Information Sciences 562 (2021) 279–303

Contents lists available at ScienceDirect

Information Sciences

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

Incremental classifier in crime prediction using bi-objective Particle Swarm Optimization

Priyanka Das [a], Asit Kumar Das [a], Janmenjoy Nayak [b], Danilo Pelusi [c], Weiping Ding [d,*]

[a] Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, India
[b] Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, India
[c] Department of Communications Sciences, University of Teramo, Italy
[d] School of Information Science and Technology, Nantong University, Nantong 226019, China

ARTICLE INFO

Article history:
Received 30 March 2020
Received in revised form 29 January 2021
Accepted 1 February 2021
Available online 10 February 2021

Keywords:
Crime report prediction
Particle Swarm Optimization
Incremental classifier
Named entity recognition
Multi-objective optimization

ABSTRACT

Nowadays, the increase in criminal activities has resulted in a massive generation of crime reports describing the details of the crime incidents. Analyzing these reports for crime type prediction helps the law enforcement agencies deal with crime prevention strategies. But it is quite a demanding and difficult task to consider these reports individually and determine their crime types. In the proposed work, an efficient classifier has been designed to analyze the crime reports which not only predict the crime types of the reports but at the same time upgrades itself with the help of new crime reports. Therefore, this task demands an incremental supervised learning technique that continuously learns the existing classifier based on the new set of reports and information already extracted from the old set of reports. Developing an incremental classifier infuses the knowledge that keep coming from the newly generated reports and help in increasing the report-discriminating power of the classifier. In this work, we have applied a Bi-objective Particle Swarm Optimization technique for generating an efficient incremental classifier for classifying and predicting the crime reports dynamically. Crime reports of different countries, such as India, the United States of America, and the United Arab Emirates, have been collected from online classified newspapers to measure the performance of the proposed as well as some state-of-the-art classifiers. Also, the method has been evaluated based on an unbiased police witness narrative crime reports and finally, a statistical test has been performed using all four considered datasets to measure the statistical significance of the proposed methodology.

1. Introduction

Recently, the sudden surge of crime has become a topic of global concern. The increase in the rate of criminal incidents proves to be a major threat worldwide. The crimes have many facets and take place in a wide variety of scenarios and environments. The criminal incidents get reported on a daily basis which help in massive growth of the volume of crime reports. The law enforcement agencies examine these reports in a methodical and standardized manner for crime investigation, which facilitate them to predict the crime trend and prevent from criminal activities in near future. The crimes can be categorized into different classes based on the subject matter, circumstances, and severity. The crimes with the least violations

* Corresponding author.
E-mail addresses: priyankadas700@gmail.com (P. Das), akdas@cs.iiests.ac.in (A.K. Das), mailforjnayak@gmail.com (J. Nayak), dpelusi@unite.it (D. Pelusi), dwp9988@163.com (W. Ding).

https://doi.org/10.1016/j.ins.2021.02.002
0020-0255/

are often termed as infractions, whereas more serious crimes are known as misdemeanors and the heinous acts of crimes are known as felonies. The classification of a crime influences both the substances and procedures of a criminal charge. For example, different types of crimes are processed towards different types of courts. Therefore, it is vital to figure out the differences between the crime types. Categorizing the reports according to the crime types is the primary step by which the police can gain an insight into the details of the crime. By the report classification techniques, a predefined label (i.e., crime type) can be assigned to a report. Though this topic has been on the research list for so long and many traditional algorithms are dealing with this problem, but it is very necessary to find an appropriate method that can handle this huge amount of data at a reasonable time. Classification can be used in criminology for procuring knowledge about the crime patterns to reduce the future crime rate.

Often the most significant features related to the crime types are identified to make the classification model [1] more efficient and effective. Artificial neural networks along with gray relational analysis [2] can be used for crime classification and prediction but the problem lies around the running time complexity of the neural networks. A criminal shortlisting scheme has been designed by using the machine learning approaches in [3]. Another recent work on rough set based incremental crime report labelling has been described in [4]. Particle Swarm Optimization (PSO) [5,6] can be used to select the significant and optimal subset of features in a comparatively efficient manner. Though there exists many research works [7,8], which describe various approaches for crime classification, to the best of the authors' knowledge, this is for the first time an incremental crime classification technique based on PSO has been explored in the crime domain. Here, we have used the PSO approach as it results in better diversification and intensification of the crime types and it also provides faster convergence than the other meta-heuristics, like Genetic Algorithm (GA) or Differential Evolution (DE). In the paper, we have initially proposed two different multi-objective PSO based classifiers to predict the crime types of new unlabelled crime reports. In many crime events, new labelled crime reports may be generated over time. These new labelled crime reports are used as the incremental portion of the dataset to upgrade the already developed classifier, hence the resultant classifier is called an incremental classifier. Here it is worthy to mention that, the newly available labelled crime reports may be of class labels different from the existing class labels and/or of the same class labels with different features. Thus, these new labelled reports may cover some new aspects of crime and such diversity can efficiently be captured by the proposed bi-objective PSO. Thus, the proposed model not only predicts the crime types of new reports but also adapts itself based on the patterns derived from the new set of reports. This is the main objective of the incremental classifier which is very much useful in the crime domain specially for crime report prediction.

1.1. Literature survey

Classification has the potential to assist in systematic crime investigation. Besides the common crimes, cybercrime is increasing at an alarming rate. Several types of cybercrimes have been introduced in [9], which demonstrates two case studies by which varying dimensions of crimes have been explored. It suggested that by looking into the minute details of the rapidly flourishing cybercrime incidents, proper preventive measures can be adapted. The crime category of several states of the USA has been predicted in [10] using two most popularly used classification techniques, like Naïve Bayes and Back Propagation algorithms. A decision tree based classifier has been designed by Aziz et al. [11] that helped the law enforcement agencies to analyse the crime patterns and predict the future crime trends. The decision tree and Naïve Bayes classifiers are compared based on the prediction accuracy in [12]. This research is based on socio-economic data collected from 1990 US Census, law enforcement data collected from the 1990 US LEMAS survey, and crime data collected from 1995 FBI UCR. A systematic and comprehensive review work explaining a classification framework for supporting decision making in crime prevention has been introduced in [7]. Forty-four crime related journals, published between 2000–2015, have been analysed in this work. Here, the crimes have been classified into two categories: violent crime and property crime. The crime analysis has been performed based on six different types of classification techniques, like prediction, classification, visualization, regression, clustering and outlier detection. This research has also shown that Bayesian, neural network and nearest neighbour based classifiers are very efficient for decision making in crime prevention.

Apart from the above mentioned works, different machine learning algorithms have also been used for crime data analysis. Short narrative texts accompanying crime records were passed into machine learning algorithms for achieving ecologically more meaningful latent crime classes [13]. This significant crime topic modelling technique has tried to identify behavioural as well as situational facts underlying the crime incidents. Sheela et al. [14] showed that text mining along with classification can be used for extracting valuable information from unstructured documents. In this paper, a crime thesaurus has been constructed and domain related terms and sentences have been identified for deriving patterns. Here, a sentence level identification of criminal events has been performed. The previously extracted patterns have been used for classifying web based documents. Murder cases have been classified by (i) a fuzzy genetic neural network based on trapezoidal and Lagrange's interpolation membership functions [15] and (ii) a hybrid algorithm based on maximum spanning tree and fuzzy neural network [16]. The system developed in [15] performed the experiment on benchmark identification of XOR problem and for the application of the classification of criminal law. Both the works in [15,16] proved to be helpful for the police, judges, lawyers and other people seeking judgements for the crimes faced by them. A rough set based reasoning approach for identification of criminal activities has been shown by Singh et al. [17]. They acknowledged the visual perception of eye witnesses in the form of symbolic representation and this further helped in acquiring knowledge about the physiological and facial characteristics of criminal which help in their identification. Das et al. [18] proposed a rough spanning tree based approach for selecting relevant features for crime data analysis. Spanning tree is also used in a forensic system called ECL-finder [19] for identifying the leaders from a criminal network. This research represented the criminal organization as a graph and the most influential nodes of the graph are chosen as the leaders based on the existence dependencies.

There is a plethora of work on crime data by using evolutionary algorithms and optimization techniques. One of such works [20] introduced a classification model that uses both the concepts of artificial neural network (ANN) and particle swarm optimization (PSO) for categorization of violent crimes. In this research work, PSO has been used for identifying the redundant and irrelevant features which have been removed from the data for achieving a better classification accuracy. Hybrid PSO [5] has also been used for selecting features for crime classification. Here, apart from the ANN and PSO, Gray Relation Analysis (GRA) has been involved to rank the selected features. This research has also compared the PSO based feature selection scheme with other evolutionary algorithms. This study identified the significant features of specific crimes and further classified the crimes into three different categories. Global optimization on newspaper-based crime reports has been demonstrated in [21], where the strength pareto based evolutionary algorithm is used for identifying crime related features from the news reports. Optimized patterns have been extracted from criminal database using the FP-growth and multi objective PSO technique [22]. Here, a certain fitness score has been considered as the threshold and the patterns having values larger than the threshold are processed for optimization. Phillips et al. [23] developed a graph based representation of criminal databases which are analysed in conjunction with socio-economic and socio-demographic factors to discover co-distribution patterns contributing to the formulation of crime. They performed the experiment by extracting patterns from heterogeneous areal aggregated datasets and observed the resulting patterns efficiently. Crime data mining based on extension classification is also described in [24]. This research developed a model that involved an extended data mining module for recognising relevant system inputs and analyzed the way these inputs interact within the system. This extended model was based on traditional methods to analyze the relation between data and intuition. Real crime dataset recorded by police in England and Wales within the time period of 1990–2011 has been taken and a theoretical model based on clustering and classification has been introduced in [25]. Features obtained from these police records have been given weights to improve the quality of the model and genetic algorithm is applied for detecting the outliers. The mode of operation that has been considered while performing the crime can also help in crime data analysis and these are called Modus Operandi features [26]. Two stage approach for crime data analysis has been done in [27]. Here, the first stage has been used for identifying named entities from crime reports and the second stage has considered the subtypes of the named entities that are usually neglected in crime analysis. But this research has shown that use of these subtypes is beneficial for a more insightful crime analysis.

Though PSO is a widely used optimization technique but it has been observed in many applications that the improved PSO algorithms provide better results than the traditional approaches. But this improvisation requires proper tuning of the parameters, known as dynamic adaptation. The work described in [28] incorporated dynamic adaptation of parameters by the utilization of an interval type-2 fuzzy inference system for searching optimal solution. This very recent research work has shown that type-2 fuzzy systems are way better for finding optimal results than type-1 fuzzy systems. This parameter adaptation with type-2 fuzzy inference system has been extended for analysing some of the widely used optimization algorithms [29]. This survey paper has shown that uncertainty can be tackled better using the type-2 fuzzy logic rather than the type-1 fuzzy logic. This research has also shown that the execution time of optimization techniques can increase if the parameters are not well adjusted. A new method for multi-objective optimization, called Fuzzy Adaptive Multi-objective Evolutionary algorithm (FAME) has been proposed in [30]. This research is all about developing a smart operator controller comprises a novel effective density estimator with polynomial complexity that helps in choosing the most promising variant of operator to optimize the search process. A comparative study based on the use of fuzzy logic for improving PSO variants for mathematical functions using co-evolution has been described in [31]. Here, a fuzzy system has been used to adjust the parameters of different PSO techniques and the problems with higher degree of complexity have been solved based on the co-evolution concept. A new swarm intelligence approach, termed as 'Artificial Bee Colony Algorithm', has been introduced in [32]. This newly developed algorithm imitates the foraging behaviour of honey bees. This is a variant of the artificial bee colony optimization technique with the addition of memory algorithm. It helps the artificial bees to memorize their previous successful experiences of foraging behaviour. Another improved Artificial Bee Colony algorithm has been built based on the gravity model [33]. This method is very useful for selecting a better neighbour of a current individual to improve the exploitation ability of the existing Artificial Bee Colony. Here, a novel solution for search equation has been proposed to chose the neighbour that plays an important role for searching process in the employed bee phase. Next, a random guiding search has been performed in the onlooker bee phase for balancing the foregoing exploitation. This newly developed algorithm, termed as ABCG, has also incorporated a multiple solution search equations, a scheme of perturbation frequency and a multiple scouts search strategy, in view of opposition-based learning. This ultimately helps in balancing between the exploitation and the exploration. Firefly algorithm has been considered for multi-stage transmission expansion planning with adequacy-security considerations in deregulated environments [34]. The transmission expansion planning (TEP) has been compared with other methods, such as genetic algorithm, particle swarm optimization, simulated annealing and differential evolution. Pattern recognition done by a modular neural network based on a multi-objective hierarchical genetic algorithm has been proposed in [35]. This study has shown that a modular neural network with granular approach and optimized parameters provides better results than normal neural networks. In this optimization approach, the fitness function has been used for minimizing the size of the dataset for training phase as well as error of the model using a multi-objective approach. Ontology technology based semantic relations have been found out by using an improvised version of PSO algorithm [36]. This improved PSO algorithm has been added with a new operator to update its crucial parameters. The formula by which the particle velocity used to get updated has been improved [37] for avoiding the premature convergence of PSO algorithm. The particle optimization performance has been adjusted through the weight factor. Particle swarm optimization techniques can also be used in industrial and electronic sectors. PSO and fuzzy logic controller has also been used for controlling speed of motors [38]. Here, the performance of the fuzzy logic controller based on PSO (FLC-PSO) has been compared with different controllers, such as PID controller, fuzzy logic controller and optimized fuzzy logic controller. It has been observed that the FLC-PSO based controller has acquired better speed of the dc motor. Also, the tuning of fuzzy scaling factors for active suspension control by PSO has been proposed in [39]. Here, the objective function of the algorithm has been defined to minimize sprung mass acceleration. This method has been evaluated by simulating the transient response to road perturbations.

