http://infor.seaninstitute.org/index.php/infokum/index
JURNAL INFOKUM, Volume 9, No. 2, Juni 2021 ISSN : 2302-9706

KNN METHOD ON CREDIT RISK CLASSIFICATION WITH BINARY PARTICLE SWARM OPTIMIZATION BASED FEATURE SELECTION

Harmoko Lubis [1], Pahala Sirait[2] , Arwin Halim[3]
Master of Information Technology, STMIK Mikroskil, Medan, Sumatera Utara, Indonesia[1,2,3]
lubisharmoko@gmail.com,sirpahala@yahoo.com, arwin@mikroskil.ac.id

Abstract

Article Info
Received, 06 Mei 2021
Revised, 01 June 2021
Accepted, 20 June 2021

Today, classification performance has become increasingly important for credit risk assessment for loss control and revenue maximization. Therefore, a classification method is required that can accurately and efficiently measure the credit risk level of prospective borrowers as the key to the credit approval process. This study contributes to the development of feature selection methods with SI algorithms that use binary representation, namely feature selection using PSO algorithms with binary representation or *Binary Particle Swarm Optimization* (BPSO) applied to credit risk classification, with classification evaluation using kNN classification method. The application of feature selection is done to eliminate excessive features, thus reducing the number of features, improving the accuracy of the model, and reducing running time. The test results showed that KNN's best accuracy of 76.40%, can be improved by bpso-based selection feature with better accuracy of 88.70%, with an accuracy improvement of 13.35%. This test showed that bpso-based selection feature technique successfully improved the accuracy of KNN classification on credit risk classification.

Keywords: Coronary Heart Classification, KNN, Selection Features, Binary PSO

1. Introduction

One of the most important procedures at bank institutions is credit risk evaluation, also known as credit management decisions involving data collection, data analysis and classification of different credit variables to arrive at credit decisions (Sánchez & Lechuga, 2016). Credit risk occurs when a borrower in a debt contract defaults or delays debt repayment, either in whole or in part (Zamore et al., 2018). Credit risk prediction, monitoring, model reliability, and effective loan processing are key to decision making and transparency (Addo et al., 2018). Reliable evaluation model for credit risk plays an important role in loss control and revenue maximization (Chen et al., 2016). To improve the quality of lending and to reduce the risks involved in this process, several credit scoring models have been developed and utilized to improve the creditworthiness assessment process (Sameer et al., 2019). Today, classification performance has become increasingly important for credit risk assessments (Abellán & Castellano, 2017).Therefore, a classification method is required that can accurately and efficiently measure the credit risk level of prospective borrowers as the key to the credit approval process (Lopez & Jeronimo, 2015).

Various credit risk classification methods or algorithms have been studied and implemented, including *artificial neural network* (ANN) (Gafarova, 2017), *k-Nearest Neighbor* (kNN) (Ivandari et al., 2017)(Kaur & Cheema, 2018), *Support Vector Machine* (SVM) (Gafarova, 2017)(Maldonado et al., 2017)(Yu et al., 2018), *Decision Tree*(Tun, 2017), and *Multilayer Perceptron*(Khashei & Torbat, 2019). The use of kNN for credit risk classification is interesting to do because kNN has a variety of advantages, namely more intuitive and easy to implement, moreover, strong interpretationability of kNN is very important to build a credit prediction model (Zhang, 2020). Besides being very simple, powerful, easy to implement and understand, kNN is very useful because it doesn't involve any assumptions about the data, plus the size of the distance that can be calculated consistently between two

examples (Kiran et al., 2018). Regarding the KNN classification method, this method can be improved by applying the selection (selection) feature that refers to the selection of a subset of features of all existing features (Zhang, 2020). The application of feature selection in kNN classification has also been tested on the classification of lung cancer prognosis with 100% accuracy results (Maleki et al., 2020).

Therefore, it is necessary to efficiently select as many *n* features because feature selection is a combinatorial issue as much as 2^n combinations. Recently, *swarm intelligence* (SI) techniques have gained a lot of attention from the feature selection community due to its simplicity and potential global search capabilities(Nguyen et al., 2020). SIyang algorithms that have been studied to select features such as *Particle Swarm Optimization*(PSO) (Abualigah et al., 2018)(Kavitha et al., 2018)(Cherrington et al., 2019), *Artificial Bee Colony* (ABC) (Keleş & Kılıç, 2018)(Kiliç & Keleş, 2018), *Ant Colony Optimization* (ACO) (Peng et al., 2018)(Ghosh et al., 2019), and so on.

