Dynamic Multi-layer Ensemble Classification Framework for Social Venues Using Binary Particle Swarm Optimization

Article *in* Wireless Personal Communications · April 2019

DOI: 10.1007/s11277-019-06156-w

Ahsan Hussain
Islamic University of Science and Technology JK
11 PUBLICATIONS 33 CITATIONS

Keshavamurthy B.N.
National Institute of Technology Goa
34 PUBLICATIONS 206 CITATIONS

Wireless Personal Communications
https://doi.org/10.1007/s11277-019-06156-w

Dynamic Multi-layer Ensemble Classification Framework for Social Venues Using Binary Particle Swarm Optimization

Ahsan Hussain[1] · Bettahally N. Keshavamurthy[1] · Ramalingaswamy Cheruku[1]

Abstract

Multi-layer ensemble frameworks perform much better as compared to individual classifiers. However, selection of a classifier and its placement, impacts the overall performance of ensemble framework. This problem becomes very difficult, if there are more classifiers and layers. To address these problems in this paper, we design "Binary Particle Swarm Optimization" method for selection and placement of right classifiers in multi-layer ensemble model. Proposed classifier weight-assignment method is implemented to prioritize the selected classifiers. The model is simulated for the classification of social-user check-ins in Location-Based Social Network datasets. The experimental results show that the proposed ensemble model outperforms the state-of-the-art ensemble methods in the literature. It can be used by security firms, high level decision makers and various governmental organizations for tracking malicious users.

Keywords Location-Based Social Networks · Social-venue classification · Machine learning · Majority voting · Dynamic multi-layer ensembles · User-checkins

1 Introduction

Location-Based Social Networks (LBSNs) like Foursquare, Google+ and Facebook, effectively are an extension of online social networks to the real world. LBSNs help us to comprehend, investigate, explore, and geographically record the spots we live in. Foursquare permits us to spatially check-in to famous places of a city, updating rich databases that hold computerized engravings of our associations at the same time [1]. Internet connectivity and location-aware smartphones are required by LBSN users for recording their presence at a particular place called as *checkin*. Friends in the vicinity sharing a common LBSN,

✉ Ahsan Hussain
ahsan.hussain@nitgoa.ac.in

Bettahally N. Keshavamurthy
bnkeshav.fcse@nitgoa.ac.in

Ramalingaswamy Cheruku
rmlswamygoud@gmail.com

[1] Department of Computer Science and Engineering, National Institute of Technology Goa, Ponda 403401, India

Published online: 12 February 2019

are notified due to the checkin. This gives rise to *venues* which are the places explicitly registered on the LBSNs. Checkins provide a sematic representation of the attributes like geographical locations along with the venue name and credentials. Foursquare that has the primary function of checkins, incorporates the dedicated Swarm application for providing the checkin ability to reorganise its business [2]. A sample Foursquare checkin is visualized in Fig. 1. Social-users are shown checking in at various venues, with the some users (black icons) visiting some common venues.

Foursquare datasets contain various features that provide the checkin information of a social-user. During preprocessing, unwanted and redundant features are removed and only significant features are selected that can improve the classification-accuracy [3]. The checkin information can be utilized for predicting the location by virtue of various machine learning techniques. Various studies prove that the single classifiers performance is modest when compared to ensemble classification methods. Therefore, various ensemble methods like bagging, boosting, random sub-space etc. are becoming a popular means of classification of large datasets [4]. Ensemble of various classifiers together perform much better by reducing the estimated error-variance. But, a standard mechanism for choosing ideal classifiers that gives the best accuracy in the ensemble, is lacking. A set of base classifiers are trained and dynamically placed in a particular order that gives the better performance. A small amount of improvement in classification accuracy can be of utmost importance for security agencies and business organizations, where small mistakes can cause a lot of damage.