1.2. Our contribution

Though enough research articles exist in the literature dealing with crime data analysis, but very few works are there which incorporate a multi-objective particle swarm optimization (MOPSO) technique [40] for classifying crime reports. The primary objective of the proposed work is to construct a rule based incremental classifier to categorize the dynamic crime reports based on their crime types for prediction of unlabelled crime reports both effectively and efficiently. The main contributions of this paper are summarized as follows:

- First of all, initial n-dimensional search space is created for developing the proposed discrete MOPSO based classification model. Also an initial population of particles is generated and considered as the initial candidate solutions of the problem.
- The method performs basic operations to move the candidate solutions gradually towards the global optimal position in the search space. It selects the non-dominated solutions based on the defined competent fitness functions in each generation and the optimal set of classification rules is obtained from the final population after the convergence of the process. We have defined the fitness functions based on accuracy and confidence of the rules which are defined in the proposed methodology section of the paper.
- Initially, Two different MOPSO based classifiers, namely dominance relationship based PSO (i.e., DRPSO) and non-dominated pareto front based PSO (i.e., NPPSO) are developed. Both are the rule based classifiers which provide optimal set of rules useful for classification of unlabelled crime reports.
- When new crime reports are generated, the MOPSO classifier is retrained using both the new dataset and existing optimal rule set obtained from the old dataset. Thus the classifier is incrementally modified and provides updated rule set. The incremental process is applied on both DRPSO and NPPSO classifiers to upgrade themselves to incremental DRPSO (i.e., INRDRPSO), and incremental NPPSO (i.e., INRNPPSO) classifiers, respectively. Based on the performance measures, it is observed that INRNPPSO outperforms the INRDRPSO and other state of the art incremental classifiers as per as the crime report prediction is concerned.

Thus, the proposed algorithm efficiently handles incremental data without reusing the old data based on which the classifier is already trained; rather the classification model is retrained using existing rule set and new dataset, which reduces the running time complexity of the algorithm. The objective functions, defined based on the existing rule set and the new dataset, are competent and independent to each other and are thus providing non-dominated solutions. So, the main objective of this paper is to develop an efficient and effective MOPSO based incremental classifier for crime report prediction. Fig. 1 shows the basic structure of the proposed MOPSO based incremental classifier.

The existing crime report dataset available in time t is an unstructured set R_1 of reports which has been preprocessed and turned into a structured crime dataset DB_1. This dataset DB_1 has been applied on the proposed multi-objective PSO (MOPSO) based classification algorithm to generate a set of rules. These rules have been used to predict the class labels of unlabelled crime reports available after t-th time. Each rule in the existing rule set may be considered as an object and the existing rule set may be considered as the representative of the dataset DB_1. Let DB_2 be the representative set of DB_1. In case of incremental dataset, in each time interval, new crime reports have been generated and these generated reports may be labelled or unlabelled. As the labelled crime reports are available, the existing rules must be updated for better prediction of the unlabelled crime reports. So the labelled reports available in time interval $(t, t + \triangle t)$ have been collected. Similarly, this set R_2 of reports has been preprocessed and dataset DB_3 has been generated. Now, $DB_4 (= DB_2 \cup DB_3)$ has been considered as the structured dataset for the set of reports $R_1 \cup R_2$ available after time $(t + \triangle t)$. Next, MOPSO based classification model has been trained on dataset DB_4 and finally, after convergence of the PSO process, the updated optimal set of rules has been obtained. Similarly, these rules are used to predict the class labels of unlabelled crime reports available after time $(t + \triangle t)$. This process has been continued whenever new data enters into the system after a certain interval of time. Thus, without regenerating the classification model, the existing model is retrained with the help of new data. This helps in developing an efficient incremental classifier for predicting the crime reports accurately in a dynamic environment where data are grown incrementally.

1.3. Summary

The rest of the paper is organised as follows: Section 2 describes the preliminary concepts of Particle Swarm Optimization. The incremental classifier generation for crime type prediction is discussed in Section 3. The experimental results and

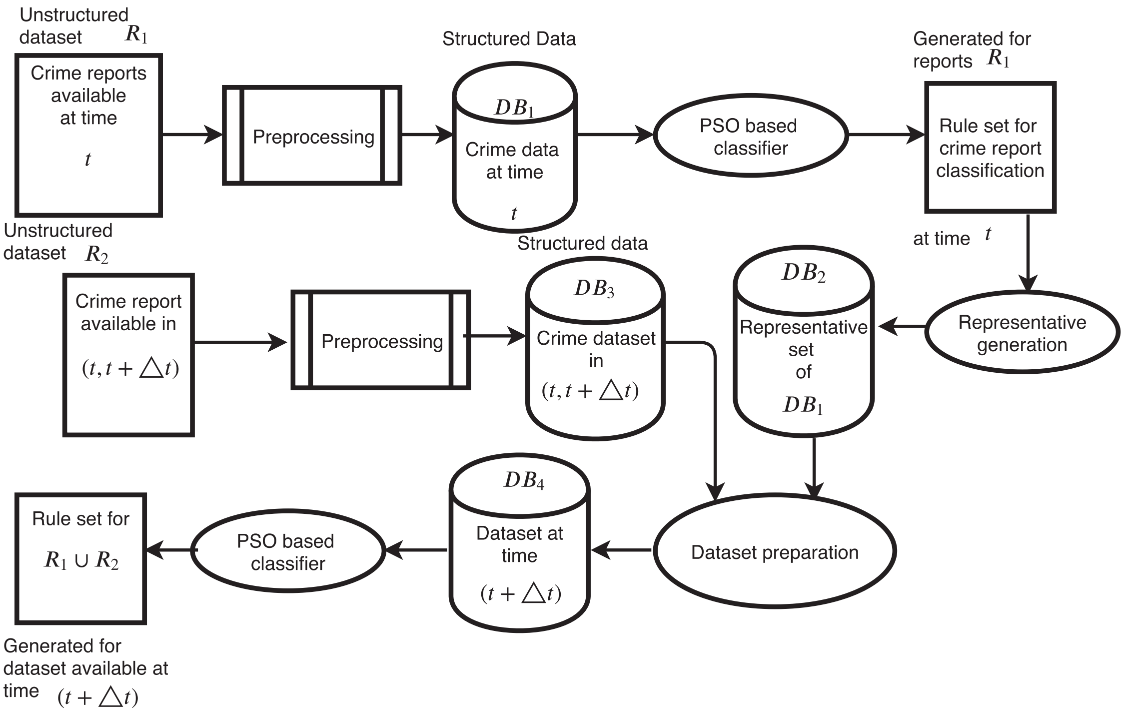


Fig. 1. Overview of Multi-objective PSO based Incremental classifier design.

efficiency of the proposed methodology is demonstrated in Section 4 and finally, the conclusion and future scope of the work are stated in Section 5.

2. Preliminary concepts of Particle Swarm Optimization

This section describes the fundamentals of the Particle Swarm Optimization (PSO) technique used in the proposed methodology. The PSO is a well known population based stochastic optimization technique, almost similar to other evolutionary algorithms, but unlike genetic algorithms, it has no crossover or mutation operators. The details related to PSO are described in the following subsections.

2.1. Basic working principle of PSO

PSO starts with a random initial population called ‘swarm’ and the candidate solutions in the population are encoded as particles in the search space. The swarm moves in the search space to find out the optimal solution by updating its generations. While updating the generations, the candidate solutions keep changing their positions depending on their own experience as well as on their neighbouring particles. Let a particle in a potential solution to the problem be represented by $P(i) = (p_{(i,1)}, p_{(i,2)}, \cdots, p_{(i,n)})$ in an n-dimensional search space. The particle $P(i)$ with a co-ordinate $p_{(i,d)}$ has a rate of change of position (i.e., velocity) $v_{(i,d)}$, where $d = 1, 2, \cdots, n$. While moving in the search space, every particle keeps a record of its best position visited, called as P_{best}. It also keeps track of the global best position attained which is called as G_{best}. Each particle is associated with a velocity vector which is modified after every generation and this modified velocity vector is used to generate new particle. Another topology, known as neighbourhood, defines the interaction of P_{best} and G_{best} with other candidate solutions in the swarm. This helps in the gradation of the velocity vector along with its position. In PSO, there are mainly three stages, namely evaluate, compare and evolve. The first stage is used to keep track the performance of all the candidate solutions. The comparison stage chooses the best particles and finally, the evolve phase generates new particles depending on the previously found best candidate solutions. A stopping criterion is provided in this optimization technique and the process is repeated until the stopping condition is matched. The aim of using this optimization approach is to find out the best particle providing the optimal solution to the problem. PSO is a kind of swarm intelligence technique that gives the overall social behaviour of swarms.

2.2. Multi-objective optimization

Unlike single objective PSO, multi-objective optimization problems handle several objective functions simultaneously to achieve the optimal solutions. Such problems consider two or more independent objective functions that have to be optimized simultaneously. These optimal solutions are known as non-dominant solutions that have been determined using the concept of Pareto optimality. Evolutionary algorithms are well suited to multi-objective optimization problems due to their ability to search for non-dominated Pareto optimal solutions. Though PSO shares many properties with evolutionary algorithms, but it has some special features, such as directed mutation, population representation and operators that take full advantage of PSO's efficiency in order to solve multi-objective problems in a comparatively simpler way. In multi-objective problems, basically there are two fundamental methods for designing PSO algorithms: (i) The first method considers each objective function separately to evaluate the particles and the position is determined similarly as determined in case of single-objective optimization problem [6]. The main challenge in this method is to integrate the objective functions in order to drive the particles towards Pareto optimal solutions and (ii) The second method considers all objective functions simultaneously to evaluate each particle and determines non-dominated best positions based on the concept of Pareto optimality. But determination of these positions is a difficult task as there may exist many non-dominated solutions in the neighbourhood of a particle out of which only one is generally selected to update its velocity.

2.3. Challenges in optimization problems

In this method, particle's best positions may be considered as the non-dominated solutions, but this choice may not always work well as (a) the desirable size of the Pareto front may exceed the swarm size and (b) two non-dominated solutions may be equally good, which arising the question of which one will be selected for determining the best position of a particle. The size problem can be solved by using an external archive that is used for storing the non-dominated solutions obtained during search. But choosing the most suitable archive member for determining the best position is difficult. Also, an external archive has a bounded size which creates difficulty to replace the existing solutions with new ones. Thus, selection of a member from the external archive and replacement of solutions in the archive are very challenging tasks for solving the optimization problems using multi-objective PSO techniques. At the same time, new solutions generated are taken into the archive in such a way that the diversity among the solutions is maintained. So it is very important to decide which one between an existing and a new solution is taken into the archive to retain the maximum possible diversity.

As mentioned earlier, in Pareto-based methods, the best position of a particle is replaced only by a new one that dominates it. If the existing best position and new one of a particle are non-dominated, then usually the new one is placed into the archive to promote swarm diversity. At this time, we must be very careful regarding the performance of the algorithm. Generally, the member is selected by assessing the quality of each archive member, based on some density estimators. The most commonly used density estimators are the Nearest Neighbour Density Estimator [41] and the Kernel Density Estimator [42] which provide estimations about the proximity and number of neighbours for a given point. Such measures help to select the archive members that promote the diversity among the solutions. In our proposed work, we have explored the multi-objective PSO using dominance relation and non-dominated pareto based schemes for crime type prediction.

3. Classifier generation for incremental crime reports

Classifier is a model generated by supervised learning technique where the model is trained using a given set of labelled data. Based on the training, the classifier can predict the class label of the unknown samples. In many real life applications including crime domain, many labelled and unlabelled instances are generated gradually over time. If these unlabelled instances are predicted using existing classifier, there will be a high possibility of model over-fitting problem as the generated model may be trained with the instances reside in some fixed regions of the searching space, i.e., the initial set of instances may not be well distributed over the search space. As the new labelled instances may be of class labels different from the existing class labels and/or of same class labels with different features, so the new instances may cover some new area of the searching space. Thus, diversity is the fundamental criteria that has the high probability to occur especially in case of the incremental dataset. The whole dataset generally provides comparatively better classifier, but the efforts already involved for generating the classifier based on initial dataset would be lost. To resolve this problem, the old model needs to be retrained using the new labelled dataset and as a result, the incremental classifier is obtained. So the incremental classifier is required when new set of data is added over time to the existing dataset, with the objective that the new model will be efficient as well as effective. PSO is an effective and widely used optimization technique that can be used considering suitable objective functions to construct a dynamic and incremental classifier for prediction of crime reports. In this proposed work, the objective is to extract the minimal number of classification rules that can predict the exact class labels of the unlabelled crime reports using bi-objective PSO and modify the rule set efficiently while new labelled crime reports are available in the system. Thus, an automated incremental classifier has been developed using bi-objective PSO to predict the crime types of unlabelled crime reports. Fig. 1 beautifully illustrates the overview of the proposed methodology. Initially, the available labelled crime reports have been preprocessed for generating a structured crime dataset which has been applied on a bi-objective PSO to generate a minimal rule set. The objective functions are defined in such a way that their domain sets consider the already generated rules and the new reports. This bi-objective PSO runs repeatedly while the new group of labelled crime reports get generated. For the first run of PSO, the objective functions have been defined using the existing crime reports only, as no rule set got generated during this time. Subsequently, the rule set has been updated from time to time and the new crime reports have been analysed dynamically and that too with a shortened training time. The performance of the model has been evaluated and compared with several state-of-the-art classifiers to demonstrate its effectiveness. The following subsections show different steps through which the proposed methodology has developed the crime report prediction model in a dynamic environment.