Although the SI algorithm has been successfully applied to the selection of features, currently the SI algorithm used mostly uses standard representations (Nguyen et al., 2020). Nguyen, et al (2020) trace that standard representation is not the most natural representation for feature selection, whereas binary representation is the most natural representation for feature selection and has not been studied much. Therefore, an SI method is required that uses binary representation for feature selection. This study contributes to the development of feature selection methods with SI algorithms that use binary representation, i.e. feature selection using PSO algorithms with binary representation or *Binary Particle Swarm Optimization* (BPSO) applied to credit risk classification. For the evaluation of the classification, the kNN classification method is used because of the advantages mentioned above.

2. Method

2.1 Data Mining

In the process of processing data into information, *data mining* is needed which will then produce a new knowledge sourced from the old data, where the decision results can be used as a reference in future decision making. *Data mining* involves discovering new, exciting, and potentially useful patterns from large data sets and implementing algorithms for the extraction of hidden information. Many other terms are used for data *mining*,for example, knowledge discovery (mining) in *databases* (KDD), knowledge extraction, data/pattern analysis, data archaeology, data dredging, and information collection(Han et al., 2011).

The purpose of any *data mining* process is to build an efficient predictive or descriptive model of a large amount of data that not only best fits or explains it, but can also generalize to new data(Mukhopadhyay et al., 2013). Based on a broad view of *data mining*functionality, *data mining* is the process of finding interesting knowledge of large amounts of data stored either in databases, data warehouses, or other information repositories .

Based on the definition *of data mining* and the definition of data *mining* function, generally *the data mining* process includes the following steps(Chen et al., 2015):

1. Data preparation: prepare data for mining. This includes the following 3 steps: integrating data across multiple data sources and cleaning up interference from data; extracting some parts of data into *data mining*system; preprocessing data to facilitate data *mining*.
2. Data mining: apply algorithms to data to find patterns and evaluate patterns of knowledge found.
3. Data presentation: visualizes data and represents knowledge mined to users. We can see the data mining in a multidimensional view(Han et al., 2011), yaitu:
    a) In the knowledge view or data mining function view, this includes characterization, discrimination, classification, grouping, association analysis, time series analysis, and pencilan analysis.
    b) In the display of the techniques used, it includes machine learning, statistics, pattern recognition, big data, support vector machines, rough sets, neural networks, and evolutionary algorithms.

c) In the application view, it covers industry, telecommunications, banking, fraud analysis, biodata mining, stock market analysis, text mining, web mining, social networking, and *e-commerce.*

The concept of learning in *Data mining* is divided into 2 kinds of learning concepts(Oded & Lior, 2010), i.e. first *Supervised Learning* which is an algorithm that makes a function based on existing training data, in this case it can be said that for this algorithm is available complete and detailed training data and classified well that will be used as a data model when conducted the test process with new test data and produce the appropriate output results expected in advance based on existing training data. The second is *Unsupervised Learning* which is an algorithm that seeks to represent or represent the pattern of an input derived from the training data and which is one of the differences with *Supervised Learning* is the absence of classifying of data input.

2.2 Artificial Neural Network

Neural Network has been successfully applied to practical classification tasks in various industries, including industrial, commercial, and scientific fields. *The Neural Network* consists of neurons that can process information. Interconnection between neurons is an algorithmic model that includes layer input, *hidden layer,*and *output layer*(Hewahi &Hamra, 2017). When using *the Artificial Neural Network*(ANN) to solve classification problems, for example, we can implement the sigmoid activation function, which is often used as a neural network threshold function to map variables between 0 and 1, to complete classification tasks. Mhatre et al.(2017) proposes that neural networks be stable because when nerve tissue elements fail, they can continue their parallel properties without problems. But for large data processing, it takes longer processing time.

ANN or Artificial Neural Network is a concept of knowledge engineering that adopts the human nervous system. This method can be used for pattern recognition, classification and forecasting. In its design, ANN has 3 parts namely input parts, processing parts and output parts (Prasetyo, 2014). This input in ANN can be vector so that calculations in ANN can be done for complex problems easily. In the process, this ANN method is used to perform forecasting and pattern recognition in *data mining*. To do so, ANN requires a training process in order to predict the class of a test data. In the process of data mining, ANN uses the activation function used to limit the output of the processing part or neurons to fit the desired limit. There are various algorithms that can be usedto use this method. One of them is *the Backpropagation*algorithm.