In this paper, we propose a Dynamic Multi-Layer Ensemble Classification (DMLEC) technique that works on dynamically choosing the classifiers as per the Binary Particle Swarm Optimization (BPSO) using a novel fitness function. In addition, a new classifier weight assignment method is proposed that updates weights for the particular classifiers as per their classification accuracy. As compared to the single-layered classifiers, the proposed DMLEC is more flexible. It can be used over big datasets for classification of various real world problems. Due to less computing errors at each level, the multi-layered scheme gives

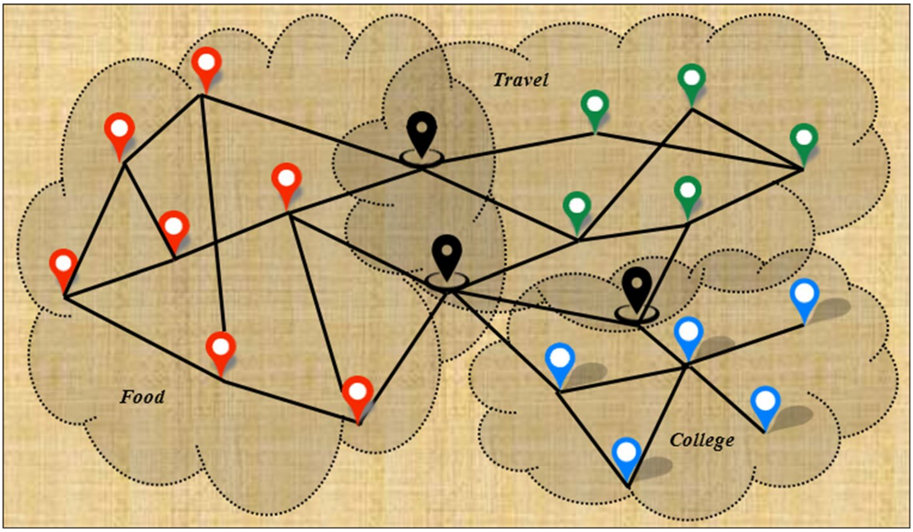

Fig. 1 Sample Foursquare checkins

better classification accuracy than the single-layered structure [5]. Users generally follow a trend in visiting and checking-in at their favorite places. The check-in information can be utilized in order to predict the location of a social-user. In addition, the proposed model can be very useful in the expert and intelligent systems' environments, where the system needs to detect malicious users present in various domains like credit card transactions, spammers, etc. It helps commercial-intelligence, where valuable insights for the high level decision makers can be provided from the electronic information.

1.1 Contributions

The main contributions of this paper are the following:

1. This is the first comprehensive study of social-venue classification in LBSNs, explicitly taking into account checkin information disclosure with respect to social-users.
2. A novel fitness function for obtaining the fitness value of each candidate solution is defined, based on the accuracy of base classifiers.
3. DMLEC model is proposed. BPSO is designed for dynamical selection and placement of classifiers in the proposed DMLEC model.
4. New classifier weight-assignment method has been proposed.

1.2 Related Work

LBSNs have been defined as the technology where location of the mobile device is used to access various information services of users through mobile networks [6]. Recently, LBSNs like Foursquare are becoming highly popular by acquiring new customers for small businesses [7, 8]. The number of Foursquare users and checkins increased from 7 million and 0.45 billion by March 2011 to 40 million users and 4.5 billion checkins by September 2013, respectively [9]. Checkin is a unique feature of an individual social-user that can lead to understanding the user-behaviour in a different perspective. However, only limited studies exist in this field that include the works of [10–12], co-relate the personality traits of individual social-users with their respective checkins.

The factors that motivate the checkin behavior of social users include visiting new venues and knowledge of already visited ones, creating new social friend connections and for leisure activities. Selecting venues to be visited depends on user-preferences about a particular venue [13]. Self-presentation is achieved using various LBSNs that help to choose particular venues and controlling the number of checkins for avoiding spamming their accounts [14]. This can help to avoid certain venues that are perceived as negative places to visit [15]. Foursquare checkins have been studied for various researches on social-user behaviour based on user activity patterns aggregating a higher degree of location-related data [14, 15].

Link mining is categorized by focusing on edges/links, nodes and complete graphs [16]. Link-based object classification based on node-attributes can be traditionally performed by using various machine learning algorithms [16, 17]. Recently, various researches on *link-prediction* in SNs have come to focus. Two of the best works include Bliss et al. [18] and Torabi et al. [19] who have used different strategies on *Twitter* and *Brightkite* datasets and achieved accuracy of 97.4% and 97.8%, respectively. Our work focus is for node prediction, where important advances have been made by Gu et al. [20] and Almallah et al. [21]. They have worked on *location-identification* and *venue-prediction* on different Foursquare

LBSN datasets and achieved best accuracy of 92.1% and 66%, respectively. Hu et al. [22] obtained accuracy of 70% using models of human mobility features on *Twitter* dataset for home location inference.

