ASM Sc. J., 12, Special Issue 5, 2019 for ICoAIMS2019, 1-9

Hyperparameters Tuning of Random Forest with Harmony Search in Credit Scoring

R.Y. Goh[1], L.S. Lee[1,2*] and M.B. Adam[1,2]

[1]*Laboratory of Computational Statistics and Operations Research, Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia*

[2]*Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia*

Correct identification of defaulters and non-defaulters in the lending industry is a crucial task for financial institutions. Credit scoring is a tool utilized for credit granting decisions. Recently, Random Forest (RF) is actively researched in credit scoring due to two main benefits, i.e. non-parametric flexibility to account for various data patterns with good classification ability and the computed features importance that can explain the attributes. Hyperparameters tuning is a necessary procedure to ensure good performance of a RF. This paper proposes the use of a metaheuristic, Harmony Search (HS), to form a hybrid HS-RF to conduct hyperparameters tuning. A Modified HS (MHS) is also proposed, forming MHS-RF, for effective yet efficient search of the RF hyperparameters. Along with parallel computing, MHS-RF effectively reduces the computational efforts of the hyperparameters tuning procedure. The proposed hybrid models are benchmarked with standard statistical models on the Lending Club peer-to-peer lending dataset. The computational results show that a well-tuned RF have better performance than statistical models, with MHS-RF reported the best performance yet being the most efficient in hyperparameters tuning of RF.

Keywords: credit scoring; random forest; harmony search

I. INTRODUCTION

Credit granting decision to new applicants is an essential risk evaluation task by financial institutions to avoid defaulters that incur cost and identify potential non-defaulters that bring profit. Credit scoring is an evaluation tool to assist the decision-making process, where it is the formal statistical method used to classify applicants for credit into 'good' and 'bad' risk classes as defined in Hand and Henley (1997).

Artificial Intelligence (AI) techniques have been the current research trend as shown in several comparative studies and review papers (Baesens *et al.*, 2003; Goh & Lee, 2019; Hand & Henley, 1997; Lessmann *et al.*, 2015; Louzada *et al.*, 2016; Thomas, 2000). Random Forest (RF) is one of the current research interests in credit scoring due to its competitiveness as recognized by Lessmann *et al.* (2015) and its benefit to explain attributes using the computed features importance as compared to other black box AI models. RF has shown its potential in the credit scoring domain via involvement in comparative studies (Florez-Lopez & Ramon-Jeronimo, 2015; Lessmann *et al.*, 2015) and application in various credit domains (Gorter, 2017; Malekipirbazari & Aksakalli, 2015; Óskarsdóottir *et al.*, 2019; Tang *et al.*, 2018). The competitiveness of RF then led to formulation of new models with RF, i.e. modified RF (Ghatasheh, 2014; Van Sang *et al.*, 2016; Ye *et al.*, 2018), hybrid RF (Jiang *et al.*, 2019; Yeh *et al.*, 2012) and new ensembles (He *et al.*, 2018; Zhang *et al.*, 2018).

Hyperparameters tuning has been the mandatory task for every application of RF in credit scoring where the techniques adopted are manual tuning of repeated trial-and-error experiments (Ghatasheh, 2014; Gorter, 2017; Florez-Lopez & Ramon-Jeronimo, 2015; Óskarsdóottir *et al.*, 2019; Van Sang *et al.*, 2016; Tang *et al.*, 2018; Ye *et al.*,

*Corresponding author's e-mail: lls@upm.edu.my

2018), examination of a particular input range (Lessmann *et al.*, 2015; Malekipirbazari & Aksakalli, 2015) and utilization of specific tuning technique (He *et al.*,2018; Jiang *et al.*, 2019). Manual tuning is the dominant way to tune RF hyperparameters which is mostly based on subjective judgement of the researchers. Examination of a particular input range is similar to Grid Search (GS) but the values are not in a fixed increasing range of a grid, and this method is readily available in some software toolbox. GS, Random Search and Particle Swarm Optimization (PSO) are the specific tuning techniques attempted for RF in this domain.

