted that the impact of humans cannot be ignored during this process. Even though all of these systems are human-made, their performance should be carefully controlled and verified by humans. However, this is not always the case during automated test response verification by OCR (Optical Character Recognition). Even though humans usually control machines, in most cases the computational power of machines is much higher than that of human beings. Thus, a machine with high performance can be helpful to improve credit scoring models. In other words, an accurate system is highly essential to predict possible relationships between lenders and their borrowers.

In the last decades, several studies have tried to assess credit scoring potential of bank clients using different predictive models [3]. A vast number of ML (Machine Learning) techniques have been used for this purpose, including support vector machines (SVM), neural networks, decision trees (DT), logistic regression, fuzzy systems, etc. Each of these studies has analyzed different datasets to show the effectiveness of their methods. Generally, finding a relationship between low and high credit risks is one of the popular research areas in financial forecasting, consisting in developing new predictive systems. Thus, the main goals of this study can be summarised as follows:

- To construct a deep multilayer-based structure providing a fast, efficient and accurate training approach for different learners.
- To verify the effectiveness of the proposed methodology by applying it to a well-known CS dataset.
- To develop a new effective fusion-based and cascaded ensemble learning optimizer based on an evolutionary computation technique (genetic algorithm).

As to the main contribution and novelty of this work, we introduce a new Deep Genetic Hierarchical Network of Learners (DGHNL) system, characterized by the four following approaches: (1) genetic layered training, (2) different machine learning algorithms used at the same time, (3) hierarchical neural network structure, and (4) deep learning-based structure. We propose a 29-layer network of learners, based on fusion of several machine learning (ML) techniques including: 10-fold cross-validation (CV), deep learning, ensemble learning, layered learning technique, supervised training, genetic algorithm (GA) as a feature selection method, optimization of the learners' parameters based on GA, and finally, genetic layered training approach for the selection of learners.

The proposed DGHNL system has a fusion-based 29-layer structure including five ML algorithms (two types of SVM algorithms, kNN, PNN, as well as a fuzzy system), two normalization techniques, two feature extraction approaches, three kernel functions, and three parameter optimization techniques based on three types of error calculation. In this work, we explore the well-known Statlog German dataset (1000 instances) [12]. However, the proposed methodology could be applied to any other CS dataset as well.

The performance of traditional machine learning algorithms is a benchmark in this domain. For this reason, improving the performance of classical algorithms is one of the major goals of many studies. Combination of such simple algorithms is a common solution to enhance the model performance. The main motivation for this study is to use an ensemble learning technique with many layers (a 29-layer structure was used in our study) to boost the performance of the CS prediction system. It has been shown that the application of evolutionary algorithms (EA) help to improve the performance of classical ML methods [28]. Hence, we have included at least one EA in our final model. Also, we used a genetic algorithm (GA) to train the learners at different layers of the model using an ensemble learning technique

The rest of the study is structured as follows. In Section 2, we reviewed the relevant papers related to CS prediction using various ML algorithms. Section 3 provides a brief research background. The proposed methodology is discussed in Section 4. The obtained results are presented in Section 5. Finally, the paper concludes by Section 6.

2. Literature review

Recently, CS-based selection methods have become very popular in the bank sector. Due to its importance, many researchers are trying to propose new approaches to decrease the risk rates. Consequently, many studies have primarily focused on various aspects of CS using different analytical solutions. For instance, ML/DM (Data Mining) techniques are widely used to assess CS. This section briefly discusses relevant work in the field.

Generally, there are two kinds of missing data in the context of credit scoring, including data missing at random (MAR) and missing not at random (MNAR) [9]. The MNAR represents the class of results that depend on the features of borrowers as well as on some unobserved features [39]. Generally, the ideal data should include information on all applicants, i.e., either accepted applicants or rejected applicants, to create accurate credit models [39]. However, in most of practical cases, we can access only the records of accepted applicants, while very little information related to the rejected applicants is available [4]. This is one of the main disadvantages of the current CS databases. Bucker et al. [4] showed that the performance of predictive models can be improved when the information on the rejected category is also available. This means that the ignorance of the rejected applicants data is not suitable to deal with challenging statistical and economic concerns.

In [39], a novel ML/DM approach was presented to investigate the reject inference. The authors introduced a kernel-free fuzzy quadratic surface support vector machines model. Chang et al. [5] applied an artificial immune network, called AINE-based classifier, to carry out a credit scoring analysis. Moreover, a comprehensive review was presented by [16] to investigate the application of statistical and intelligent methods to the bankruptcy prediction problem from 1968 to 2005. These methods can be categorized into 9 main groups. They are as follows:

- statistical techniques,
- neural networks (NNs),
- case-based reasoning,
- decision trees (DTs),
- operational research,
- evolutionary approaches,
- rough set based techniques,
- other techniques: fuzzy logic, support vector machine (SVM) and isotonic separation,
- soft computing.

Ensemble learning (EL) is a well-known approach playing a remarkable role in optimizing the performance of machine learning and data mining algorithms [32]. These methods combine different traditional classifiers in order to provide more accurate and better performing models. EL-based methods have been widely used in a variety of applications such as medical science, energy, marketing, destination prediction, image classification, sentiment analysis, time series prediction, and credit scoring [20,21,28,30]. These studies showed that ensemble learning techniques are efficient methods for optimization purposes. Soto et al. [37] applied EL with interval type-2 fuzzy neural networks (IT2FNN) to analyze time series data. In another study by the same research group [36], the ANFIS ensemble model with a genetic algorithm having type-1 fuzzy and interval type-2 integrators was used to predict chaotic time series.

Zhang et al. [48] proposed a hybrid-based method to study CS. In this regard, an ensemble-based system was introduced using five classifiers of credit scoring (support vector machine (SVM), logistic regression (LR), neural network (NN), gradient boosting decision trees (GBDT) and finally random forest(RF)). Furthermore, a GA was used to choose the classifiers. The proposed model was called CF-GA-Ens. It was used to investigate three credit scoring datasets: Australian, German, and Japanese CS data. In another study, Abelln and Castellano [1] showed that ensembles of techniques provided better results in terms of CS prediction. To this end, ensembles of classifiers were applied on various datasets, including Australian, German, Polish, Iranian, University of California, San Diego (UCSD), and Japanese CS data. Chen et al. [6] proposed an ensemble-based model using a logistic algorithm, backpropagation (BP) neural networks and the Adaboost algorithm (named Logistic-BP-AdaBoost model: A-L-B) for forecasting credit risks. Tripathi et al. [40] applied a feature reduction approach with ensemble classification. These authors combined the multilayer feedforward neural networks (MLFFNN), naive Bayes (NB), quadratic discriminant analysis (QDA), distributed time delay neural network (DTNN), and time delay neural network (TDNN) methods using the majority voting and weighted voting ensemble learning techniques.

A new CS model proposed in [41] is based on the weighted voting and cluster-based feature selection. An extensive comparative performance of individual and ensemble classifiers was studied in [34]. Setiono et al. [33] introduced a new approach to select the samples by applying an ensemble of neural network for CS prediction. Ala'raj and Abbod [2] investigated CS data using a new hybrid ensemble model. The proposed hybrid model was based on multivariate adaptive regression splines (MARS) and Gabriel neighbourhood graph editing (GNG), as a new complex combiner named ConsA. A new heterogeneous ensemble CS model based on the bstacking approach was proposed by Xia et al. [45]. This model combined the bagging technique with the stacking approach.

A cascade ensemble contains several ensemble learning techniques [49]. This methodology can remarkably enhance the performance of classical ensemble learning techniques. Zhang et al. [49] showed the effectiveness of their cascaded ensemble classifier system for the recognition of handwritten digits. Xu et al. [46] applied cascade ensemble learning (called CELearning) to model human activity recognition (HAR), obtaining some outstanding results. Labao et al. [17] proposed cascaded ensemble architecture of deep learning to detect automatically fish underwater. In the study of Riordon et al. [31], a cascade ensemble of several support vector machines (CE-SVMs) was used to classify sperm samples into several common shape-based categories, suggested by the World Health Organization (WHO).

The base learners used in our work are support vector machines (SVM), k-nearest neighbors (kNN), probabilistic neural networks (PNN) and fuzzy systems (FS).

3. Brief research background

In this work, we have applied several well-known algorithms and optimization techniques. The innovative elements of our study are summarized below: genetic layered training, boost diversity, a new hierarchical and deep learning structure. The main steps involved are as follows:

Step 1: Normalization: we used z-score normalization (i.e., data standardization with mean data value = 0 and data standard deviation = 1) and data rescaling to the range [0,1] approaches in this step.
Step 2: Feature extraction: two types of feature extractions were applied: no extraction, and principal component analysis (PCA).
Step 3: Feature selection: no selection and genetic selection (genetic algorithm (GA) for feature selection) were used. The parameters of GA are indicated in Table 1.
Step 4: Cross-validation: stratified 10-fold cross validation approach was applied.
Step 5: Machine learning algorithms: five machine learning algorithms including different kinds of SVM (C-SVC, nu-SVC), probabilistic neural network (PNN), k-nearest neighbors (kNN), and the Takagi-Sugeno fuzzy system, were used.
Step 6: Optimization of parameters: GA was used to optimize the parameters of learners. Three kinds of error calculation were used: (1) acceptance features coefficient + sum of errors in training and testing groups, (2) sum of percentage errors in training and testing groups, (3) sum of errors in testing groups.

