Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 1027518, 17 pages
https://doi.org/10.1155/2022/1027518

Research Article

Towards Tax Evasion Detection Using Improved Particle Swarm Optimization Algorithm

Houri Mojahedi, Amin Babazadeh Sangar, and Mohammad Masdari

Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

Correspondence should be addressed to Amin Babazadeh Sangar; bsamin2@liveutm.onmicrosoft.com

Received 24 January 2022; Revised 29 April 2022; Accepted 8 August 2022; Published 5 September 2022

Academic Editor: Kehui Sun

This paper employs machine learning algorithms to detect tax evasion and analyzes tax data. With the development of commercial businesses, traditional algorithms are not appropriate for solving the tax evasion detection problem. Hence, other algorithms with acceptable speed, precision, analysis, and data decisions must be used. In the case of assets and tax assessment, the integration of machine learning models with meta-heuristic algorithms increases accuracy due to optimal parameters. In this paper, intelligent machine learning algorithms are used to solve tax evasion detection. This research uses an improved particle swarm optimization (IPSO) algorithm to improve the multilayer perceptron neural network by finding the optimal weight and improving support vector machine (SVM) classifiers with optimal parameters. The IPSO-MLP and IPSO-SVM models using the IPSO algorithm are used as new models for tax evasion detection. Our proposed system applies the dataset collected from the general administration of tax affairs of West Azerbaijan province of Iran with 1500 samples for the tax evasion detection problem. The evaluations show that the IPSO-MLP model has a higher accuracy rate than the IPSO-SVM model and logistic regression. Moreover, the IPSO-MLP model has higher accuracy than SVM, Naive Bayes, k-nearest neighbor, C5.0 decision tree, and AdaBoost. The accuracy of IPSO-MLP and IPSO-SVM models is 93.68% and 92.24%, respectively.

1. Introduction

Machine learning is one of the ideal ways to reduce the operational costs and costs of business processes. It also accelerates work and provides better services to the customers [1]. Studies have shown that machine learning could reduce costs by 20 to 25% in the banking industry and information technology (IT) operations, infrastructure, and maintenance operations, generate new revenue in industries and services, and increase customer acquisition and retention in various areas. By transforming human processes into intelligent and automated processes, companies can focus their resources on more valuable activities, such as providing better products and services to customers and detecting tax evasion [2]. As one of the most important sources of government revenue, tax currently plays a vital role in the economy of any country [3]. By using various tax policies, governments can use tax tools and adjust their various economic policies to achieve their most important goals such as social justice, proper distribution of income, and elimination of the class gap between different classes of society, stabilization of prices, reduction of unemployment, economic prosperity, and increase of investment [4, 5].

Enforcing the correct tax law is an excellent way to increase government revenue and modernize countries' tax systems, which can only be achieved through accurate design and proper implementation of intelligent systems, particularly the design and implementation of suitable training systems for training tax organizations [6]. The emergence of tax is perhaps the most important economic event of the last decades of the twentieth century, and its importance is increasing rapidly. However, some people refuse to pay taxes and are looking for ways to evade them [7], which hurts the budget revenues of businesses and governments. Therefore, private and public businesses should focus on construction and manufacturing activities rather than discovering ways to evade taxes [8].

Tax evasion is a global phenomenon whose disruption affects society as a whole. This phenomenon can be described as a deliberate act on tax returns to obtain illegal financial benefits and reduce tax liability. The internal tax code defines tax fraud under the IRC [9]. According to this article, any person who intentionally attempts to evade or defeat any tax imposed by the Internal Revenue Service (IRS) will be recognized guilty, and other penalties shall be provided with them by law. Recent studies have estimated that governments worldwide lose about $ 500 billion annually due to tax evasion.

One of the most critical consequences of tax evasion is economic and social injustice. Tax evasion changes the ability of economic competition in favor of tax evaders. Another consequence of tax evasion is the intensification and spread of this phenomenon due to disruption in the economic security required to expand economic activities and investment [10]. By predicting some ways for tax evasion and finding appropriate solutions, its spread can be primarily prevented, but the most critical factor in preventing tax evasion is the people's awareness of the importance of tax payment.

Unfortunately, auditing a tax return is a slow and costly process. Due to the lack of software and hardware platforms for receiving tax returns and electronic payments, the electronically classified data of companies could not be obtained in previous years. For this reason, a shift to provide an intelligent software system could not detect tax evasion and design a suitable criterion for it [11]. By implementing various software and hardware infrastructures in the country's tax affairs organization, various intelligent structures can be designed and developed along with the above systems. Therefore, intelligent prediction models based on machine learning methods [12] to detect tax fraud can be used to increase the precision and efficiency of auditing [13].

Tax agencies use two methods to investigate tax fraud: the auditors' experience and rule-based systems. A rule-based system, often in a set of if-then series, detects fraud cases [14]. These rules are developed through a complex process in which auditors identify a tax fraud case after investigation and generalize its characteristics, including a set of rules based on tax fraud knowledge. However, these traditional methods have two significant drawbacks. First, they are mainly dependent on past experiences, so they cannot detect new methods of fraud. Second, the subjective judgment of the experts makes the basics of knowledge expensive for providing, maintaining, and updating rule-based systems. Therefore, a new solution to detect tax evasion is the use of machine learning techniques that discover the extraction and generation mechanisms of knowledge from a significant amount of data to detect fraudulent behavior [15].

With the development of machine learning and meta-heuristic algorithms, problem-solving in various fields such as optimization [16, 17], prediction [18], detection [19], classification [20], and clustering [21] is performed with a more accurate process. Meta-heuristic algorithms are widely used in optimization problems due to their high efficiency and various solutions [22]. In particular, the PSO algorithm [23] has shown high efficiency by changing the position and velocity of particles [24, 25]. This paper uses the improved MLP and PSO called IPSO-MLP, SVM and improved PSO called IPSO-SVM, and logistic regression algorithm to detect tax evasion. The IPSO-MLP model uses IPSO to adjust weights, and the IPSO-SVM model employs IPSO to adjust the SVM parameters that play a significant role in the precision of classification. One of the significant challenges in the multilayer artificial neural network is the optimal selection of neural weights that can be solved with meta-heuristic algorithms. Optimal selection of the classification parameters is also essential to increase SVM precision. Meta-heuristic algorithms such as the PSO algorithm can solve problems with reasonable speed and precision by exploring optimal solutions [26]. The models proposed in this paper have not been used in the previous studies on tax evasion; therefore, they are presented as new models for tax evasion detection (TED). Using machine learning algorithms can significantly increase the accuracy and robustness of TED and design detection systems without the need to detect linear relationships. Moreover, the advantage of an improved algorithm is that it can directly extract the optimal response. The main objectives of this paper are as follows:

(1) Providing IPSO-MLP model based on the improvement of the MLP weights for tax evasion detection. The IPSO algorithm aims to improve the neurons' weights in the MLP network, implement the data training steps correctly, and reduce the amount of output error.

(2) Providing IPSO-SVM model based on the improvement of the SVM parameters for tax evasion detection. The SVM model highly depends on the value of the initial parameters. If their correct value is determined, it will increase the detection accuracy and accurate separation of instances into different classes.

(3) Using machine learning methods for tax evasion detection and comparing their results with the IPSO-MLP model.

The general structure of the paper is organized as follows: Section 2 reviews the previous studies, and Section 3 illustrates the IPSO algorithm and IPSO-based hybrid models. In Section 4, relevant simulations are performed. Finally, Section 5 provides conclusions and future research directions for this work.

2. Review of the Literature

This section reviews previous studies conducted on tax evasion detection. As mentioned earlier, machine learning algorithms play an essential role in tax evasion detection, and most studies have used a combination of machine learning algorithms.

For example, a study in the field presented an architecture for the problem of financial fraud detection by Chinese commercial companies, which included communication with the experts in the field, use of data mining

algorithms, design instructions for data mining systems, and integration of knowledge of the experts in the field. The proposed architecture used the C5.0 decision tree. The dataset contained samples of 500 commercial companies during one year, and each sample had 100 characteristics. After classification, the training dataset was divided into two parts, including 460 positive samples and 40 negative samples. The implementation precision of the C5.0 decision tree was 85–90% [27].

Another study implemented eight models based on different combinations of the decision tree and logistic regression (LR) for value-added tax (VAT) in India from 2003 to 2004. The samples included 402 sales agents. The results indicated that all the models developed through data mining were better than the random selection method [28].

Moreover, researchers used association rules for Taiwan data to design an evasion detection model from VAT from 2003 to 2004. They evaluated data on two different datasets with 1934 and 1543 samples and employed eight different rules to detect fraudulent samples. The precision of the association rules was >80. According to the results, the designed model increased the tax evasion detection, and therefore, it could be used to effectively reduce or minimize losses due to VAT evasion [29].

In addition, scholars [30] used an intelligent system that combined an MLP-ANN, support vector machine (SVM), and logistic regression (LR) with a harmony search algorithm (HS) to detect tax evasion of companies taken from the Iranian National Tax Administration (INTA). Learning rate is one of the essential factors in MLP, which ranges between 0 and 1. Moreover, the optimal number of iterations was optimized to prevent network over-learning and the increase in network error. By increasing the number of iterations, the amount of error was reduced, but increasing the number of iterations should be systematic to reduce the amount of network error and prevent the training time. HSA was used to find the parameters of the SVM and MLP classification models. This model was tested using a 10-fold iterative validation structure with datasets, including 2451 and 2053 test samples from a two-year tax return and 1118 and 906 samples as data from the food and textile sectors. Even if the data contained actual values, network training would result in high error rates if the data were not normalized. Data normalization was performed according to the following equation:

$$N = \frac{(UN - \mu_{UN})}{\sigma_{UN}}, \tag{1}$$

where UN is the financial variable before normalization, μUN is the UN average, σUN represents the standard deviation (SD), and N is the normalized financial variable.