Despite manual tuning being the most common approach, the study by He *et al.* (2018) utilizing PSO for an automated tuning process is perceived as a starting point for the metaheuristics approach (MA) in tuning RF hyperparameters. Along with some competitive hyperparameters tuning performance presented recently using Genetic Algorithm (GA) (Yu *et al.*, 2011), PSO (Danenas & Garsva, 2015) and Artificial Bee Colony (ABC) (Hsu *et al.*, 2018), MA is a potential technique to be considered for hyperparameters tuning task. To the best of our knowledge, Harmony Search (HS) has yet to be hybridized with RF. Harmony Search (HS) is a metaheuristic first developed by Geem *et al.* (2001). The evolutionary process to seek for optimal solutions is inspired by the music improvisation process, where musicians tune instruments' pitch to achieve perfect harmony. In this study, HS is proposed to tune the RF hyperparameter. A Modified Harmony Search (MHS) is formulated by modifying the HS operators, which is then hybridized with RF for a more efficient search of hyperparameters. Parallel computation with MHS-RF is developed to achieve lower computational effort. Lastly, the two proposed models are assessed with standard statistical credit scoring models namely, Logistic Regression (LOGIT), Backward Stepwise Logistic Regression (STEP) and Linear Discriminant Analysis (LDA), on the latest 36-month loan application of Lending Club peer-to-peer lending.

This paper is organized as follows. Section II describes the algorithm of HS and modifications made on the MHS. Section III hybridizes HS and MHS with RF, developing HS-RF and MHS-RF respectively. Section IV elaborates on the experimental setup. Then, Section V reports the computational results together with detailed discussions. Finally, Section VI concludes the study and provides possible future directions.

II. HARMONY SEARCH

A. Standard HS

A standard HS algorithm imitates the music improvisation process to play from existing music pieces, to adjust pitch according to known music pieces, or to randomly compose new notes. There are five procedures in a standard HS:

i) **Definition of objective function and the required parameters,** i.e. Harmony Memory Size (HMS), Harmony Memory Considering Rate ($HMCR$), Pitch-Adjustment Rate (PAR), bandwidth (bw), and maximum iterations (max_iter).

ii) **Initialization of Harmony Memory (HM)** with HMS number of possible candidate solutions which are generated from uniform distribution based on decision variables range.

iii) **Improvisation to generate new harmony** with $HMCR$ for exploration to control random selection of new harmony from HM, PAR for exploitation to control improvement of new solution to neighbouring values with step size of bw (continuous variables) or one step to the left or right (discrete variables), and $1 - HMCR$ for randomization.

iv) **Update HM** if the new solution has better fitness than the worst solution in HM by replacing it.

v) **Termination** of the repeating third and fourth procedure until max_iter has reached.

B. Modified HS

In seeking for a good quality solution, HS requires a good balance between the exploration and exploitation parameters. Besides, to improve the computational efficiency, HS should converge to a good solution in lesser number of iterations. The main modifications of the proposed MHS are:

i) Elitism selection during memory consideration:

The selection of new harmony is no longer a random selection from HM, but with a tendency to select a better quality harmony. Harmony vectors in HM are divided into two groups, i.e. elite ($g1$) and non-elite ($g2$), where $g1$ consists of harmony vectors with better performance than $g2$. Each harmony vector in HM takes an index number from the sequence of $\{1,\ HMS\}$. Since HM is sorted in the order of best to worst performance, harmony vectors with lower index number indicate their potential as the candidates in the elite group. The first quartile, $q1 = round(0{:}25 \times (HMS + 1))th$ term of the index sequence is computed and acts as the cutoff to divide HM into the elite and non-elite groups where $g1 \in \{1, q1\}$ and $g2 \in \{(q1 + 1),\ HMS\}$. Note that decimal values for $q1$ is rounded up because index values are discrete.

An extra parameter $elit$ is included to allocate a proper weightage on the elite group, so the selected new harmony has higher probability to originate from the elite group. With a probability $elit$, a new harmony is selected from the elite group. If the selection is from the non-elite group, two harmonies will be picked. Then, the better one will be the new harmony. Hence, a better harmony is always selected. Note that a low quality harmony, when joining with other harmony or being adjusted, may also produce good harmony. Thus, $elit$ cannot be too high to ensure a balance to seek from elite and non-elite group. Detailed selection process is illustrated as in Algorithm 1.