Classifier ensembles can be used for classification of social-venues. Ensemble technique leads to a much improved classification accuracy due to the diversity and independency of the individual classifiers constituting the ensemble [23]. Providing a generalized approach to ensure classifier diversity is a major challenge as it ensures the capacity to cover the individual errors. Each classifier used in the approach has its own valuable contributions in the final classification accuracy. *K*-Nearest Neighbors Classification (KNC) is one of the most trivial classification methods [24] that classifies an object by neighbor-majority vote to its nearest neighbor. The disadvantage of using KNN is that available data is required by the classifier that can sometimes lead to considerable overhead for a larger training dataset. Decision Tree Classifier (DTC) resembles human reasoning and logically combines a simple test sequence [24] where numeric/nominal attributes are compared against possible threshold values in every test. Decision tree algorithm J48 is a C4.5 decision tree used for classification by creation of a binary tree that is applied on each tuple in the dataset. The main aim is to divide the whole dataset into a range, depending on the attribute values. It classifies using decision trees or the generated-rules from them [25]. Naive Bayes (NB) is based on probability which gives an excellent accuracy for classification and prediction [24]. For performing correlation analysis between different feature sets (NB drawback), Linear Discriminant Aanlysis (LDA) and Logistic Regression (LR) are used in the proposed ensemble model. LDA classifier has a linear decision surface, does not require any parameters for algorithm tuning and is inherently multiclass classifier [26]. LR is simple, easily interpretable and accepted worldwide for better classification [26]. Random Forest Classifier (RFC) proves to be the most important among all base classifiers in our proposed DMLEC as it can deal with uneven data sets that have missing variables and always gives lower classification error [27]. Suppost Vector Machine (SVM) has advantages like better classification, regression, and outlier detection, effective in high dimensional space and takes decisions based on various kernel functions [28]. This enables the SVM algorithm to increase the classification and prediction accuracy by expertly handling the over-fitting issue. Therefore, all classifiers with varying features are brought together to produce the ensemble that performs much better in comparison to the individual base classifiers.

Classifier ensemble methods are based on the combination of individual set of classifiers to obtain a collective classification decision [29]. The reasons for the use of ensemble classifiers [30] include statistical (vote-averaging), computational (better approximation to the true unknown function in local searching) and representational scope (coping with complex decision boundaries). Refinement of classification tasks can be performed using different classifiers for different layers with each layer having different set of classifiers. Among the various ensemble techniques, Majority Voting (weighted and un-weighted) [31, 32], Bagging [33], Boosting [34], and Random Subspace [35] are most widely used in the literature.

The rest of the paper is organized as follows: the proposed BPSO based dynamic classifier selection is discussed in Sect. 2. Experimental and comparative analysis of results obtained from proposed DMLEC model are presented in Sect. 3. Finally, Sect. 4 concludes the paper.

2 Proposed Methodology

The proposed framework consists of 4 major phases, as presented in Fig. 2. The first phase is the pre-processing phase which involves selection of best attributes from the datasets, followed by basic data cleaning and imputation. This processed data is partitioned into training and testing datasets. The training dataset is fed to the training phase in which various chosen base classifiers are trained. Best classifiers for testing phase are dynamically selected based on classifiers training accuracies using the BPSO approach in phase 3. In testing phase, the selected classifiers are arranged in two layers as per the classifier