3.1. Dataset preparation

The crime report datasets collected from various classified online newspapers contain many noisy and irrelevant information and so are passed through some of the necessary preprocessing steps, like stopword removal and lemmatization. Having an idea of the context of the reports, a list of stopwords have been prepared and further these stopwords have been removed from the whole dataset for each country. It is very important to get rid of these stopwords as they are mostly considered as noise and their presence create hindrance in the classification task. The dataset is also made to pass through lemmatization which removes the suffixes and prefixes and keeps the root words only. Then tokenization is done to split the sentences of the whole dataset into multiple tokens and subsequently Parts-Of-Speech (POS) tagging is performed for all the tokens. All these essential preprocessing steps are performed with the help of Natural Language Toolkit module available in Python [43]. The POS tagging identifies all the parts-of-speech present in the documents that helps in noun phrase chunking. The noun phrases are also known as named entities which can be paired with each other for crime data analysis. Now, as the proposed work is based on predicting the crime types, therefore, we have considered the PER-PER (person-person) domain of named entity pairs. The intervening context words of the entity pairs define the crime types. Next, all the intervening context words from each named entity pair are collected and together these are termed as a phrase. For example, if we consider a sentence like, "*Nita* has been brutally murdered by *Ramen*", here the italicised words *Nita* and *Ramen* are two named entities categorized as PER(person) and the entity pair domain is PER-PER domain. Now, the intermediate context words "has been brutally murdered by" together are considered as a single phrase. All such phrases of the collected crime reports are considered to give the structured form of the reports. To get the structured form, each report is represented by an n dimensional vector using the GloVe word embedding model [44]. Let, a report be represented by phrases $\{P_1, P_2, \cdots, P_w\}$. Then all the context words in a phrase have been vectorized using the GloVe word embedding model and their average value gives the vector representation of the phrase. Finally, the average score of the phrase vectors of a report have been considered as the vector representation of the report. This form is known as the structured form of the unstructured text report. Let the structured dataset corresponding to the set of crime reports be represented by a decision system, $DS = (R, C, D)$, where R is the set of reports, $C = \{C_1, C_2, \cdots, C_n\}$ is the set of n condition features by which the reports are characterized, and D is the set of decision features representing the crime types. The condition feature-values of each report represent an n dimensional vector. As the dataset generated contains the real values, and the proposed PSO based algorithm considers only the discrete values, so we have discretized the continuous dataset using the Python based Orange tool [45]. Finally, for the proposed work, the dataset is partitioned into two subsets, R_1, and R_2, such that $R = R_1 \cup R_2$, where R_1 is the old dataset that contains 70% of the reports, and R_2 is the new or incremental dataset that contains remaining 30% of the reports. The data subsets are generated randomly from the dataset R. Thus, the system DS is divided into two subsystems, $DS_1 = (R_1, C, D_1)$ and $DS_2 = (R_2, C, D_2)$ such that $D = D_1 \cup D_2$. Initial classifier has been generated based on DS_1 and incremental classifier has been generated based on DS_2 and rule set of initial classifier. Both the classifiers have been evaluated using 10-fold cross validation technique.

3.2. Initial classifier generation

Initially, the PSO algorithm deals with the subsystem $DS_1 = (R_1, C, D_1)$ to extract the minimal set of classification rules. Each extracted rule has two components, namely antecedent and consequent. The condition features (in feature set C) with their values are the part of the antecedent whereas, the decision feature (in set D) with their corresponding values or class labels (i.e., crime types) are the part of consequent of the rules. For an example, let a rule is represented as $(C_1 = v_1 \wedge C_2 = v_2) \rightarrow (D = l)$, where C_1, C_2 are condition features in C with values v_1, v_2, respectively and $(D = l)$ is a class with label (i.e., crime type) l. Therefore, a rule is constructed with some rule components, like $(C_1 = v_1), (C_2 = v_2)$. The steps followed in the proposed bi-objective PSO method for generating classifier are described in the following subsections.

3.2.1. Initial population generation

First of all, the PSO method is applied on dataset $DS_1 = (R_1, C, D_1)$ for generating the initial classifier. The PSO algorithm is invoked for each crime type separately in parallel processing environment using Keras library in Python. We have discussed the proposed method for the reports of crime type $D_1 = l_1$ as follows. Let R_{11} be the subset of reports in R_1 with crime type $D_1 = l_1$ which are selected with the help of relational algebra based projection (π) operation [46], as defined in (1). Thus, Eq. (1) selects all the reports (i.e., n-dimensional vectors) of crime type l_1 from the set R_1 of reports. So, R_{11} is a set of n-dimensional objects.

$$R_{11} = \pi_C(\sigma_{D_1=l_1}(R_1)) \quad (1)$$

Now, let us assume that in R_{11}, the feature C_i in C has c_i distinct values, $c_{i1}^1, c_{i1}^2, \cdots, c_{i1}^{c_i}$ and is defined as $C_i = \{(C_i = c_{i1}^1), (C_i = c_{i1}^2), \cdots, (C_i = c_{i1}^{c_i})\}; \forall i = 1, 2, \cdots, n$. Now all the distinct feature values, (i.e., $(c_1 + c_2 + \cdots + c_n)$), are indexed by natural numbers as follows:

$$\begin{aligned} index(C_i = c_{i1}^j) &= j;\ \text{for } 1 \leqslant j \leqslant c_1;\ \text{if } i = 1 \\ &= c_1 + j;\ \text{for } 1 \leqslant j \leqslant c_2;\ \text{if } i = 2 \\ &= c_1 + c_2 + j;\ \text{for } 1 \leqslant j \leqslant c_3;\ \text{if } i = 3 \\ &= c_1 + c_2 + \cdots + c_{n-1} + j;\ \text{for } 1 \leqslant j \leqslant c_n;\ \text{if } i = n \end{aligned}$$

We prepare an n-dimensional search space from where the proposed PSO algorithm looks for appropriate set of particles, i.e., the minimal set of rules capable to classify the crime reports with class label $D_1 = l_1$. The search space contains the set $S = \{P_1 \cup P_2 \cup \cdots \cup P_n\}$ of particles, where $P_1 = \bigcup_{i=1}^n C_i, P_2 = \bigcup_{i,j=1;i<j}^n C_i \times C_j, P_3 = \bigcup_{i,j,k=1;i<j<k}^n C_i \times C_j \times C_k, \cdots\cdots,$ $P_n = C_1 \times C_2 \times \cdots \times C_n$.

So, the search space S contains particles of 1-dimension, 2-dimension, $\cdots$, n-dimension, which are in $P_1, P_2, \cdots, P_n$, respectively. The particles which are not of n-dimension, are made n-dimensional by appending remaining dimensions with zeros. Thus all the particles in the search space S become n-dimensional. Before applying the discrete PSO, the particles are encoded by the index values of their components. For example, if a particle in the population is denoted as $\{(C_1 = c_{11}^5), (C_2 = c_{21}^7), (C_3 = c_{31}^3)\}$, then it is encoded as $\{5, c_1 + 7, c_1 + c_2 + 3\}$. The initial population is a set of predetermined number, (say, m) of candidate solutions, collected randomly from the search space S.

3.2.2. Velocity and position of a particle

As mentioned earlier, each candidate solution (which is nothing but a rule) is a particle that swarms through the search space to find the optimal values of the objective functions. Each particle in the population maintains its position, velocity and individual best position during its flying in the search space. In addition, the swarm maintains its global best position. So the PSO algorithm consists of three basic steps, like evaluate fitness value of each particle, update individual and global best, and update velocity and position of each particle. These three steps are repeated until some stopping condition is met. Let, $v_i(t)$ and $x_i(t)$ are the velocity and position of the i-th particle at time t, and $v_i(t+1)$ and $x_i(t+1)$ are the velocity and position of the i-th particle at time $(t+1)$, respectively. Then the velocity of the particle at $(t+1)$-th time is given by (2), where w is the inertia coefficient, $a_1, a_2 (0 \leqslant a_1, a_2 \leqslant 2)$ are acceleration coefficients, $r_1, r_2 (0 \leqslant r_1, r_2 \leqslant 1)$ are random values used for every velocity update, $x_i^{best}(t)$ is the individual best position of the particle, and $g(t)$ is the swarm's best position as of time t.

$$v_i(t+1) = w v_i(t) + a_1 r_1 [x_i^{best}(t) - x_i(t)] + a_2 r_2 [g(t) - x_i(t)] \quad (2)$$

The inertia coefficient w helps to keep the particle along the same direction towards which it was directed initially. The lower value of w speeds up the convergence process and its upper value encourages exploring the search space. Usually, the value is considered in between 0.8 and 1.2. The acceleration coefficient a_1 is known as cognitive coefficient that acts as the particle's memory. It helps the particle to return to its individual best regions of the search space. Its value is usually set very close to 2 which limits the size of the step the particle takes toward its individual best position. The other acceleration coefficient a_2 is known as social coefficient that causes the particle to move to the best regions the swarm has found so far. Its value is also set very close to 2, which limits the size of the step the particle takes toward the global best position. The position of the particle is updated in $(t+1)$-th time using Eq. (3), where $x_i(t+1)$ and $v_i(t+1)$ are the new position and velocity of the particle and $x_i(t)$ is its old position.

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3)$$

In the proposed methodology, i-th particle p_i in the population P is an n-dimensional vector that represents its initial position, say $x_i(0)$. Initially, both the particle's best position, $x_i^{best}(0)$ and swarm's best position, $g(0)$ are set by the initial position of the particle, and the initial velocity of the particle is set as $v_i(0) = \{1, 1, \cdots, 1\}$, i.e., an n-dimensional vector with all 1. After each iteration, velocity of the particle is being changed and at $(t+1)$-th iteration, velocity of the particle p_i is computed using Eq. (2) and subsequently, its new position is obtained using Eq. (3). From Eq. (2), it is clear that, the new position vector contains the real values. So we round off the values to get the nearest integer value, which helps us to find the new particle in the search space.

3.2.3. Objective function

The fitness value of each particle is measured by defining two objective functions, that is why, the method is called Bi-objective PSO. One fitness value is defined in terms of the rule accuracy. Rule accuracy is different than the classification accuracy, since it is not computed over the whole training set, but only over all reports whose feature values match with the antecedent of the rule. Rule accuracy is computed by dividing the number of times the rule is in a correct set by the number of times it is in a match set. Let, antecedent part of the rule r is matched with t instances of the training set represented by the subsystem DS_1 out of which only s instances are matched with consequent part of the rule. Then the Accuracy is computed using Eq. (4).

$$Accuracy(A) = \frac{s}{t} \tag{4}$$

Thus, rule accuracy can be thought of as a 'local accuracy' of the model. Now suppose the rule accuracy is very good, say almost 80% and the 20% error occurs due to many reports of class value other than the consequent of the rule match with the antecedent part of the rule. In that case, antecedent of the rule is similar to many different type of reports and so its discriminating power to distinguish the reports based on report type is not sufficiently high, though its Accuracy is high. That is why, we have considered another objective function, Confidence of a rule. Here, consequent part of rule r is $(D_1 = l_1)$. Let $p = |R_{11}|$ is the number of reports of the training set which are of crime type $(D_1 = l_1)$, and out of these p reports, q reports have the feature values similar to the values of the antecedent part of the rule. The confidence of the rule r is given by Eq. (5).

$$Confidence(C) = \frac{q}{p} \tag{5}$$

Thus, accuracy provides the strength of the rule based on training subset of different crime types and confidence gives the strength of the rule based on the training subset of same crime types. So they are not dependent on each other, rather they compete to each other, both of which need to be maximized for rule based classifier design.

3.2.4. Rule set generation

In the proposed work, bi-objective PSO is used for rule set generation in crime type prediction. One of the simplest bi-objective PSO methods is to handle two objective functions separately, which is equivalent to two single objective PSO. In case of a single objective PSO, if the fitness of the particle at new position is better than its current position, then the particle moves to the new position. So which particles would be in the population for the next iteration depend only on the particles current position, individual best position and the swarm's best position. These lead to the lack of sharing information among the particles in the population except that each particle shares the global best position. In the work, we have proposed two different schemes of bi-objective PSO for rule set generation. In first scheme, the problem of lack of sharing information has been tried to resolve considering dominance relationship among the parent and offspring particles. In second scheme, a non-dominated sorting PSO methodology has been devised. Both the schemes perform an initialization method, named as INIT(S, P), to initialize the position, best position and initial velocity of each particle p_i in P, which are denoted by $x_i(0), x_i^{best}(0)$ and $v_i(0)$, respectively. Also, the initial global best position $g(0)$ of the swarm is initialized by the initial position of the first particle in the initial population. So, the INIT(S, P) method mentioned in Algorithm 1 takes generated search space S as input, initializes the particles in P and returns P together with the initial global best position $g(0)$. By returning P, we mean that the method returns all of $x_i(0), x_i^{best}(0)$ and $v_i(0)$ of each particle p_i in P. In each iteration the position and velocity of the particles as well as the swarm's best position are modified and the initial set of rules is obtained from the final population after the convergence of the PSO.