The results of experimental data showed that the MLP model in combination with HSA had better detection than other combinations so its precision for food and textile datasets was 90.07% and 82.45%, respectively. Moreover, sensitivity was 85.84% and 84.85% for food and textile datasets, respectively, and specificity was 90.34% and 82.26% for food and textile datasets.

Furthermore, researchers proposed a model based on linear regression and SVM to detect high-risk taxpayers. Therefore, they collected tax data from 2010 to 2015 in the INTA. The steps of linear regression were as follows: formulating the regression formula: $Y_i = a + bX_i + E_i$, selecting the latest data, obtaining tax income for taxpayers, calculating the average taxable income of taxpayers, and calculating the goodness-of-fit $(Y_i = a + b \times X_i)$ and regression model for taxpayers. People who had a moderate amount of high regression prediction for different years were considered high risk. Tax experts' output accuracy test indicated that high precision could be obtained by combining the SVM and linear regression models [31].

Due to the recent development and large volume of data stored in tax systems, a tool is needed to process the stored data and detect fraudsters based on the information obtained from it. In this regard, some scholars used the parallel Bayesian network to detect forgers [32]. The Bayesian network is a directional graph in which nodes represent variables $(X_1, \ldots, X_n)$. The dataset used in their study included 10028 records. The results showed that the fraud of taxpayers with a complementary sheet was about 57.9% [32].

A colored network-based model (CNBM) was proposed to describe economic behaviors, social relationships, and taxpayer transactions and establish an interaction network [33]. China-based National Tax Information System (NTICS) is involved in a large volume of transactions and data. For example, there are more than 31,910,000 taxpayers and 48,000 tax offices across China. The first stage aimed to detect suspicious groups from a heterogeneous information network based on the CNBM to detect suspicious business relationships. Suspicious groups were extracted in the first group, called the suspicious mining group (MSG). The second stage, identifying tax evasion (ITE), performs all transactions related to suspicious business relationships to detect tax evasion in a set of suspicious groups using traditional methods. To evaluate the effectiveness of the CNBM model in the MSG phase, a simulated network based on the business relationships was implemented based on the graph theory and actual data-based experiments for all nodes. Experimental results indicated that the CNBM model could improve efficiency in the possible tax evasion detection in the MSG phase [33].

A deep learning network-based model for tax evasion detection was also proposed, in which some features were extracted based on the maximum conditional difference (CMMD) for the conditional probability distribution (CPD) [34]. In the deep learning network, different layers and distribution adapters were used to identify suspicious samples. According to the findings, the deep network model had better detection precision than the conventional artificial neural network.

Another study presented a regression model using commercial primary tax information and the rate of tax evasion by suspicious commercial sellers [35]. The sellers were categorized into different seller groups using Benford's law, and the type of classification was determined after implementing the k-medoids clustering algorithm on a set of sellers. In the k-medoids algorithm, before calculating the

distance of other data from each cluster center, the K point was randomly selected from n data as the cluster's center with the specified center as the median. Then, each point was assigned to the nearest cluster. This iterative method for changing cluster centers was continued to achieve the best clustering. Auditors use Benford's law as a simple and effective tool to detect fraud in fraudulent audit methods. This law includes a set of statistical principles to determine the extent of dispersion of numbers used in specific rows of digits in the sample set. Equation (2) was used to give suspicious points to clusters [35].

$$m * \frac{\sum_{i=1}^{m}\left(W_i * 1000^{\varnothing(c)}\right)}{\left(m + \sum_{i=1}^{m} W_i\right)}, \tag{2}$$

where m is the total number of edges (or transactions) in clusters c, and W is the weight of the edges. $\varphi(c)$ is the mean value of the absolute deviation from Benford's law for W.

The dataset used in this study was provided by the Commercial Tax Office of Telangana state, India. The results of this study helped tax enforcement agencies in preventing tax evasion.

Another study used random forest, MLP-ANN, SVM, and logistic regression algorithms to evaluate risk and detect tax evasion [36]. Therefore, an integrated social network of taxpayers was modeled. In an economic transaction (u, v), node u is the seller, and node v is the buyer. The taxpayer social network, built from data from the Tuscany region of Italy in 2014, included about 700,000 nodes and 1,800,000 edges. The random forest model had the best results in terms of accuracy (74.29), AUCROC (74.29), precision (75.42), and F1 (76.73), while the best value for the recall criterion belonged to the MLP model (75.63).

A graph-based network model called TED-TNR used the weighted adjacency matrix for tax evasion detection [37]. This model used three different vectors A, S, and X. A is the matrix of taxpayers' traits. S is the similarity matrix of the taxpayers' features and can be calculated by measuring similarities such as cosine similarity. X contains the final values for the taxpayers based on a value obtained from A and S. Therefore, 9,422,952 transaction samples were evaluated in the wholesale and retail industrial groups. The transaction network was a directional weighting network that included 323,587 nodes and 1,430,821 edges. Indicators such as company size, registered capital, and investment ratio were the main outlines of the trading network. The results demonstrated that the detection precision of the TED-TNR model was higher than conventional and ANN models [37].

In addition, researchers proposed a deep learning model called the transferable tax evasion detection method based on positive and unlabeled learning (TTED-PU) to identify the suspected tax evasion samples [38]. They used a transfer learning method based on the semiregulatory method using positive and negative samples to predict untested samples. In this model, the gradient reduction method was applied to find the weight of neurons from derivation rules. Evaluation on 20,444 samples showed that the TTED-PU model had a lower error.

Moreover, a model was suggested based on the error back-propagation artificial neural network and the CHAID decision tree for tax evasion detection [39]. Hence, BP-ANN injected tax samples into the algorithm, and differences in training data were detected by increasing and decreasing weights and deviations. One of the critical goals in ANN was to find the appropriate weight for different layers and actually to estimate the ANN parameters. The BP algorithm is a method for calculating weights that can be calculated from two forward and backward paths, and this forward and backward path is iterated to achieve the best estimate of the network parameters and is considered a training process. In the CHAID tree, all values of the characteristics of the target variable were evaluated using the chi-squared statistical criterion. In this algorithm, the statistically similar values are related to each other according to the target variable. Evaluation of 12,458 different samples revealed that the percentage of accuracy of the CHAID decision tree was higher than that of BP-ANN [39].

Furthermore, scholars proposed a model based on MLP-ANN to help tax fraud detection on personal income tax returns (IRPF, in Spanish) [40]. In this network, neurons of each layer are related to the neurons of the previous layer, but this relationship is not necessarily under the same conditions but with different weights. The MLP-ANN output was defined according to the following equation:

$$Y_i = f\left(\sum_{i=1}^{N} w_{ij} x_i + b_j\right), \tag{3}$$

where x_i is the node value i of the previous layer, b_j is the bias of the node j in the current layer, w_{ji} is the connection weight of x_i and y_j, N is the number of nodes in the previous layer, and f is the activation function in the current layer. In the learning phase, 70% of the data were used for the training phase, and 30% for the testing phase. The dataset included 2,000,000 samples, of which 1,350,974 were for the training phase and the rest for the testing phase. The precision of MLP-ANN was >80%.

Researchers also analyzed the tax return data of a group of commercial sellers in Telangana (India) based on graph clustering [41]. In graph clustering, the top-down method is used, and each sample is assigned to a cluster closer to the samples. The closest Euclidean distance for clustering was used to identify similar samples. The results showed that clustering affected the tax samples, and suspicious samples were detected by clustering [41]. Another study used the machine learning classification approach to detect fraudulent samples of government-linked companies in Malaysia [42]. Therefore, researchers applied LR, SVM, KNN, MLP, DT, and random forest models to detect and classify the samples. The 24-feature dataset included fraudulent companies from 2010 to 2016. The findings indicated that the detection precision of the random forest model and DT was higher than in other models [42].

Another research aimed to identify companies that experienced fraudulent financial statements between 2002 and 2013 [43]. Hence, two regression tree (CART) and Chi-squared automatic interaction detector (CHAID) algorithms

Table 1: Advantages and disadvantages of the proposed models for tax evasion detection.

Refs.	Models	Tools	Data	Validation method	Advantages	Disadvantages
[27]	C5.0	Microsoft.NET framework	500	Accuracy	*Careful classification Proportion of positive and negative *quick update	Increase the depth of the tree
[28]	Decision tree, logistic regression	Statistical	402	Prediction efficiency (PE), examination effort (EF), strike rate (SR)	LR is easier to implement, interpret, and very efficient to train	Overfitting
[29]	Association rule	DB miner	—	Accurate rate, error rate	It is appropriate for low transaction dataset	It needs multiple passes over the dataset
[30]	MLP, SVM, LR, HSA	Statistical	4504	Accuracy, sensitivity, specificity, AUROC	Fast convergence, increase efficiency, increase detection accuracy	Overfitting MLP is sensitive to feature scaling
[31]	Linear regression, SVM	Statistical	—	Accuracy	SVM is more effective in high dimensional spaces, proper performance of SVM in memory usage	Kernel function is not easy long training time for large datasets
[32]	Bayesian networks	Statistical	10028	Speedup	A strong and mathematically coherent framework for the analysis	The memory utilization is more
[33]	Colored network-based model (CNBM)	Framework	31,910,000	Accuracy	Accurate detection of samples segmentation of samples based on the weight of the samples	In CNBM model, the selection of optimal parameters values is required
[34]	Conditional maximum mean discrepancy (CMMD)	Coding	—	Accuracy	Accurate detection of samples	Increasing complexity
[35]	LR, k-medoids	Statistical	—	Mean absolute deviation (MAD), root mean square error (RMSE)	Discover the exact center for the samples	High prediction complexity for large datasets
[36]	MLP, SVM, logistic regression, random forest	Statistical	700,000	Accuracy, AUCROC, precision, F1-score, recall	Fast convergence, Increase efficiency, Increase detection accuracy	Overfitting
[37]	Transaction network representation	Light-GBM	9,422,952	True positive, true negative, false negative, false positive, error rate, precision, recall, F-measure, ROC	Quick calculation time Search space is correct. Inexpensive testing of each instance	With big dataset, the prediction stage might be slow
[38]	Deep learning	Light-GBM	20444	Accuracy, F1-Score, AUC	No need for labeling of data Suitable for bulk data Learning of layers based on calculate of individual neurons	Increasing complexity
[39]	BP-ANN, CHAID tree	Intelligent miner	12458	Accuracy rate, error rate	Low prediction complexity for large datasets	The memory utilization is more
[40]	ANN-MLP	Statistical	2,000,000	Sensitivity	Fast convergence, Increase efficiency	MLP may suffer from over fitting
[41]	Clustering	Statistical	—	Accuracy	Simple execution	The memory utilization is more
[42]	LR, SVM, KNN, MLP, DT, RF	Statistical	—	Accuracy, precision, recall	KNN finds the k-nearest data points in the training set	Kernel function is not easy
[43]	CART, CHAID	Statistical	—	Accuracy rate, error rate	High accuracy Discover important features	Stuck in the local optimal