Since the improvement of the CS prediction performance was the main goal of our study, we have designed a more accurate prediction system, based on techniques that increase the accuracy of classical ML methods, such as deep learning, ensemble learning, and evolutionary computation. The ensemble learning techniques combine classic algorithms to strengthen the performance of the entire system. These techniques have been widely applied in different domains and showed a greater accuracy as compared to single classical algorithms [22–24,38]. The most important data mining techniques used in ensemble learning include Bagging (Bootstrap aggregation), Boosting (AdaBoost), Stacked Generalization (Stacking), Random Forest, and Mixtures of Experts.

4. Methodology

A general view of the proposed DGHNL (Deep Genetic Hierarchical Network of Learners) system is discussed in this section.

4.1. Performance evaluation metrics

Different evaluation metrics can be used to evaluate the performance of ML/DM techniques. They include sensitivity, specificity, precision, Matthews correlation coefficient (*MCC*), sum of errors, *F*1 score, *AUC*, and accuracy. In this study we used two of them: (1) Accuracy (*ACC*) and (2) Sum of errors ERR_{sum} [28].

$$ACC = \left(\sum_{i=1}^{N} \frac{TP + TN}{TP + FP + TN + FN} \right) \times 100\% / N \tag{1}$$

where:

N is the number of sets considered during the CV prediction (10-fold, stratified),
TP is the number of True Positives,
TN is the number of the True Negatives,
FP is the number of the False Positives, and
FN is the number of the False Negatives.

Moreover, using the confusion matrix the sum of errors (ERR_{sum}) can be also computed. The ERR_{sum} is the total number of incorrect classifications.

Table 1
Detailed information about genetic algorithm in the DGHNL system.

Feature selection and learner parameters optimization	
Combination of the CV approach (10-fold, stratified) and the *GA* for feature selection process as well as learner parameters optimization	
Genetic Algorithm	• The population of individuals size: 500; • The gene representation kind: vectors of floating-points; • Chromosome structure for an individual, for example, for *SVM* learner, kind: $nu-SVC$, kernel function kind: *RBF*, vector of floating-point of the construction $[g_1, g_2, f_1, \ldots, f_{96}]$, where g_1 – the first gene, specifies the first parameter value, γ, g_2 – the second gene, specifies the second parameter value, ν, and $f_1, \ldots, f_{96}$ – 96 genes (values in the range of [0,1], specifies the selection of features, rounded to the values: a) 0 – rejected feature and b) 1 – accepted feature. For the other learner kinds: *SMV C-SVC, PNN, kNN, Fuzzy system* and other parameters the chromosome contains appropriate genes number - g, that specifies the values of optimized parameters; • Preliminary population for each generation: uniform, random; • The range selection of the gene values related to the preliminary population: local range of gene values related to each learner parameter (see Table 2 - *Optimized parameters section*), values selected experimentally based on global (wider) range. For feature selection, range = [0,1]; • The targeted value (target value) of the fitness function: 0; • Selection of the number of generations (maximum number) for each layer: 30 for the first layer and 50 for the layers 2 to 29; • Crossover type: intermediate for first and 29th layer, and scattered for the layers 2 to 28; Crossover probability: 0.7; • Mutation type and its probability: uniform and 0.1; • The number of best survival individuals without any change: 5; • The scaling method of the fitness function value: ranking; • Method for selection of parents: tournament; • The equation for calculating the fitness function: Sum of errors in testing groups: $ERR_T = \frac{err_{Lsum}}{1000} + err_{Tsum}$ (3) Sum of percentage errors in testing and training groups: $ERR_{\%} = (err_{L\%} + err_{T\%})/2$ (4) Sum of errors in testing and training groups along with the acceptance features coefficient: $ERR_{sum} = err_{Lsum} + err_{Tsum} + \frac{F_a}{F}$ (5) where: err_{Lsum} – sum of errors in all 10 training groups; err_{Tsum} – sum of errors in all 10 testing groups; $err_{L\%}$ – error percentage in all 10 training groups (sum of errors in all training groups to sum of all instances in all training groups: $err_{L\%} = \frac{err_{Lsum}}{L_{sum}} \cdot 100\%$); $err_{T\%}$ – error percentage in the 10 testing groups (sum of errors in all testing groups to sum of all instances in all testing groups: $err_{T\%} = \frac{err_{Tsum}}{T_{sum}} \cdot 100\%$); $\frac{F_a}{F} = C_F$ – acceptance of the coefficient of features (formula 2 in Section 4.1);

The acceptance feature coefficient (C_F) is the ratio of the number of accepted features F_a and the sum of all features F, given as percentage. In addition, the effectiveness of genetic feature selection can be estimated using the following formula:

$$C_F = \frac{F_a}{F} \times 100\%, \tag{2}$$

where:

F_a is the number of accepted features, and
F is the sum of all features.

4.2. General view of the proposed model

This study introduces a new machine learning-based approach to analyze the borrowers data. Generally, the proposed methodology includes 6 major steps (normalization, extraction of features, selection of features, CV technique, ML methods, and optimization of the learners parameters). Fig. 1 shows a schematic view of our model. In the following section, we briefly explain each of these steps.

4.3. The proposed methodology

The motivation to design the DGHNL system derives from the structure and functioning of the human brain (layered structure, bipolar signals, various connections between nodes, similarly processing information in nodes/neurons etc.).

Philosophy of the DGHNL system. Major characteristics of the system are as follows:

- Learners are considered as neurons connected in a network.

Table 2
Detailed information about learners in the DGHNL system.

Learners	
1 layer 108 optimized, trained, and tested learners - *experts*: two learner kinds (C-SVC or nu-SVC) · three kernel function kinds · two normalization kinds · two extraction of features kinds · three error calculation kinds and 3 learners (kNN, PNN, Fuzzy system) · two normalization kinds · two extraction of features kinds · three error calculation kinds	
2–26 layer 45 (nine for C-SVC answers + nine for nu-SVC + nine for kNN + nine for PNN + nine for Fuzzy system) optimized, trained, and tested learners - *judges*: one learner kind (nu-SVC) · three kernel function kinds · three error calculation kinds	
27–29 layer nine optimized, trained, and tested learners - *judges*: one learner kind (nu-SVC) · three kernel function kinds · three error calculation kinds	
Basic parameters	
SVM	• Type of the learner: nu-SVC or C-SVC; • Kind of kernel function: RBF (radial-Gaussian) or polynomial or sigmoid; • Selection of one output from the set of {1, 2};
PNN	• The type of the activation function: a radial activation function (Gaussian) - competition; • Algorithm for training: using distance for mapping of training set; • Objective function calculator: the sum of square errors (SSE); • Topology (neurons): 24 (number of borrowers' attributes) – 900 – 2; Biases: 1 – 0; • Selection of two outputs from the set of {0, 1};
kNN	• Metric for calculating the distance: *Minkowski*; • Selection of one output from the set of {1, 2};
Fuzzy system	• Type: Sugeno; • Selection of one output from the set of {1, 2};
Optimized parameters A broader range was used to find out the experimentally final parameter ranges	
SVM	• For nu-SVC kind only: the width of each margin will be defined by using the parameter ν ($-n$) from the range [0.005; 0.6] for RBF (radial basis function), polynomial, and sigmoid kernel functions. The resolution is 10^{-14}: for the first layer = 1500 values ($30 \cdot 500$) or for 2–29 layer = 25,000 values ($50 \cdot 500$); • For C-SVC type only: the parameter *cost* ($-c$) defines margins width from range: [0.1; 10] for kernel functions: RBF and polynomial, or [0.1; 20] for kernel function - sigmoid. The resolution is 10^{-14}: for the first layer = 15,000 values ($30 \cdot 500$) or for 2–29 layer = 25,000 values ($50 \cdot 500$); • The *degree* parameter ($-d$) defines range of kernel function spread: [0.01; 100] only for polynomial kernel function. The resolution is 10^{-14}: for the first layer = 15,000 values ($30 \cdot 500$) or for 2–29 layer = 25,000 values ($50 \cdot 500$); • The γ parameter ($-g$) defines range of kernel function spread: [0.001; 0.1] for polynomial kernel function, or [0.01; 100] for RBF kernel function, or [0.001; 1] for sigmoid kernel function. The resolution is 10^{-14}: for the first layer = 15,000 values ($30 \cdot 500$) or for 2–29 layer = 25,000 values ($50 \cdot 500$); • The parameter *coef*0 ($-r$) defines range of kernel function spread: [0.01; 5] for polynomial kernel function, or [0.01; 10] for sigmoid kernel function. The resolution is 10^{-14}: for the first layer = 15,000 values ($30 \cdot 500$) or for 2–29 layer = 25,000 values ($50 \cdot 500$);
PNN	• The *spread* parameter defines spread of RBF, for kernel of network, from the range [0.01; 10]. The resolution is 10^{-14}, $500 \cdot 30 = 15000$ values;
kNN	• The *number of nearest neighbors* parameter related to the range of [1; 100]. The resolution is 10^{-14}, $500 \cdot 30 = 15000$ values; • The calculation of the Minkowski distance will be affected by *exponent* parameter related to the range of [0.1; 1000]. The resolution is 10^{-14}, $500 \cdot 30 = 15000$ values;
Fuzzy system	• The *radii* parameter can be considered as a vector which a cluster centers range of influence will be determined in each of the data dimensions, from the range of [0.01; 100]. The resolution is 10^{-14}, $500 \cdot 30 = 15000$ values;