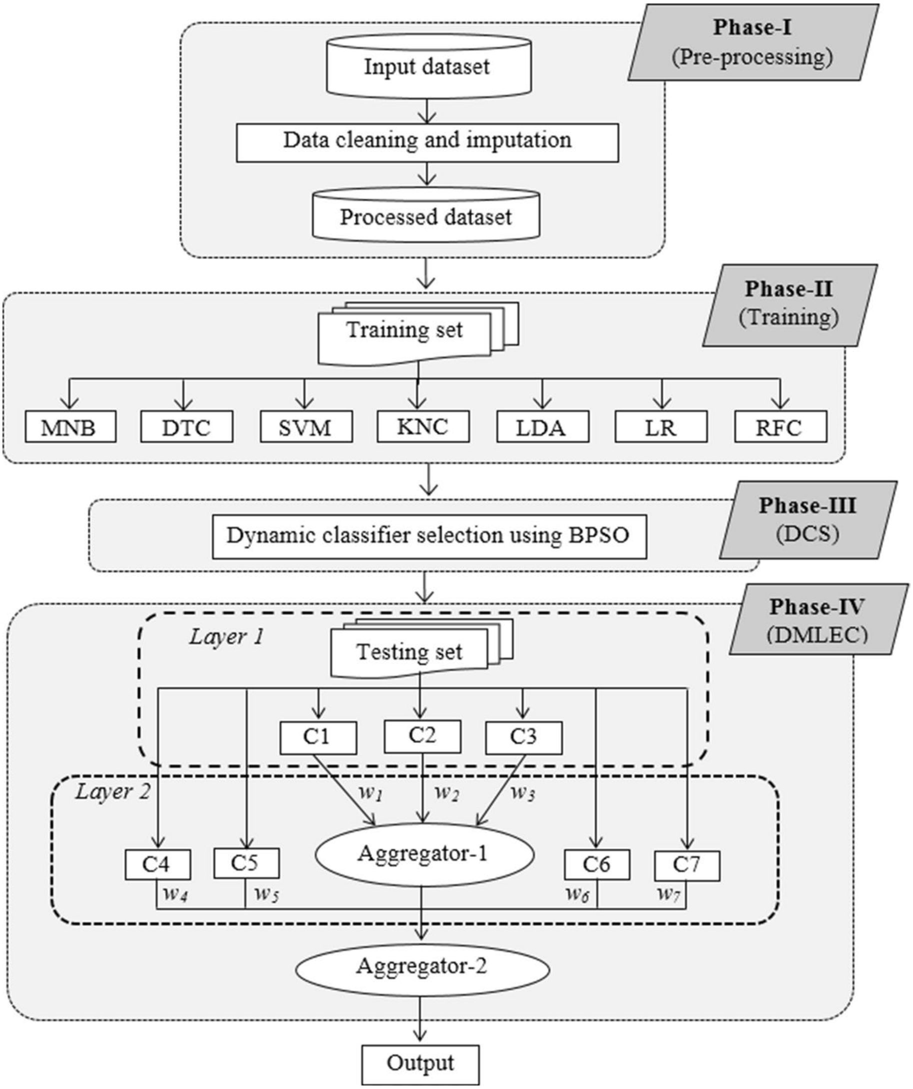

Fig. 2 Proposed DMLEC framework

encoding scheme. In layer 1, an ensemble of 3 base classifiers is used. The output of layer 1 is obtained by using majority voting principle and it is passed to layer 2. In layer 2, the layer 1 output is combined with outputs of 4 more base classifiers. Final output is obtained by applying majority voting principle on layer 2 outputs, which gives the classification accuracy.

We summarize the proposed DMLEC model by the following four steps:

1. Perform pre-processing over the datasets. Generate initial set of base learners with varying characteristics in order to obtain the individual accuracy rates on the processed data.
2. Select the base learners with best accuracy among all base classifiers for training the model over various partitions of the training datasets.
3. Apply the BPSO approach using the novel fitness function and the classifier encoding scheme for arranging the best classifiers at the right places in the model.
4. Integrate the selected base learners from the previous step with different classifier combinations in the testing phase. This integration involves the use of weighted and un-weighted majority voting principle, to form the DMLEC model for obtaining the final classification accuracy.

2.1 Pre-processing Phase

In this phase, basic data mining operations like data-acquisition and pre-processing of the input LBSN datasets is performed. Data cleaning, replacement of missing data with substituted values and data normalization is carried out. After pre-processing, we trimmed our datasets to having 9 attributes each, that include essential check-in information like the exact location and time of the social users visiting a particular venue. The combination of these essential attributes including the user-id, was used for training and testing purposes. By selecting only desired attributes for LBSN analysis, venue-classification and prediction by the proposed model efficiently resolves the storage problem.

