A Hybrid Meta-cognitive Restricted Boltzmann Machine Classifier for Credit Scoring

Savitha Ramasamy
Data Analytics Department
Institute for Infocomm Research
Agency for Science, Research and Technology
Singapore
Email: ramasamysa@i2r.a-star.edu.sg

Kanagasabai Rajaraman
Data Analytics Department
Institute for Infocomm Research
Agency for Science, Research and Technology
Singapore
Email: kanagasa@i2r.a-star.edu.sg

***Abstract*—Meta-cognition with self-regulated learning equips a machine learning algorithm to make judicious decisions about every sample in the training data set. Due to this capability, meta-cognitive machine learning algorithms exhibit better generalization behavior. In the past, numerous works have focused on studying the effect on meta-cognition for learning patterns from data in a supervised fashion. In this paper, we attempt to study the effect of meta-cognition towards efficient feature representation in a Restricted Boltzmann Machine (RBM). To this end, we develop a Meta-cognitive Restricted Boltzmann Machine (McRBM) that decides *what-to-represent*, *how-to-represent* and *when-to-represent* for each sample in the training data set through its Sample deletion, Feature Representation and Sample Reserve strategies, respectively. The meta-cognitive component of McRBM helps to improve the generative training of RBM. The features thus generated by RBM are then used in classification through back propagation learning, which is derived based on the hinge-loss error function. The McRBM is used to solve credit scoring problem using the German credit data set, Australian credit data set and the KAGGLE credit data set. Performance of McRBM is compared against the Support Vector Machines (SVM), Extreme Learning Machines (ELM), Multi-layer Perceptron with Back Propagation (MLP-BP), and Classification Restricted Boltzmann Machine (ClassRBM). Performance results show superior classification ability of McRBM.**

I. Introduction

Research in human learning [1] emphasizes that self-regulation enables humans to learn by planning, monitoring and evaluating their progress seamlessly using Meta-cognition. In [2], meta-cognition has been defined as the higher-order thinking that enables a learner to evaluate his/her knowledge with reference to the world, identify the need and make personalized strategies to acquire new knowledge. Thus, the self-awareness of the learner and knowledge about one's own knowledge are critical to meta-cognition.

Most of the traditional approaches in machine learning learn every sample presented to the network. Recently meta-cognition has been adapted in the machine learning literature and several meta-cognitive neural networks have been derived [3]–[9], based on the Nelson and Narens model of meta-cognition [1]. These meta-cognitive neural networks have a cognitive and a meta-cognitive component. While the cognitive component represents knowledge, the meta-cognitive component has a dynamic model of the cognitive component and enables a self-regulatory learning mechanism for the meta-cognitive neural network. The self-regulatory learning mechanism of the meta-cognitive component uses certain knowledge measures to dynamically monitor the performance of the cognitive component and chooses appropriate learning strategy for each sample in the training set, thus controlling the learning ability of the cognitive component. It has been shown in the studies by [3]–[10] that the meta-cognitive component improves the generalization capability of the network through identifying and learning novel knowledge, while avoiding repetitive knowledge.

All the aforementioned studies were aimed at learning patterns from data samples, as against representing the distribution of samples. The Restricted Boltzmann Machine is a probabilistic model that models the probability distribution of the input features represented by the input nodes using a set of stochastic features (variables) represented by the hidden nodes. Training an RBM to solve classification problems involve a generative training in an unsupervised manner to represent the probability distribution of the samples, followed by a discriminative training using a supervised learning algorithm (eg. feedforward neural network). In the discriminative training phase, the RBM learnt features obtained from the generative training are used to solve the regression or classification task. Feature representation with RBM using all the samples in the training data set may bias the representation towards the densely populated sample space of the training data set. Therefore, in this paper, we develop a Meta-cognitive Restricted Boltzmann Machine (McRBM) and study the effect of meta-cognition in effective representation of sample distribution. McRBM has a cognitive Restricted Boltzmann Machine (RBM) to represent the probability distribution of the input samples and a meta-cognitive component with a self-regulated representational ability to decide *what-to-represent* and *when-to-represent*. In each epoch of the learning process, as each batch of samples is presented to the RBM, the meta-cognitive component uses the error between the input features and the RBM represented features to choose one of the following representational strategies:

- Sample deletion: If the distribution from which the cur-

Fig. 1. The overall block diagram of the McRBM with the generative and discriminative training phases.

rent sample is drawn is well-represented by the RBM, the sample is deleted from the data set.