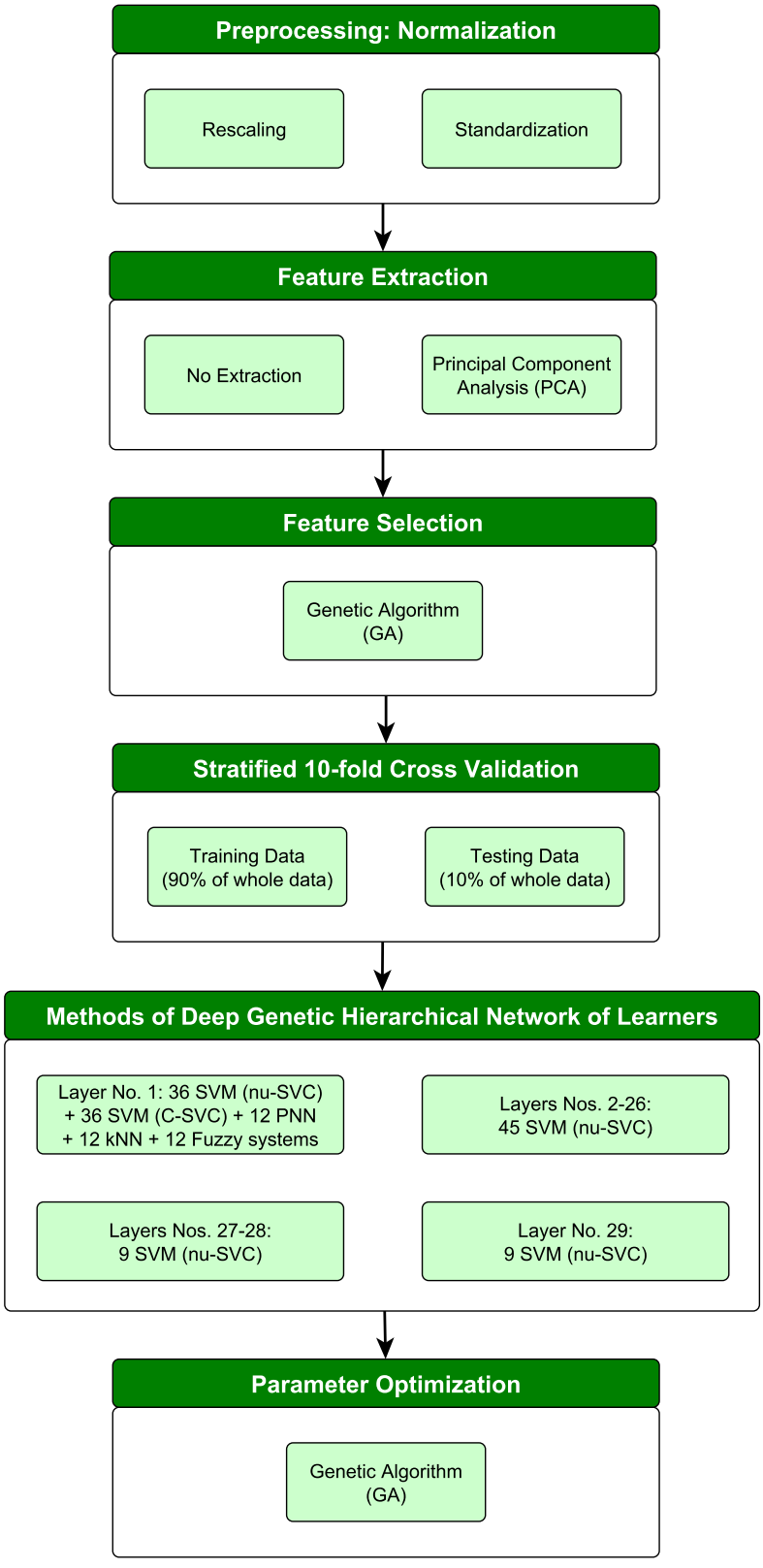

Fig. 1. A schematic view of our model applied to the German credit scoring data.

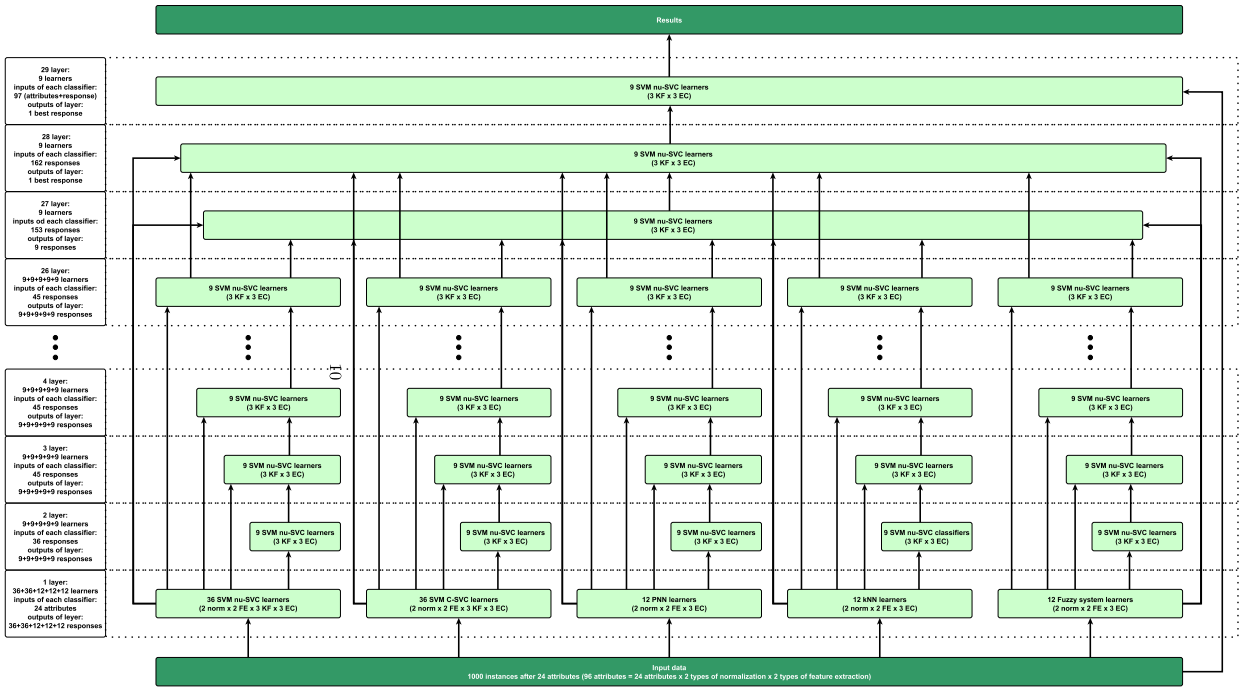

Fig. 2. A schematic view of the combination between different layers, information fusion and flow in the DGHNL system. Legend: norm - normalization type, FE - type of feature extraction, KF - kernel function type, and EC type of error calculation.

- A layered learning approach is similar to deep learning in which learning progresses in different steps.
- Hierarchical-base structure which has a multi-layered hierarchical structure of ensemble learning techniques presents an appropriate flow as well as fusion of information through the tutoring mechanism.
- Using a tutoring-based on the hierarchical structure, learners (neurons) can be trained from subsequent layers by transmission answers from prior layers and also from the 1st layer.
- A genetic layered training optimizes the structure of DGHNL by removing wrong answers from learners (bad experts):
  - Connections between learners from adjacent layers are optimized (feature selection) using a genetic algorithm which deletes some undesirable connections. Similarly, unconnected learners of the system (neurons) are removed from it (selection of learners).
  - The system records all feedbacks from the genetic algorithm (genetic optimization) in the training as well as from a cross-approach, which is similar to back-connections in the brain.
- Diversity:
  - A variety of learners performing preprocessing, extraction of features, and connection between learners can be considered as an analogy to the various kinds of neurons, data processing, and irregular connections between neurons, which belong to the neocortex of the brain.
  - A variety of component learners with a 108-learner model in the 1st layer of the proposed DGHNL system was used, including five kinds of learners, two kinds of normalization techniques, two kinds of feature extraction approaches, three kinds of kernel functions, and finally three kinds of parameter optimizations corresponding to three kinds of error calculation. In the layers 2 to 26, the proposed system had 45 models of learners consisting of five kinds of learners, three kinds of kernel functions, and three kinds of error calculation), while in the layer 27 to 29, 9 models of learners including three kinds of kernel functions and three kinds of error calculations were used.
  - A variety of data preprocessing and feature extraction techniques used in the first layer included two types of normalization approaches and two types of feature extraction techniques.
  - A variety of connections: all possible connections between the learners do not occur between layers.
- Bipolarity is used to transmit binary data {0; 1}, which is similar to the action potential of nerve cells values.
- A deep learning approach (deep multilayered structure of the DGHNL system) based on the deep learning theory; a network having more than 2 layers in its structure is considered as a deep learning system. This is similar to the neocortex having 7 layers.
- An abstract-based learning for internal feature extraction which transforms information in subsequent layers of its structure of the system acquires intricate features increasingly. Name: deep - because the structure of designed system consists of 29 layers, genetic - because in this study genetic algorithm plays an important role, hierarchical - because