The *Backpropagation* algorithm is one of the algorithms used to conduct training on the ANN method. This algorithm is nonlinear which can solve various complex problems. This algorithm has a high mathematical basis and is trained using guided learning methods where the results or objectives are already known. In this algorithm, the network will be given a pair of patterns that are the desired inputs and patterns. When the pattern is inserted into the network, the weights will be changed to minimize the difference in the output pattern with the desired pattern. This training is done repeatedly so as to meet the desired pattern, this algorithm supports ann type that is multi *layer* or commonly called Multi Layer *Precepton* (MLP). In this algorithm consists of 3 *layers* namely *layerinput*, hidden *layer* and output *layer*.

2.3 Support Vector Machine

Support Vector Machine(SVM) is an algorithm that works with *nonlinear mapping* that serves to transform initial *training* data into new, higher dimensions. In this new dimension, SVM will find *optimum hyperplanelinear*. By *mappingnonlinear* to a higher dimension, data from two classes will always be separated by a *hyperplane*. This method will find *hyperplane* by using support *vectors* and *margins*(Han et al., 2011). SVM can solve *nonlinear* problems and can handle high-dimensional data sets, but the interpretation is not strong. With a set of training examples, each training instance marked as belonging to one or the other of two categories, SVM created a model that assigns a new instance to one of two categories, making it a *non-probabilistic*binary linear classifier(Zhang, 2020).

This method was first presented in 1992 by Vapnik, Boser, and Guyon at the *Workshop on Computational*Learning. SVM theory introduces a new strategy by finding the best *hyperplane* in the input space. The first SVM principle was a *linear classifier*,but SVM was later developed to be able to work on non-linear problems by inserting the kernel. The development of SVM stimulates research interest in the field of pattern recognition in developing the potential capabilities of SVM methods both theoretically and in terms of application. Nowadays SVM has been successfully applied in solving practical problems. The concept of SVM can be explained simply as an attempt to find the best *hyperplane* that serves as a two-class separator in the input space.

Classification problems can be interpreted as an attempt to find a line that separates the two groups. The best *hyperplane* separator between the two classes can be found by measuring the margin of the *hyperplane* and finding its maximum point. Margin is the distance between the *hyperplane* and the closest *pattern* of each class. *This* closest pattern is referred to as vector *support.* Efforts to locate this *hyperplane* are at the heart of svm's learning process.

2.4 Feature Selection Using PSO

PSO is a *population-based stochastic optimization* technique (fish, bees, birds etc.), put forward by Russell C. Eberhart and James Kennedy in 1995 inspired by the social behavior of the movement of birds or fish. PSO has been successfully applied in various research fields and many applications, including specific applications with specific needs, such as: function optimization, sudoku games, fuzzy system control, including Artificial *Neural Networks* (ANN" "training"), solving suplay chain problems (Habibi, 2017)and many other applications. This is because PSO has a method of solving problems quickly and simply and gives better results when compared to other methods.

PSO simulates the behavior of a flock of birds. As in the following scenario: there is a group of birds randomly looking for food in an area, where there is only one piece of food in the area being searched. All birds do not know how far the food is. Then the best strategy for finding food is to follow the birds that are closest to the food. Pso adopts such scenarios and applies them to solve optimization problems.

In PSO, every single solution referred to as "bird" in the search for space we call "particles" (or individuals). Each particle "flies" following the current*optimum particles.* Particles store traces of their position in problem *space.* The traces of the position are interpreted as *the best solution,*or *fitness* in GA that he has obtained so far. The value, namely *fitness value,*called *pbest* is also stored. In addition *to pbest* which belongs to the individual concerned, also stored the best value of the individual around it (local*best*), called *lbest*. If an individual takes into account all individuals in the population in which he or she is located, then the best value in question is the best overall value (global*best)*and is called *gbest.* Furthermore, there is an acceleration between the *pbest* location and the *pbest* location of each individual. After finding the two best values, update the particle speed and position with the following equation (Tuegeh et al., 2009):

$$v_{ij}^{k+1} = \omega_k * v_{ij}^k + c_1 * rand * \left(pbest_{ij}^k - x_{ij}^k\right) + c_2 * rand * \left(gbest_{ij}^k - x_{ij}^k\right) \quad (5)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (6)$$

2.5 Feature Selection using *Binary*-PSO (BPSO)

Pso was originally proposed in the form of a kontiniu representation. In implementing a PSO to complete binary optimization is to keep using the form of kontiniu representation and change the continuous position to the binary position. The sigmoid function is widely used for this task because it can convert any continuous value to a continuous value in the range [0,1], which is then converted to a binary value by comparing it with a random number or threshold(Nguyen et al., 2020). A random value between [0,1] is used to convert the continuous value obtained into a binary value. The above approach has been applied to achieve feature selection(Qasim &Algamal, 2018) and (Yadav et al., 2018).