- Feature representation: If the current sample is novel, it is used to update the probability distribution already represented by the RBM.
- Sample reserve: If the current sample satisfies neither of the above conditions, it is pushed to the reserve end of the data set, and is reserved for use in the future.

The feature representation strategy ensures better representation of the data set distribution, and the sample deletion strategy avoids over-representation of the densely populated sample space. A feedforward neural network with back propagation learning uses the features generated by RBM to perform classification task. The backpropagation learning algorithm of the feedforward neural network is derived based on the hinge-loss error function that is a better predictor of the posterior probability than the mean-squared error function [11].

We demonstrate the classification ability of the McRBM by evaluating its performance to score credits, using three data sets, namely, the German Credit scoring data set and the Australian credit scoring data set from the UCI machine learning repository ([12]) and the KAGGLE credit data set (https://www.kaggle.com/c/GiveMeSomeCredit). In all these problems, the performance of McRBM is compared against those of Support Vector Machines (SVM), Multi-layer Perceptron with Back Propogation (MLP-BP), Extreme Learning Machines (ELM), and Classification Restricted Boltzmann Machines (ClassRBM) [13]. From the results of the performance study, it can be seen that the meta-cognitive component and the hinge-loss error function of McRBM enables it with better generalization ability than other state-of-the-art classifiers, including ClassRBM.

The paper is organized as follows: We present the learning algorithm of the Meta-cognitive Restricted Boltzmann Machine in Section II. The classification ability of McRBM in credit scoring is presented in III. Finally, Section IV summarizes the study.

II. A Hybrid Meta-cognitive Restricted Boltzmann Machine

Let $\left\{\left(\mathbf{x}^1, c^1\right), \ldots, \left(\mathbf{x}^t, c^t\right), \ldots \left(\mathbf{x}^N, c^N\right)\right\}$ be the training data set, where $\mathbf{x}^t \in \Re^m = [x_1^t, \ldots, x_j^t, \ldots, x_m^t]$ are the inputs of the t-th sample and $c^t \in \{1, 2, \ldots, s\}$ is its corresponding class label, s is the total number of classes, m is the number of input features, and N is the total number of samples. The objective of the McRBM is to approximate the functional relationship between the inputs and their targets as closely as possible. We define the coded class labels $\mathbf{y}^t = [y_1^t, \cdots, y_s^t]$ as

$$y_i^t = \begin{cases} 1 & \text{if } \; c^t = i \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots s; \tag{1}$$

The McRBM has a generative and discriminative training phases as shown in Fig. II. In the generative training phase, a meta-cognitive restricted Boltzmann machine is trained in an unsupervised fashion to efficiently generate stochastic features that are most representational of the input data. The discriminative training phase involves training a feedfoward neural network using backpropagation learning algorithm based on the features generated by McRBM. The backpropagation learning algorithm of the feedforward neural network is derived based on the hinge-loss error function.

A. Generative Training: Meta-cognitive Restricted Boltzmann Machine

The generative training phase of McRBM involves two components, namely, a cognitive component and a meta-cognitive component. A Restricted Boltzmann Machine (RBM) [14] with one hidden layer is the cognitive component of McRBM. The self-regulated representational ability of the meta-cognitive component decides *what-to-represent*, *how-to-represent* and *when-to-represent* for the cognitive RBM.