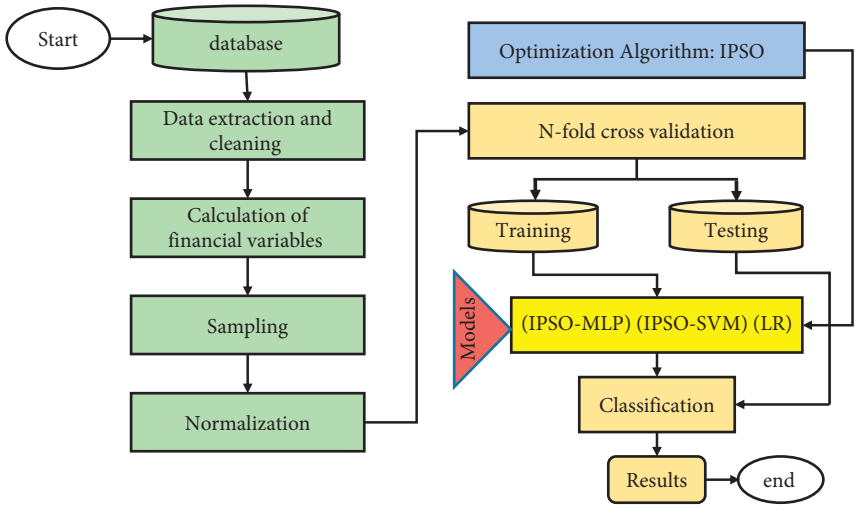

Figure 1: Block diagrams of the proposed models.

were used to select the main variables of fraud detection. The second stage combined CART, CHAID, deep belief network, support vector machine, and artificial neural network to create models for detecting fraudulent financial statements. According to the results, the detection performance of the CHAID-CART model with 87.97% precision was better than other models. Table 1 presents the advantages and disadvantages of the proposed models for tax evasion detection. Each model has some advantages and disadvantages that lead to success or inefficiency. According to the analysis of the literature review, it was concluded that artificial neural networks had better detection and minor error due to the pattern recognition capability, optimal relationship between input and output data, less sensitivity to errors in input data and training of neurons, parallel processing, fewer input data, and faster and easier verification process in detecting and predicting the relationship between tax evasion factors.

In ANN, the error and trial method is mainly used to determine the optimal number of hidden layers, and therefore, a structure with the least number of hidden layers must be selected with an acceptable degree of error. The fewer hidden layers of a network take less time to train a network.

Moreover, the number of neurons in the hidden layers has a significant effect on ANN function. The use of a small number of neurons leads to inaccurate learning of most samples by the ANN. On the other hand, the presence of many neurons results in the preservation of patterns and thus prevents the neural network from learning to detect their basic features. According to the analysis, it was concluded that issues such as the number of hidden layers and number of neurons should be considered in ANN.

Furthermore, precise feasibility and ease of implementation are determinants for choosing the appropriate model for tax evasion detection (TED). As mentioned in the extensive previous literature, the use of machine learning models such as SVM and MLP is recommended due to the increased precision compared to previous models such as decision trees (DTs). According to this and other advantages, including flexibility, efficiency, and precision of instance detection, this study uses machine learning models for TED.

3. Proposed Models

This section explains the IPSO-MLP, IPSO-SVM, and LR models. The IPSO-MLP model uses IPSO to find the weight of the MLP network, and the IPSO-SVM model employs IPSO to find the SVM parameters. Figure 1 depicts a block diagram of hybrid models.

Data extraction and cleaning: first, the data were extracted from the tax administration in an Excel file. Then, unsuitable and scattered data were identified and deleted.

Calculation of financial variables: the dataset includes dependent and independent variables. The dependent variable is a binary variable of 0–1, so that 1 represents the presence of tax evasion, and 0 shows the absence of tax evasion. Independent variables, which are the most important, are classified according to personal taxes. The measurement of a class variable is defined based on the following equation [44]:

$$C = \frac{(\text{TAXIN} - \text{ACCIN})}{\text{ACCIN}} \times 100, \tag{4}$$

where C is the percentage difference between the included tax expressed profit difference and the profit included deterministic tax in year t. The ACCIN and TAXIN parameters are the included tax expressed profit difference at the end of the fiscal year, and the profit included deterministic tax at the end of the fiscal year, respectively. If there is a 15% difference between the profit included deterministic tax and the included tax expressed profit difference of the business unit, then it will be considered as tax evasion of the business unit.

Sampling: data samples are collected from the tax database, and records that are most likely to be involved in tax evasion are selected.

Normalization: data normalization is performed for all proposed models. In the proposed models, the samples are

first to read from the dataset file, and then, the preprocessing operation is performed. Standardization was performed in the preprocessing stage to normalize the data in a specific range. In general, data in different change ranges cannot positively affect each other or the model. Therefore, the data should be in an equal range (e.g., they should be 0 to 1). The normalization operation on the data is defined based on the following equation:

$$N_i = 0.5 \times \left[\frac{x_i - x_{\text{mean}}}{x_{\max} - x_{\min}} \right] + 0.5, \tag{5}$$

where the parameter N_i is the normalized values, x_i represents the actual values, x_{mean} is the average of the actual values, $x_{\max}$ is the maximum actual values, and $x_{\min}$ refers to the minimum actual values.

The 10-fold cross-validation method is used to perform the training and test process. Therefore, each dataset is divided into ten parts, and nine parts are used in each implementation as a training group and one part as a test group.

Optimization algorithm: the main goal in this step is to maximize the precision of the classification. The use of the IPSO algorithm in the MLP network training and optimization of SVM parameters accelerates the operation and increases the precision of the results.

3.1. IPSO Algorithm. The particle swarm optimization (PSO) algorithm is a social search algorithm inspired by the social behavior of birds when searching for food [23]. There are several particles in this algorithm that seek to optimize an optimization problem in a search space. Each particle calculates the goodness-of-fit function in its current position. Then, it selects an optimal direction for movement by comparing information about its current position and the best position it has ever been in and information about the best particles in the group. Hence, all the particles choose the movement direction, and one step of the algorithm is completed after each movement.

In the PSO algorithm, the position of particle I is defined as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. Particle velocity is also defined as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. The goodness-of-fit function for particles in the population is evaluated and compared with the value of the previous best result of the same particle and the best particle in the whole population. In the PSO algorithm, the particles move to the optimal regions under the influence of their experience and knowledge (Pbest) and the knowledge of their neighboring particles to achieve the best solutions. After finding these two optimal values, the particle moves according to (6) and (7) by updating its speed and position.

$$\begin{aligned} V_i^{t+1} &= \omega \times V_i^t + c_1 \times \text{rand}_1 \times \left(p_{best(i)} - X_i^t \right) \\ &\quad + c_2 \times \text{rand}_2 \times \left(g_{\text{best}} - X_i^t \right), \end{aligned} \tag{6}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1}, \tag{7}$$

where $i = (1, 2, \ldots, N_{\text{pop}})$. N_{pop} represents the population size, and $p_{\text{best}(i)}$ is the best response found by the particle i. g_{best} is the best response in the whole group. Parameters c_1 and c_2 are learning parameters whose values can be defined in the range 0–2. The functions rand_1 and rand_2 are two random numbers with uniform probability in the range 0–1. Changes in V_i^{t+1} are in the range $[V_{\min}, V_{\max}]$ so that Vmax is the maximum speed allowed for the particles. The inertia coefficient ω is used to control the search balance of the algorithm between exploration and exploitation. The population size matrix is defined according to the following equation:

$$\text{pop}_{ij}^t = \begin{bmatrix} x_{1,1}^t & x_{1,2}^t & \cdots & \cdots & x_{iD}^t \\ x_{2,1}^t & x_{2,2}^t & \cdots & \cdots & x_{iD}^t \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{\text{pop},1}^t & x_{\text{pop},2}^t & \cdots & \cdots & x_{\text{pop},iD}^t \end{bmatrix}. \tag{8}$$

An essential factor in the PSO algorithm is the conversion of continuous mode to discrete mode. In the discrete state, the movement of the particles is limited to 0 and 1. The parameter v, whose value is mapped to the range 0–1, determines the value of x (position), which means the probability that $x = 1$. Implementing the IPSO model, the particle velocity is mapped to a value between zero and one using the bounded sigmoid function according to (9). Finally, particle i in the d-dimensional dimension is updated according to the following equation:

$$S(v_i) = \text{sigmoid}(v_i) = \frac{1}{1 + e^{-v_i}}, \tag{9}$$

$$x_{i+1} = \begin{cases} 0, & \text{if rand } 0 \geq S(v_i), \\ 1, & \text{if rand } 0 < S(v_i). \end{cases} \tag{10}$$

According to (11) and (12), the learning factors are improved in this paper to encourage the movement of particles in the whole search space and strengthen the convergence rate.