Algorithm 1

selection ( )
$g1 \in \{1, q1\}$
$g2 \in \{(q1 + 1), HMS\}$

if $(U(0,1) \leq elit)$
    $ind1 = round\big(g1_{min} + U(0,1) \times (g1_{max} - g1_{min})\big)$
    $ind = ind1$
else
    $ind2 = round\big(g2_{min} + U(0,1) \times (g2_{max} - g2_{min})\big)$
    $ind3 = round\big(g2_{min} + U(0,1) \times (g2_{max} - g2_{min})\big)$
    if $(ind2 \leq ind3)$
        $ind = ind2$
    else
        $ind = ind3$
return ind

ii) Dynamic HMCR and PAR with step function

HMCR follows a step function with the range $[HMCR_{min}, HMCR_{max}]$. Selection probability starts from low to high. Thus, the search process at the start will have higher diversification with $HMCR_{min}$, then increased by step following a step function until $HMCR_{max}$ is reached in the later search process. PAR follows a step function with the range $[PAR_{min}, PAR_{max}]$. In contrast with HMCR, adjustment probability starts from high to low. This results in high exploitation at the start with PAR_{max}, then decreased by step following a step function until PAR_{min} is reached and remained the same till the end.

In utilizing the step function, several components i.e. $HMCR$ range, PAR range, $HMCR$ increment, PAR decrement, and step size $step$ have to be determined. The $HMCR = \{0.70, 0.95\}$ is set based on the recommended value in Yang (2009). To align with the step function for $HMCR$, $PAR = \{0.10, 0.35\}$ is used. The interval for increment and decrement of $HMCR$ and PAR respectively is set at 0.05 as this small interval is sufficient to cover the whole range for these two operators. The step size $step$ determines the number of iterations for $HMCR$ and PAR to maintain before shifting to another value in the range until both operators reach a plateau. The setting of $step$ depends on the search range size and smaller $step$ is preferable as the main aim is to have faster convergence with active exploration and exploitation in the early stage. Thus, $step$ is set to enable both $HMCR$ and PAR to reach a plateau within the first quarter part of the total iterations. Detailed settings are enclosed in Section III-A, Table 1.

iii) Additional termination criteria

The termination criteria used in this study are the maximum number of iterations (max_$iter$), convergence of HM, and non-improvement on the best solution for a fixed number of consecutive iterations ($cons\_no\_imp$). HS procedure will stop when any of the criterion is met.

The modifications in MHS has resulted in several extra parameters, i.e. $elit$, $HMCR_{min}$, $HMCR_{max}$, PAR_{max}, PAR_{min}, $step$ and $cons\_no\_imp$. Modifications (i) and (ii) result in major changes in the improvisation step. The differences are summarized in Algorithm 2 and 3 for HS and MHS respectively.

Algorithm 2

For every decision variable i do
if $(U(0,1) \leq HMCR)$
  $ind = int(U(0,1) \times HMS) + 1$
  $x_i' = x_i^{ind}$
  if $(U(0,1) \leq PAR)$
    $x_i' = x_i' + U(-1,1) \times bw$
    $x_i' = x_i' \pm 1$
else
  $x_i' = LB_i + U(0,1) \times (UB_i - LB_i)$
  $x_i' \in \{LB_i, UB_i\}$

Algorithm 3

$HMCR = HMCR_{iter}$ **a**
$PAR = PAR_{iter}$ **b**
For every decision variable i do
if $(U(0,1) \leq HMCR)$
  $ind = selection(\ \ )$ **c**
  $x_i' = x_i^{ind}$
  if $(U(0,1) \leq PAR)$
    $x_i' = x_i' + U(-1,1) \times bw$
    $x_i' = x_i' \pm 1$
else
  $x_i' = LB_i + U(0,1) \times (UB_i - LB_i)$
  $x_i' \in \{LB_i, UB_i\}$

a: Increasing step function for HMCR instead of a static preset value
b: Decreasing step function for PAR instead of a static preset value
c: Elitism selection from HM instead of random selection

III. HYBRID MODELS FORMULATION

Random Forest is an ensemble model with the collection of decision trees using the bootstrap aggregation technique, or more commonly known as the bagging technique. Trees are grown with binary splitting algorithm with Gini Impurity, $GI = 1 - \sum_{i=1}^{k} p_i^2$ as the splitting criteria, where i is the number of classes and p_i is the proportion of instances belonging to the respective class.

During the tree growing process, to avoid correlations in between the trees, only a subset of the variables are required for splitting. End results of the classification is based on the majority votes from all the collected trees in the forest. The two hyperparameters to be tuned in RF are the number of trees ($ntree$) and number of variables from available attributes ($mtry$).