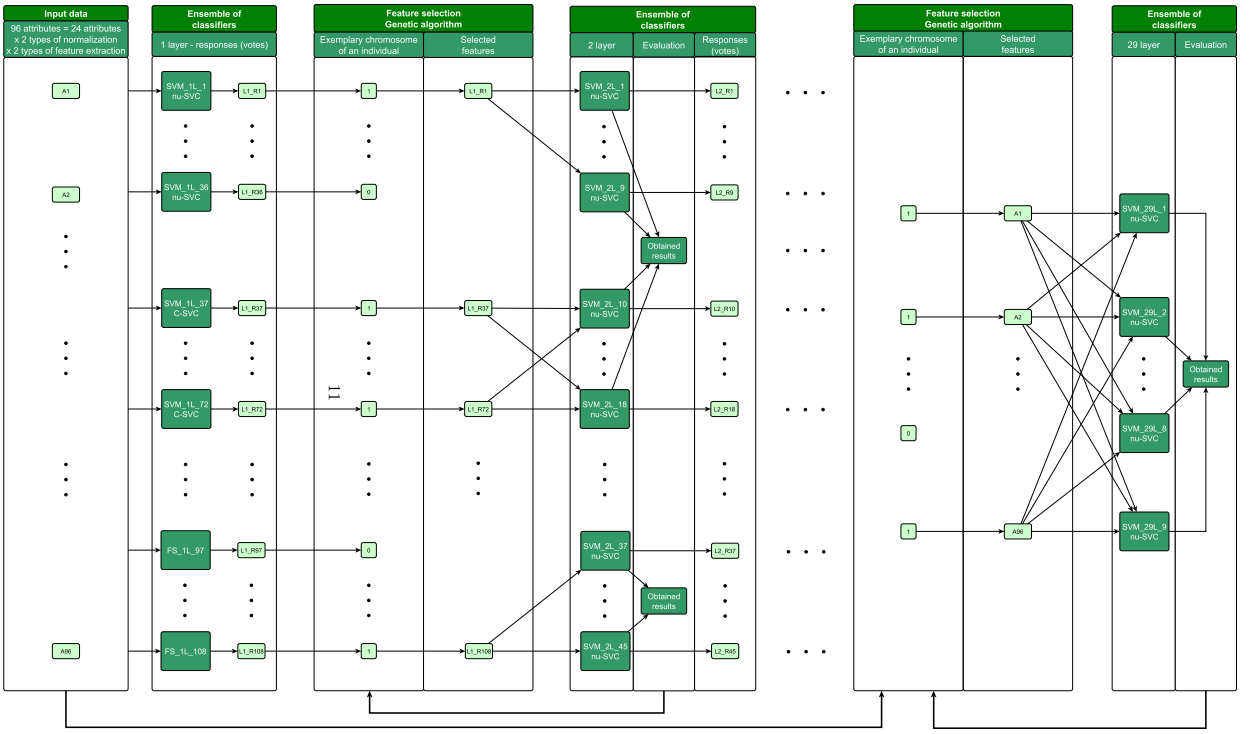

Fig. 3. A schematic view of the genetic layered training (feature selection based on GA) applied for combining learners, for examples chromosomes of individuals, and data for a single borrower.

the system structure is hierarchical, and network of learners - because the designed system consists of many nodes (learners) connected in a network.

- A layered-based learning supervised training for 108 learners was applied at the first layer. Then, based on the replies obtained from 108 models of different learners, constructed at the first layer, a supervised genetic training of 45 learners from the second layer was carried out. After that, similarly, learning goes through other layers (i.e., through all 29 layers).
- CV approach - the GA and 10-fold CV approach were coupled.

The 1st and 29th layers:

- At the first and 29th layers of the system, a genetic-based feature selection approach was applied for selection of the most important features and for optimizing the parameters of 108 learners.
- By using a genetic algorithm (see Table 1) the optimization was performed to choose the input features as well as the parameters of all learners.

The layers 2 to 28:

- A genetic layered training approach was applied to adjust the structure of ensemble-based learners from the layers 2 to 28. It includes the feature selection based on either votes of experts or judges from layers 1 to 27, based on the reference replies. At this stage, the rejection of incorrect votes (responses) of learners from the layers 1 to 27 was the main task of the genetic algorithm taking into account the errors in testing and training sets. Only reliable votes (answers) were accepted at this stage. The detailed information is provided in Fig. 3.
- Votes - each learner (expert) had only one output with the value of “1” or “2”.

A deep learning mimic mechanism in the neocortex of the brain is the main inspiration for the DGHNL system. As mentioned earlier, the DGHNL system combines several techniques such as deep learning, ensemble learning, and evolutionary computation. The proposed methodology is a continuation of the method introduced by Pławiak in [26,28,29].

While designing the system, we focused on achieving the highest possible accuracy with least complexity (i.e., the number of layers and nodes) in the system. The highest accuracy was obtained for 29 layers and the performance did not increase further by increasing the number of layers in the system.

In the optimization and training phase of the system, its strength relied on the diversity of nodes (component learners). This diversity was achieved by using: (a) two kinds of data preprocessing, (b) two kinds of feature extraction, (c) three kinds of kernel functions for learners, (d) five kinds of learners, and (e) three kinds of error calculations (fitness functions in the genetic algorithm presented in Table 1):

- acceptance features coefficient + sum of errors in training and test groups,
- sum of percentage errors in training and testing groups,
- sum of errors in testing groups.

Moreover, we used the two following strategies to prevent overfitting:

- 3 kinds of fitness functions (Table 1) - errors in testing groups are more important than errors in training groups, and
- a CV method (10-fold, stratified).

General information about the genetic algorithm used in our DGHNL system for feature selection and learner parameters optimization is shown in Table 1. We chose the parameters of the genetic algorithm in an experimental way by testing them as explained in Table 1. Based on our own experience [15,26,27,29] and taking into account the mechanisms of selective pressure and exploration (we move from a wider to a narrower space of solutions). The presented parameter values are suitable for an optimal performing of the DGHNL system. The entire structure of the DGHNL system contains 2496 parameters being optimized.

The algorithm view of the proposed system is presented in Algorithm 1 (DGHNL - Part I) and Algorithm 2 (DGHNL - Part II). Algorithm 1 (DGHNL - Part I) presents the functioning of the proposed system at the first layer and Algorithm 2 (DGHNL - Part II) presents the system functioning at the layers 2 to 29. Fig. 2 gives a schematic view of the proposed methodology, including all connections between different layers, information flow and its fusion. A scheme of genetic layered training (applied for selection of features using GA), used to combine the learners (as, for examples, chromosomes of individuals and data for a single borrower), is shown in Fig. 3.

Algorithm 1: DGHNL - Part I.

```
for 1 to 2 (normalization types) do                                              /* Normalization */
    Execute normalization (rescaling and standardization; Section 3)
    for 1 to 2 (extraction types) do                                             /* Feature extraction */
        Execute feature extraction (none and PCA; Section 3)
        Prepare testing, training sets and matrices with reference answers based on stratified 10-fold cross-validation and
         pre-processed data; Section 3;
        for 1 to 5 (learners types) do                                           /* First layer */
            Set the basic parameters values of the learners (SVM C-SVC and SVM nu-SVC and kNN and PNN and Fuzzy system; Section
             3 Table 2)
            if SVM C-SVC or SVM nu-SVC then
                for 1 to 3 (number of kernel functions) do
                    Set kernel function (polynomial and RBF and sigmoid) for SVM learner
                    Execute Algorithm steps from 14 to 42
                end
            end
            Set genetic algorithm parameters (Section 3; Table 1)
            Prepare an preliminary individuals population                        /* Begin of GA */
            for 1 to 30 (no. of generations) do
                for 1 to 500 (no. of individuals) do
                    Execute selection of features (Section 3)                    /* Selection of features */
                    Set the optimized learners parameter values.
                    for 1 to 10 (no. of sets; Section 3) do                      /* CV */
                        Create the model of the learner
                        Define the learner responses
                        Define the number of errors
                    end
                    Compute the total number of errors for the testing and training sets
                    for 1 to 3 (error calculation) do
                        Define the value of the fitness function (ERR_T and ERR_% and ERR_sum; Table 1) for learner
                    end
                    Save the related values of learner parameters, fitness function, selected features and matrices with learner responses
                end
                if fitness function == 0 then
                    Give out the best individual with related matrix of responses
                    break
                else
                    Execute the individual selection
                    Use of genetic operators: mutation and crossover
                    Prepare a new individuals population
                end
            end                                                                  /* End of GA */
            Give out the best individual with related matrix of responses
            Prepare the confusion matrix
            Compute the ACC for the learner
        end
    end
end
```