1) Cognitive component: Restricted Boltzmann Machine: A Restricted Boltzmann Machine (RBM) [14] has stochastic visible and hidden layers, connected through symmetric weights. The neurons in the visible layer ($\mathbf{v} = [v_1 \ldots v_m]^T$) of a RBM correspond to the input features of the data ($\mathbf{x}^t$). The neurons in the hidden layer ($\mathbf{h} = [h_1 \ldots h_K]^T$) model the probability distribution of the inputs. Let the symmetric connection weights between the visible and the hidden layers of the RBM be given by $w_{ji}; \; j = 1, \ldots, m, \; i = 1, \ldots, K$.

Then the energy of the joint configuration of ($\mathbf{v}$, $\mathbf{h}$) is given by:

$$E(\mathbf{v}, \mathbf{h}) \;\; := \;\; -\sum_{i,j} v_j w_{ji} h_i - \sum_j c_j v_j - \sum_i b_i h_i \tag{2}$$

where w_{ji} are the symmetric connecting weights, $c_j, j = 1, \ldots, m$ and $b_i, i = 1, \ldots, K$ are the bias of the visible and the hidden neurons, respectively. The joint probability distribution of the neurons in the input and hidden layers is:

$$P(\mathbf{v}, \mathbf{h}) \;\; = \;\; \frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h})} \tag{3}$$

where Z is the partition function (normalization constant) defined by:

$$Z \;\; = \;\; \sum_i \sum_j exp^{-E(\mathbf{v}, \mathbf{h})} \tag{4}$$

The neurons in the same layer of a RBM are not connected. Hence, the conditional probability of a configuration of the visible neurons $\mathbf{v}$, given a configuration of the hidden neurons $\mathbf{h}$, is given by:

$$P(\mathbf{v}|\mathbf{h}) \;\; = \;\; \prod_{j=1}^{m} P(v_j|\mathbf{h}) \tag{5}$$

Similarly, the conditional probability of a configuration of the hidden neurons $\mathbf{h}$, given a configuration of the visible $\mathbf{v}$, is given by:

$$P(\mathbf{h}|\mathbf{v}) = \prod_{i=1}^{K} P(h_i|\mathbf{v}) \quad (6)$$

The individual activation probabilities are given by:

$$P(h_i = 1|\mathbf{v}) = \sigma\left(b_i + \sum_{j=1}^{m} w_{ji} v_j\right) \quad (7)$$

$$P(v_j = 1|\mathbf{h}) = \sigma\left(c_j + \sum_{i=1}^{K} w_{ji} h_i\right) \quad (8)$$

The objective of the generative training phase is to model the marginal distribution of the visible units $P(\mathbf{v})$. In maximum likelihood, the aim is to minimize the negative log probability of the training data,

$$L_{gen} = -\sum_{v \in N} log\ P(\mathbf{v}|(w_{ji}, b_i, c_j)) \quad (9)$$

Although the generative RBM represents the probability distribution of the inputs, the representation does not consider the imbalances of the data in the input space. Therefore, samples from the densely populated input space are well represented, while samples from sparsely populated input space are not well represented. This may affect the generalization ability of the RBM. However, efficient representation requires that the approximation of the distribution is even across all the regions of the input space. Below, we provide a meta-cognitive paradigm that enables efficient representation through a self-regulatory representational mechanism.

2) Meta-cognitive component: Self-regulatory representational mechanism: The meta-cognitive component of the McRBM is similar to the meta-cognitive component of the meta-cognitive fully complex-valued radial basis function network described in [10]. The meta-cognitive component of McRBM measures the feature representation ability of the RBM for a given sample through the representational error (E_{cd}) (10), and uses this error to decide *what-to-represent*, *when-to-represent* and *how-to-represent* in a self-regulatory representational mechanism.

$$E_{cd} = \sqrt{\frac{1}{m}(\mathbf{x}^t - \widehat{\mathbf{x}}^t)^2} \quad (10)$$

where $\mathbf{x}^t$ is the input of the sample t (corresponding to the visible neuron of sample t), and $\widehat{\mathbf{x}}^t$ is the RBM represented stochastic features (corresponding to the responses of the hidden neurons) for the sample t. The decision making is enabled through a set of strategies namely, the sample deletion, the feature representation and the sample reserve strategies. These strategies are briefly described below:

- **Sample Deletion**: This criteria addresses the *what-to-represent* component of meta-cognition. IF $E^t_{cd} < E_d$, where E_d is the delete threshold, then the sample t is deleted from the training data set. The threshold E_d is chosen based on the desired accuracy.
- **Feature Representation**: This criteria addresses the *how-to-represent* component of meta-cognition. If the feature representation condition given by:

$$\text{IF} \quad E^t_{cd} \geq E_l \quad (11)$$

is satisfied in the current epoch, then the current sample is included to represent the learned features at the hidden layer of the ClassRBM in the current epoch. Here, E_l is the representational feature update threshold that is self-regulated based on the representational error for the sample in the current epoch:

$$\text{IF } E^t_{cd} \geq E_l, \text{ THEN } E_l := \delta E_l - (1-\delta) E^t_{cd} \quad (12)$$
$$(13)$$

where δ is the slope at which the thresholds are self-regulated.
- **Sample Reserve**: This criteria decides when-to-represent a sample. A sample that does not satisfy either of the above criteria is reserved for now, and may be a candidate for representation in future. This is enabled by the self-regulatory nature of the representational feature update threshold.

For complete details of the meta-cognitive strategies, one may refer to [10].

B. Discriminative Training: Feedforward Neural Network

The features generated by the generative McRBM are used in classification through discriminative training. A Multi-Layer Perceptron (MLP) with sigmoidal activation function is used as the discriminative network for classification. During the training of the discriminative MLP, the gradients and hence, parameter updates are based on the hinge-loss error function defined by [11] as:

$$eo^t_i = \begin{cases} 0 \ \text{ if } \ (y^t_i . \widehat{y}^t_i) > 0 \\ y^t_i - \widehat{y}^t_i \quad \text{otherwise} \end{cases} \quad i = 1, 2, \cdots s \quad (14)$$

where, $\widehat{y}^t_i$ is the predicted output of the network. The hinge-loss error function is a better predictor of the posterior probabilities (than the mean squared error function).

III. Performance Study: Credit Scoring

Credit scoring is the problem in financial lending/banking business and it helps to estimate the probability that a borrower may default in the future. Several models have been applied to the problem of credit scoring and the performances of these models have been studied and compared by [15], [16]. However, credit scoring is an ongoing problem of research and industrial interest. The credit scoring problem has been solved using a Students-t hidden Markov models, Random Forests, fuzzy logistic regression and Spline based survival model by [17], [18], [19] and [20], respectively. Ensemble approaches have been shown to be efficient classifiers than single classifers and the problem has been solved using ensemble approaches by [21], and [22]. In addition to these, [23] used a classifier consensus system comprising of an ensemble of logistic regression, neural networks, support vector machines, random forests, decision trees and nave Bayes classifier, to solve the problem. Researchers have also attempted to exploit the efficient representation of the deep learning techniques for the problem of credit scoring. [24] have solved the credit scoring problem using ClassRBM. In this section, we study the performance of McRBM for credit scoring, using three credit scoring data sets: The Australian credit data set, the German credit data set and the KAGGLE credit data set (https://www.kaggle.com/c/GiveMeSomeCredit). The Australian and German credit data sets are available in the UCI machine learning repository, and are made available by [12]. We first present the related literature on credit analytics, then describe the data sets, and finally, present the results of McRBM in classifying the candidates according to their credit scores.