A. HS-RF and MHS-RF

The HS-RF and MHS-RF follow the same procedure as in HS and MHS respectively, with the RF classification task being the objective function. Area Under Receiver Operating Characteristics (AUC) is the fitness function for both models. The full procedures of HS-RF and MHS-RF, as well as their differences are detailed as follows:

Step 1: Define objective function (eqn 1) and parameters of HS and MHS (Table 1).

The objective function is the RF classification function with two decision variables that corresponds to the two hyperparameters i.e. $ntree$ and $mtry$. The search range for $ntree$ is chosen to be discrete values of $x_1 \in \{1, 5\}$, where these values are then converted to the corresponding hundred. This search range is selected as it is often attempted by researchers. The search range of the second decision variable is discrete values of $x_2 \in \{1, a\}$, where a is the total number of attributes available. This search range is chosen because the hyperparameter $mtry$ is the random subset of variables from the total available attributes.

$$\begin{aligned} \max f(x) = \ & \text{majority vote}\left\{\hat{C}_n^{ntree}(x)\right\} \\ \text{s.t} \quad & x_1 \in \{1, 5\} \\ & x_2 \in \{1, a\} \end{aligned} \tag{1}$$

Table 1. Parameters Settings

HS-RF	MHS-RF
$HMS = 10$	$HMS = 10$
$HMCR = 0.70$	$elit = 0.70$
$PAR = 0.20$	$HMCR_{iter} = \{0.70, 0.95\}$
$max\_iter = 100$	$PAR_{iter} = \{0.10, 0.35\}$
	$step = 5$
	$max\_iter = 100$
	$no\_cons\_imp = 25$

Step 2: Initialization of Harmony Memory

Each harmony vector in HM has two decision variables. Every harmony vector is evaluated with the fitness function and sorted from the best to worst. Since the decision variables to solve RF are discrete, the harmony vectors are sampled directly from the search range as in equation (1). Both HS-RF and MHS-RF have the same HM.

Step 3: Improvisation

The improvisation procedure for HS-RF and MHS-RF are summarized as in Algorithm 4 and 5 respectively.

Algorithm 4

for i in (1:2)
  if $(U(0,1) \leq HMCR)$
    $ind = int(U(0,1) \times HMS) + 1$
    $x_i' = x_i^{ind}$
    if $(U(0,1) \leq PAR)$
      $x_i' = x_i^{ind} \pm 1$
  else
    $x_i' \in \{min(x_i), max(x_i)\}$

Algorithm 5

$HMCR = HMCR_{iter}$ **a**
$PAR = PAR_{iter}$ **b**
for i in (1:2)
  if $(U(0,1) \leq HMCR)$
    $ind = selection(\ )$ **c**
    $x_i' = x_i^{ind}$
    if $(U(0,1) \leq PAR)$
      $x_i' = x_i^{ind} \pm 1$
  else
    $x_i' \in \{min(x_i), max(x_i)\}$

a, **b**: Refer Table 1
c: Refer Algorithm 1

Step 4: Update HM by evaluating and comparing the fitness function of the new harmony with the worst harmony in HM. Replace the worst harmony if the new harmony has better fitness. Same procedure for both HS-RF and MHS-RF.

Step 5: Repeat **Step 3** and **4** until max_*iter* is reached for HS-RF whereas MHS-RF involves two additional criteria i.e. HM converges or *no_cons_imp* reached.

B. Parallel Computing

The proposed hybrid MHS-RF aims for quality yet faster convergence. MHS-RF is executed sequentially over 5 independent tasks resulting from the cross validation procedure. To further enhance the computational efficiency, parallel computing with master-slave concept is employed.

Initially, master generates sub-tasks via data preparation and splitting to be assigned to 5 slaves for independent and simultaneous execution. Lastly, each slave returns required performance measures (refer Section IV-B) to compute their average. Algorithm 6 summarizes the parallel computation. Note that both sequential and parallel execution have the same seeding set to ensure identical results are obtained since the main aim is to enhance the computational time.