Algorithm 2: DGHNL - Part II.

```
for 2th to 26th layer do                                                    /* 2-26th layer */
    Prepare testing and training sets based on CV method (10-fold, stratified) and responses from previous layers (Fig. 2)
    for 1 to 5 (number of response groups; Fig. 2) do
        for 1 to 3 (kernel functions) do
            Set kernel function (polynomial and RBF and sigmoid) for one learner SVM nu – SVC
            Set the SVM nu – SVC learner basic parameters values.
            Set genetic algorithm parameters (Table 1)
            Prepare an preliminary individuals population                   /* Begin of GA */
            for i ← 1 to 50 (generation number) do
                for j ← 1 to 500 (individual number) do
                    Execute selection of learner                            /* Selection of features */
                    for 1 to 10 (no. of sets) do                                                        /* CV */
                        Create the model of the SVM nu – SVC learner
                        Define the SVM nu – SVC learner responses
                        Define the number of errors
                    end
                    Compute the total number of errors for the testing and training sets
                    for 1 to 3 (error calculation) do
                        Define the value of the fitness function (ERR_T and ERR_% and ERR_sum; Table 1) for SVM nu – SVC learner
                    end
                    Save the related values of learner parameters, fitness function, selected features and matrices with SVM nu – SVC
                     learner responses
                end
                if fitness function == 0 then
                    Give out the best individual with related matrix of responses
                    break
                else
                    Execute individual selection
                    Use of genetic operators: mutation and crossover
                    Prepare a new population of individuals
                end
            end                                                             /* End of GA */
            Give out the best individual with related matrix of responses
            Prepare the confusion matrix
            Compute the ACC for the SVM nu – SVC learner
        end
    end
end
for 27th to 29th layer do                                                   /* 27-29th layer */
    Create testing and training sets based on CV method (10-fold, stratified) and responses from previous layers (Fig. 2)
    Execute Algorithm steps from 49 to 79
end
```

5. Results

In this section we present experimental results of our study. Before that, the dataset used in this work is briefly described.

5.1. Dataset description

This study used the popular Statlog German Credit (SGC) dataset. The data is available at the UCI Machine Learning Repository website [12]. We used the numeric version of this dataset. It includes 1000 borrowers records (instances) grouped into two classes: class of accepted/good applicants (Class 1 with 700 records) and class of rejected/bad applicants (Class 2 with 300 records). All instances have 20 input attributes including 13 categorical features and 7 numerical features. Moreover, the class (accepted or rejected applicants) from each record is the output of the current research. We transformed the categorical attributes into numerical attributes. Moreover, some indicator variables were also considered, this raising the number of attributes to 24 numerical input values [13]. More details about the SGC data are given in Tables 3 and 4.

5.2. Experiments

In this study, the MATLAB R2014b environment and the LIBSVM open source library were used to implement the proposed methodology. The proposed method was carried out on an IBM PC computer equipped with a 4.0 GHz Intel Core i7-6700K CPU and 32 GB of RAM. The study was divided into two major experiments: i) in the first experiment, single learners were used to investigate the performance of CS prediction, and ii) in the second experiment, the proposed methodology was applied with different numbers of layers. More information about the obtained results of these two experiments is given in the following sections.

Table 3
Description of the data in testing and training groups used in the CV method (10-fold, stratified).

		German Credit Data - 20(24) attributes		
			CV (10-fold, stratified)	
			Sets 1–10	
Class	Description	Number of instances	Training group	Testing group
C1	Accepted credit	700 (70.0%)	630	70
C2	Rejected credit	300 (30.0%)	270	30
	Sum	1000 (100.0%)	900 (90%)	100 (10%)

Table 4
Attributes applied to assess the credit risk in the German credit dataset [12,13].

Attribute	Description	Type	Range of values
Inputs			
G1	Status of existing checking account	Categorical	1–4
G2	Duration in months	Numerical	4–72
G3	Credit history	Categorical	0–4
G4	Purpose	Categorical	0–10
G5	Credit account	Numerical	276–18424
G6	Savings account/bonds	Categorical	1–5
G7	Present employment since	Categorical	1–5
G8	Instalment rate in percentage of disposable income	Numerical	1–4
G9	Personal status and sex	Categorical	1–5
G10	Other debtors/guarantors	Categorical	1–3
G11	Present residence since	Numerical	1–4
G12	Property	Categorical	1–4
G13	Age in years	Numerical	19–75
G14	Other instalment plans	Categorical	1–3
G15	Housing	Categorical	1–3
G16	Number of existing credits at this bank	Numerical	1–4
G17	Job	Categorical	1–4
G18	Number of people being liable to provide maintenance for	Numerical	1–2
G19	Have telephone or not	Categorical	1–2
G20	Foreign worker	Categorical	1–2
Output			
G21	Class attribute	Categorical	1–2

Table 5
Comparison of the best outcomes achieved for single learners from the 1st layer: Fuzzy system, kNN, PNN, and 2 types of SVM learners.

Coefficient	Learners						
	kNN	PNN	Fuzzy	SVM	SVM	SVM	SVM
Model type	–	–	–	*nu – SVC*	*C – SVC*	*nu – SVC*	*C – SVC*
Normalization	*Standardization*	*Standardization*	*Standardization*	*Standardization*	*Standardization*	*Rescaling*	*Rescaling*
Feature extraction	*None*	*None*	*PCA*	*None*	*None*	*None*	*None*
Errors	err_t	err_t	err_t	err_t	err_t	err_t	err_t
Kernel function	–	–	–	*Polynomial*	*Polynomial*	*RBF*	*RBF*
Results for training sets							
$\boldsymbol{ERR_{sum}}$	***1923***	***1291***	***1922***	***1345***	***1602***	***1465***	***1381***
ACC	78.63%	85.66%	78.64%	85.06%	82.20%	83.72%	84.66%
Results for test sets							
$\boldsymbol{ERR_{sum}}$	***224***	***215***	***214***	***203***	***202***	***202***	***201***
ACC	77.60%	78.50%	78.60%	79.70%	79.80%	79.80%	79.90%

5.3. First experiment: single methods

In the first experiment, single fuzzy system, kNN, PNN, nu-SVC (with Polynomial, RBF and Sigmoid kernel functions) and C-SVC (with Polynomial, RBF, and Sigmoid kernel functions) methods are applied. The obtained results are reported in Table 5.

Table 6
Comparison of the best results obtained for the subsequent layers of the DGHNL method (best learner from the 1st layer and best meta-learners from layers 2 to 29, also see Fig. 2).

Coefficient	Layers								
	1	2	3	14	23	26	27	28	29
Model type	$C-SVC$	$nu-SVC$	$nu-SVC$	$nu-SVC$	$nu-SVC$	$nu-SVC$	$nu-SVC$	$nu-SVC$	$nu-SVC$
Errors	err_t	err_t	err_t	err_{pro}	err_{sum}	err_t	err_t	err_t	err_{sum}
Kernel function	*RBF*	*Sigmoid*	*RBF*	*RBF*	*Polynomial*	*Polynomial*	*Sigmoid*	*Polynomial*	*Sigmoid*
Results for training sets									
$\boldsymbol{ERR_{sum}}$	***1381***	***210***	***0***	***0***	***0***	***0***	***0***	***0***	***24***
ACC	84.66%	97.67%	100%	100%	100%	100%	100%	100%	99.73%
Results for test sets									
$\boldsymbol{ERR_{sum}}$	***201***	***153***	***133***	***89***	***77***	***77***	***61***	***56***	***54***
ACC	79.90%	84.70%	86.70%	91.10%	92.30%	92.30%	93.90%	94.40%	94.60%

Table 5 shows that the best accuracy of 85.66% was obtained with the PNN learner and that the accuracy of 79.90% was obtained with C-SVC (using the RBF kernel function) for the stratified 10-fold cross-validation technique.

5.4. Second experiment: subsequent DGHNL system

In the second experiment, the DGHNL system was used as the hybrid of all learners considered in the first experiment (i.e., Fuzzy system, kNN, PNN, nu-SVC, and C-SVC). The obtained results are illustrated in Table 6.

As reported in Table 6, in the training sets of the DGHNL system with 3, 14, 23, 26, 27, and 28 layers, the best possible accuracy of 100% was obtained, whereas the DGHNL system (nu-SVC with Sigmoid kernel function) with 29 layers had the highest accuracy in testing phase, compared to other layers, with the accuracy of 94.60%.

5.5. Running time

We have evaluated the running time required for executing the whole structure of the DGHNL system:

- optimization time = 9830 minutes on average,
- training time = 3784 seconds on average,
- classification time = 0.038 seconds on average.

It can be noted that the time required for data classification is very low. Hence, the system can be used to operate in real time.

5.6. Discussion

Fig. 4 presents the confusion matrix of the DGHNL system for the test datasets. These results confirm that the proposed system is able to predict the CS data accurately.

It can be seen from Fig. 4 that the DGHNL system showed a better performance for Class 2 with 98.40% of accuracy than for Class 1 with 93.30% of accuracy.

Fig. 5 presents the receiver operating characteristic (ROC curve) for single methods (nu-SVC, C-SVC, PNN, kNN, and fuzzy system) applied within the DGHNL system. The following area under curve (AUC) values for DGHNL = 0.9138, nu-SVC = 0.7195, C-SVC = 0.7193, PNN = 0.6798, kNN = 0.6762, and fuzzy system = 0.6986 were obtained for the test datasets. Fig. 6 presents the percentage errors for 10 test datasets using 10-fold CV carried out for the DGHNL system (error values range from 1% for 7th test set to 10% for 3rd test set). The following F1-Score values for DGHNL = 0.9326, nu-SVC = 0.7362, C-SVC = 0.7365, PNN = 0.6988, kNN = 0.6930, and the fuzzy system = 0.7154 were obtained for the test datasets.