$$c_1 = c_{1,\max} - \left(\frac{k_{\max} - k}{k_{\max}} + c_{1f} \right) \times \left(c_{1,\max} - c_{1,\min} \right), \tag{11}$$

$$c_2 = c_{2,\min} + \left(\frac{k_{\max} - k}{k_{\max}} + c_{2f} \right) \times \left(c_{2,\max} - c_{2,\min} \right), \tag{12}$$

where $c_{1,\max}$, $c_{1,\min}$, $c_{2,\max}$, and $c_{2,\min}$ are the initial constants. $k_{\max}$ is the maximum iteration. c_{1f} and c_{2f} values are determined based on the initial and final values of the learning coefficients c_1 and c_2, respectively. Many algorithms rely on fixed values to generate and search for new solutions. These values play a crucial role in the generation of optimal solutions. If the value is constant and moves in the problem space, the search may reach a final solution within a reasonable time, which may lead to the ignorance of good solutions in the vicinity of local points. In IPSO, the balance between global particle search and local search depends mainly on the learning coefficients. If the amount of learning coefficients is large, the particles are updated in a large area, which develops the global exploration of the algorithm. In contrast, local search plays an essential role in the

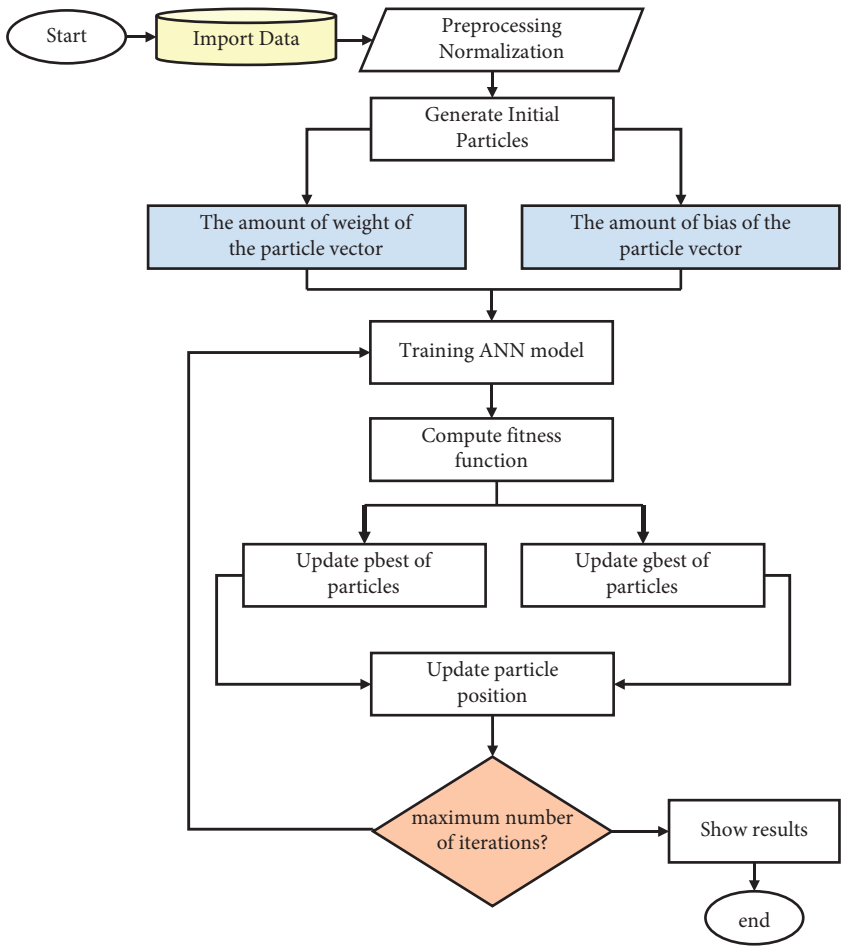

Figure 2: Flowchart of the IPSO-MLP model.

optimization process if the learning factor is negligible. The learning factor is updated during optimization according to the number of iterations to prevent the early convergence of particles and accelerate exploration.

3.2. IPSO-MLP Model. The multilayer artificial neural network mainly consists of three layers (input, middle, and output) [45]. The first layer receives n properties of input $x_1, x_2 \ldots, x_n$, which are processed by subsequent layers. The input layer only receives samples from the dataset and acts as an independent variable. Therefore, the number of input layer neurons is determined based on the number of independent variables. Hidden layers perform intermediate calculations and enable the output layer to predict the optimal response. The output layer acts as a dependent variable, and the number of neurons depends on the number of dependent variables. Each layer consists of nodes connected to all nodes of the next layer, except the input layer, whose nodes contain input properties.

This section explains the combination steps of the MLP artificial neural network with the IPSO algorithm as flowcharts and algorithms. As mentioned earlier, we use the IPSO algorithm to detect tax evasion for increasing the precision, accuracy, and training speed of the MLP artificial neural network. The purpose of training MLP artificial neural networks is to find the size of the weights to minimize the training data error. Hence, MLP artificial neural network training can be considered an optimization problem to optimize the weight coefficients of neurons to achieve the minimum training error. The random production of the initial particle population takes place in the IPSO algorithm. Random production of the initial population is simply the random determination of the initial location of the particles with a uniform distribution in the search space. The position of a particle in the IPSO algorithm is represented by x, which contains n elements such as $(x_1, x_2, \ldots, x_n)$. The next step is to select the number of initial particles. Empirically, the initial population size of 30 to 50 particles is an ideal choice and works well for almost all engineering problems. Then, the objective function must be evaluated. Each particle representing the solution to the problem under study must be evaluated at this stage. The fitness value of each particle is calculated to minimize the error. In the next step, the best position for each particle is determined, and then, the best position among all particles is determined. All particles' position and capability vector are updated, and the particles are directed to the new position. Figure 2 shows the IPSO-MLP model flowchart.

01) Start
02) Entering data: Tax data enters the hybrid model cycle to detect tax evasion.
03) Pre-processing and normalization: Pre-processing and normalization operations equalize the data and delete the outlier data.
04) Creating the initial population of particles: The initial population is created based on the IPSO algorithm.
05) Creating solution vectors based on the number of weights and biases.
05) Training the MLP network
06) Calculating the goodness-of-fit function: The goodness-of-fit function must have the minimum error to find an optimal solution with the best amount of weight and bias.
07) Optimizing the weight of nodes
08) Updating the position of particles
- Updating pbest and gbest

09) Reviewing the conditions of the program
10) Showing the results
11) The end

Figure 3: The pseudocode of the IPSO-MLP model.

According to Figure 2, after entering the data into the IPSO-MLP model, the data are prepared for training using cleaning and normalization. The solution lengths of weights and biases are designed for the MLP network based on the number of weights and the number of biases according to (13). The IPSO-MLP model uses 80% of the data for training and 20% for testing.

$$X = \{w, b\} = \left\{w_{11}, w_{12}, \ldots, w_{ij}, b_1, b_2, \ldots, b_j\right\}, \tag{13}$$

where n is the number of input nodes, w_{ij} is the weight from node i to node j, and b_j is bias.

In general, traditional methods such as the back-propagation algorithm and other gradient methods are used to train artificial neural networks. If the function is nonlinear and complex in these methods, they cause weakness and inefficiency in detection precision. In the back-propagation algorithm, a newly calculated output value is compared with the actual value each step, and the weights and biases of the network are corrected according to obtained error so that at the end of each iteration, the size of the resulting error is less than the value obtained in the previous iteration. This minimization is based on the movement of the gradient vector of the network error squares function, which is obtained by deriving a chain from the error function to each network parameter. Although the back-propagation algorithm is widely used to train artificial neural networks, using this method leads to problems in some cases. These barriers include slow convergence in the training process and early convergence in local minimums. Figure 3 depicts the pseudocode of the IPSO-MLP model.

There are several algorithms for training the multilayer artificial neural network. This paper uses the improved PSO algorithm. In an artificial neural network, the initial values of the weights are of particular importance, and all the values of the weights are selected randomly before the training begins. MLP training aims to achieve the highest classification, approximation, or prediction precision for training and experimental samples. Assuming that the number of input nodes is equal to (N), the number of hidden nodes is equal to (H), and the number of output nodes is (O), then the output of the hidden node i is defined according to (14). The sigmoid activation function in the hidden layer is used in this paper. The sigmoid function maps the value of neurons from 0 to 1 to normalize the total weight of the neurons.

$$f\left(S_j\right) = \text{Sigmoid}\left(s_j\right) = \frac{1}{1 + e^{-\left(\sum_{i=1}^{N} w_{ij}.x_i + b_j\right)}}, \quad j = 1, 2, \ldots, H, \tag{14}$$

where n is the number of input nodes, w_{ij} is the weight from node i in the input layer to node j in the hidden layer, b_j is the bias (threshold) of the hidden node j, and x_i is input i. After calculating the output of the hidden nodes, the final output can be defined according to the following equation:

$$O_k = \sum_{i=1}^{N} w_{kj}.f\left(S_j\right) + b_k, \quad k = 1, 2, \ldots, O, \tag{15}$$

where w_{kj} is the weight from the hidden node j to the output node k, and b_k is the bias (threshold) of the output node k. It should be noted that MSE is used based on (16) to determine the optimal values for weights and biases to reduce the error in the training and optimization process.

$$\text{MSE} = \frac{\sum_{i=1}^{n} \left(O_i - \widehat{O}_i\right)^2}{n}, \tag{16}$$

where O_i is the actual output of the input sample i, $\widehat{O}_i$ is the predicted output of the input sample i, and n is the number of samples.

Network output is calculated each step, and the weights are corrected according to their difference with the desired output to minimize the error value. MSE aims to minimize the discrepancy between the results of the hybrid model and the actual data.