Algorithm 6

Master: Data preparation and partitioning
do_parallel
  for i in (1:5)
    Slave: Step 1 – 5 of MHS-RF
return AUC, ACC, ACC*
Master: mean(AUC), mean(ACC), mean (ACC*)

IV. EXPERIMENTAL SETUP

A. Credit Dataset Preparation

The dataset used in this study is retrieved from the Lending Club (LC) peer-to-peer lending which is available online (https://www.lendingclub.com/info/download-data.action) on 4 April 2019. The LC website provides a huge database of customers range from year 2007 till 2019. For the experiment, a sample of the 36-month term loan of first quarter of 2016 is taken because the issued loans at this period have just reached maturity on April 2019, posing as the most recent loans that reached maturity.

To prepare the credit dataset, this experiment focuses only on loan status that are fully paid and charged off. Variables having all empty values or more than 5% missing values are removed, and variables with less than 1% of missing value have the whole instance being removed as it is only a small loss of information. Missing data is imputed with mean (mode) for numerical (categorical) attributes. The resulting dataset has 49,461 issued loans and 23 variables, with 42,178 fully paid and 7,463 charged off customers. Table 2 shows the attributes description of the dataset.

Table 2. Attributes in LC dataset

Attributes	Type
loan_amnt	Numerical
annual_inc	Numerical
dti	Numerical
delinq_2_year	Numerical
earliest_cr_line*	Numerical
inq_last_6mths	Numerical
open_acc	Numerical
pub_rec	Numerical
revol_util	Numerical
total_acc	Numerical
last_credit_pull_d**	Numerical
acc_now_delinq	Numerical
chargeoff_within_12mths	Numerical
delinq_amnt	Numerical
pub_rec_bankruptcies	Numerical
tax_liens	Numerical
inq_fi	Numerical
num_tl_120dpd_2m	Numerical
pct_tl_nvr	Numerical
home_ownership	Categorical
verification_status	Categorical
purpose	Categorical
initial_list_status	Categorical

The full name of the attributes details can be found in the LCDataDictionary.xls file in the LC website.
*Transformed to how many years since first credit line opened.
**Transformed to how many months since LC pulled credit.

Numerical attributes are standardized by subtracting the column mean and dividing the standard deviation. Categorical attributes are converted to numerical attributes with the weight-of-evidence (WOE) transformation as discussed in Thomas (2000). 5-fold cross validation is applied on the dataset, and a holdout set is prepared as the validation set for hyperparameters tuning procedure to avoid overfitting problem.

B. Performance Evaluation

Accuracy (ACC) is the proportion of correctly classified instances in the data. ACC is a direct evaluation of the model performance but it is threshold-variant, where the value changes when the threshold, τ changes. Determining τ is usually assumption-based and posed as a problem in making concrete conclusion with ACC alone (Baesens *et al.*, 2003). Hence, a threshold-invariant measure, AUC is also reported in this study. AUC gives a better picture on the discriminating ability of a model as it is the probability that a classifier will rank a randomly chosen positive example than a randomly chosen negative example (Fawcett, 2006).

Friedman test is conducted to test the significance of AUC between the compared models across the 5 test sets (from cross validation) for each dataset. Friedman test has been a popular significance test as it has been used in benchmark study (Lessmann *et al.*, 2015). Post-hoc Nemenyi test is applied if there is significant difference reported from Friedman test.

V. RESULTS AND DISCUSSIONS

The proposed models are coded in R 3.5.1 and executed on 2.70GHz Intel(R) Core(TM) i7-7500CPU with 4.00GB RAM under Windows 10 operating system. For parallel computation, the parallel environment is initiated with the 'doParallel' library in R 3.2.5 and executed on Linux based operating system using IBM system X360 M4 server with ten nodes of 2.0GHz Intel Xeon 6C processors. The experimental results are discussed based on model performances and computational time as reported in Table 3. Model performances are evaluated with AUC and accuracy with two thresholds (ACC of default and ACC* specified for maximum accuracy). The two proposed hybrid models are compared with statistical models LOGIT, STEP and LDA, as well as RF tuned using Grid Search (GS). The search range of RF with GS is the same as the input range of the hybrid models as described in Section III-A. MHS-RF (P) denotes the parallelized MHS-RF, with the same seeding applied, hence resulting in only the difference in computational time. The best performances are reported in bold.