However, because the continuous search mechanism is still being implemented, the above approach also undergoes early convergence as in continuous PSOs.

Tran et al.(2017) propose representations that can achieve feature selection and feature discritization. In the proposed representation, each element in the position vector is used as a cut point to distinguish the features of the original real value. If the element value is outside the specified range, the related feature is discarded. The proposed representation helps psos to select a small number of features and achieve better classification performance than using standard representations. However, the proposed algorithm is changed from a continuous value to a binary value, which may lose discriminatory information. In addition, a list of intersections needs to be createdhas been predetermined for each feature. A standard PSO representation consists of real value elements, which can be called continuous representations. If the value of the element is greater than the θthreshold, the corresponding feature is selected. Otherwise, the feature will be removed.

The BPSO to be applied to this study uses the "flipping" opportunity value introduced by (Xue et al., 2014)to replace the speed at whicheach particle renews during the evolutionary process. indicates the "flipping" opportunity value, which is a d-dimensional vector$p$ $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$ $p_{id}x_{id}$ $x_{id}^{t+1} = 1$ $x_{id}^{t} = 0$ (d$x_{id}^{t+1} = 0$ represents many features). indicates the$x_{id}^{t} = 1$ "flipping" opportunity value for the *i*particle. To update the position of particles used the formula (3). calculated based on the last position of the particle, *ppbest* and *gbest* taking into account the following formulas (7) and (8)(Xue et al., 2014):

$$x_{id}^{t+1} = \begin{cases} 1 - x_{id}^{t}, & if\ random() < p_{id} \\ x_{id}^{t}, & untuk\ yang\ lainnya. \end{cases} \quad (7)$$

$$p_{id} = p_0 + p_{pd} + p_{gd} = 1 \quad (8)$$

3. Results and Discussion

3.1 Preprocessing Data

The study conducted the test using a Credit German dataset published by the University of California Irivine (UCI) available in the UCI *Machine Learning* Repository. The data sethas 20 attributes and is divided into 2 class categories with 1000 instances. Meanwhile, the partial view of the data content is shown by Table 1 as follows:

Table 1. Data Set View After Transformation

Data	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Class
1	A11	6	A34	A43	1169	A65	A75	4	A93	A101	4	A121	67	A143	A152	2	A173	1	A192	A201	1
2	A12	48	A32	A43	5951	A61	A73	2	A92	A101	2	A121	22	A143	A152	1	A173	1	A191	A201	2
3	A14	12	A34	A46	2096	A61	A74	2	A93	A101	3	A121	49	A143	A152	1	A172	2	A191	A201	1
4	A11	42	A32	A42	7882	A61	A74	2	A93	A103	4	A122	45	A143	A153	1	A173	2	A191	A201	1
5	A11	24	A33	A40	4870	A61	A73	3	A93	A101	4	A124	53	A143	A153	2	A173	2	A191	A201	2
6	A14	36	A32	A46	9055	A65	A73	2	A93	A101	4	A124	35	A143	A153	1	A172	2	A192	A201	1
7	A14	24	A32	A42	2835	A63	A75	3	A93	A101	4	A122	53	A143	A152	1	A173	1	A191	A201	1
8	A12	36	A32	A41	6948	A61	A73	2	A93	A101	2	A123	35	A143	A151	1	A174	1	A192	A201	1
9	A14	12	A32	A43	3059	A64	A74	2	A91	A101	4	A121	61	A143	A152	1	A172	1	A191	A201	1
10	A12	30	A34	A40	5234	A61	A71	4	A94	A101	2	A123	28	A143	A152	2	A174	1	A191	A201	2
11	A12	12	A32	A40	1295	A61	A72	3	A92	A101	1	A123	25	A143	A151	1	A173	1	A191	A201	2
12	A11	48	A32	A49	4308	A61	A72	3	A92	A101	4	A122	24	A143	A151	1	A173	1	A191	A201	2
13	A12	12	A32	A43	1567	A61	A73	1	A92	A101	1	A123	22	A143	A152	1	A173	1	A192	A201	1
14	A11	24	A34	A40	1199	A61	A75	4	A93	A101	4	A123	60	A143	A152	2	A172	1	A191	A201	2
15	A11	15	A32	A40	1403	A61	A73	2	A92	A101	4	A123	28	A143	A151	1	A173	1	A191	A201	1
16	A11	24	A32	A43	1282	A62	A73	4	A92	A101	2	A123	32	A143	A152	1	A172	1	A191	A201	2
17	A14	24	A34	A43	2424	A65	A75	4	A93	A101	4	A122	53	A143	A152	2	A173	1	A191	A201	1
18	A11	30	A30	A49	8072	A65	A72	2	A93	A101	3	A123	25	A141	A152	3	A173	1	A191	A201	1
19	A12	24	A32	A41	12579	A61	A75	4	A92	A101	2	A124	44	A143	A153	1	A174	1	A192	A201	2
20	A14	24	A32	A43	3430	A63	A75	3	A93	A101	2	A123	31	A143	A152	1	A173	2	A192	A201	1