3.3. IPSO-SVM Model. The support vector machine (SVM) is a nonstatistical binary classifier based on regulatory classifications for data analysis [46]. The goal of the support vector machine is to maximize the margin of the hyperplane,

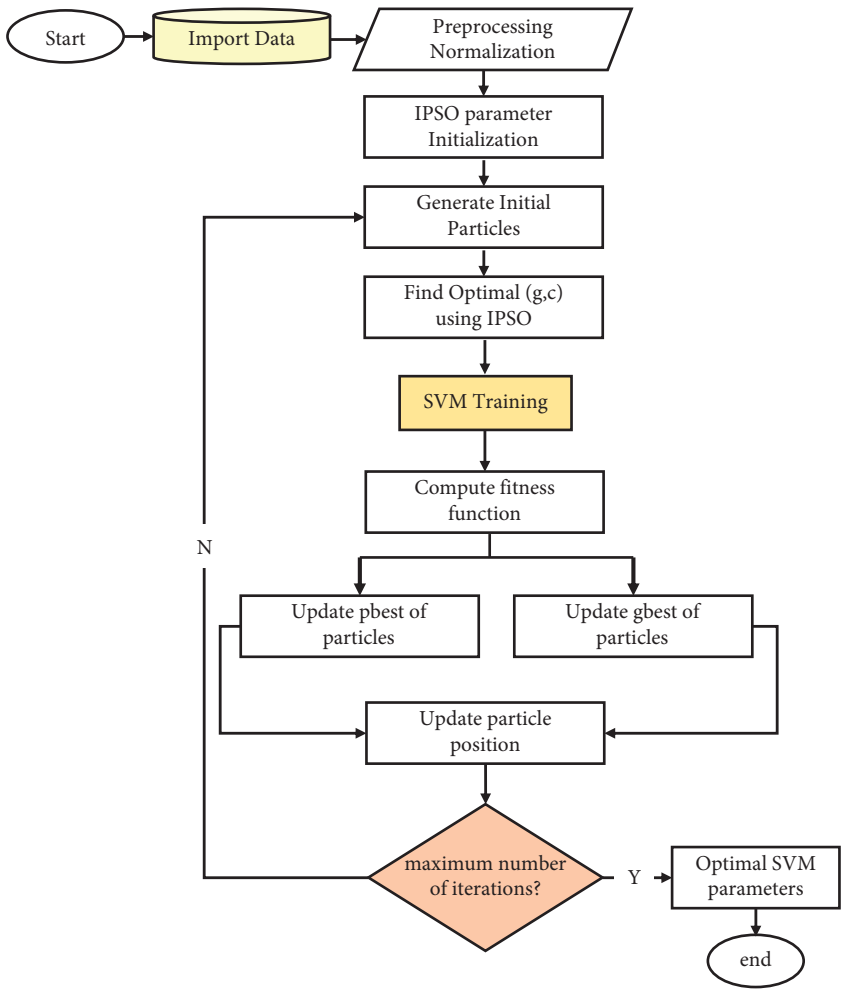

Figure 4: Flowchart of the IPSO-SVM model.

which maximizes the separation between samples. The training points near the separating hyperplane are called support vectors, which are used to identify the margin between classes. This algorithm uses an optimal linear decision margin to separate classes. If the training points are defined as $[x_i, y_i]$, the input vector is defined as $x_i \in R^n$, and the class value is $y_i \in \{1-, 1\}$, $i = 1, 2, \ldots, i$ so that the data can be separated nonlinearly, and the decision rules are defined by the optimal hyperplane for binary decision classes to separate the samples according to the following equation:

$$Y = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i K\left(x \times x_i\right) + b\right), \tag{17}$$

where Y is the output of the equation, and y_i the value of the training sample class and x_i are the parameters α_i and b to determine the hyperplane.

The function $K(x \times x_i)$ is a kernel function that generates inner multiplication to produce machines with different nonlinear surfaces in the data space. Therefore, the concept of classifier margin is used to select the best separating hyperplane in the SVM. If the norm of the vector w is expressed with $\|w\|$, then d is the margin defined for the distance between two classes according to the following equation:

$$d = \frac{2}{\|w\|}. \tag{18}$$

The SVM algorithm is a method to separate and identify two classes by a separating hyperplane defined on the training data. In the SVM algorithm, the decision margin must be able to classify all the samples correctly. Such a decision margin with the ability to classify all samples correctly is defined by solving the finite optimization problem according to the following equation:

$$\begin{cases} \text{Minimize: } \dfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \partial_i, \\ y_i\left(w^T.x_i + b\right) \geq 1, \end{cases} \tag{19}$$

where w is the weight vector or normal optimal hyperplane vector, and b is the oblique vector representing the distance from the hyperplane to the origin. C is the margin adjustment parameter to balance maximizing the margin and minimizing the classification error, which is always greater than zero. The variable $\partial > 0$ is considered to be the interference between the training data. According to (20), the

radial base kernel function transmits data to a space with higher dimensions. The $x_i - x_j$ parameter is the Euclidean distance between two feature vectors, and the user defines g as the kernel width.

$$k\left(x_i, x_j\right) = \exp\left(-gx_i - x_j^2\right), \quad g > 0. \tag{20}$$

The input parameters for SVM are adjusted using the improved PSO algorithm in this paper. Proper selection of the parameters C and g in the support vector machine algorithm is of high importance because they increase the precision of detection and prediction of the SVM. In particular, parameter optimization is an essential step in SVM classification. Figure 4 shows the IPSO-SVM model flowchart.

In the solution vectors, the values x_1 and x_2 are searched in the range $[C_1$–$C_2]$ and $[g_1 - g_2]]$. The position of particles in the problem space is changed according to personal experience and the experience of the best neighbor. Parameters C and g of the SVM classification are defined by mapping x_1 and x_2 according to the following equation:

$$\begin{aligned} C &= C_1 + x_1 \times (C_2 - C_1), \\ g &= g_1 + x_2 \times (g_2 - g_1). \end{aligned} \tag{21}$$

The population in the iteration t is defined as $\{x_1^t, \ldots, x_i^t, .., x_{NP}^t\}$ so that each particle is defined as $x_i^t = (x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t)$ where NP is the population size and D is the dimensions of each particle. In the hybrid model, each particle is defined as $x_i^t = (C_i^t, g_i^t)$. The goodness-of-fit function is evaluated based on the accuracy criterion. The best accuracy value with maximum iterations is displayed in the output.

3.4. Logistic Regression Algorithm. Logistic regression [47] is a particular form of linear regression in which the response variable is discrete. Like linear regression, there are one or more independent variables in this type of regression, based on which the probability of each of the two-state variable levels of the dependent variable can be calculated. The logistic regression model for the independent variables p is defined according to the following equation:

$$P(Y = 1) = \frac{1}{1 + e^{-\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p\right)}}, \tag{22}$$

where Y is the probability that the dependent variable is equal to one, $\beta_0 + \beta_1 + \beta_2 + \cdots + \beta_p$ is the estimated coefficient of the variable in the model by logistic regression, and $x_1 + x_2 + \cdots + x_p$ is the independent variable in the model. Using the estimated features, the probability of presence for each response variable is defined according to (23) so that $P(Y = 1)$ is the probability of a response variable. The margin between the presence and absence of the response variable is 0.5, which classifies the response variable into zero or one class. If the value of the response margin is closer to one, it represents the presence and probability of more positivity.

$$g(x) = Ln\left(\frac{P(Y = 1)}{1 - P(Y = 1)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p. \tag{23}$$

The advantage of logistic regression over other regressions that obtain model coefficients with total squares is that a linear relationship between independent and dependent variables should not exist. Moreover, it does not require a normal distribution between variables, assumes that the variables have equal statistical variances, and generally includes fewer hypotheses.

3.5. Computational Complexity. This section explains the computational complexity of IPSO-MLP. The complexity of IPSO mainly depends on factors, including population size, the maximum number of iterations, the number of variables, and the number of iteration loops. The temporal complexity of MLP is equal to $O\ (n \times m)$, where n is the number of neurons, and m is the number of layers. In addition, the computational complexity of PSO equals $(I \times P \times D)$, where I is the maximum iteration, P represents the population size, and D is the particle size. The computational complexity of IPSO in the learning factor stage equals O(N). Therefore, the overall complexity of IPSO-MLP equals $(I \times (n \times m + N + P \times D))$. In general, the complexity of the SVM algorithm is $O(n^2)$ where n is the number of training instances; hence, the overall complexity of IPSO-MLP is equal to $(I \times (n^2 + N + P \times D))$.

3.6. Evaluation Criteria. Precision, recall, F1, and accuracy criteria are widely used for classification. Evaluations made by the proposed models for a customer's validity may be bad validity (positive) or good validity (negative). Therefore, the following four situations may occur for a customer:

(a) The prediction result is tax evasion, but the customer includes tax evasion based on the empirical classification called true positive

(b) The prediction result is tax evasion, but the customer includes the absence of tax evasion based on the empirical classification, called false positive

(c) The result of the prediction is the absence of tax evasion, but the customer includes the absence of tax evasion classification based on the empirical classification, which is called true negative

(d) The result of the prediction is the absence of tax evasion, but the customer includes tax evasion based on the empirical classification, which is called false negative

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100, \\ F1 &= \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}, \\ \text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}} \times 100, \end{aligned} \tag{24}$$

Table 2: Initialization of parameters.

Parameters	Value	Algorithm
Number of particles	50	IPSO
Iteration	500	IPSO
C1	1	IPSO
C2	2	IPSO
w	1	IPSO
Neuron input layer	8	MLP
Neuron hidden layer	10	MLP
Number of hidden layers	3	MLP
Training data	80%	MLP
Testing data	20%	MLP
Activation function	Sigmoid	MLP
Learning rate	0.15	MLP
C	1–10	SVM
g	0–1.0	SVM

Table 3: Evaluation of models based on classification dataset.