A. *Credit Scoring data sets used in this study*

The details of the data set, along with the the number of classes, the number of training and testing samples, and their imbalance factors are presented in Table I. From the table, it can be seen that the German credit and Australian credit data sets have 1000 samples with 24 features and 690 samples with 14 features, respectively. The KAGGLE credit data set has 150000 samples with 10 features. All the three data sets are unbalanced data sets. We measure the extent of imbalance through the Imbalance Factor ($I.F.$) defined as:

$$I.F. \quad = \quad 1 - \frac{s}{N} \min_{i=1 \cdots s} \; N_i \tag{15}$$

where s is the total number of classes and N_i is the number of samples in class i. The KAGGLE credit data is highly unbalanced with an $I.F.$ of 0.86632.

TABLE I
DESCRIPTION OF THE CREDIT SCORE DATA SETS

Problem	No. of features	No. of classes	No. of samples	$I.F.$
German Credit	24	2	1000	0.4
Australian Credit	14	2	690	0.1101
KAGGLE Credit	10	2	150000	0.86632

The Australian credit and German credit data sets are complete, with no missing values. However, the KAGGLE credit data set has missing values of the features monthly income and number of dependents, for a few samples. As both these features depend on the age of the candidate, we group the data sets based on the age group. We then obtain the average of the monthly income of each age group to fill in the missing monthly income of candidates of same age group. Similarly, we compute the average of the number of dependents of each age group to fill in the missing number of dependents of the same age group.

B. *Credit Scoring using McRBM*

This study is conducted by training the networks with 70% of the samples and testing the network with the remaining 30% of the samples, for all the three credit scoring data sets. We present the results of the McRBM classifier, in comparison with the SVM, NN-MLP, ELM, ClassRBM classifiers on the three credit data sets (listed in Table I) in Table II. We measure the performances of the classifiers using the overall efficiency (η_O) and the average efficiency (η_A) are defined as:

$$\eta_O \quad = \quad \frac{\sum_{i=1}^{s} q_{ii}}{N} \text{X}100\% \tag{16}$$

$$\eta_A \quad = \quad \frac{1}{s} \sum_{i=1}^{s} \frac{q_{ii}}{N_i} \text{X}100\% \tag{17}$$

where q_{ii} and N_i are the the number of correctly classified samples in class i and the total number of samples in class i, respectively. In addition, we also compare the performances of the classifiers using True Positive Rate (TPR), True Negative Rate (TNR), and Geometric mean accuracy (Gmean)[TP: True Positives; FN: False Negatives; TN: True Negatives; FP: False Positives].

$$\text{TPR} \quad = \quad \frac{\text{Number of TP}}{\text{Number of TP} + \text{Number of FN}} \tag{18}$$

$$\text{TNR} \quad = \quad \frac{\text{Number of TN}}{\text{Number of TN} + \text{Number of FP}} \tag{19}$$

$$\text{Gmean} \quad = \quad \sqrt{\text{TPR} \times \text{TNR}} \tag{20}$$

From the table, it can be seen that the McRBM outperforms the other classifiers, viz., the SVM, ELM, NN-MLP, and ClassRBM classifiers in the classification, by atleast 1%. The meta-cognitive component helps to improve the classification accuracy over the ClassRBM classifier, and the hinge-loss error function helps the McRBM classifier to perform better classification using unbalanced data sets. It must be noted that the McRBM deleted 36, 124 and 7 samples in the German, Australian and KAGGLE data sets, respectively, during the training. The training and testing overall and average efficiencies of the McRBM are better than those of the SVM, NN, ELM and ClassRBM in the German and Australian credit data sets by at least 3%. However, in the highly unbalanced KAGGLE data set, the training overall and average efficiencies of the McRBM is higher than those of the other state-of-the-art classifiers. However, although the testing overall efficiency is less than that of the ELM and ClassRBM, the testing average efficiency is higher. It must be noted in a highly unbalanced problem, classification is usually biased towards the class with higher samples, and the average efficiency is the optimal indicator of the classifier performance. Performance comparisons also show that the TPR, TNR and Gmean of McRBM is higher than those of SVM, NN, ELM, and ClassRBM. The meta-cognitive component of McRBM during the generative phase and the hinge loss error function during the discriminative phase help in improving the performance of McRBM.