The results presented in Table 7 indicate that the proposed DGHNL system provides the best accuracy (94.60%) among the competing methods in the literature.

Table 5 reports the results obtained by single machine learning method in terms of the accuracy. They are as follows: 79.90% of accuracy obtained for C-SVC, 79.80% of accuracy obtained for nu-SVC, 78.60% of accuracy obtained for fuzzy system, 78.50% of accuracy obtained for PNN, and 77.60% of accuracy obtained for kNN. Importantly, the tutoring effect (a deep multilayer structure of the DGHNL system, i.e., layers 2 to 28) helps to boost the classification accuracy significantly using the training datasets (see Table 6). The application of single artificial intelligence methods resulting in a unique layer of the proposed DGHNL system allowed us to get the accuracy of 79.90% in the best case. However, after adding the 2nd layer to DGHNL (thus creating an ensemble of learners) we were able to increase the system accuracy by 5% (up to 84.70%). Finally, by using the deep learning approach, 27 subsequent layers were added to the system, allowing us to augment the system accuracy by 10% (up to 94.60%).

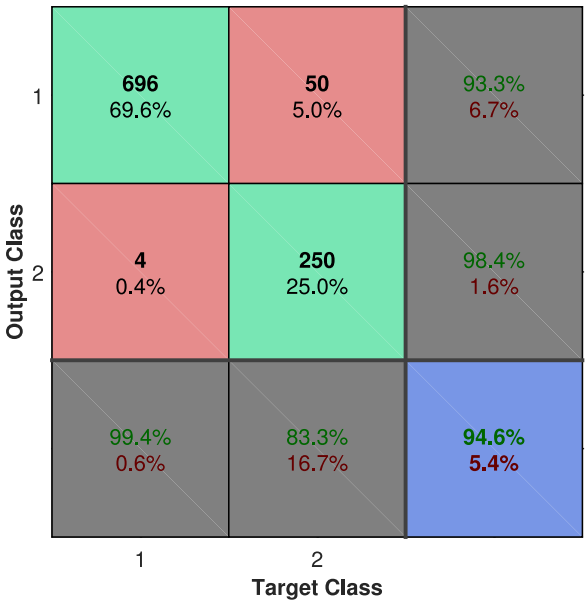

Fig. 4. Confusion matrix of the DGHNL method for the testing datasets.

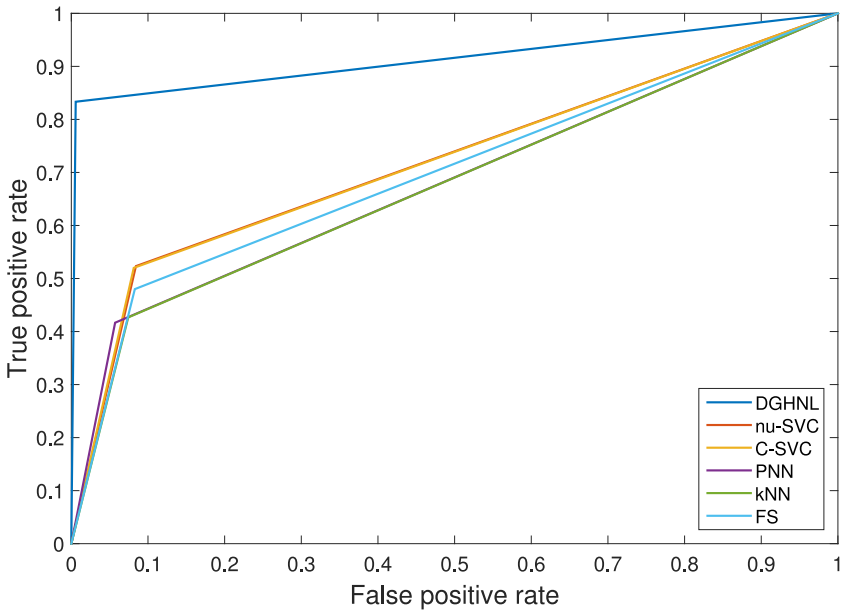

Fig. 5. ROC curves for single ML methods (nu-SVC, C-SVC, PNN, kNN, fuzzy system), used within the DGHNL system, obtained for the test datasets.

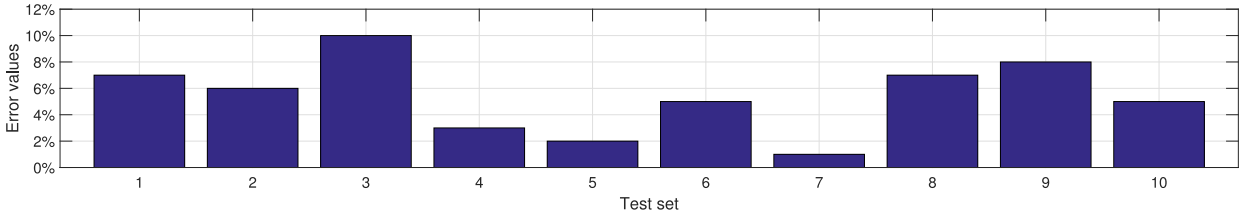

Fig. 6. Graph of (%) versus each fold of ten-fold CV.

Table 7
Comparison of the outcomes achieved for prediction of credit scoring for the Statlog German credit approval dataset.

Number	Authors	Year	Cross-validation	Feature extraction/selection	Method	ACC [%]
1.	Peng et al. [25]	2008	10-fold	–	Multicriteria Convex Quadric Programming (MCQP) (libSVM)	94.00
2.	Damrongsakmethee and Neagoe [7]	2019	Not reported	Principal Component Analysis (PCA) and ReliefF algorithm	Support Vector Machine (SVM)	91.67
3.	Tsai [43]	2014	5-fold	SOM	Homo. Classifier (MLP or CART or LR) Ensembles (weighted voting)	88.93
4.	Krishna and Ravi [14]	2019	10-fold	Adaptive Differential Evolution (ADE)	SVM	85.30
5.	Luo, Cheng, Hsieh [19]	2009	10-fold	–	Clustering-launched classification (CLC)	84.80
6.	Sivasankar [35]	2019	10-fold	Rough set-based	Weight-adjusted boosting ensemble method (WABEM)-SVM	83.85
7.	Zhang, Zhou, Leung, Zheng [47]	2010	10-fold	–	Vertical bagging decision trees	81.64
8.	Chang, Yeh [5]	2012	10-fold	–	Artificial immune classifier based on the artificial immune network (AINE-based classifier)	79.50
9.	Alaraj, Abbod [2]	2016	5-fold	Multivariate Adaptive Regression Splines (MARS)	Heterogeneous hybrid ensemble with consensus approach combination rule	79.00
10.	Tsai [42]	2009	5-fold	FA (factor analysis)	MLP	78.76
11.	Gorzczany and Rudziki [10]	2016	10-fold	–	Fuzzy rule-based classifier with multi-objective evolutionary optimization	78.50
12.	Han et al. [11]	2019	5-fold	–	Gaussian mixture model	75.21
13.	Wang and Tu [44]	2019	5-fold (for training)	LpMKL	SVM	74.40
14.	Zhang et al. [50]	2019	10-fold	SMCNOC	SVM (RBFK)	73.60
15.	Lahoti et al. [18]	2019	Not reported	–	iFair-b	73.00
	Proposed method - DGHNL	**2018**	**10-fold**	**None + PCA**	**Deep Genetic Hierarchical Network of Learners (29-layer system)**	**94.60%**

Fig. 3 indicates that the genetic layer training is an important aspect of the proposed methodology. The training approach has a significant impact on the optimization of the performance of the DGHNL system. The genetic training allowed us to: (a) remove unnecessary attributes in the 1st and 29th layers of the system, and (b) choose optimal nodes (learners) between the layers 2 to 28 of the system. The good results provided by the proposed DGHNL system are mainly due to: (a) diversity, quality and speed of the learners (nodes) included in the system, (b) hierarchical structure of the system, including a genetic layered training technique and tutoring effect, and (c) feature selection and feature extraction in the system layers.

Another key to success of the proposed system (Fig. 2) is a proper information fusion and flow model. Regarding the nodes (component learners), other learners having diversity are preferred. In this work, we have empirically tried and selected the learners and kernel functions in order to obtain the highest performance.

A slightly worse result has been obtained for the system consisting of SVM learners only (2 types + 3 kernel functions). The best results have been obtained using all three kernel functions from the LIBSVM library (except of the linear kernel function). We have obtained a greater diversity of learners, and hence a higher system accuracy, by using 3 types of SVM kernel functions, compared to other types of learners. Therefore, we can conclude that the optimal solution (in terms of the accuracy, complexity and system optimization time) may be obtained by considering a network with only two types of SVMs.

The system learns during the training phase (modifies its own structure), e.g. by removing incorrect responses. Using techniques preventing overfitting (CV, 3 types of fitness function), the acquired knowledge can be generalized to samples from real-world applications.