Datasets	Instances	Features	Classes	Models	Accuracy
Heart Cleveland	303	14	2	IPSO-MLP	87.15
				IPSO-SVM	86.49
				LR	84.61
Hepatitis	155	19	2	IPSO-MLP	81.11
				IPSO-SVM	80.64
				LR	79.29
Diabetes	768	8	2	IPSO-MLP	85.26
				IPSO-SVM	84.19
				LR	83.07
Cancer	699	9	2	IPSO-MLP	98.56
				IPSO-SVM	98.25
				LR	96.13
Heart Stalog	270	13	2	IPSO-MLP	87.35
				IPSO-SVM	86.07
				LR	85.34
Lung cancer	32	56	2	IPSO-MLP	90.49
				IPSO-SVM	89.27
				LR	88.41
Sonar	208	60	2	IPSO-MLP	95.76
				IPSO-SVM	94.08
				LR	93.56

where TP is the number of correct records of positive cases (tax evasion), FP represents the number of incorrect records of positive cases, FN is the number of incorrect records of negative cases, and TN refers to the number of correct records of negative cases (no tax evasion).

Table 4: The results of the models based on different criteria.

Models	Precision	Recall	F1	Accuracy
SVM	88.67	89.16	88.91	90.58
KNN	81.70	82.42	82.06	85.33
C5.0	84.11	85.61	84.85	88.66
NB	85.33	86.67	85.99	88.89
MLP	89.82	90.45	90.13	91.33
Adaboost	88.00	88.32	88.16	89.65
IPSO-MLP	93.25	93.78	93.51	**93.68**
IPSO-SVM	92.64	92.75	92.69	92.24
LR	91.80	82.34	86.81	67.00

The bold value indicates the highest accuracy.

4. Evaluation and Analysis

This section evaluates three proposed methods (IPSO-MLP, IPSO-SVM, and LR) based on machine learning for tax evasion detection. As mentioned earlier, this paper uses the dataset of the General Administration of Tax Affairs of West Azerbaijan Province in 2019 with different groups of 1500 samples and nine features (gross taxable income, including expressive tax net income, related tax, tax exemptions, tax discount, tax payable, payments made, taxable balance, and class feature). The models are implemented in MATLAB 2017b. One of the most critical parts of determining the optimal structure of a multilayer artificial neural network is to determine the number of hidden layers and the number of neurons in each hidden layer to achieve the minimum error. Table 2 presents the initial values of the parameters to run the models.

4.1. Applied Study. This section evaluates the models based on the classification dataset. The models' performance has been tested using seven reference datasets. These datasets have been taken from the machine learning repository (UCI) (https://archive.ics.uci.edu/ml/datasets.php), and Table 3 reports their specifications. These datasets have been chosen because they have been mainly used to prove the experimental performance of algorithms. This paper has used the classification dataset to show the efficiency of the IPSO-MLP, IPSO-SVM, and LR models to determine the percentage accuracy. According to the results of Table 3, the accuracy percentage of the IPSO-MLP model on Heart Cleveland is 87.15. However, the accuracy percentage of IPSO-SVM and LR models is lower than that of the IPSO-MLP model.

The accuracy of the IPSO-MLP model on cancer is 98.56. The MLP performance depends on the choice of various parameters such as the initial weight and size of hidden nodes. Optimal adjustment of the parameters of an artificial neural network, including the selection of the appropriate initial weights, leads to solving slow and early convergence problems of the training process. According to this study, it can be concluded that selecting optimal weights and the number of hidden nodes helps MLP performance to increase the accuracy of classification detection.

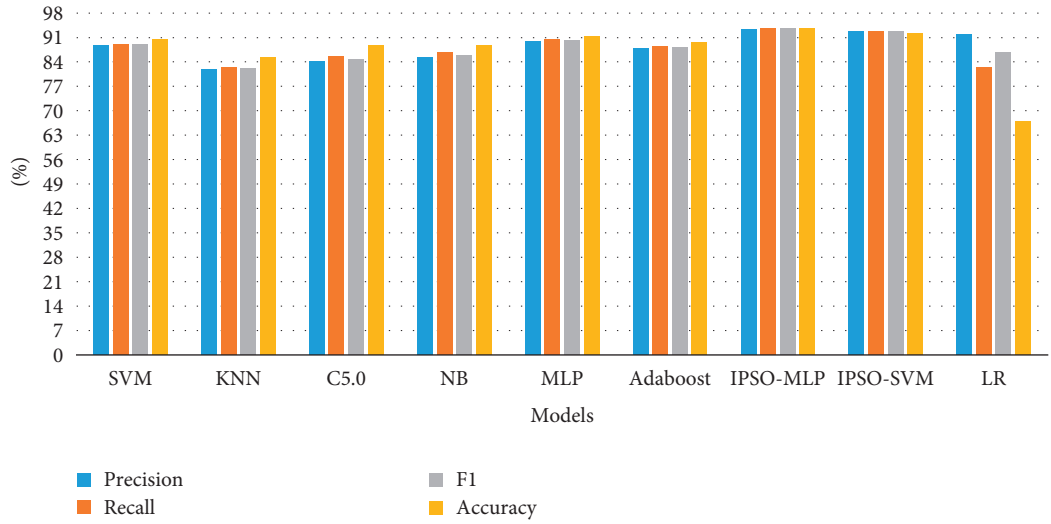

Figure 5: Comparison diagram of the models based on different criteria.

Table 5: Results of the IPSO-MLP model based on the number of different layers.

Models	Number of layers	Precision	Recall	F1	Accuracy	MSE
IPSO-MLP	3	93.25	93.78	93.51	**93.68**	0.033995
	5	90.35	91.72	91.03	**91.47**	0.036564
	7	90.65	90.23	90.44	**91.19**	0.035241

4.2. Evaluation of Models. Table 4 and Figure 5 show the results of the models based on different criteria. To evaluate the detection precision of the IPSO-MLP, IPSO-SVM models, SVM, KNN, C5.0, NB, MLP, and AdaBoost models are used for comparison. According to the results, the accuracy of the IPSO-MLP model is 93.68, which is higher than the other models. Moreover, the SVM and MLP models have higher detection precision than KNN, C5.0, NB, and LR models. The precision and recall percentages in the IPSO-MLP model are 93.25 and 93.78, respectively. The precision and recall percentages in the IPSO-SVM model are 92.64 and 92.75, respectively. The precision and recall percentages in the LR model are 91.80 and 82.34, respectively. The precision, recall, and accuracy percentages in the MLP model are 89.82, 90.45, and 91.33, respectively.

According to Figure 5, hybrid models have a higher percentage of detection precision. The IPSO-MLP and IPSO-SVM models exhibit higher efficiency and precision using IPSO. The strength and efficiency of the MLP model lie in its internal structure. If the internal structure of the MLP is appropriately trained, the MLP output will be high precision.

Table 5 reports the results of the IPSO-MLP model based on the number of different layers. According to the results, it is clear that the IPSO-MLP model with three layers has a higher percentage of accuracy. Different hidden layers are used in MLP, and the optimal number is determined to minimize errors. The process starts with a small number of layers, and additional layers continue until increasing the layers does not improve the error. When there are 5 and 7 layers, the accuracy percentage is 91.47 and 91.19, respectively. In contrast, the 3-layer IPSO-MLP model has a lower MSE error rate and higher detection precision. The MSE value is the mean value of the best combination of the connection weights and bias values.

Figure 6 shows the run diagram of the IPSO-MLP model. According to the Figure, the horizontal axis represents the epochs, and the vertical axis represents the MSE value. The run of IPSO-MLP is shown based on the training, validation, and testing stages. It can be observed that the amount of MSEs of training, validation, and testing is gradually decreased. The error value in Figure 6(b) is lower than in Figure 6(a). According to the results, it can be concluded that the more the number of epochs, the amount of error will be less.

Figure 7 depicts a comparison graph of the IPSO-MLP and IPSO-SVM models based on different runs. As shown in the figure, it is clear that the IPSO-MLP model has a higher percentage of accuracy in all runs.

Figure 8 shows a comparison graph of the IPSO-MLP and IPSO-SVM models based on the number of iterations of the IPSO algorithm. As shown in the figure, it is clear that the IPSO-MLP model has a higher percentage of accuracy in all iterations. The accuracy of the IPSO-MLP and IPSO-SVM model with 100 iterations is 90.47 and 89.75, respectively. The accuracy percentage with 300 iterations is 92.35 and 91.48, respectively. The IPSO algorithm with a reinforcement learning rate prevents local optimization and early convergence in the PSO algorithm. With increasing iteration time, the global search capability increases by IPSO at high iterations and thus improves the convergence speed.

This paper examines different machine learning models and confirms that IPSO-MLP is a good model for tax evasion detection. It should be noted that the IPSO-SVM model performs better than models such as SVM, KNN, NB, and

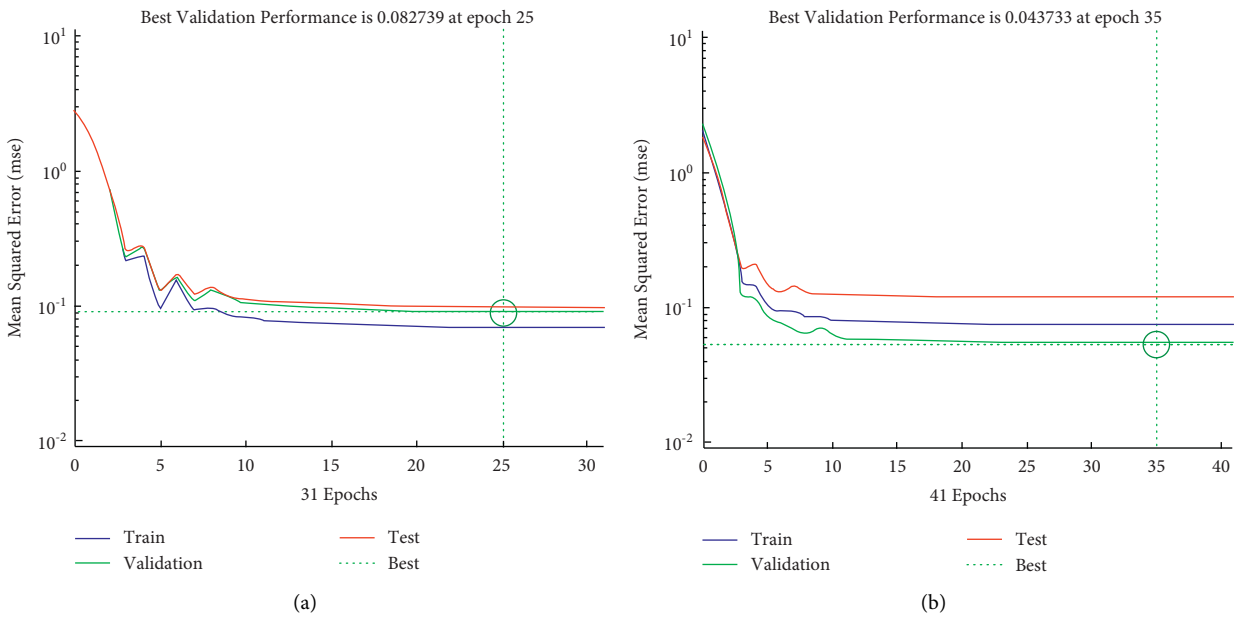

Figure 6: Run of the IPSO-MLP model based on MSE for training, validation, and testing stages.