Algorithm 1: INIT (*S; P*)

Input: S=the search space
Output: P=the initial population
begin
  P = Randomly generated initial population from search space S;
  Let $P = \{p_1, p_2, \cdots, p_m\}$ be the set of m particles;
  $t = 0$; /*$t = 0$ indicates the initial stage of the particles */
  for $i = 1$ *to* m **do**
    Let $x_i(t) = (x_{i1}, x_{i2}, \cdots, x_{in})$;/* $x_i(0)$ = initial position of p_i */
    $x_i^{best}(t) = x_i(t)$; /* initially, initial position is its best position */
    $v_i(t) = (1, 1, \cdots, 1)$; /* initial velocity of p_i */
  $g(t) = x_1(0)$; /* $g(t)$ is the Swarm's best position*/
  Return (P, t, g(t)); /* P contains both initial position, best position and velocity arrays*/

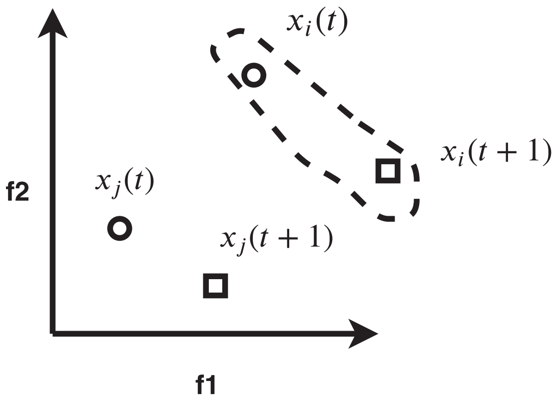


Fig. 2. Dominance relationships among four particles.

A. Dominance Relationship based PSO: First, we compute the fitness of the particles based on considered objective functions. The proposed dominance relationship based bi-objective PSO (DRPSO) helps to share the information among each other during generation of the new population of particles. But the question is which two particles are involved in the sharing process. That may be any pair of particles, which is time consuming and so only a k-nearest neighbours of each particle are selected for sharing their information with that particle. Let for each particle p_i, the k-nearest neighbour particles be $q_{i1}, q_{i2}, \cdots, q_{ik}; \forall i = 1, 2, \cdots, m$. So, p_i is used to share its information with each particle q_{ij} $\forall j = 1, 2, \cdots, k$. In this case, it may happen that both particle (i.e., current position) and its offspring (i.e., new position) may or may not be there in the new population. To explain the idea, let us consider an example with particle p_i and one of its neighbour particle p_j, where $x_i(t)$ and $x_j(t)$ are their respective current positions and $x_i(t+1)$ and $x_j(t+1)$ are their respective new positions. For all these four instances, the fitness functions f_1 (say, accuracy) and f_2 (say, confidence) are computed and the instances are plotted considering the fitness functions as the 2-D coordinate axes, as shown in Fig. 2.

Fig. 2 shows that $x_i(t)$ and $x_i(t+1)$ are non-dominated to each other, and $x_j(t)$ and $x_j(t+1)$ are also non-dominated to each other. But $x_j(t+1)$ is dominated by both $x_i(t)$ and $x_i(t+1)$, and $x_j(t)$ is dominated by $x_i(t)$. So, if all four instances are compared, then $x_i(t)$ and $x_i(t+1)$ are the two better instances to be retained for the next generation. Thus, one parent instance and another offspring instance are selected for the new population. Here, though information among four instances are shared but the useful non-domination relationships among them are not fully captured. In our proposed DRPSO method, considering p_i as one parent, individually each q_{ij} is used as another parent, $\forall j = 1, 2, \cdots, k$ and two better instances are retained as the new particles. So, after one iteration, from the set of $(k+1)$ particles $\{p_i, q_{i1}, q_{i2}, \cdots, q_{ik}\}, k$ particles, say, $\{p_1^{i,1}, p_2^{i,1}, \cdots, p_k^{i,1}\}$ are obtained. We repeat the same process considering $p_1^{i,1}$ as one parent and each of the remaining generated particles as another parent. This provides $(k-1)$ particles $\{p_1^{i,2}, p_2^{i,2}, \cdots, p_{k-1}^{i,2}\}$. Finally, after k-th iteration, only one particle $p_1^{i,k}$ with two instances are obtained. These two instances are basically two particles which are the instances of the initial particle set $\{p_i, q_{i1}, q_{i2}, \cdots, q_{ik}\}$. If one instance dominates the other one then the new position $x_i(t+1)$ of particle p_i at time $(t+1)$ is set by the position of the non-dominated instance; otherwise it is set arbitrarily by the position of any one of the two instances. Thus, the new position of the particle is not only determined by its own information and the global best information, rather it is determined by sharing the information of its k-nearest neighbour particles. If the new position $x_i(t+1)$ of the particle p_i is more fit than its previous position $x_i(t)$, then its best position, $x_i^{best}(t+1)$ and swarm's best position, $g(t+1)$ are set by the new position of the particle. Thus, after each iteration, the particles in the population either remains in the same position or moves to some other position based on the objective functions for searching the optimal solution. So after a certain number of iterations, the particles in the population give their best individual positions. Each particle in this population is a rule for class $(D_1 = l_1)$ and the population may contain many common particles, so we select only distinct particles, i.e., the final population provides at most m rules for a given crime type. The final population providing the initial classification rule set for the crime type $(D_1 = l_1)$ is described in Algorithm 2. Similarly, the rules are generated for other class labels or crime types in parallel processing environment.

Algorithm 2: DRPSO (R_{11}, S, P)

Input: R_{11}=Reports with class '$D_1 = l_1$', S=the search space and P=the initial population
Output: P=Final Population that represents classification rules for crime type '$D_1 = l_1$'
begin
  INIT(S, P, t); /*initialize the particles*/
  do
    for $i = 1$ *to* m **do**
      Compute new position $x_i(t+1)$ of particle p_i using eq. (2) and eq. (3);
      Compute fitness functions $f_1^i(t)$ and $f_2^i(t)$ for $x_i(t)$ using eq. (4) and (5), respectively;
      Similarly compute $f_1^i(t+1)$ and $f_2^i(t+1)$ for $x_i(t+1)$;
    $g(t+1) = g(t)$;
    for $i = 1$ *to* m **do**
      Let Neigh $(p_i) = \{q_{i1}, q_{i2}, \cdots, q_{ik}\}$;
      do
        $P_i^{new} = \phi$; /*New particles generated from p_i for the new population*/ $c = 0$;
        for *each* $p_j \in$ *Neigh* (p_i) **do**
          $c = c + 1$;
          Let $P_{ij} = \{x_i(t), x_i(t+1), x_j(t), x_j(t+1)\}$
          /*$t-th$ and $(t+1)-th$ instances of p_i and p_j*/
          Compute dominance relationship among particles in P_{ij} based on their fitness values;
          $P_c^{i,1}$ = Set of two better instances in P_{ij} as identified by Figure (2);
          $P_i^{new} = P_i^{new} \cup P_c^{i,1}$;
        Let $p_i = P_i^{new}[1]$; /*the first particle in P_i^{new}*/
        Neigh $(p_i) = P_i^{new} - \{p_i\}$; /*remaining particles in P_i^{new}*/
      while *Neigh* (p_i) *is not empty;*
      Let p_i contains two new instances x_i^1 and x_i^2;
      if x_i^1 *dominates* x_i^2 *w.r.t both the fitness functions* **then**
        $x_i(t+1) = x_i^1$;
      else
        if x_i^2 *dominates* x_i^1 *w.r.t both the fitness functions* **then**
          $x_i(t+1) = x_i^2$;
        else
          $x_i(t+1) = x_i^1$ or x_i^2, selected randomly;
      $p_i = x_i(t+1)$; /*new position of particle p_i based on dominance relation*/
      if $x_i(t+1)$ *dominates* $x_i(t)$ **then**
        $x_i^{best}(t+1) = x_i(t+1)$
      else
        $x_i^{best}(t+1) = x_i(t)$;
      if $x_i^{best}(t+1)$ *dominates* $g(t+1)$ **then**
        $g(t+1) = x_i^{best}(t+1)$;
    $t = t + 1$;
  while $t <$ *a predefined threshold;*
  Return P; /*each particle in the population P is a rule*/

B. Non-Dominated Pareto Front based PSO: In case of dominance relationship based PSO, non-dominance relationship among the particles is not fully captured. As a result, obtained solutions may not be well-distributed along the non-dominated pareto front. This problem leads the model over-fitting which is resolved by the proposed non-dominated pareto front based Particle Swarm Optimization technique (NPPSO). In NPPSO, we have adopted the non-dominated sorting based genetic algorithm (NSGA II), where the entire population is divided into non-dominated pareto fronts. Instead of considering only k-nearest neighbours of the particles, all m offspring and local best of all m particles in the population are combined to form a temporary population of $2m$ particles. Next, non-dominance relationships among the particles are measured by generating the non-dominated pareto fronts, as shown in Fig. 3. Initially, the entire temporary population is divided into two sets, the non-dominated set and the dominated set. All the particles in the non-dominated set are placed in Front 1 and called the particles of the best non-dominated set, as they are not dominated by any of the $2m$ particles in the population.

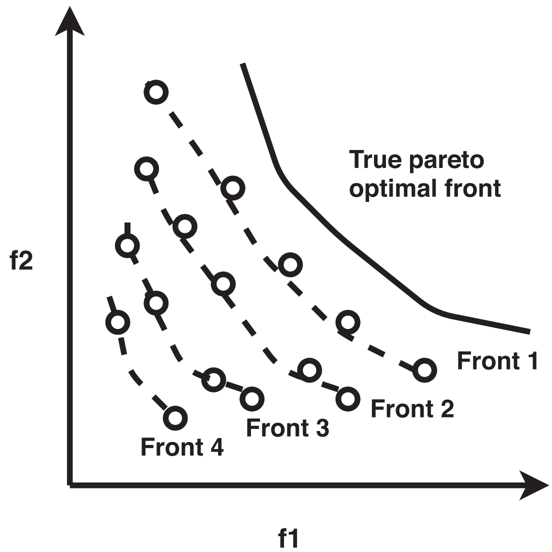


Fig. 3. Non-dominated fronts of the particles in a population.

To generate the next front (i.e., Front 2), we consider the particles of the temporary population which are not in Front 1, similarly they are divided into two sets, the non-dominated set and the dominated set. The particles of non-dominated set form the second pareto front, Front 2. The same process is continued until all the particles of the temporary population are exhausted. After generating the non-dominated pareto fronts of all considered $2m$ particles, next step is to select the next generation population of m particles from these $2m$ particles. Since, Front 1 is closest to the true pareto optimal front and particles in this front are non-dominated to each other and dominate all the particles in the temporary population, so generally all the particles in Front 1 are selected in the new population. Next, the particles from Front 2 are selected for the new population, and repeat the same process for other fronts in order until all m particles of the new population are selected. During this selection process, following observations are noticed and accordingly the selection process is modified.

(i) It may happen that out of all $2m$ particles, Front 1 contains more than m particles. In this case, which m particles will be selected?
(ii) Is it always good to select all particles from a front (say, Front 1) even if it contains less than or equal to m particles?
(iii) If we need to select the particles from other fronts, then from how many fronts and in which ratio the particles will be selected?