We achieved the effect of network tutoring using successive layers. This effect has been obtained after many experiments involving: a) different optimization of component learners (different fitness functions - error calculation metrics), b) diversity of component learners, and different network structures (connections between nodes).

Like a neural network, the described DGHNL system is generic, but it must be re-trained and optimized for every new data types. Our results show that such training is fast and effective. The concept of the described method (based on similar training but differing in the structure of the system) has also been successfully experimented using ECG data [26,29] and Australian credit database [28]. The results obtained (on different types of data) confirm that the presented idea is accurate and robust.

The proposed system can incorporate any rules supplied by bankers and make accurate decisions. With this tool, banks can save a lot of money by reducing the staff and maximize the profits. In addition, the fast execution time allows one to implement this model on mobile devices to help the customers to evaluate the possibility of getting the loan. The only limitation of our model is its structural complexity. Similar to an artificial neural network, the knowledge is not transmitted

explicitly and sometimes it is difficult to analyze the system's decisions in our model. However, by analyzing the responses at the nodes (experts and judges) of the proposed system, one can get access to knowledge based on which it makes final decisions.

6. Conclusions

In the bank industry, a low risk credit scoring is essential. This paper presents a novel methodology, called Deep Genetic Hierarchical Network of Learners (DGHNL), which can work with various kinds of learners. The proposed DGHNL system was applied to analyze the Statlog German credit approval dataset with 1000 records, available at the UCI machine learning repository. A novel training approach for learners has been described. It is based on genetic layered training. By using stratified 10-fold cross-validation, our DGHNL system showed the highest accuracy rate on the Statlog German credit data, compared to the existing work in the field, with the accuracy of 94.60%. Obviously, our new system also has some limitations. For example, as in any other deep learning model, the long-term training and optimization are required to build up a complex system structure. In the future, we intend to test the proposed system with different credit scoring datasets. Also, other evolutionary algorithms may be used in our model in order to optimize its performance. We can also try to increase the accuracy of the proposed DGHNL system by using the blind-fold cross validation technique. Also, other classification algorithms could be employed within DGHNL.

Declaration of Competing Interest

None.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ins.2019.12.045.

CRediT authorship contribution statement

Paweł Pławiak: Conceptualization, Data curation, Formal analysis, Writing - original draft. **Moloud Abdar:** Writing - original draft. **Joanna Pławiak:** Writing - original draft. **Vladimir Makarenkov:** Writing - original draft. **U Rajendra Acharya:** Writing - original draft.

References

[1] J. Abelln, J.G. Castellano, A comparative study on base classifiers in ensemble methods for credit scoring, Expert Syst. Appl. 73 (2017) 1–10 http://www.sciencedirect.com/science/article/pii/S0957417416306947.
[2] M. Ala'raj, M.F. Abbod, A new hybrid ensemble credit scoring model based on classifiers consensus system approach, Expert Syst. Appl. 64 (Supplement C) (2016) 36–55 http://www.sciencedirect.com/science/article/pii/S0957417416303621.
[3] C. Bravo, L.C. Thomas, R. Weber, Improving credit scoring by differentiating defaulter behaviour, J. Oper. Res. Soc. 66 (5) (2015) 771–781, doi:10.1057/jors.2014.50.
[4] M. Bcker, M. van Kampen, W. Krmer, Reject inference in consumer credit scoring with nonignorable missing data, J. Banking Finance 37 (3) (2013) 1040–1045 http://www.sciencedirect.com/science/article/pii/S037842661200341X.
[5] S.-Y. Chang, T.-Y. Yeh, An artificial immune classifier for credit scoring analysis, Appl. Soft Comput. 12 (2) (2012) 611–618 http://www.sciencedirect.com/science/article/pii/S1568494611004261.
[6] X. Chen, C. Zhou, X. Wang, Y. Li, The credit scoring model based on logistic-bp-adaboost algorithm and its application in p2p credit platform, in: X. Li, X. Xu (Eds.), Proceedings of the Fourth International Forum on Decision Sciences, Springer Singapore, Singapore, 2017, pp. 119–130.
[7] T. Damrongsakmethee, V.-E. Neagoe, Principal component analysis and relieff cascaded with decision tree for credit scoring, in: R. Silhavy (Ed.), Artificial Intelligence Methods in Intelligent Algorithms, Springer International Publishing, Cham, 2019, pp. 85–95.
[8] D. Durand, Risk Elements in Consumer Instalment Financing, National Bureau of Economic Research, Inc, 1941. https://EconPapers.repec.org/RePEc:nbr:nberbk:dura41-1.
[9] A.J. Feelders, Credit scoring and reject inference with mixture models. intelligent systems in accounting, Finance Manage. 9 (1) (2000) 1–8.
[10] M.B. Gorzaczany, F. Rudziski, A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability, Appl. Soft Comput. 40 (Supplement C) (2016) 206–220 http://www.sciencedirect.com/science/article/pii/S1568494615007553.
[11] X. Han, R. Cui, Y. Lan, Y. Kang, J. Deng, N. Jia, A gaussian mixture model based combined resampling algorithm for classification of imbalanced credit data sets, Int. J. Mach. Learn. Cybern. (2019), doi:10.1007/s13042-019-00953-2.
[12] H. Hofmann, Uci machine learning repository, statlog (german credit data) data set, https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).
[13] A. Khashman, Neural networks for credit risk evaluation: investigation of different neural models and learning schemes, Expert Syst. Appl. 37 (9) (2010) 6233–6239 http://www.sciencedirect.com/science/article/pii/S0957417410001405.
[14] G.J. Krishna, V. Ravi, Feature subset selection using adaptive differential evolution: an application to banking, in: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, in: CoDS-COMAD '19, ACM, New York, NY, USA, 2019, pp. 157–163, doi:10.1145/3297001.3297021.
[15] W. Książek, M. Abdar, U.R. Acharya, P. Pławiak, A novel machine learning approach for early detection of hepatocellular carcinoma patients, Cognit. Syst. Res. 54 (2019) 116–127 http://www.sciencedirect.com/science/article/pii/S1389041718308714.
[16] P.R. Kumar, V. Ravi, Bankruptcy prediction in banks and firms via statistical and intelligent techniques a review, Eur. J. Oper. Res. 180 (1) (2007) 1–28 http://www.sciencedirect.com/science/article/pii/S0377221706008769.
[17] A.B. Labao, P.C. Naval, Cascaded deep network systems with linked ensemble components for underwater fish detection in the wild, Ecol. Inf. 52 (2019) 103–121 http://www.sciencedirect.com/science/article/pii/S1574954118303078.
[18] P. Lahoti, K.P. Gummadi, G. Weikum, ifair: learning individually fair data representations for algorithmic decision making, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE), 2019, pp. 1334–1345, doi:10.1109/ICDE.2019.00121.