TABLE II
PERFORMANCE RESULTS ON THE CREDIT SCORING DATA SETS: OVERALL AND AVERAGE EFFICIENCIES

Data set	Classifier	K	Training		Testing				
			η_O	η_A	η_O	η_A	TPR	TNR	Gmean
German Credit Dataset	SVM	534	76.429	66.679	74.667	61.378	0.3255	0.8878	0.5376
	NN	60	98.571	97.573	72.333	65.105	0.4574	0.8446	0.6216
	ELM	80	78.429	69.950	73.333	66.355	0.5	0.8271	0.6431
	ClassRBM	80	77.428	63.346	74.000	56.738	0.4418	0.8271	0.6045
	McRBM	**80**	**83.571**	**73.993**	**77**	**61.297**	**0.5116**	**0.8458**	**0.6578**
Australian Credit Dataset	SVM	192	85.507	86.263	85.507	86.048	0.7946	0.9263	0.8579
	NN	60	94.824	94.767	84.058	83.727	0.7917	0.8828	0.836
	ELM	80	88.199	87.875	85.990	85.881	0.8438	0.8738	0.8587
	ClassRBM	50	86.128	86.391	85.507	86.021	0.8953	0.8264	0.8602
	McRBM	**50**	**86.335**	**86.373**	**88.889**	**88.807**	**0.8762**	**0.9**	**0.8881**
KAGGLE Credit Dataset	SVM	6340	69.970	59.430	72.240	60.018	0.5771	0.8982	0.72
	NN	60	63.896	62.287	74.200	63.017	0.6165	0.8792	0.7363
	ELM	100	75.112	73.144	87.667	73.768	0.501	0.7593	0.6167
	ClassRBM	100	75.687	74.048	86.160	74.789	0.6	0.8975	0.73384
	McRBM	**100**	**76.4618**	**75.1587**	**84.8133**	**75.9574**	**0.6573**	**0.8619**	**0.7527**

IV. CONCLUSIONS

Meta-cognitive neural networks are known to exhibit better generalization behavior, owing to their self-regulatory learning ability that help them decide *what-to-learn*, *when-to-learn* and *how-to-learn*. This paper presents a efficient feature representation framework through a meta-cognitive restricted Boltzmann Machine (McRBM). Training in McRBM is enabled through a generative phase and a discriminative phase. Generative training involves efficient generation of stochastic features through a meta-cognitive RBM. A cognitive RBM and a meta-cognitive self-regulatory representational mechanism are the two components of the generative McRBM. The self-regulatory representational mechanism chooses either of sample deletion, feature representation or sample reserve strategies, for every sample in each epoch, depending on how well the distribution from which the sample is drawn is represented by the network. The discriminative training is enabled through a MLP trained through back propagation, which is derived based on a hinge-loss error function. Performance studies on benchmark and real-world problems, in comparison with state-of-the-art algorithms show improved generalization ability of McRBM.

REFERENCES

[1] T. O. Nelson and L. Narens, *Metacognition: Core Readings*, t. o. nelson (ed.) ed. Allyn and Bacon: Boston, 1980, ch. Metamemory: A theoritical framework and new findings, pp. 9 – 24.

[2] D. P. Josyula, H. Vadali, B. J. Donahue, and F. C. Hughes, “Modeling metacognition for learning in artificial systems,” *Proceedings of 2009 World Congress on Nature and Biologically Inspired Computing, (NABIC 2009)*, 2009.

[3] G. S. Babu and S. Suresh, “Meta-cognitive neural network for classification problems in a sequential learning framework,” *Neurocomputing*, vol. 81, pp. 86–96, 2012.

[4] ——, “Sequential projection-based metacognitive learning in a radial basis function network for classification problems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 2, pp. 194–206, 2013.

[5] R. Savitha, S. Suresh, and H. J. Kim, “A meta-cognitive learning algorithm for an extreme learning machine classifier,” *Cognitive Computation*, vol. 6, no. 2, pp. 253–263, 2014.