Though the selection of particles from the fronts closer to true pareto front drives the particles of the population towards the true front, but it lacks to select the particles from other fronts, which causes the problem of lateral diversity. Resolving this problem, we can overcome the model over-fitting problem too of the generated classifier. In the proposed NPPSO method, model over-fitting problem is solved using the Niching method [47]. This method is popularly used in genetic algorithm to maintain the population diversity. There are many Niching methods in the literature, we have used a niche count method for new population generation. Niche count n_i of a particle p_i is the number of particles existing within the σ_{share} distance from it. It is the Euclidean distance generally specified by the user. If two candidate solutions are in a same pareto front, then the solution with smaller niche count is more preferable than the other one to become a member of the next generation. This helps to maintain more diversity during selection of solutions from the non-dominated fronts. Obviously, the performance of the model is highly dependent on this user specified parameter. Fonseca et al. [47] proposed a method for dynamic updation of σ_{share}, which we have used in our work to set its value dynamically. The value of σ_{share} is set dynamically using Eq. (6), where f_1^{ub}, f_1^{lb} are the upper and lower bound of objective function f_1, whereas f_2^{ub}, f_2^{lb} are the upper and lower bound of objective function f_2 for the whole population and m is the population size. From Eq. (6), it is observed that, as population size increases, the value of σ_{share} decreases, which implies that we select more particles with less niche count from the non-dominated solutions, i.e., we are selecting the particles from the less crowded regions. The final population providing the initial classification rule set for the crime type $(D_1 = l_1)$ is described in Algorithm 3.

$$\sigma_{share} = \frac{f_2^{ub} - f_2^{lb} + f_1^{ub} - f_1^{lb}}{m-1} \tag{6}$$

Algorithm 3: NPPSO (R_{11}, S, P)

```
Input: R_11= Reports with class 'D_1 = l_1', S=the search space and P=the initial population
Output: P=Final Population that represents classification rules for crime type 'D_1 = l_1'
begin
    INIT(S, P, t); /*initialize the particles*/
    do
        P' = φ /*offspring set generated from P*/
        for i = 1 to m do
            Compute offspring p'_i, i.e., x_i(t+1) of p_i, i.e., x_i(t) in P using eq. (2) and (3)
            P' = P' ∪ {p'_i}
        P_T = P ∪ P' /*temporary population of 2m particles*/
        Let P_T = {p_1, p_2, ..., p_m, p_{m+1}, ..., p_{2m}}; /*here, p_{m+i} = p'_i ∀i = 1, 2, ..., m*/
        g(t+1) = g(t)
        for i = 1 to 2m do
            Compute f_1^i(t) and f_2^i(t) of p_i in P_T using eq. (4) and (5), respectively;
        c = 0; /*Number of non-dominated pareto fronts*/
        do
            c = c + 1;
            Let F_c = set of non-dominated particles in P_T w.r.t f_1^i(t) and f_2^i(t);
            Front - c contains all particles in F_c;
            P_T = P_T - F_c;
        while P_T is not empty;
        Compute σ_share using eq. (6), considering all particles in P ∪ P';
        P = φ; /*new population for next generation*/
        for i = 1 to c /*select particles from every front */ do
            for j = 1 to |F_i| /*compute niche count of p_j */ do
                n_ij = No. of particles in front-i within σ_share distance of P_j;
            Arrange all particles of front - i in ascending order of their niche count;
            if i == 1 then
                k = ⌈0.8m⌉; /*k = no. of particles selected from each front*/
            else
                k = 0.2m|F_i| / Σ_{l=2}^{c} |F_l|;
            if |F_i| ≤ k then
                P = P ∪ F_i;
            else
                G_i = first k particles from front -i;
                P = P ∪ G_i
        for each p_i in P do
            if p_i,i.e., x_i(t+1) dominates x_i(t) then
                x_i^best(t+1) = x_i(t+1); else
                    x_i^best(t+1) = x_i^best(t);
            if x_i^best(t+1) dominates g(t+1) then
                g(t+1) = x_i^best(t+1);
        t = t + 1
    while t < a predefined threshold
    Return P /*each particle in the population P is a rule*/
```

3.3. Generation of incremental classifier

Once the initial classifier is generated, class labels of all unlabelled crime reports are predicted using that classifier. But labelled crime reports are generated frequently and so after certain interval of time, the initial classifier needs to be retrained based on the existing as well as new labelled reports. We can train the classifier from the scratch combining the old and new labelled reports together, but it reduces the efficiency of the model as the previous training of the model is not under consideration. So the objective of this section is to retrain the model considering the new reports and the information extracted by the model from the old reports. The information already extracted from the old reports are stored in terms of classification rules. Each rule of the model is treated as an n-dimensional object generated from the old reports. So all the m rules (actually, number of rules is at most m) are considered as m objects representing the old set of reports. Now the model is retrained by the same PSO algorithm with dataset containing these m objects and new set of reports. For each class, the search space is already generated using subsection 3.2.1 with the help of data object of that class. For new data objects of same class, similar process is applied to create new subspace and added it with the existing search space. Thus, the search space becomes larger for each class in case of an incremental dataset. But the population size remains same (i.e., m), considering arbitrarily m particles from the incremental search space. Also, new velocity and position of the particles are computing using Eq. (2) and (3), as computed earlier. The main concern in incremental classifier design is the computation of objective functions. Here also accuracy and confidence are considered as the objective function. In case of initial classifier design, whole training set is used to compute the values of the objective functions. But in incremental classifier design, previous training dataset is replaced by the representatives obtained from the classification rules. So, if we compute objective functions by the similar approach then the objective functions will be biased to the incremental portion of the dataset, as we are giving equal weight to new objects and rules representing objects. As one rule is representing many initial training objects so each rule should get more importance than any new object. Keeping it in mind, we have defined both the objective functions namely, incremental accuracy (IA) and incremental confidence (IC). We are discussing only the process of computing accuracy of a particle in case of an incremental dataset in the following section, as similar method is applied for computation of incremental confidence.

3.3.1. Incremental accuracy

Let the incremental training set be denoted by $IT = (R_t, C, D_t)$ so that $R_t = R_i \cup R_2$, where R_i is the set of rules generated by initial classifier, R_2 is the new set of reports and D_t is the set of class labels in R_t. $D_t = D_1 \cup D_2$ contains all the class labels (i.e., D_1) of the initial training report set R_1 represented by R_i and the new class labels (i.e., D_2) appear in new set of reports R_2. If some class labels in D_t are new (i.e., not in initial training set), then the accuracy calculation of the particles for these classes is exactly similar to the previous technique used during initial classifier design. Here, the accuracy is computed based only on the new reports of corresponding class using Eq. (4). But many new reports are available whose classes are same to that of initial set of reports. In this subsection, we have mainly described the accuracy calculation of the particles (i.e., rules) of those classes.

Let us consider a rule $r : \{< r_1, r_2, \cdots, r_i >, < c >\}$, where $r_1, r_2, \cdots, r_i$ are the all i-rule components known as antecedent and c is the consequent of the rule r. Here, c is not only in D_2 but also in the class label set D_1 of initial set of reports. Let, antecedent part of r is matched with set T of instances of the training set R_t out of which only set S of instances are matched with consequent part of the rule r. so $S \subseteq T$. Now partition the set T of instances into four subsets T_1, T_2, T_3 and T_4 such that T_1, T_2 contain the set of instances representing rules of initial classifier with and without class c, respectively and T_3, T_4 contain the set of instances of the new report set with and without class c, respectively. Similarly, S is partitioned into S_1 and S_2 such that S_1 is the set of instances representing rules of initial classifier and S_2 is the set of instances of the new report set. The partitions of set of reports T, S ($S \subseteq T$) obtained using rule r are shown in Fig. 4.

Now the incremental accuracy (IA) of the rule r is defined by Eq. (7) where, $A_1(r)$ and $A_2(r)$ are the accuracy of the rule r computed from new report set R_2 and old report set R_i, respectively and δ is the weight factor used to set the contribution of $A_1(r)$ and $A_2(r)$ towards $IA(r)$.

$$IA(r) = \delta A_1(r) + (1 - \delta) A_2(r) \tag{7}$$

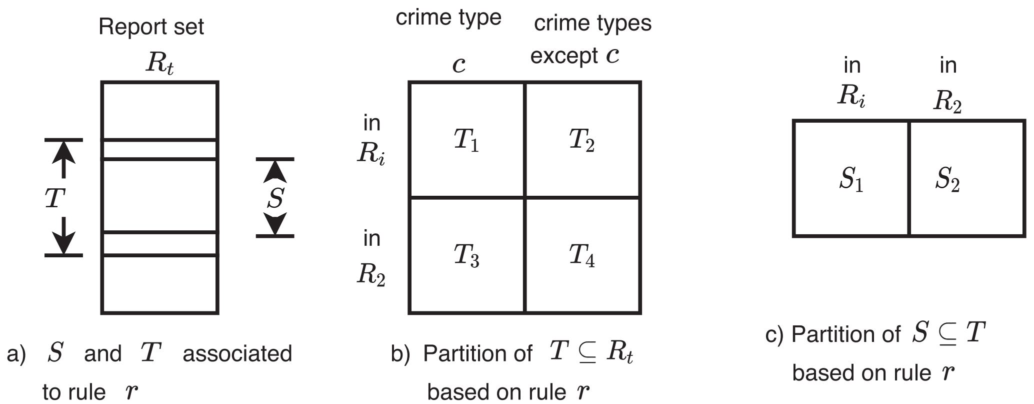


Fig. 4. Partition of T and S based on antecedent and consequent of rule r.

$A_1(r)$ and $A_2(r)$ are computed as follows:(i) From Fig. 4, it is observed that for new report set R_2, the consequent part of rule r is satisfied by S_2 and antecedent part of rule r is satisfied by T_3 and T_4. So, $A_1(r)$ is defined by the Eq. 8 which provides the accuracy of rule r based on new report set R_2.

$$A_1(r) = \frac{|S_2|}{|T_3| + |T_4|} \tag{8}$$

(ii) For computing $A_2(r)$, it is noteworthy to mention that each report is denoted by n-dimensional vector. So for all new reports in R_t, there are features values for all n features in C. But for all initial reports only rules are considered as the representative objects.So these objects may not necessarily contain all n feature values. In reality, very few features (i.e., number of rule components) of a rule contain the valid values, all other feature values are filled by a special symbol, to distinguish them easily from feature values. Obviously T_1 consists of only one instance corresponding to rule r, because if other instances exist in T_1 then they must be redundant as they are by default implied by that instance. To be more specific, without loss of generality, assume that $r_1, r_2, \cdots, r_i$ of rule r are the first i feature values of the instances of T_1. Since all the instances are of same class c and $r_1, r_2, \cdots, r_i$ are the features values of all the instances of T_1, so only the instance $< r_1, r_2, \cdots, r_i >$ representing by rule r must be there in T_1 and all other rules in T_1 are redundant which are not in initial rule set. Here, $s_1, s_2, \cdots, s_{n-i}$ are special symbol, say X represents the invalid values of the corresponding features. Thus, the accuracy of the rule r for initial training set is obtained by only these i features.Let, initial accuracy of the rule r be α, where $0 \leqslant \alpha \leqslant 1$. This accuracy is contributed by i features out of n features. So, we set a term $(\alpha)^{i/n}$ for set T_1 to contribute towards incremental accuracy. But in set T_2, there may be many initial rules of class type other than c. Let $T_2 = \{I_1, I_2, \cdots, I_P\}$, where each I_j $(for\ j = 1, 2, \cdots, P)$ in T_2 is the initial rule of class other than c. Let, β_j is the accuracy of I_j obtained during initial classifier generation in Section 3.1. Also let, β_j contains γ_j number of rule components which are different from all i-rule components of rule (i.e., particle) r. So these γ_j components (each one for a feature in C) are responsible for disagreeing to the class type c. Thus, this instance in T_2 reduces the accuracy of the particle r and this reducing factor is considered as $(\beta_j)^{\gamma_j/n}$. Thus, the total reducing factor is$\sum_{j=1}^{P}(\beta_j)^{\gamma_j/n}$. So, $A_2(r)$ is defined by Eq. (9).

$$A_2(r) = (\alpha)^{i/n} - \sum_{j=1}^{P}(\beta_j)^{\gamma_j/n} \tag{9}$$

Similarly, the incremental accuracy is measured for all particles in the population. The value of δ in Eq. (7) is considered within the range $0 < \delta < 0.5$, as initial dataset is huge and one initial rule represents many instances of the dataset, i.e., more importance is given to initial rules than new reports (i.e., new objects) during computation of incremental accuracy of the particles. Similar technique is used during computation of incremental confidence and so we have omitted the redundant discussion.

Once both the objective functions are defined, all other techniques of bi-objective PSO applied are similar as discussed in Section 3.1. For the incremental dataset, similar to the initial dataset, two different multi objective PSO algorithms, namely (i) INRDRPSO, an incremental dominance relationship based PSO and (ii) INRNPPSO, an incremental non dominated pareto front based PSO, are applied. After the convergence of the algorithm(s), the modified set of rules is obtained from the final population of particles. Performances of both INRDRPSO and INRNPPSO based classifiers have been evaluated to measure their effectiveness. Similarly, when a new set of labelled crime reports becomes available, the updated rule set has been considered as the initial rule set and the same incremental classifier generation algorithm discussed in this subsection is applied. Thus, the classifier has been upgraded after every interval of time.

4. Experimental results

The proposed work is focused on classification of the crime reports according to the crime types. For this purpose, we have considered the crime reports of the United States of America (USA), United Arab Emirates (UAE) and India following the same approach as mentioned in [1]. Online classified newspapers, like "The Times of India", "The Hindu" and "The Indian Express" report crime events of India. Other websites, namely "The Guardian", "Courts - The National", "Khaleej Times" and "news24" have a huge collection of crime reports for USA and UAE. A customised Python program has been used as a web crawler that extracts useful reports from the newspaper websites. For extracting crime reports from "The Guardian" website, an API-key has been generated initially by registering on the website as a developer. The API call [1] returns a json file containing near about 100 reports from 2017–07-25 to 2017–08-24 tagged with the term "murder". Python `requests` package is then used to generate the GET requests and the obtained json data is properly formatted using regular expressions. The crawled crime reports are from January 2007 to July 2017. Here, the collected reports are tagged with their crime types, which are the class labels of the reports. There exist different domains of named entity pairs in the crime reports but in this work, only the PER-PER domain of entity pairs has been considered for incremental classifier generation as most of the crime types are identified from this domain. The details of the newspaper extracted data for the USA, UAE and India are given in Table 1, where

[1] "http://content.guardianapis.com/search?from-date=2017–07-25&to-date=2017-08-24&show-fields=all&page-size=100&q=murder&api-key=< API-key>"

Table 1
Details of the dataset for PER-PER domain of entity pairs.

Dataset	70% data			30% data		
	No. of entity pairs	No. of reports	No. of class	No. of entity pairs	No. of reports	No. of class
INDIA	3,387	40,570	12	1,585	18,381	12
USA	4,712	45,393	24	2,056	20,228	24
UAE	5,320	46,931	13	2,367	20,921	13

Table 2
Confusion Matrix.