[19] S.-T. Luo, B.-W. Cheng, C.-H. Hsieh, Prediction model building with clustering-launched classification and support vector machines in credit scoring, Expert Syst. Appl. 36 (4) (2009) 7562–7566 http://www.sciencedirect.com/science/article/pii/S0957417408006829.
[20] P. Melin, M. Pulido, Optimization of ensemble neural networks with type-2 fuzzy integration of responses for the dow jones time series prediction, Intell. Autom. Soft Comput. 20 (3) (2014) 403–418, doi:10.1080/10798587.2014.893047.
[21] P. Melin, J. Soto, O. Castillo, J. Soria, A new approach for time series prediction using ensembles of anfis models, Expert Syst. Appl. 39 (3) (2012) 3494–3506 http://www.sciencedirect.com/science/article/pii/S0957417411013492.
[22] L. Ogiela, R. Tadeusiewicz, M.R. Ogiela, Cognitive analysis in diagnostic dss-type it systems, in: L. Rutkowski, R. Tadeusiewicz, L.A. Zadeh, J.M. Żurada (Eds.), Artificial Intelligence and Soft Computing – ICAISC 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 962–971.
[23] M.R. Ogiela, R. Tadeusiewicz, Nonlinear processing and semantic content analysis in medical imaging-a cognitive approach, IEEE Trans. Instrum. Meas 54 (6) (2005) 2149–2155, doi:10.1109/TIM.2005.858566.
[24] D. Panek, A. Skalski, J. Gajda, R. Tadeusiewicz, Acoustic analysis assessment in speech pathology detection, International Journal of Applied Mathematics andComputer Science 25 (3) (2015) 631–643. https://content.sciendo.com/view/journals/amcs/25/3/article-p631.xml.
[25] Y. Peng, G. Kou, Y. Shi, Z. Chen, A multi-criteria convex quadratic programming model for credit data analysis, Decis. Support Syst. 44 (4) (2008) 1016–1030 http://www.sciencedirect.com/science/article/pii/S0167923607002126.
[26] P. Pławiak, Novel genetic ensembles of classifiers applied to myocardium dysfunction recognition based on ecg signals, Swarm Evol. Comput. 39 (2018) 192–208 http://www.sciencedirect.com/science/article/pii/S2210650217306053.
[27] P. Pławiak, Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neural system, Expert Syst. Appl. 92 (2018) 334–349 https://www.sciencedirect.com/science/article/pii/S0957417417306292.
[28] P. Pławiak, M. Abdar, U.R. Acharya, Application of new deep genetic cascade ensemble of svm classifiers to predict the australian credit scoring, Appl. Soft Comput. 84 (2019) 105740 http://www.sciencedirect.com/science/article/pii/S1568494619305216.
[29] P. Pławiak, U.R. Acharya, Novel deep genetic ensemble of classifiers for arrhythmia detection using ecg signals, Neural Comput. Appl. (2019), doi:10.1007/s00521-018-03980-2.
[30] M. Pulido, P. Melin, O. Castillo, Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the mexican stock exchange, Inf. Sci. 280 (2014) 188–204 http://www.sciencedirect.com/science/article/pii/S0020025514005246.
[31] J. Riordon, C. McCallum, D. Sinton, Deep learning for the classification of human sperm, Comput. Biol. Med. 111 (2019) 103342 http://www.sciencedirect.com/science/article/pii/S0010482519302112.
[32] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (4) (2018) e1249 https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1249. https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1249.
[33] R. Setiono, A. Azcarraga, Y. Hayashi, Using sample selection to improve accuracy and simplicity of rules extracted from neural networks for credit scoring applications, Int. J. Comput. Intell. Appl. 14 (4) (2015) 1550021, doi:10.1142/S1469026815500212.
[34] P. Singh, Comparative study of individual and ensemble methods of classification for credit scoring, in: 2017 International Conference on Inventive Computing and Informatics (ICICI), 2017, pp. 968–972, doi:10.1109/ICICI.2017.8365282.
[35] E. Sivasankar, C. Selvi, S. Mahalakshmi, Rough set-based feature selection for credit risk prediction using weight-adjusted boosting ensemble method, Soft Comput. (2019), doi:10.1007/s00500-019-04167-0.
[36] J. Soto, P. Melin, O. Castillo, Time series prediction using ensembles of anfis models with genetic optimization of interval type-2 and type-1 fuzzy integrators, Int. J. Hybrid Intell. Syst. 11 (3) (2014) 211–226, doi:10.3233/HIS-140196.
[37] J. Soto, P. Melin, O. Castillo, A new approach for time series prediction using ensembles of it2fnn models with optimization of fuzzy integrators, Int. J. Fuzzy Syst. 20 (3) (2018) 701–728, doi:10.1007/s40815-017-0443-6.
[38] R. Tadeusiewicz, L. Ogiela, M.R. Ogiela, Cognitive analysis techniques in business planning and decision support systems, in: L. Rutkowski, R. Tadeusiewicz, L.A. Zadeh, J.M. Żurada (Eds.), Artificial Intelligence and Soft Computing – ICAISC 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 1027–1039.
[39] Y. Tian, Z. Yong, J. Luo, A new approach for reject inference in credit scoring using kernel-free fuzzy quadratic surface support vector machines, Appl. Soft Comput. 73 (2018) 96–105 http://www.sciencedirect.com/science/article/pii/S1568494618304812.
[40] D. Tripathi, R. Cheruku, A. Bablani, Relative performance evaluation of ensemble classification with feature reduction in credit scoring datasets, in: D. Reddy Edla, P. Lingras, V. K. (Eds.), Advances in Machine Learning and Data Science, Springer Singapore, Singapore, 2018, pp. 293–304.
[41] D. Tripathi, D.R. Edla, V. Kuppili, A. Bablani, R. Dharavath, Credit scoring model based on weighted voting and cluster based feature selection, Procedia Computer Science 132 (2018) 22–31 http://www.sciencedirect.com/science/article/pii/S1877050918307877. International Conference on Computational Intelligence and Data Science.
[42] C.-F. Tsai, Feature selection in bankruptcy prediction, Knowl.-Based Syst. 22 (2) (2009) 120–127 http://www.sciencedirect.com/science/article/pii/S0950705108001536.
[43] C.-F. Tsai, Combining cluster analysis with classifier ensembles to predict financial distress, Inf. Fusion 16 (Supplement C) (2014) 46–58 http://www.sciencedirect.com/science/article/pii/S1566253511000807.
[44] T. Wang, X. Tu, Multiple kernel learning using nonlinear lasso, IEEJ Trans. Electr. Electron. Eng. 14 (5) (2019) 760–767, doi:10.1002/tee.22862.
[45] Y. Xia, C. Liu, B. Da, F. Xie, A novel heterogeneous ensemble credit scoring model based on bstacking approach, Expert Syst. Appl. 93 (2018) 182–199 http://www.sciencedirect.com/science/article/pii/S0957417417306966.
[46] S. Xu, Q. Tang, L. Jin, Z. Pan, A cascade ensemble learning model for human activity recognition with smartphones, Sensors 19 (10) (2019) 2307, doi:10.3390/s19102307.
[47] D. Zhang, X. Zhou, S.C. Leung, J. Zheng, Vertical bagging decision trees model for credit scoring, Expert Syst. Appl. 37 (12) (2010) 7838–7843 http://www.sciencedirect.com/science/article/pii/S0957417410003544.
[48] H. Zhang, H. He, W. Zhang, Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring, Neurocomputing 316 (2018) 210–221 http://www.sciencedirect.com/science/article/pii/S0925231218309056.
[49] P. Zhang, T.D. Bui, C.Y. Suen, A novel cascade ensemble classifier system with a high recognition performance on handwritten digits, Pattern Recognit. 40 (12) (2007) 3415–3429 http://www.sciencedirect.com/science/article/pii/S003132030700146X.
[50] Z. Zhang, J. He, G. Gao, Y. Tian, Sparse multi-criteria optimization classifier for credit risk evaluation, Soft Comput. 23 (9) (2019) 3053–3066, doi:10.1007/s00500-017-2953-4.

P. Pławiak, B.Eng., M.Sc., Ph.D., was born in Ostrowiec, Poland, in 1984. He obtained his B.Eng. and M.Sc. degree in Electronics and Telecommunications and his Ph.D. degree with honors in Biocybernetics and Biomedical Engineering at the AGH University of Science and Technology, Cracow, Poland, in 2012 and 2016, respectively. He is the Head of the Department of Information and Communications Technology and an Assistant Professor in Cracow University of Technology, Cracow, and Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice Poland. He has published more than 20 papers in refereed international SCI-IF journals. He is a reviewer of many prestigious and reputed journals. His research interests include machine learning and computational intelligence (e.g., artificial neural networks, genetic algorithms, fuzzy systems, support vector machines, k-nearest neighbors, and hybrid systems), ensemble learning, deep learning, evolutionary computation, classification, pattern recognition, signal processing and analysis, data analysis and data mining, sensor techniques, medicine, biocybernetics, and biomedical engineering.

M. Abdar, received his bachelors degree in computer engineering from Damghan University, Iran, in 2015. He also received the masters degree in computer science and engineering from the University of Aizu, Aizu, Japan, in 2018. Currently, he is working toward his Ph.D. degree at the Department of Computer Science, University of Quebec in Montreal, Montreal (QC), Canada. He has written several papers in the fields of data mining, machine learning, sentiment analysis and user modeling in some refereed international journals and conferences. His research interests include data mining, machine learning, ensemble

learning, evolutionary algorithms and user modeling. He is a recipient of the Fonds de Recherche du Quebec Nature et Technologies Award (ranked 5th among 20 candidates in the second round of selection process) in 2019.

J. Pławiak, B.Eng., M.Sc., was born in Ostrowiec, Poland, in 1988. She obtained her B.Eng. degree in Automatic Control and Robotics at the AGH University of Science and Technology, Cracow, Poland, in 2012, and her M.Sc. degree in Management, specialty Applied Computer Science at the School of Management and Banking, Cracow, Poland, in 2016. Her research interests include machine learning and computational intelligence (e.g., artificial neural networks, support vector machines, k-nearest neighbors), classification, pattern recognition, data analysis and data mining, economics, banking and biomedical engineering.

V. Makarenkov, Ph.D. is a Full Professor and Director of a graduate Bioinformatics program at the Department of Computer Science at the University of Quebec in Montreal. His research interests are in the fields of Bioinformatics, Data mining, Software Engineering, and Artificial Intelligence.

U Rajendra Acharya, Ph.D., DEng. is a senior faculty member at Ngee Ann Polytechnic, Singapore. He is also (i) Adjunct Professor at Taylors University, Malaysia, (ii) Adjunct Faculty at Singapore Institute of Technology- University of Glasgow, Singapore, and (iii) Associate faculty at Singapore University of Social Sciences, Singapore. He received his Ph.D. from National Institute of Technology Karnataka (Surathkal, India) and DEng from Chiba University (Japan). He has published more than 400 papers, in refereed international SCI-IF journals (345), international conference proceedings (42), books (17) with more than 25,000 citations in Google Scholar (with h-index of 83), and ResearchGate RG Score of 46.9. He is ranked in the top 1% of the Highly Cited Researchers (2016,2017,2018 and 2019) in Computer Science according to the Essential Science Indicators of Thomson. He has worked on various funded projects, with grants worth more than 4 million SGD. He has three patents and in the editorial board of many journals. He has served as guest editor for many journals. His major academic interests are in biomedical signal processing, biomedical imaging, data mining, visualization and biophysics for better healthcare design, delivery and therapy. Please visit https://scholar.google.com.sg/citations?user=8FjY99sAAAAJ&hl=en Google Scholar for more details.