Table 2. Comparison of KNN and BPSOKNN Best Accuracy with variations

k-fold	Akurasi	
	KNN	BPSOKNN
2	73.60	88.60
5	75.60	88.60
10	**76.40**	**88.70**
20	75.80	88.30
rata-rata	75.35	88.55

In the performance measurement of knn classification approach with Binary PSO or BPSOKNN based selection feature, test scenarios are performed using many k-neighbor variations on KNN, and k-fold cross validation variations. Variations of many neighbors k on knn to be tested that is ranging from k = 1 to k = 30. For k-fold cross validation variations to be tested are 2-fold, 5-fold, 10-fold, and 20-fold.

3.2 Discussion

1. 2-fold Validation Test

In tests with 2-fold validation, the data set will be divided into 2 kelompok, namely training data and testing data. Of the total instances of 1000 data is divided into 2 parts, where each section consists of 500 data. In the 1st fold, that is, when the 1st part (the first 500 data) becomes the test data (testing data) then the rest (the next 500 data) becomes training data. Furthermore, accuracy is calculated

based on the portion of the data in conventional KNN classification models without selection features and on KNN models with BPSO-based selection features. In the 2nd fold is when the 2nd part (500 second data) becomes test data (testing data) then the rest is used as training data. Furthermore, it will also calculate accuracy based on the portion of the data. It is then calculated the average accuracy of the 2 repeating accuracy and the average accuracy obtained from each Classification model can be seen in Table 4. 10. In Table 4.10, it can be seen that the KNN classification model obtained the highest accuracy of 73.60% at k=12, and the BPSOKNN classification model obtained the highest accuracy of 88.60% at the same k value. It also indicates that in the use of 2-fold validation, the best k value on KNN models and BPSOKNN models is 12, and bpso-based selection feature techniques managed to improve the accuracy of knn classification by 15%.

2. 5-fold Validation Test Results

In tests with 5-fold validation, the data set will be divided into 5 groups, then classified into training data and testing data. Of the total instances of 1000 data divided into 5 parts, each part consists of 200 data. In the 1st fold, that is when the 1st part (200 first data) becomes test data (testing data) then the rest (800 next data) becomes training data (training data). Sample training data is shown in Table 4.14 and Table 4.15. Furthermore, accuracy is calculated based on the portion of the data in conventional KNN classification models without selection features and on KNN models with BPSO-based selection features. In the2nd fold is when the 2nd part (200 second data) becomes test data (testing data) then the rest is used as training data. Furthermore, it will also calculate accuracy based on the portion of the data. Thus continued until fold to 5. Then calculated the average accuracy of the 5 repetitions and the average accuracy obtained from each model klasification

3. 10-fold Validation Test Results

In tests with 10-foldvalidation, the data set will be divided into 10 groups, then classified into training data and testing data. Of the total instances of 1000 data is divided into 10 parts, where each part consists of 100 data. In the 1st fold, that is when the 1st part (100 first data) becomes test data (testing data) then the rest (900 next data) becomes training data (training data). Furthermore, accuracy is calculated based on the portion of the data in conventional KNN classification models without selection features and on KNN models with BPSO-based selection features. In the2nd fold is when the 2nd part (100 second data) becomes test data (testing data) then the rest is used as training data. Sample training data is shown in Table 4.16 and Table 4.17. Furthermore, it will also calculate accuracy based on the portion of the data. Thus continued until fold to 10. It is then calculated the average accuracy of the 10 repetitions and the average accuracy obtained from each k lasification model can be seen inTable 4. 16. In Table 4.16 below, it can be seen that the KNN classification model obtained the highest accuracy of 76.40% at k=17, and the BPSOKNN classification model obtained the highest accuracy of 88.70% at k=19 and k=26. It also indicates that in the use of 10-fold validation on KNN models, the best k value is 17, while on bpsoknn models, the best k values are 19 and 26. The accuracy results obtained indicate that bpso-based selection feature techniques successfully improved the accuracy of KNN classification by 12.30%.