Actual	Predicted	
	Positive	Negative
Positive	T_p	F_n
Negative	F_p	T_n

Table 3
Performance of initial classifiers based on 70% data described in Table 1.

Country	DRPSO				NPPSO			
		($w = 1.1$,	$a_1 = 1.9$,	$a_2 = 1.8$)		($w = 1.2$,	$a_1 = 1.8$,	$a_2 = 1.8$)
	A	P	R	F	A	P	R	F
INDIA	77	76	74	75	**81**	**83**	**83**	**83**
USA	74	75	76	75	**77**	**76**	**79**	**77**
UAE	72	73	75	74	**78**	**77**	**80**	**78**
PWN	73	75	76	75	**74**	**76**	**77**	**76**

70% data have been considered for initial classifier generation and remaining 30% of the data have been considered for incremental classifier generation. It is observed from Table 1 that, though number of entity pairs is the largest for crime reports of the UAE, but the variety of crime types i.e., number of class labels is the highest for crime reports of the USA.

Apart from the newspaper reports as described in Table 1, investigators and law enforcement agencies often depend on interviews with the victims, witnesses and very often with the criminals. These interviews yield narrative reports that are basically stored by the police as pertinent data for crime analysis. Here, we have also considered one of such police and witness narrative reports (PWN) for evaluation of the proposed classifier. The dataset chosen for evaluation is mentioned in [48]. These crime related information from police reports are available in several newspapers, blogs and forums like "SFGate Crime" (https://www.sfgate.com/crime/), "TheLAW.com" (https://www.thelaw.com/forums/), "Baltimore Crime" (http://baltimorecrime.blogspot.com/), "Secret Witness" (http://secretwitness.com/), "True Crime Blog" (https://laurajames.typepad.com/) etc. The witness narrative reports have been extracted from law related forums like "ChatLawInfo" (http://chat.lawinfo.com/), "FreeAdvice" (http://forum.freeadvice.com/) and "ExpertLaw" (http://www.expertlaw.com/forums/index.php). Once these data have been extracted, we have focused on collecting the PER-PER domain of entities. Both the initial and incremental classifiers are validated by 10-fold cross validation technique based on all four considered datasets.

Here, the evaluation of the proposed classifier has been performed in several stages. Initially, Section 4.1 describes the performance of initial classifiers and incremental classifiers. Section 4.2 compares the incremental classifier with some traditional as well as recently developed classifiers and Section 4.3 provides a ranking of the classifiers. The proposed work has been implemented using Python 3.6 with its inbuilt multiprocessing libraries, like Keras and several modules in a computer running Ubuntu GNU/Linux version 16.04 LTS on an Intel(R) Core i3-5005U CPU @ 2.00 GHz processor.

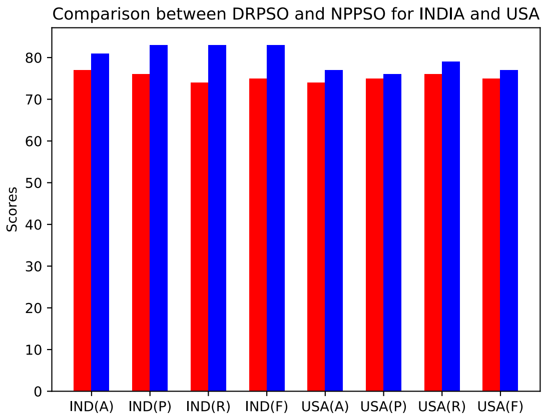
4.1. Evaluation

The classifier has been trained by 70% data and the initial classification rules have been generated. To evaluate the initial classifier based on the three parameters namely, w, a_1 and a_2, we have used 10-fold cross validation and the evaluation has been done based on four evaluation metrics, such as Precision (P), Recall (R), F-measure (F) and Accuracy (A), computed using Eq. (10)–(13), where T_p, T_n, F_P and F_n are the terms True Positive, True Negative, False Positive and False Negative, respectively. These values are obtained by computing the confusion matrix as shown in Table 2.

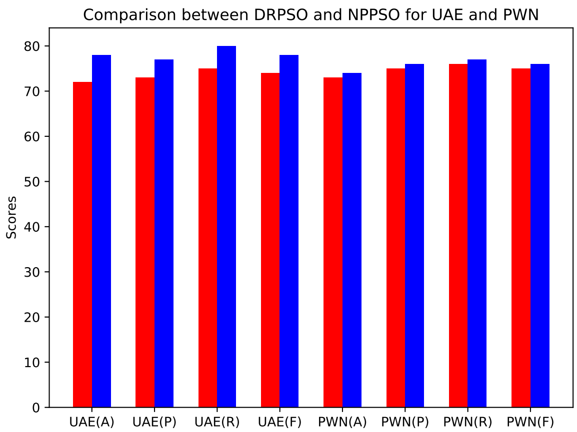
The parameters w, a_1 and a_2 are varied during execution of the program within the range [0.8, 1.2], [0.8, 1.2], [1.6, 2.0], and [1.6, 2.0], respectively and the best values obtained for DRPSO are $w = 1.1, a_1 = 1.9, a_2 = 1.8$ and for NPPSO are $w = 1.2, a_1 = a_2 = 1.8$. Table 3 shows the results of both DRPSO and NPPSO, applied on the initial 70% data as described in Table 1. It is observed from the results that comparatively better performance has been achieved by NPPSO for all the data-

Table 4
Performance of incremental classifiers based on respective initial classifiers and 30% data described in Table 1.

Country	INRDRPSO				INRNPPSO			
		($w = 1.2$,	$a_1 = 1.9$,	$a_2 = 1.9$)		($w = 1.2$,	$a_1 = 1.9$,	$a_2 = 2.0$)
	A	P	R	F	A	P	R	F
INDIA	75	74	76	75	**80**	**82**	**80**	**81**
USA	73	72	71	71	**75**	**74**	**76**	**75**
UAE	71	71	70	70	**74**	**73**	**72**	**72**
PWN	76	77	77	77	**78**	**79**	**81**	**80**



(a) For dataset of India and USA



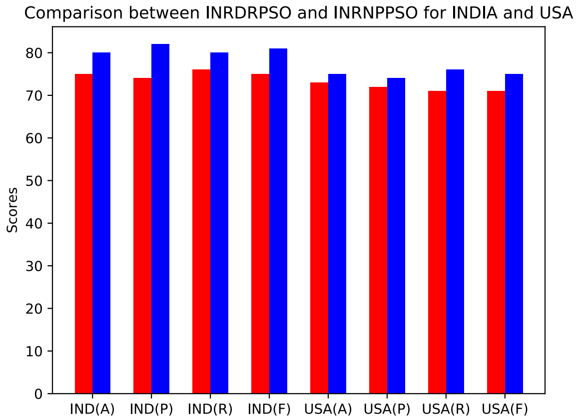
(b) For dataset of UAE and PWN

Fig. 5. Accuracy, Precision, Recall and F-measure comparison between DRPSO and NPPSO.

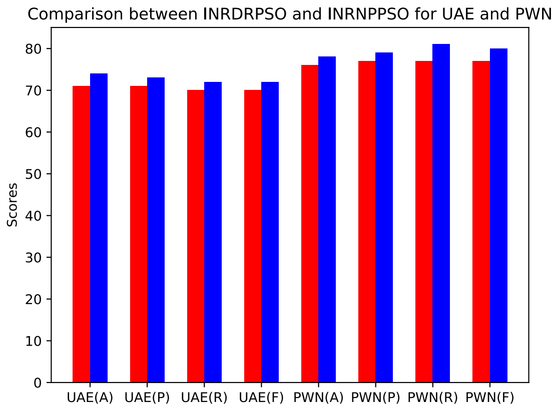
sets. The highest classification accuracy of 81% has been achieved by NPPSO method for crime data of India, whereas 78% and 77% accuracy are achieved on crime dataset of UAE and USA, respectively. The highest F-measure of 83% has been obtained for crime reports of India by this method. Again, for PWN dataset, the highest accuracy and F-measure are of 74% and 76%, respectively.

$$\text{Precision(P)} = \frac{T_p}{T_p + F_p} \tag{10}$$

$$\text{Recall(R)} = \frac{T_p}{T_p + F_n} \tag{11}$$



(a) For dataset of India and USA



(b) For dataset of UAE and PWN

Fig. 6. Accuracy, Precision, Recall and F-measure comparison between INRDRPSO and INRNPPSO.

$$\mathrm{F-measure(F)} = \frac{2\mathrm{PR}}{P+R} \tag{12}$$

$$\mathrm{Accuracy(A)} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{13}$$

Remaining 30% reports have been considered as incremental reports and both the DRPSO and NPPSO based initial classifiers have been retrained to generate INRDRPSO and INRNPPSO based incremental classifiers, respectively. To compute the values of objective functions, (i.e., incremental accuracy (IA) and incremental confidence (IC)), the parameter δ (as defined in (7)) is experimentally set as 0.25. Here also 10-fold cross validation is used to measure the performance of the classifiers. The parameters set up and respective performance metrics of both the methods are listed in Table 4. From Table 4, it is observed that INRNPPSO method dominates INRDRPSO in terms of all performance measure metrics for all datasets. Here, the highest classification accuracy of 80% has been achieved by INRNPPSO for crime reports of India, and the highest F-measure of 81% has also been achieved for the same country. Again, for PWN dataset, the highest accuracy and F-measure are of 78% and 80%, respectively.

Fig. 5 shows the bar plot for comparison between the proposed DRPSO and NPPSO methods based on first 70% data of four different datasets. Fig. 5a shows the plot for datasets of India and USA and Fig. 5b for the datasets of UAE and PWN datasets. The plot shows comparison using the four external evaluation metrics, like Accuracy (A), Precision (P), Recall (R) and F-measure (F). It has been visualized from both the plots that all the evaluation metrics have achieved better results for the NPPSO method compared to the DRPSO method. Likewise, Fig. 6 shows the bar plot on the remaining 30% data being acknowledged as the incremental data. Fig. 6a shows the bar plot for the datasets of India and USA, whereas Fig. 6b describes the plotting for UAE and PWN datasets. The plot shows the comparative results of all four metrics like, Accuracy, Precision, Recall and F-measure for the proposed incremental classifiers, INRDRPSO and INRNPPSO. Here also it is observed that

Table 5
Comparison of INRNPPSO with some traditional classifiers for India and USA dataset.

Country	Classifier	Accuracy	Precision	Recall	F-measure
INDIA	INRNPPSO	**80**	**82**	80	81
	NB	74	76	76	76
	KSTAR	72	73	76	74
	SVM	78	81	**83**	**82**
	AB	69	72	71	71
	BG	64	66	71	68
	PART	71	70	73	71
	J48	69	72	74	73
	SPSO	73	74	74	74
	ANN	79	78	81	79
	IPSO	76	77	82	79
	MLP	**80**	79	**83**	81
	RF	67	70	72	71
	LR	69	71	75	73
	DT	70	74	75	74
USA	INRNPPSO	**75**	74	**76**	**75**
	NB	67	69	71	70
	KSTAR	69	70	75	72
	SVM	74	**76**	75	**75**
	AB	65	70	71	70
	BG	66	67	71	69
	PART	70	72	72	72
	J48	69	72	74	73
	SPSO	71	70	74	72
	ANN	67	71	74	72
	IPSO	69	71	74	72
	MLP	72	73	**76**	74
	RF	69	72	75	73
	LR	71	73	75	74
	DT	62	66	67	66

Table 6
Comparison of INRNPPSO with some traditional classifiers for UAE and PWN dataset.

Country	Classifier	Accuracy	Precision	Recall	F-measure
UAE	INRNPPSO	74	73	72	72
	NB	63	67	68	67
	KSTAR	64	67	69	68
	SVM	73	74	**76**	**75**
	AB	65	66	69	67
	BG	70	72	74	73
	PART	66	65	68	66
	J48	67	66	69	67
	SPSO	71	70	72	71
	ANN	**76**	**75**	74	74
	IPSO	74	73	73	73
	MLP	74	74	**76**	**75**
	RF	67	68	69	68
	LR	63	65	68	66
	DT	63	64	67	65
PWN	INRNPPSO	**78**	**79**	**81**	**80**
	NB	73	72	72	72
	KSTAR	71	73	72	72
	SVM	77	**79**	78	78
	AB	69	71	70	70
	BG	65	67	68	67
	PART	71	70	69	69
	J48	69	70	71	70
	SPSO	71	71	73	72
	ANN	74	74	76	75
	IPSO	72	73	73	73
	MLP	73	72	74	73
	RF	68	70	69	69
	LR	67	69	68	68
	DT	69	71	72	71

Table 7
Comparison of INRNPPSO with some recent classifiers.