2.2 Training Phase

The pre-processed LBSN datasets are partitioned into training and testing datasets. The training data is fed to the training phase in which various chosen base classifiers are trained. These classifiers are dynamically selected for the testing phase, based on their individual classification accuracies for the datasets using BPSO.

2.3 Adopting BPSO Algorithm for Classification Rule Mining

This section discusses how BPSO is adapted for classifier selection tasks (i.e., use BPSO for selecting right classifier in right place). This section also discusses the proposed novel fitness function and the proposed encoding scheme for classifier selection task.

2.3.1 BPSO

Particle Swarm Optimization (PSO) is an evolutionary computation technique [36, 37], that is inspired from the social behavior of flocking of birds. Multiple candidate solutions are tried (number of flying particles) until a best solution is obtained (tracing the bird paths). Thus, each particle considers its own best solution and the best among all the currently obtained solutions from the swarm is selected. For modifying the positions, all PSO particles take into account their current locations, their current speeds, the distance to their personal best solution, *pbest*, and the distance to the global best solution, *gbest*.

Kennedy and Eberhart proposed the Binary version of PSO (BPSO) in 1997 [38]. Two varying components that distinguish between the continuous and binary versions of PSO include a different position updating procedure and a new transfer function. The particle positions in binary search spaces are switched between 1 and 0 using the position updating procedure. In addition, the continuous search space is mapped to a binary search, with the help of the transfer function. The original version of BPSO has the drawback of getting stuck in local minima [39]. To solve this problem, certain modifications have been introduced.

PSO was mathematically modeled as follow:

$$v_i^{t+1} = wv_i^t + c_1 \times rand \times (pbest_i - x_i^t) + c_2 \times rand \times (gbest_i - x_i^t) \tag{1}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2}$$

where v_t^i is the velocity of particle i at iteration t, c_j is an acceleration coefficient, w is a weighting function, *rand* is a random number between 0 and 1, x_i^t is the current position of particle i at iteration t, $pbest_i$ is the best solution that the ith particle has obtained so far, and *gbest* indicates the best solution the swarm has obtained so far.

The ability to explore the PSO is given by the initial segment (wv_i^t) of Eq. 1. It is followed by $[c_1 \times rand \times (pbest_i - x_i^t)]$ representing the private thinking and the last part $[c_2 \times rand \times (gbest_i - x_i^t)]$ gives the particle-collaboration. The process starts by placement of the particles in the problem space randomly. Equation 1 is used to calculate the velocities of various particles at each iteration. The location of different particles in the problem space is calculated using Eq. 2, after defining the particle-velocities. Until the end criteria is satisfied, the process of changing the position or location of different particles is continued.

Various problems like dimensionality reduction and feature selection, have intrinsic non-continuous binary search spaces [40, 41]. Moreover, the conversion of continuous variables to binary variables helps various problems with continuous real search space to be converted into binary problems. Irrespective of the various types of binary problems, a binary search space has its individual structure with some drawbacks. A binary search space can be thought-out as a hypercube, wherein the agents (particles) of a BPSO algorithm can translocate to closer and farther hypercube corners by various number of bit-flippings [35]. So, the basic concepts of velocity and position updating process need to be modified while designing BPSO.

In the continuous PSO version, mobility of particles in the search space is by utilizing position vectors within the continuous real domain. Using the Eq. 2, position updation for particles is implemented by adding velocities to positions. At the same time, in case of a discrete binary space, there is an entirely different meaning to the position updation for particles [42]. Here, the process of position updation for particles cannot be performed using Eq. 2, as it deals with only two numbers (1 and 0). Thus, we require to find out a method by which the velocities of particles can be used to change the positions of the agents from 1 to 0 and vice versa. Therefore, it can be stated that a link needs to be devised between velocity and position, as well as revising the position updating Eq. 2.

The process of position updation deals with switching between 1 and 0 values in discrete binary spaces. The agent-velocities determine how this switching takes place. The main problem is to use the idea of real space velocity for updating the particle-positions in the binary space. This can be achieved by changing the agent position with the probability of its velocity [35, 43, 44]. Velocity values are mapped to probability values for updating the particle-positions using a transfer function.