4. 20-fold Validation Test Results

In tests with 20-fold validation, the data set will be divided into 20 groups, then classified into training data and testing data. Of the total instances of 1000 data divided into 20 parts, each part consists of 50 data. In the 1st fold, that is when the 1st part (50 first data) becomes test data (testing data) then the rest (950 next data) becomes training data (training data). Furthermore, accuracy is calculated based on the portion of the data in conventional KNN classification models without selection features and on KNN models with BPSO-based selection features. In the2nd fold is when the 2nd part (50 second data) becomes test data (testing data) then the rest is used as training data. Furthermore, it will also calculate accuracy based on the portion of the data. Thus continued until fold to 20. It is then calculated the average accuracy of the 20 repetitions of validation and the average accuracy obtained from each k lasification model can be seen in Table4. 17. In Table 4.17,it can be seen that the KNN

classification model obtained the highest accuracy of 75.80% at k=11, and the BPSOKNN classification model obtained the highest accuracy of 88.30% at the value of k=17. It also indicates that in the use of 20-fold validation on KNN models, the best k value is 11, whereas on the BPSOKNN model, the best k value is 17. The accuracy result also informs that bpso-based selection feature technique successfully improved the accuracy of KNN classification by 12.50%.

4. Conclusions

This enelitian contributes to the development of feature selection methods with SI algorithms that use binary representation, i.e. feature selection using PSO algorithms with binary representation or Binary Particle Swarm Optimization (BPSO) applied to credit risk classification. The selected features are then evaluated with the KNN classification model. Performance of KNN classification algorithm with BPSO-based selection feature compared to the performance of conventional KNN classification algorithm without selection feature by testing uci benchmark data set. The results showed that the best accuracy of KNN was in the use of 10-fold validation, which is 76.40%, and in the use of the same k-fold validation (10-fold) bpsoknn classification model obtained the best accuracy that lebih large by 88.70%, with the increase in accuracy produced bpsoknn compared to KNN is 13.35%. This test showed that bpso-based selection feature techniques successfully improved the accuracy of KNN classification on credit risk classification, and the higher F1-score achievement of BPSOKNN algorithm from KNN indicates that bpso-based selection feature performance has performed well.

Reference

[1] Abellán, J., & Castellano, J. G. (2017). A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Systems with Applications*.

[2] Abualigah, L. M., Khader, A. T., & Hanandeh, E. S. (2018). A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *Journal of Computational …*.

[3] Addo, P. M., Guegan, D., & Hassani, B. (2018). Credit risk analysis using machine and deep learning models. *Risks*.

[4] Chakraborty, A., & Kar, A. K. (2017). Swarm intelligence: A review of algorithms. *Nature-Inspired Computing and Optimization*.

[5] Cherrington, M., Airehrour, D., Lu, J., & ... (2019). Particle Swarm Optimization for Feature Selection: A Review of Filter-based Classification to Identify Challenges and Opportunities. *2019 IEEE 10th*

[6] Fong, S., Wong, R., & Vasilakos, A. V. (2015). Accelerated PSO swarm search feature selection for data stream mining big data. *IEEE Transactions on Services …*.

[7] Fränti, P., & Sieranoja, S. (2018). K-means properties on six clustering benchmark datasets. *Applied Intelligence*.

[8] Hewahi, N. M., & Hamra, E. A. (2017). A hybrid approach based on genetic algorithm and particle swarm optimization to improve neural network classification. *Journal of Information Technology*

[9] Ivandari, I., Chasanah, T. T., & ... (2017). Data Attribute Selection with Information Gain to Improve Credit Approval Classification Performance using K-Nearest Neighbor Algorithm. *... Journal of Islamic* .

[10] Gafarova, L. (2017). *Usage of Artificial Neural Network and Support Vector Machine model for classification of Credit Scores*. dspace.khazar.org.

[11] Hassanat, A. B., Abbadi, M. A., Altarawneh, G. A., & ... (2014). Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach. *ArXiv Preprint ArXiv*.