Country	Classifier	Accuracy	Precision	Recall	F-measure
INDIA	INRNPPSO	**80**	**82**	**80**	**81**
	[14]	72	69	69	69
	[20]	70	68	69	68
	[5]	70	68	68	68
	[22]	74	73	74	73
	[6]	68	70	72	71
	[46]	74	72	70	71
USA	INRNPPSO	**75**	**74**	76	**75**
	[14]	68	67	71	69
	[20]	70	72	76	74
	[5]	71	**74**	**77**	**75**
	[22]	72	**74**	74	74
	[6]	68	69	70	69
	[46]	70	71	75	73
UAE	INRNPPSO	74	73	72	72
	[14]	66	64	69	66
	[20]	70	71	72	71
	[5]	72	73	72	72
	[22]	70	70	**74**	72
	[6]	70	72	70	71
	[46]	**75**	**74**	73	**73**
PWN	INRNPPSO	**78**	**79**	**81**	**80**
	[14]	71	72	72	72
	[20]	69	71	73	72
	[5]	72	74	75	74
	[22]	71	72	73	72
	[6]	70	71	71	71
	[46]	70	71	72	71

INRNPPSO performs better compared to INRDRPSO. From Fig. 6b, it is observed that the proposed classifier INRNPPSO provides better results for the relatively unbiased police and witness narrative reports. Table 4 shows that INRNPPSO has achieved 78% classification accuracy and 80% F-measure for the PWN reports. These results are far better than those achieved by INRDRPSO. In a brief, we can conclude that INRNPPSO is more effective than INRDRPSO, as it provides better performance in terms of all four performance measure metrics. INRNPPSO is also better than the proposed static classifiers, like NPPSO and DRPSO as it does not train the classifier from the beginning based on the whole dataset, but it retrains from the existing classifier based on the incremental portion of the dataset to upgrade it into an incremental classifier. Thus we are considering INRNPPSO as our final proposed incremental classifier which is used for the crime report prediction.

4.2. Comparison based on performance

As INRNPPSO outperforms the other proposed classifier INRDRPSO, so we have compared only the INRNPPSO based incremental classifier with some state-of-the-art traditional as well as recently developed classifiers. Here, we have acknowledged traditional classifiers [2] such as (i) probability based classifier – Naïve Bayes (NB), (ii) instance based classifier – K^* (KSTAR), (iii) kernel based classifier – Support Vector Machine (SVM), (iv) ensemble classifiers – Random Forest (RF), AdaBoost (AB) and Bagging (BG), (v) rule based classifier – J48, PART and Decision Tree (DT), (vi) metaheuristic classifer – Static PSO (SPSO) and incremental PSO (IPSO) (vii) Neural Network based classifier - Logistic Regression (LR), Multilayer perceptron (MLP), Artificial neural network (ANN). For traditional classifiers, whole dataset with 10-fold cross validation is used for performance measurement and for INRNPPSO based classifier (as described in Table 4), the initial NPPSO based classifier and remaining 30% reports with 10-fold cross validation are considered. The evaluation results are shown in Table 5 and Table 6. The highest score obtained for each dataset is marked by bold face. It is observed from the table that while considering the crime data of INDIA, the highest classification accuracy of 80% and precision of 82% have been obtained by INRNPPSO classifier, where as the highest recall of 83% and F-measure of 82% have been obtained by the SVM classifier. The MLP classifier also provides the highest Accuracy of 80% as same as INRNPPSO and recall value of 83% as same as SVM. Thus the proposed INRNPPSO and the existing MLP classifiers outperform the other classifiers. Again, while considering the crime data of USA, the highest classification accuracy of 75%, recall of 76%, and F-measure of 75% have been achieved by INRNPPSO. Here, SVM has resulted in achieving the highest Precision of 76% and F-measure of 75%. MLP also provides the highest Recall of 76% values as ours. Overall these three classifiers perform better than others. ANN provides the highest classification accuracy of 76% and precision of 75% for UAE crime data where the highest recall of 76% and F-measure of 75% have been obtained by SVM and MLP. But the proposed

[2] Available in Python

Table 8
Average ranking of INRNPPSO and traditional classifiers based on Accuracy.

Classifiers	INDIA	USA	UAE	PWN	Avg Rank
INRNPPSO	1	1	2	1	**1.25**
NB	5	7	10	4	6.50
KSTAR	7	6	9	6	7.00
SVM	3	2	3	2	2.50
AB	10	9	8	7	8.50
BG	12	8	5	10	8.75
PART	8	5	7	6	6.50
J48	10	6	6	7	7.25
SPSO	6	4	4	6	5.00
ANN	2	7	1	3	3.25
IPSO	4	6	2	5	4.25
MLP	1	3	2	4	2.50
RF	11	6	6	8	7.75
LR	10	4	10	9	8.25
DT	9	10	10	7	9.00

Table 9
Average ranking of INRNPPSO and traditional classifiers based on F-measure.

Classifiers	INDIA	USA	UAE	PWN	Avg Rank
INRNPPSO	2	1	4	1	2.00
NB	4	5	7	5	5.25
KSTAR	5	4	6	5	5.00
SVM	1	1	1	2	**1.25**
AB	7	5	7	7	6.50
BG	8	6	3	10	6.75
PART	7	4	8	8	6.75
J48	6	3	7	7	5.75
SPSO	5	4	5	5	11.50
ANN	3	4	2	3	3.00
IPSO	3	4	3	4	3.50
MLP	2	2	1	4	2.25
RF	7	3	6	8	6.00
LR	6	2	8	9	6.25
DT	5	7	9	6	6.75

Table 10
Average ranking of INRNPPSO and recent classifiers based on Accuracy.

Classifiers	INDIA	USA	UAE	PWN	Avg Rank
INRNPPSO	1	1	2	1	**1.25**
[14]	3	5	5	3	4.00
[20]	4	4	4	5	4.25
[5]	4	3	3	2	3.00
[22]	2	2	4	3	2.75
[6]	5	5	4	4	4.50
[46]	2	4	1	4	2.75

Table 11
Average ranking of INRNPPSO and recent classifiers based on F-measure.

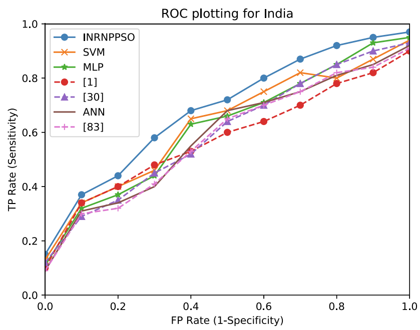
Classifiers	INDIA	USA	UAE	PWN	Avg Rank
INRNPPSO	1	1	1	1	**1.00**
[14]	4	7	4	3	4.50
[20]	5	3	2	3	3.25
[5]	5	2	1	2	2.50
[22]	2	5	2	3	3.00
[6]	3	6	4	4	4.25
[46]	3	4	3	4	3.50

Table 12
Overall ranking of the classifiers including both Accuracy and F-measure.

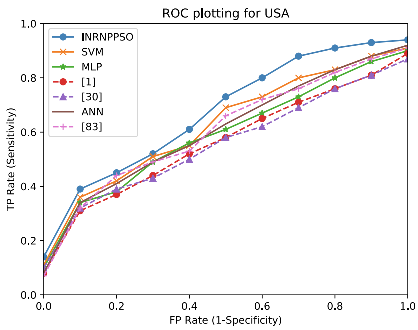
Classifiers	Rank based on Accuracy	Rank based on F-measure	Final rank
INRNPPSO	1.25	2.00	**1.62**
NB	6.50	5.25	5.87
KSTAR	7.00	5.00	6.00
SVM	2.50	1.25	**1.87**
AB	8.50	6.50	7.50
BG	8.75	6.75	7.75
PART	6.50	6.75	6.62
J48	7.25	5.75	6.50
SPSO	5.00	11.50	8.25
ANN	3.25	3.00	**3.12**
IPSO	4.25	3.50	3.87
MLP	2.50	2.25	**2.37**
RF	7.75	6.00	6.87
LR	8.25	6.25	7.25
DT	9.00	6.75	7.87
[14]	4.00	4.50	4.25
[20]	4.25	3.25	3.75
[5]	3.00	2.50	**2.75**
[22]	2.75	3.00	**2.87**
[6]	4.50	4.25	4.37
[46]	2.75	3.50	**3.12**

method has also achieved very good results for this dataset. On the other side, while considering the PWN dataset, INRNPPSO has achieved the highest score for all performance metrics. It has obtained the highest accuracy of 78%, the highest precision of 79%, the highest recall of 81% and finally, the highest F-measure of 80%. Here, along with INRNPPSO, SVM has also obtained the highest precision score of 79%. So as a whole, INRNPPSO provides very promising result for all four crime report datasets.

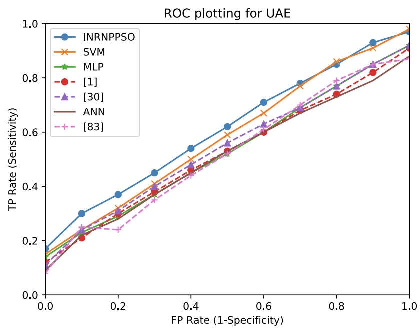
Next, we have considered some existing works shown in [14,20,5,22,6,46] to compare them with our INRNPPSO based classifier. Here, the performance has been measured using four basic evaluation metrics and the results are shown in Table 7.



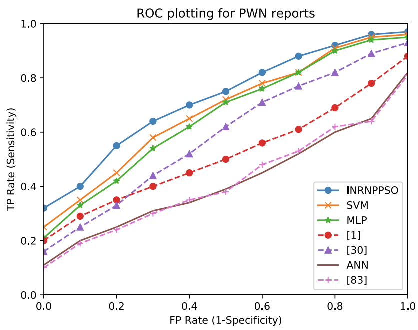
(a) ROC plotting for India dataset



(b) ROC plotting for USA dataset



(c) ROC plotting for UAE dataset



(d) ROC plotting for PWN dataset

Fig. 7. ROC for crime datasets of India, USA, UAE and PWN reports.

The best results are marked in bold face. It is observed from the table that, in most of the cases, INRNPPSO provides the best results on an average with respect to all four performance metrics for INDIA. INRNPPSO has scored the highest accuracy of 80% for Indian crime reports. Apart from this, it has also achieved the precision of 82%, recall of 80% and most importantly 81% F-measure. For crime reports related to USA, the highest accuracy and F-measure of 75%, precision of 74% have been achieved by INRNPPSO. Similar type of results in case of precision of 74%, recall of 77% and F-measure of 75% have been achieved by the method described in [5]. The highest precision also has been obtained by [22]. Only in case of UAE crime data, the highest accuracy of 75%, precision of 74%, and F-measure of 73% have been obtained by the method mentioned in [46] and the best recall of 74% has been achieved by the method [22]. Also, it has been observed from the results that even if we are considering a relatively unbiased and real criminal reports, i.e., the PWN dataset (these are called relatively unbiased as these are interviews narrated into reports), still the proposed INRNPPSO approach provides better results than most of the existing works. Unlike, the other three newspaper crawled reports, here, no existing classifier outperforms the proposed INRNPPSO. Therefore, from the results it can be concluded that the proposed approach is powerful enough for predicting crime types even from the narrated crime reports.

4.3. Comparison based on statistical analysis

The proposed INRNPPSO classifier is compared with other classifiers based on some statistical measures. Here, the classifiers have been given a rank based on their accuracy and F-measure for all four datasets and this ranking has been done based on the evaluation scores obtained by both the traditional and recently developed classifiers. The ranking based on accuracy and F-measure of traditional classifiers is shown in Tables 8 and 9, respectively. Similarly, the recently developed classifiers have been ranked and shown in Tables 10 and 11.

It is observed from Tables 8 and 9 that INRNPPSO secures the first rank based on accuracy, whereas SVM secures the first rank when ranked based on F-measure. But, from Tables 10 and 11, it is observed that in both the cases of accuracy and F-measure, INRNPPSO secures the first rank.

Finally, the average rankings of the classifiers are shown in Table 12. It is observed from the Table 12 that INRNPPSO classifier outperforms the existing traditional and recently developed classifiers, followed by SVM, MLP, [5,22], jointly ANN and [46]. The best results are marked in bold.

Once the ranking of the classifiers has been done, we have considered the five best classifiers among all the classifiers used for comparison. These 5 classifiers, namely SVM, MLP, [5,22], and jointly ANN and [46] have been compared with proposed INRNPPSO based on the Receiver Operating Characteristics (ROC) curve. The *X*-axis in a ROC curve represents the false positive rate (*FpR*), whereas the *Y* axis represents the true positive rate (*TpR*) measured using Eq. (15) and (14), respectively. The bend present in the top left side of the ROC plotting is known as the 'ideal' point, where the *FpR* is zero and *TpR* is one. The ideal point depicts that classifier with larger area under the curve obtains better accuracy than the classifier with less larger area under the curve. Optimal results can be achieved by maximization of the true positive rates and minimization of the false positive rates. Therefore, the inclination present in the ROC curve is an important characteristic for analysing the performances of the classifiers. The resultant ROC curves for all classifiers (including the proposed one) are shown in Fig. 7. It is observed from the figure that the curve representing the proposed work has the largest area under the curve and hence, it represents the best classifier among all other classifiers considered in this experiment.

$$\mathrm{TpR} = \frac{T_p}{T_p + F_n} \tag{14}$$

$$\mathrm{FpR} = \frac{F_p}{F_p + T_n} \tag{15}$$

Here, non-parametric Friedman test [49] has also been performed for statistical analysis of the proposed method, where the samples are assumed to be random, single, independent from each other and continuous. If the p value of this test is very less then the hypothesis can be rejected. The primary objective behind performing this test is to check the presence of significant differences among the considered classification algorithms. This test also ranks the classification algorithms i.e., the best performing algorithm gets the highest rank and the poorest one securing the lowest. Friedman test compares the average ranks of the algorithms and thus the Friedman statistic has been computed. The presence of any statistically significant difference implies that some of the hypotheses in the methodology have different distribution from one another and the task is to identify those pairs of the algorithms which have proved to be significantly different. This has been done using 4 post hoc tests, namely Nemenyi (pNeme), Shaffer (pShaf), Bergmann (pBerg) and Hommel (pHolm) [49]. The proposed INRNPPSO classifier is combined individually with other high ranked classifiers, namely SVM, MLP, [5,22], ANN and [46]. The unadjusted p values and four adjusted p values corresponding to the four post hoc comparisons obtained in each combination are placed in Table 13. As the significance level is considered as $\alpha = 0.10$, the p values below 0.10 represent that those algorithms have significant differences. From the table, it is observed that most of the values are below 0.10 (which are marked in bold), which state that the proposed INRNPPSO classifier is significantly different from other classifiers and thus, the probability that the proposed classification method is statistically significant is more than 90%, which demonstrates the usefulness of the proposed incremental classification algorithm.