Kennedy and Eberhart [35] proposed the use of original BPSO to allow PSO to operate in binary problem spaces. Here, the flight of the particles in the binary search space for their position vectors is limited to only either 1 or 0 due to the role of velocity. For transforming all the real velocity values to probability values in the interval [0, 1] is accomplished by using a transfer function.

If the transfer function is smoother, it provides a high probability of changing the position for a large absolute value of the velocity and vice versa. Hence, we have chosen V4 from the family of V-shaped transfer functions [45], as shown in Fig. 3. It is represented by the following Eq. 3.

$$T(v_i^k(t)) = \frac{2}{\pi} arc \tan\left(\frac{\pi x}{2}\right) \tag{3}$$

where $v_i^k(t)$ indicates the velocity of particle i at iteration t in kth dimension. After converting velocities to probability values, position vectors could be updated with the probability of their velocities as follows:

$$X_i^k(t+1) = \begin{cases} 0, & \text{if } rand < T(v_i^k(t+1)) \\ 1, & \text{if } rand \geq T(v_i^k(t+1)) \end{cases} \tag{4}$$

BPSO algorithm is constituted of the following major steps:

1. Initialize all particles with random values.
2. Repeat steps 3–5 until meeting the end condition.
3. Equation 1 defines the velocities for all particles.
4. Calculate probabilities for changing the elements of position vectors using Eq. 3.

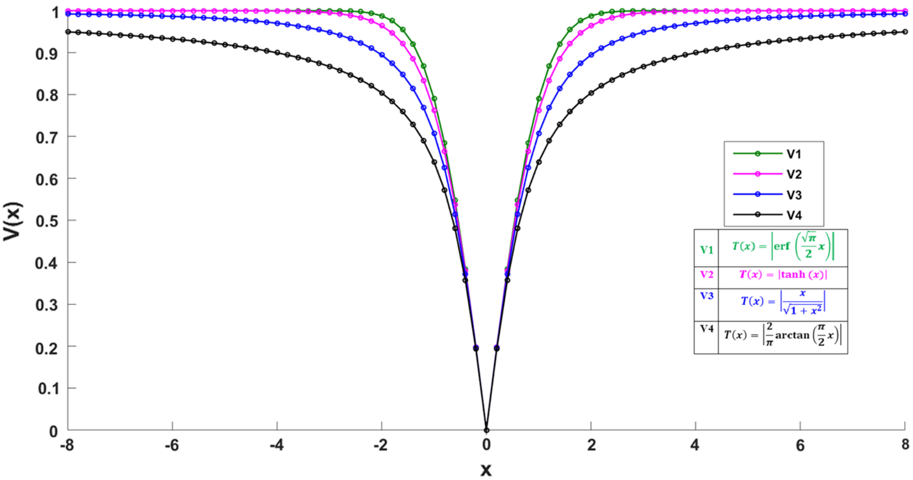

Fig. 3 Family of V-shaped transfer functions

5. With the help of rules given in Eq. 4, update the elements of position vectors.

2.3.2 Novel Fitness Function

A novel fitness function has been designed in the proposed approach to compute the fitness value of each candidate solution. This function is presented in the following equation.

$$Fitness = \frac{1}{N}\sum_{i=1}^{N} Accuracy_i \tag{5}$$

where N is the total number of classifiers in the ensemble. For a better solution fitness function needs to be maximized, i.e., high fitness value indicates a good candidate solution.

2.3.3 Encoding Method

In classifier selection process, each particle represents a candidate solution that maps best classifier to best places in DMLEC framework. Each particle P_i for 2-layered model is encoded, as shown in Fig. 4. It is clear that each layer is encoded as septuple $< C_1, C_2, C_3, C_4, C_5, C_6, C_7 >$. In each septuple, flag variable indicates the corresponding classifier is present in the respective layer or not.