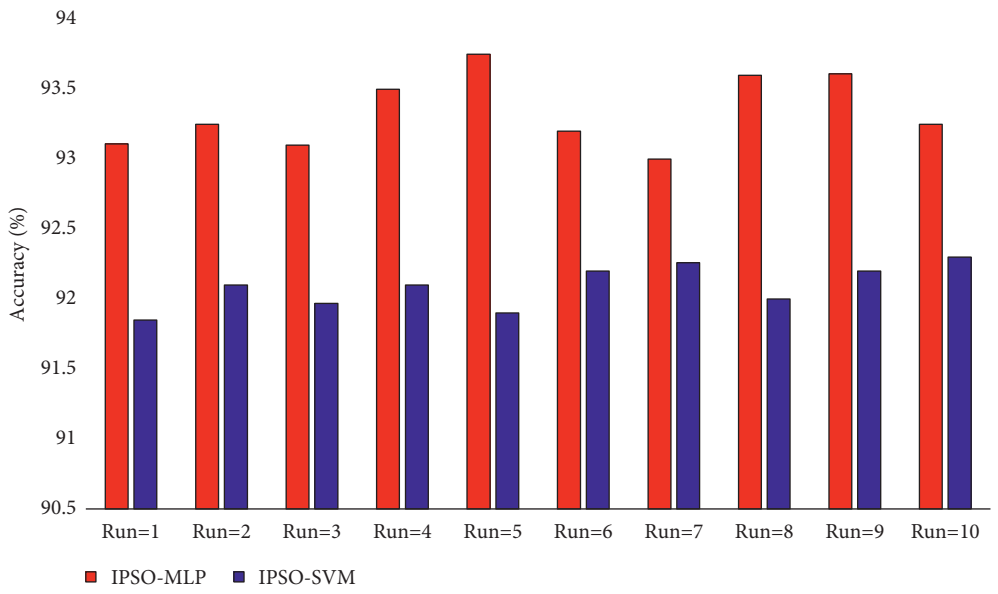

Figure 7: Comparison of the IPSO-MLP and IPSO-SVM models based on different runs.

C5.0. In general, it can be concluded that the combination of machine learning algorithms increases detection precision.

Table 6 compares IPSO with the genetic algorithm (GA), artificial bee colony (ABC) [48], firefly algorithm (FA) [49], and imperialist competitive algorithm (ICA) [50]. The parameters included in the algorithms were set as follows: the maximum number of iterations is 500, and the population size is 50. Each algorithm was run ten times independently. Table 6 presents the average of the results obtained by each algorithm. According to the table, it is clear that the IPSO algorithm has a higher percentage of accuracy than that of GA, ABC, FA, and ICA. The accuracy percentage of IPSO-MLP, GA-MLP, ABC-MLP, FA-MLP, and ICA-MLP was 93.68, 93.11, 93.26, 91.94, and 93.27, respectively. In addition, the accuracy percentage of IPSO-SVM, GA-SVM, ABC-SVM, FA-SVM, and ICA-SVM was 92.24, 91.43, 91.74, 92.38, and 91.53, respectively. IPSO was used to extract the optimal MLP and SVM parameters for changing the learning coefficients. MLP is a predictive model for establishing a mapping relationship between input and output instances.

According to the analyses, the IPSO-MLP model showed the highest classification among the compared models when the performance of tax evasion detection models was evaluated. It was found that the LR model classified data

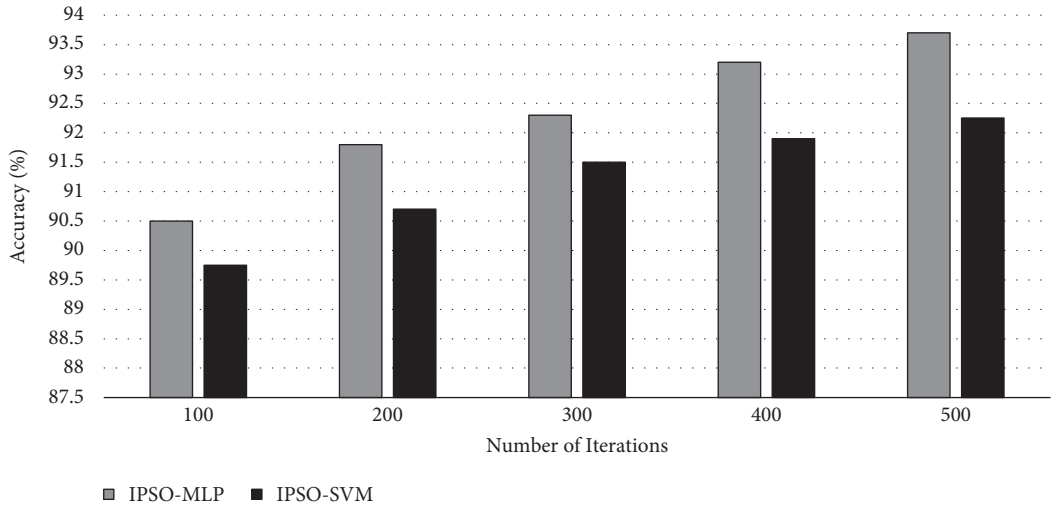

Figure 8: Comparison of the IPSO-MLP and IPSO-SVM models based on IPSO iterations.

Table 6: Comparison of IPSO algorithm with other algorithms.

Models	Precision	Recall	F1	Accuracy
IPSO-MLP	93.25	93.78	93.51	**93.68**
IPSO-SVM	92.64	92.75	92.69	92.24
LR	91.80	82.34	86.81	67.00
GA-MLP	92.87	92.93	92.90	93.11
GA-SVM	91.34	91.57	91.45	91.43
ABC-MLP	92.91	93.19	93.05	93.26
ABC-SVM	91.16	91.38	91.27	91.74
FA-MLP	91.07	91.46	91.26	91.94
FA-SVM	90.89	91.22	91.05	92.38
ICA-MLP	92.90	93.13	93.01	93.27
ICA-SVM	90.72	90.89	90.80	91.53

The bold value indicates the highest accuracy.

inefficiently with a minimum accuracy (67%). The higher accuracy belonged to the FA-SVM model, which is superior to previous models. The ABC-MLP and ICA-MLP models showed the same accuracy values as the IPSO-MLP model. However, the IPSO-MLP model revealed better detection due to improved learning factors.

4.3. Evaluate Statistical Analyses Such as ANOVA. Analysis of variance (ANOVA) is not very efficient and accurate in this paper. For example, the LR method is used in this paper, and ANOVA was implemented to some extent, though the precision of ANOVA detection cannot be compared to machine learning algorithms.

5. Conclusion and Further Research

Tax evasion is a main problem of the tax system in most countries of the world. Due to the importance of tax evasion, it is essential to use methods that can identify tax evasion cases for the tax administration. Since machine learning algorithms have predictive and classification features, the decision-making process in financial issues can be facilitated. Moreover, neural networks provide low-cost algorithmic solutions and facilitate analysis because they do not require different statistical assumptions. This paper investigates the efficiency and ability of machine learning methods in the field of tax evasion detection. Therefore, this system is implemented by the tax administration dataset using the 10-fold cross-validation method and an iterative training, testing, and validation method. The paper results on 1500 tax samples indicate that tax evasion may be detected using machine learning methods. The accuracy of the IPSO-MLP model is over 93%. In addition, the IPSO-MLP error value is 0.033995. Evaluation of the hidden layer active neurons and training of the artificial neural network model demonstrate that 30 iteration cycles with ten hidden layer neurons as an optimal artificial neural network are suitable for tax evasion detection.

Furthermore, the IPSO-SVM, SVM, and MLP models perform well. Therefore, future research should investigate the importance of population initialization in the PSO algorithm for convergence rate and the quality of the final solution. Moreover, opposition-based learning can be used to increase diversity in the initial population. Consequently, the whale and gray wolf optimization algorithms may be used in the exploration phase of the PSO algorithm, and each of them can be tested separately.

Data Availability

Our proposed method applies the dataset collected from the general administration of tax affairs of the West Azerbaijan province of Iran.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This manuscript is part of the research in the Ph.D. thesis of Houri Mojahedi and my cooperation with Dr. Amin

Babazadeh Sangar and Dr. Mohammad Masdari. This research is self-funding.

References

[1] I. Lee and Y. J. Shin, "Machine learning for enterprises: applications, algorithm selection, and challenges," *Business Horizons*, vol. 63, no. 2, pp. 157–170, 2020.

[2] H. H. Le and J.-L. Viviani, "Predicting bank failure: an improvement by implementing a machine-learning approach to classical financial ratios," *Research in International Business and Finance*, vol. 44, pp. 16–25, 2018.