[6] K. Subramanian, A. K. Das, S. Suresh, and R. Savitha, “A meta-cognitive interval type-2 fuzzy inference system and its projection based learning algorithm,” *Evolving Systems*, vol. 5, no. 4, pp. 219–230, 2014.

[7] G. S. Babu, R. Savitha, and S. Suresh, “A projection based learning in meta-cognitive radial basis function network for classification problems,” *Proceedings of the International Joint Conference on Neural Networks*, p. (Accepted), 2012.

[8] R. Savitha, S. Suresh, and N. Sundararajan, “A meta-cognitive learning algorithm for a fully complex-valued relaxation network,” *Neural Networks*, vol. 32, pp. 209–218, 2012.

[9] K. Subramanian, S. Suresh, and R. Savitha, “A metacognitive complex-valued interval type-2 fuzzy inference system,” *IEEE Transactions on Neural Networks and their Learning Systems*, vol. 25, no. 9, pp. 1659–1672, 2014.

[10] R. Savitha, S. Suresh, and N. Sundararajan, “Metacognitive learning in a fully complex-valued radial basis function neural network,” *Neural Computation*, vol. 24, no. 5, pp. 1297 – 1328, 2012.

[11] S. Suresh, N. Sundararajan, and P. Saratchandran, “Risk-sensitive loss functions for sparse multi-category classification problems,” *Information Sciences*, vol. 178, no. 12, pp. 2621–2638, 2008.

[12] C. Blake and C. Merz, “UCI repository of machine learning databases,” *Department of Information and Computer Sciences, University of California, Irvine, [URL: http://archive.ics.uci.edu/ml/]*, 1998.

[13] H. Larochelle, M. Mandel, R. Pascanu, and Y. Bengio, “Learning Algorithms for the Classification Restricted Boltzmann Machine,” *Journal of Machine Learning Research*, vol. 13, pp. 643–669, 2012.

[14] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[15] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, “Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research,” *European Journal of Operational Research*, vol. 247, no. 1, pp. 124–136, 2015.

[16] B. Baesens, T. V. Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, “Benchmarking state-of-the-art classification algorithms for credit scoring,” *The Journal of the Operational Research Society*, vol. 54, no. 6, pp. 627–635, 2003.

[17] A. Petropoulos, S. Chatzis, and S. Xanthopoulos, “A novel corporate credit rating system based on Student’s-t hidden Markov models,” *Expert Systems with Applications*, vol. 53, pp. 87–105, 2016.

[18] M. Malekipirbazari and V. Aksakalli, “Risk assessment in social lending via random forests,” *Expert Systems with Applications*, vol. 42, no. 10, pp. 4621–4631, 2015.

[19] S. Y. Sohn, D. H. Kim, and J. H. Yoon, “Technology credit scoring model with fuzzy logistic regression,” *Applied Soft Computing*, vol. 43, pp. 150–158, 2016.

[20] S. R. Luo, X. Kong, and T. T. Nie, “Spline based survival model for credit risk modeling,” *European Journal of Operational Research*, vol. 253, no. 3, pp. 869–879, 2016.

[21] R. F. Lopez and J. M. R. Jeronimo, “Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment: A correlated-adjusted decision forest proposal,” *Expert Systems with Applications*, vol. 42, no. 13, pp. 5737–5753, 2016.
[22] H. S. Xiao, Z. Xiao, and Y. Wang, “Ensemble classification based on supervised clustering for credit scoring,” *Applied Soft Computing*, vol. 43, pp. 73–86, 2016.
[23] M. Alaraj and M. F. Abbod, “Classifiers consensus system approach for credit scoring,” *Knowledge Based Systems*, vol. 104, pp. 89–105, 2016.
[24] M. J. Tomczak and M. B. Zie, “Classification Restricted Boltzmann Machine for comprehensible credit scoring model,” *Expert Systems with Applications*, vol. 42, no. 4, pp. 1789–1796, 2015.