Table 13
Results for post hoc comparison.

Hypothesis	Unadjusted *p*	*p*Neme	*p*Holm	*p*Shaf	*p*Berg
INRNPPSO × SVM	**0.00345**	**0.00578**	**0.00547**	0.10498	**0.00498**
INRNPPSO × MLP	**0.00239**	**0.03644**	0.20945	**0.00671**	**0.02587**
INRNPPSO × [5]	**0.00403**	0.13642	0.13502	**0.00161**	**0.01028**
INRNPPSO × [22]	**0.01423**	**0.02135**	**0.02341**	**0.01957**	0.20234
INRNPPSO × ANN	**0.00314**	0.12620	**0.01643**	0.10365	**0.00431**
INRNPPSO × [46]	**0.00413**	**0.00327**	**0.00426**	**0.00395**	**0.00382**

5. Conclusion and future work

Development of an incremental classifier is of immense importance in a real life problem like crime investigation as it learns more and more concepts about crime trends over time from the continuously generated crime reports. The proposed incremental classification method uses both dominance relationship based PSO (INRDRPSO) and non-dominated pareto front based PSO (INRNPPSO). Both the methods not only incrementally train the classifier by the set of crime reports with same set of crime types, but can also equally train the classifier if new types of crime reports are added progressively. It is experimentally observed that INRNPPSO outperforms the INRDRPSO. To demonstrate the effectiveness of INRNPPSO, it has been compared with some traditional as well as recently developed classifiers based on different performance measurement metrics and statistical analysis. The proposed work has also been applied on a relatively unbiased witness and police narrative reports. The proposed method is efficient as it does not learn the classifier from the scratch using the whole dataset, rather it learns the existing classifier based on the incremental portion of the dataset to upgrade it into an incremental classifier. The objective functions are defined on the domain set consists of the rule set and new instances of dataset. They are defined in such a way that, if any existing rule can correctly predict the new object then its fitness value increases, otherwise it decreases. So the definitions of the objective functions are very crucial in the proposed work. There is always a scope of defining better objective functions for the proposed work. But the main demerit of the PSO technique is its greater time complexity. This demerit has been surpassed with the help of parallel processing environment of Keras library in Python. Sometimes, an ensemble of classifiers is more beneficial than a single classifier. In the proposed scheme, we have selected a single set of rules from the non-dominated pareto fronts based on some predefined parameters. But multiple such classifiers can be generated (each from a set of rules) and we can ensemble them to construct an optimal classifier and this remains as the future scope of this work. Now-a-days, deep neural network takes an important role for document classifications, as it automatically generates the features inside the hidden layers of the model where the extracted features are comparatively raw features and may have more semantic relationships among themselves. At the same time, the text documents may contain many imprecise and inconsistent entities, which may be tackled using fuzzy logic. So deep fuzzy neural network may be a suitable classification model. The present work does not solve the class imbalance problem, which may also be the future scope of this paper.

CRediT authorship contribution statement

Priyanka Das: Conceptualization, Methodology, Writing - original draft. **Asit Kumar Das:** Supervision, Validation. **Janmenjoy Nayak:** Writing - review & editing. **Danilo Pelusi:** Review and suggestion. **Weiping Ding:** Review and Suggestion.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] P. Das, A.K. Das, Graph-based clustering of extracted paraphrases for labelling crime reports, Knowl.-Based Syst. 179 (2019) 55–76.
[2] S. Prakash, J. Mercy, M.K. Salugu, K. Vineeth, Optimization of drilling characteristics using grey relational analysis (gra) in medium density fiber board (mdf), Mater. Today: Proc. 2 (2015) 1541–1551.
[3] A.A.S.U.A.K.D.M.P. Munasinghe, R.A.H.D. Perera, CRIMINAL SHORT LISTING AND CRIME FORECASTING BASED ON MODUS OPERANDI, Technical Report, University of Colombo School of Computing, 2010.
[4] P. Das, A.K. Das, Rough set based incremental crime report labelling in dynamic environment, Appl. Soft Comput. 85 (2019) 105811.
[5] S. Anuar, A. Selamat, R. Sallehuddin, Hybrid particle swarm optimization feature selection for crime classification, New Trends in Intelligent Information and Database Systems (2015) 101–110.

[6] S. Sengupta, A.K. Das, Particle swarm optimization based incremental classifier design for rice disease prediction, Comput. Electron. Agricul. 140 (2017) 443–451.
[7] R.M.M.. M.A.J. Noor Maizura Mohamad Noor, Siti Haslini Ab Hamid, M.S. Hitam, A review on a classification framework for supporting decision making in crime prevention, J. Artif. Intell. 8 (2015) 17–34.
[8] T. Ward, E. Carter, The classification of offending and crime related problems: a functional perspective, Psychol. Crime Law 25 (2019) 542–560.
[9] S. Gordon, R. Ford, On the definition and classification of cybercrime, J. Comput. Virol. 2 (2006) 13–20.
[10] A. Babakura, M.N. Sulaiman, M.A. Yusuf, Improved method of classification algorithms for crime prediction, in, Int. Symposium Biometrics Secur. Technol. (2014) 250–255.
[11] A. Nasridinov, S.-Y. Ihm, Y.-H. Park, A decision tree-based classification model for crime prediction, in: Information Technology Convergence, 2013, pp. 531–538.
[12] Aida Mustapha, Payam Hassany Shariat Panahy, Rizwan Iqbal, Masrah Azrifah Azmi Murad, N. Khanahmadliravi, An experimental study of classification algorithms for crime prediction, Indian, J. Sci. Technol. 6 (2013) 4219–4225.
[13] D. Kuang, P.J. Brantingham, A.L. Bertozzi, Crime topic modeling, Crime Sci. 6 (2017) 12.
[14] J. Sheela, A. Vadivel, Criminal event detection and classification in web documents using ann classifier, Int. J. Signal Process. Syst. 4 (2016) 382–388.
[15] M.M. Janeela Theresa, V. Joseph Raj, Fuzzy based genetic neural networks for the classification of murder cases using trapezoidal and lagrange interpolation membership functions, Appl. Soft Comput. 13 (2013) 743–754.
[16] M.M. Janeela Theresa, V. Joseph Raj, A maximum spanning tree-based dynamic fuzzy supervised neural network architecture for classification of murder cases, Soft. Comput. 20 (2016) 2353–2365.
[17] A.K. Singh, N. Baranwal, G.C. Nandi, A rough set based reasoning approach for criminal identification, Int. J. Mach. Learn. Cybern. (2017) 1–19.
[18] P. Das, A.K. Das, J. Nayak, Feature selection generating directed rough-spanning tree for crime pattern analysis, Neural Comput. Appl. (2018) 1–17.
[19] K. Taha, P.D. Yoo, Using the spanning tree of a criminal network for identifying its leaders, IEEE Trans. Inf. Forensics Secur. 12 (2017) 445–453.
[20] M.S. Anuar, A. Selamat, R. Sallehuddin, Particle Swarm Optimization Feature Selection for Violent Crime Classification (2014) 97–105.
[21] P. Das, A.K. Das, An application of strength pareto evolutionary algorithm for feature selection from crime data, in: 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, pp. 1–6.https://doi.org/10.1109/ICCCNT.2017.8203948.
[22] S. Agrawal, V. Sejwar, Crime identification using fp-growth and multi objective particle swarm optimization, in: 2017 International Conference on Trends in Electronics and Informatics, 2017, pp. 727–734.
[23] P. Phillips, I. Lee, Mining co-distribution patterns for large crime datasets, Expert Syst. Appl. 39 (2012) 11556–11563.
[24] W. Tao, Crime data mining based on extension classification, Software Engineering and Knowledge Engineering: Theory and Practice (2012) 383–390.
[25] S.M. Rasoul Kiani, A. Keshavarzi, Analysis prediction of crimes by clustering and classification, Int. J. Adv. Res. Artif. Intell. 4 (2015) 11–17.
[26] S.U.R.W. Malith Munasinghe, Harsha Perera, Machine learning based criminal short listing using modus operandi features, in, in: Proceedings of the International Conference on Advances in ICT for Emerging Regions, 2015, pp. 69–76.
[27] P. Das, A.K. Das, A two-stage approach of named-entity recognition for crime analysis, in: 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, pp. 1–5.
[28] F. Gaxiola, P. Melin, F. Valdez, J. Castro, A. Manzo-Martnez, Pso with dynamic adaptation of parameters for optimization in neural networks with interval type-2 fuzzy numbers weights, Axioms 8 (2019) 1–21.
[29] F. Valdez, A review of optimization swarm intelligence-inspired algorithms with type-2 fuzzy logic parameter adaptation, Soft. Comput. 24 (2020) 215–226.
[30] A. Santiago, B. Dorronsoro, A.J. Nebro, J.J. Durillo, O. Castillo, H.J. Fraire, A novel multi-objective evolutionary algorithm with fuzzy logic based adaptive selection of operators: fame, Inf. Sci. 471 (2019) 233–251.
[31] F. Valdez, J.C. Vazquez, P. Melin, O. Castillo, Comparative study of the use of fuzzy logic in improving particle swarm optimization variants for mathematical functions using co-evolution, Appl. Soft Comput. 52 (2017) 1070–1083.
[32] X. Li, G. Yang, Artificial bee colony algorithm with memory, Appl. Soft Comput. 41 (2016) 362–372.
[33] W. Li Xiang, X. Lei Meng, Y. Zhen Li, R. Chun He, M. Qing An, An improved artificial bee colony algorithm based on the gravity model, Inf. Sci. 429 (2018) 49–71.
[34] A. Rastgou, J. Moshtagh, Application of firefly algorithm for multi-stage transmission expansion planning with adequacy-security considerations in deregulated environments, Appl. Soft Comput. 41 (2016) 373–389.
[35] P. Melin, D. Snchez, Multi-objective optimization for modular granular neural networks applied to pattern recognition, Inf. Sci. 460–461 (2018) 594–610.
[36] L. Shao, Y. Bai, Y. Qiu, Z. Du, Particle swarm optimization algorithm based on semantic relations and its engineering applications, Syst. Eng. Procedia 5 (2012) 222–227.
[37] Z. Li, R. Tan, B. Ren, Research on particle swarm optimization of variable parameter, in: Advances on P2P, Parallel, Grid, Cloud and Internet Computing, 2017, pp. 25–33.
[38] R. Rahmani, M.S. Mahmodian, S. Mekhilef, A.A. Shojaei, Fuzzy logic controller optimized by particle swarm optimization for dc motor speed control, IEEE Student Conference on Research and Development (SCOReD) 2012 (2012) 109–113.
[39] J. Hurel, A. Mandow, A. Garca-Cerezo, Tuning a fuzzy controller by particle swarm optimization for an active suspension system, IECON 2012–38th Annual Conference on IEEE Industrial Electronics Society (2012) 2524–2529.
[40] D. Wang, D. Tan, L. Liu, Particle swarm optimization algorithm: an overview, Soft. Comput. 22 (2018) 387–408.
[41] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Trans. Evol. Comput. 6 (2002) 182–197.
[42] K. Deb, D.E. Goldberg, An investigation of niche and species formation in genetic function optimization, in, in: Proceedings of the Third International Conference on Genetic Algorithms, 1989, pp. 42–50.
[43] E. Loper, S. Bird, Nltk: The natural language toolkit, in: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, vol. 1, 2002, pp. 63–70.https://doi.org/10.3115/1118108.1118117.
[44] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[45] J. Demšar, T. Curk, A. Erjavec, T. Črt Gorup, M. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, A. Toplak, M. Starič, L. Štajdohar, L. Umek, J. Žagar, M. Žbontar, B. Zupan Žitnik, Orange: Data mining toolbox in python, J. Mach. Learn. Res. 14 (2013) 2349–2353.
[46] A.K. Das, J. Sil, An efficient classifier design integrating rough set and set oriented database operations, Appl. Soft Comput. 11 (2011) 2279–2285.
[47] Y. Li, Y. Chen, J. Zhong, Z. Huang, Niching particle swarm optimization with equilibrium factor for multi-modal optimization, Inf. Sci. 494 (2019) 233–246.
[48] C.H. Ku, A. Iriberri, G. Leroy, Crime information extraction from police and witness narrative reports, in: IEEE conference on Technologies for Homeland Security, 2008, pp. 193–198.
[49] N. Settouti, M.E.A. Bechar, M.A. Chikh, Statistical comparisons of the top 10 algorithms in data mining for classification task, Int. J. Interactive Multimedia Artif. Intell. 4 (2016) 46–51.