[3] K. Adandohoin, "Tax transition in developing countries: do value added tax and excises really work?" *International Economics and Economic Policy*, vol. 18, no. 2, pp. 379–424, 2021.

[4] T.-H. Chen and R.-C. Chang, "Using machine learning to evaluate the influence of FinTech patents: the case of Taiwan's financial industry," *Journal of Computational and Applied Mathematics*, vol. 390, Article ID 113215, 2021.

[5] A. Mohamed, M. K. Najafabadi, Y. B. Wah, E. A. K. Zaman, and R. Maskat, "The state of the art and taxonomy of big data analytics: view from new big data framework," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 989–1037, 2020.

[6] K. A. Blaufus, A. N. Möhlmann, and A. N. Schwäbe, "Stock price reactions to news about corporate tax avoidance and evasion," *Journal of Economic Psychology*, vol. 72, pp. 278–292, 2019.

[7] N. Varela, L. P. Carrasquilla Díaz, and O. B. Pineda Lezama, "Design and implementation of a system to determine tax evasion through de stochastic techniques," *Procedia Computer Science*, vol. 175, pp. 647–652, 2020.

[8] M. Cooper and Q. T. Nguyen, "Multinational enterprises and corporate tax planning: a review of literature and suggestions for a future research agenda," *International Business Review*, vol. 29, no. 3, Article ID 101692, 2020.

[9] D. Barker, D. C. Ling, and M. Petrova, "The benefits and costs of tax deferral: an analysis of section 1031 exchanges," *Journal of Real Estate Literature*, vol. 28, no. 1, pp. 1–29, 2020.

[10] P. Castellón González and J. D. Velásquez, "Characterization and detection of taxpayers with false invoices using data mining techniques," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1427–1436, 2013.

[11] P. M. S. Choi, C. Y. Chung, and D. Kim, "Corporate tax, financial leverage, and portfolio risk," *The North American Journal of Economics and Finance*, vol. 54, Article ID 101264, 2020.

[12] M. Hosseinzadeh, A. M. Rahmani, B. Vo, M. Bidaki, M. Masdari, and M. Zangakani, "Improving security using SVM-based anomaly detection: issues and challenges," *Soft Computing*, vol. 25, no. 4, pp. 3195–3223, 2021.

[13] E. Hemberg, J. Rosen, G. Warner, S. Wijesinghe, and U. M. O'Reilly, "Detecting tax evasion: a co-evolutionary approach," *Artificial Intelligence and Law*, vol. 24, no. 2, pp. 149–182, 2016.

[14] Z. Chen, L. D. Van Khoa, E. N. Teoh, A. Nazir, E. K. Karuppiah, and K. S. Lam, "Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: a review," *Knowledge and Information Systems*, vol. 57, no. 2, pp. 245–285, 2018.

[15] Z. Zheng, W. Wei, C. Liu, W. Cao, L. Cao, and M. Bhatia, "An effective contrast sequential pattern mining approach to taxpayer behavior analysis," *World Wide Web*, vol. 19, no. 4, pp. 633–651, 2016.

[16] K. Asghari, M. Masdari, F. S. Gharehchopogh, and R. Saneifard, "Multi-swarm and chaotic whale-particle swarm optimization algorithm with a selection method based on roulette wheel," *Expert Systems*, vol. 38, no. 8, Article ID e12779, 2021.

[17] S. Gharehpasha and M. Masdari, "A discrete chaotic multi-objective SCA-ALO optimization algorithm for an optimal virtual machine placement in cloud data center," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 10, pp. 9323–9339, 2021.

[18] S. C. Nayak and B. B. Misra, "A chemical-reaction-optimization-based neuro-fuzzy hybrid network for stock closing price prediction," *Financial Innovation*, vol. 5, no. 1, pp. 38–34, 2019.

[19] J. D. J Rocha Salazar, "Money laundering and terrorism financing detection using neural networks and an abnormality indicator," *Expert Systems with Applications*, vol. 169, Article ID 114470, 2021.

[20] A. O. Adewumi and A. A. Akinyelu, "A survey of machine-learning and nature-inspired based credit card fraud detection techniques," *International Journal of System Assurance Engineering and Management*, vol. 8, no. S2, pp. 937–953, 2017.

[21] M. Masdari and S. Barshandeh, "Discrete teaching-learning-based optimization algorithm for clustering in wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 5459–5476, 2020.

[22] G Dhiman, "SSC: a hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications," *Knowledge-Based Systems*, vol. 222, Article ID 106926, 2021.

[23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95-international conference on neural networks*, IEEE, Perth, WA, Australia, 27 November 1995 - 01 December 1995.

[24] S. Rengasamy and P. Murugesan, "PSO based data clustering with a different perception," *Swarm and Evolutionary Computation*, vol. 64, Article ID 100895, 2021.

[25] X. Zhang, Q. Lin, W. Mao, S. Liu, Z. Dou, and G. Liu, "Hybrid particle swarm and grey wolf optimizer and its application to clustering optimization," *Applied Soft Computing*, vol. 101, Article ID 107061, 2021.

[26] A. Mohammadzadeh, "Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing," *Evolutionary Intelligence*, pp. 1–29, 2020.

[27] F Yu, Z. Qin, and X.-L. Jia, "Data mining application issues in fraudulent tax declaration detection," in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693)*, IEEE, Xi'an, China, 05-05 November 2003.

[28] M. Gupta and V. Nagadevara, "Audit selection strategy for improving tax compliance: application of data mining techniques," in *Proceedings of the Foundations of Risk-Based Audits. Proceedings of the eleventh International Conference on e-Governance*, Hyderabad, India, December. 2007.

[29] R.-S. Wu, C. Ou, H Lin, S. I. Chang, and D. C. Yen, "Using data mining technique to enhance tax evasion detection performance," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8769–8777, 2012.

[30] E. Rahimikia, S. Mohammadi, T. Rahmani, and M. Ghazanfari, "Detecting corporate tax evasion using a hybrid intelligent system: a case study of Iran," *International Journal of Accounting Information Systems*, vol. 25, pp. 1–17, 2017.

[31] M. S. Rad and A. Shahbahrami, "Detecting high risk taxpayers using data mining techniques," in *Proceedings of the 2016 2nd*

International Conference of Signal Processing and Intelligent Systems (ICSPIS), IEEE, Tehran, Iran, 14-15 December 2016.

[32] M. S. Rad and A. Shahbahrami, "High performance implementation of tax fraud detection algorithm," in *Proceedings of the 2015 Signal Processing and Intelligent Systems Conference (SPIS)*, IEEE, Tehran, Iran, 16-17 December 2015.

[33] F. Tian, T. Lan, K. M. Chao et al., "Mining suspicious tax evasion groups in big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2651–2664, 2016.

[34] R. Wei, B. Dong, Q. Zhang, and X. Zhu, "Unsupervised conditional adversarial networks for tax evasion detection," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 09-12 December 2019.

[35] J. Mathews, P. Mehta, and S. Kuchibhota, "Regression analysis towards estimating tax evasion in goods and services tax," in *Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, IEEE, Santiago, Chile, 03-06 December 2018.

[36] W. Didimo, L. Grilli, G. Liotta, L. Menconi, F. Montecchiani, and D. Pagliuca, "Combining network visualization and data mining for tax risk assessment," *IEEE Access*, vol. 8, pp. 16073–16086, 2020.

[37] Y. Wu, B. Dong, Q. Zheng, and R. Wei, "A novel tax evasion detection framework via fused transaction network representation," in *Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, IEEE, Madrid, Spain, 13-17 July 2020.

[38] F. Zhang, B. Shi, and B. Dong, "TTED-PU: a transferable tax evasion detection method based on positive and unlabeled learning," in *Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, IEEE, Madrid, Spain, 13-17 July 2020.

[39] C.-H. Lin, I. C. Lin, C. H. Wu, Y. C. Yang, and J. Roan, "The application of decision tree and artificial neural network to income tax audit: the examples of profit-seeking enterprise income tax and individual income tax in Taiwan," *Journal of the Chinese Institute of Engineers*, vol. 35, no. 4, pp. 401–411, 2012.

[40] C. Pérez López, M. J. Delgado Rodríguez, and S. de Lucas Santos, "Tax fraud detection through neural networks: an application using a sample of personal income taxpayers," *Future Internet*, vol. 11, no. 4, p. 86, 2019.

[41] P. Mehta, "Detecting tax evaders using TrustRank and spectral clustering," in *Proceedings of the International Conference on Business Information Systems*, Springer, 2020.

[42] R. A. Rahman, S. Masrom, and N. Omar, "Tax avoidance detection based on machine learning of malaysian government-linked companies," *International Journal of Recent Technology and Engineering*, vol. 8, 2019.

[43] S. Chen, "Detection of fraudulent financial statements using the hybrid data mining approach," *SpringerPlus*, vol. 5, no. 1, pp. 89–16, 2016.

[44] M. Dastgir and M. Qaribi, "Using data mining techniques to enhance tax evasion detection performance," *Iranian National Tax Administration (INTA)*, vol. 23, no. 28, p. 0, 2016.

[45] H. You, Z. Ma, Y. Tang et al., "Comparison of ANN (MLP), ANFIS, SVM, and RF models for the online classification of heating value of burning municipal solid waste in circulating fluidized bed incinerators," *Waste Management*, vol. 68, pp. 186–197, 2017.

[46] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[47] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B*, vol. 20, no. 2, pp. 215–232, 1958.

[48] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical report-tr06, Erciyes university, engineering faculty*, 2005.

[49] X.-S. Yang, *Nature-inspired Metaheuristic Algorithms*, Luniver press, 2010.

[50] M. Abdollahi, A. Isazadeh, and D. Abdollahi, "Imperialist competitive algorithm for solving systems of nonlinear equations," *Computers & Mathematics with Applications*, vol. 65, no. 12, pp. 1894–1908, 2013.