Table 3. Model Performances

	AUC	ACC	ACC*	Time
LOGIT	0.7468	0.8488	0.8508	9.77s
STEP	0.7468	0.8485	0.8508	5.11min
LDA	0.7481	0.8451	0.8498	6.55s
RF	0.7702	0.8579	0.8593	7.96hrs
HS-RF	0.7706	**0.8586**	0.8592	9.56hrs
MHS-RF	**0.7712**	0.8582	**0.8597**	4.57hrs
MHS-RF (P)	**0.7712**	0.8582	**0.8597**	1.43hrs
$\chi^2_{friedman} = 22.701, (0.000385)$				

For AUC, ACC and ACC*, models from RF family have consistently outperformed the statistical models, indicating the flexibility of RF in capturing the dataset pattern. Among the three different tuning techniques for RF, the proposed

HS-RF and MHS-RF have consistently performed better than the GS-tuned RF, showing HS and MHS are competitive for hyperparameters tuning. The highest AUC and ACC* reported from MHS-RF indicate that MHS is effective in the tuning process. All ACC* are higher than ACC which is due to the threshold setting to obtain maximum accuracy, implying a change of threshold will change the accuracy evaluation, resulting in HS-RF to be the best in ACC while MHS-RF to be the best in ACC*. RF family is in general better than the statistical models, with HS-RF and MHS-RF are better than GS-tuned RF.

AUC is the main focus in this study to evaluate the performance due to its threshold invariant property. The Friedman test statistics and p-value (in parenthesis) is reported in the last row of Table 3. The results shows significant differences between the five models. The corresponding post-hoc test with the p-values are reported in Table 4, where pairs with significant difference at 5% significance are bolded.

Table 4. Post-hoc Nemenyi Test

	LOGIT	STEP	LDA	RF	HS-RF	MHS-RF
LOGIT	-	-	-	-	-	-
STEP	1.00	-	-	-	-	-
LDA	1.00	0.77	-	-	-	-
RF	0.11	**0.048**	0.77	-	-	-
HS-RF	**0.032**	**0.010**	0.30	1.00	-	-
MHS-RF	**0.018**	**0.004**	0.24	1.00	1.00	-

The AUC in Table 3 show better performance of RF family as compared to statistical models, and also better performance of the proposed hybrid models than the GS-tuned RF. According to the post-hoc Nemenyi test, there are only four pairs with significant difference i.e. both proposed hybrid models are significantly better than LOGIT and STEP. This implies that the hybrid HS-RF and MHS-RF are able to improve RF performances.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

This study presents HS and MHS to be hybridized with RF for hyperparameters tuning. Both are proposed as an alternative to the usual manual repeated trial-and-error or GS approach in past research to tune RF hyperparameters. HS operators pose the flexibility to be modified to develop MHS for effective yet efficient hyperparameters search.

The experiment is conducted on a real-life dataset from the Lending Club peer-to-peer lending loans which just reached maturity. Evaluation of the models based on AUC and ACC have shown the non-parametric RF performs better than standard statistical models of LOGIT, STEP and LDA. AUC is viewed as the main focus of performance evaluation due to its threshold-invariant property. HS-RF and MHS-RF have shown improvements compared with GS-tuned RF; indicating the effectiveness of HS and MHS as a potential MA to tune hyperparameters. The ability of MHS to perform hyperparameters tuning in a much shorter time shows the flexibility of HS to be modified. In addition, parallel computing has further enhanced computational efficiency. Statistical models may be more efficient than RF that requires hyperparameters tuning. But, to achieve a higher performance measure, additional procedure such as identification of interaction terms may be time-consuming as well.

HS and MHS have been demonstrated as a competitive tool in hyperparameters tuning for RF. In consideration of both model performance and computational effort, MHS-RF is concluded as the potential alternative for credit scoring. For possible future directions, the HS can be hybridized for hyperparameters tuning with other AI techniques as well. Besides, this study does not solve the black box property of RF. Possible future work can be focused to incorporate the features importance from RF for rules extraction.

VII. ACKNOWLEDGEMENT

This research was supported by Geran Putra-Inisiatif Putra Siswazah (GP-IPS/2018/9646000) (Universiti Putra Malaysia). The authors would like to thank reviewers for their time to thoroughly review and provide constructive comments for improvements of the manuscript.

VIII. REFEREENCES

Baesens, B, Van Gestel, T, Viaene, S, Stepanova, M, Suykens, J & Vanthienen, J 2003, 'Benchmarking state-of-the-art classification algorithms for credit scoring', *Journal of the Operational Research Society,* vol. 54, no. 6, pp. 627-635.

Danenas, P & Garsva, G 2015, 'Selection of support vector machines based classifiers for credit risk domain', *Expert Systems with Applications*, vol. 42, no. 6, pp. 3194-3204.

Fawcett, T 2006, 'An introduction to ROC analysis', *Pattern Recognition Letters,* vol. 27, no. 8, pp. 861-874.

Florez-Lopez, R & Ramon-Jeronimo, JM 2015, '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.

Geem, ZW, Kim, JH & Loganathan, GV 2001, 'A new heuristic optimization algorithm: harmony search', *Simulation*, vol. 76, no. 2, pp. 60-68.

Ghatasheh, N 2014, 'Business analytics using random forest trees for credit risk prediction: A comparison study', *International Journal of Advanced Science and Technology,* vol. 72, pp. 19-30.

Goh, RY & Lee, LS 2019, 'Credit scoring: a review on support vector machines and metaheuristic approaches', *Advances in Operations Research,* vol. 2019, pp. 1-30.

Gorter, D 2017, 'Added Value of Machine Learning in Retail Credit Risk Financial Engineering and Management', PhD thesis, University of Twente, Enschede, Netherlands.

Hand, DJ & Henley, WE 1997, 'Statistical classification methods in consumer credit scoring: a review', *Journal of the Royal Statistical Society: Series A (Statistics in Society),* vol. 160 no. 3, pp. 523-541.

He, H, Zhang, W & Zhang, S 2018, 'A novel ensemble method for credit scoring: Adaption of different imbalance ratios', *Expert Systems with Applications,* no. 98, pp. 105-17.

Hsu, FJ, Chen, MY & Chen, YC 2018, 'The human-like intelligence with bio-inspired computing approach for credit ratings prediction', *Neurocomputing,* vol. 279, pp. 11-18.

Jiang, C, Wang, Z & Zhao, H 2019, 'A prediction-driven mixture cure model and its application in credit scoring', *European Journal of Operational Research,* no. 277, vol. 1, pp. 20-31.

Lessmann, S, Baesens, B, Seow, HV & Thomas, LC 2015, 'Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research', *European Journal of Operational Research*, no. 247 vol. 1, pp. 124-136.

Louzada, F, Ara, A & Fernandes, GB 2016, Classification methods applied to credit scoring: Systematic review and overall comparison', *Surveys in Operations Research and Management Science,* vol. 21, pp. 117-134.

Malekipirbazari, M & Aksakalli, V 2015, 'Risk assessment in social lending via random forests', *Expert Systems with Applications,* vol. 42, no. 10, pp. 4621-4631.

Óskarsdóottir, M, Bravo, C, Sarraute, C, Vanthienen, J & Baesens, B 2019, 'The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics', *Applied Soft Computing,* vol. 74, pp. 26-39.

Tang, L, Cai, F & Ouyang, Y 2018, 'Applying a nonparametric random forest algorithm to assess the credit risk of the energy industry in China', *Technological Forecasting and Social Change,* vol. 144, pp. 563-572.

Thomas, LC 2000, 'A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers', *International Journal of Forecasting*, vol. 16, no. 2, pp. 149-172.

Van Sang, H, Nam, NH & Nhan, ND 2016, 'A novel credit scoring prediction model based on feature selection approach and parallel random forest', *Indian Journal of Science and Technology,* vol. 9,

no. 20, pp. 1-6.

Yang, XS 2009, *Music-Inspired Harmony Search Algorithm: Theory and Applications,* Berlin: Springer, pp. 1-14.

Ye, X, Dong, LA & Ma, D 2018, 'Loan evaluation in P2P lending based on random forest optimized by genetic algorithm with profit score', *Electronic Commerce Research and Applications,* vol. 32, pp. 23-36.

Yeh, CC, Lin, F & Hsu, CY 2012, 'A hybrid KMV model, random forests and rough set theory approach for credit rating', *Knowledge-Based Systems,* vol. 33, pp. 166-172.

Yu, L, Yao, X, Wang, S & Lai, KK 2011, 'Credit risk evaluation using a weighted least squares SVM classifier with design of experiment for parameter selection', *Expert Systems with Applications,* vol. 38, no. 12, pp. 15392-15399.

Zhang, H, He, H & Zhang, W 2018, 'Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring', *Neurocomputing,* vol. 316, pp. 210-221.