2.3.4 Proposed Classifier Weight-Assignment Method

The prediction performance of the aggregators (weighted voting schemes) is less-explored even after a wide usage for various applications. In this direction, we present an extensive empirical analysis of un-weighted and weighted majority voting schemes using our special function on weight assignment in the proposed classification model for LBSNs. In this section, we define a novel weight-assignment function, which is used in weight-updation with the help of Eq. 7. It helps in obtaining weights of base classifiers as per their classification accuracies that are to be used in the DMLEC model. Classifier with high accuracy attains higher weight and vice versa. The novel weight-calculation function used for weight assignment in the proposed model is given in the following Eq. 6:

$$W^i_{Adj} = W^i_{Org} \times \left(1 + \frac{1}{2}\log_{10}(Accuracy_i)\right) \tag{6}$$

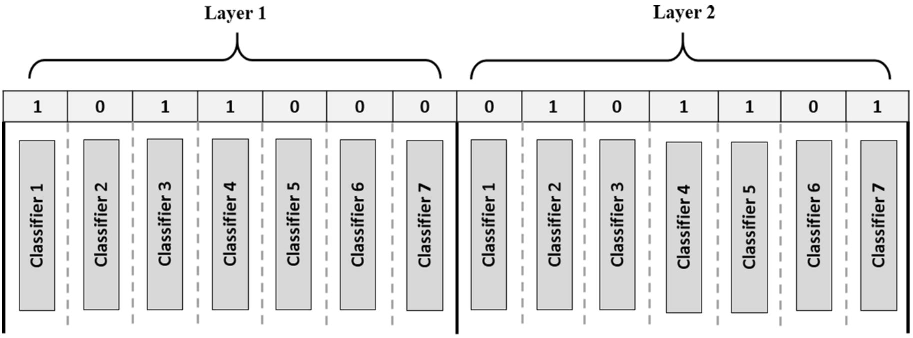

Fig. 4 Proposed encoding scheme

where W_{Adj} is the adjusted weight of the classifier i, W_{Org} is the original weight of the classifier and *Accuracy* is the classification accuracy of the base classifier.

The following weight-assignment function is used for obtaining the optimum weights for the aggregators as per the respective accuracies. It implies that the weight of a particular classifier ranges between 1 and 2.

$$W^i_{Adj} = \begin{cases} W^i_{Org}, & \text{If } Accuracy_i = 0; \\ W^i_{Org} \times 2, & \text{If } Accuracy_i = 100; \\ W^i_{Org} \times \left(1 + \frac{1}{2}\log_{10}(Accuracy_i)\right), & \text{otherwise.} \end{cases} \tag{7}$$

The output of each aggregator is calculated using below equation.

$$Final\ output = \sum_{i=1}^{N} W_i \times C_i \tag{8}$$

where W_i is the weight assigned to ith classifier C_i and N is the total number of classifiers.

2.4 Proposed Dynamic Multi-layer Ensemble (DMLEC) Model

The proposed DMLEC model is based on the concept of utilizing the decisions made by BPSO approach for dynamically selecting the diverse base classifiers. Each classifier in the framework has complementary qualities that helps in classifying the LBSN datasets accurately for venue prediction. The model is flexible in utilizing the classifiers in various combinations at different layers. Each individual classifier predicts a discrete value as a result based on the output. In addition, we also supply weights to the classifiers in accordance to our novel weight-assignment method. Finally, the aggregator uses the majority vote for the final classified output.

Each classifier used in the proposed approach has its own valuable contributions in the final classification accuracy and it is mostly difficult to find the best generalized classifier. To improve this drawback, various inaccurate and weak classifiers are combined to form a strong ensemble of classifiers. This technique leads to a much improved classification accuracy due to the diversity and independency of the individual classifiers constituting the ensemble [24]. Providing a generalized approach to ensure classifier diversity is a major challenge as it ensures the capacity to cover the individual errors.

As shown in the framework (Fig. 2), the proposed model consists of 2 layered ensemble of classifiers in the Phase-IV (DMLEC phase). Different combinations of the classifiers are dynamically selected using the proposed classifier encoding scheme in both layers. Layer-1 ensemble consists of 3 base classifiers whose output is obtained using majority voting aggregator (Aggregator-1). Similarly, layer-2 consists of 4 base classifiers and the result obtained using the Aggregator-1. The output of layer-2 is also obtained using the majority voting aggregator (Aggregator-2) that gives the final classification accuracy. Different variations in the proposed approach are carried out using all four possible combinations for both aggregators (weighted and un-weighted) to obtain all the final output. The weight for each classifier in aggregators is calculated using the proposed classifier weight-assignment method, described in the previous section. The output of the proposed DMLEC is the final classification for a given dataset instance.

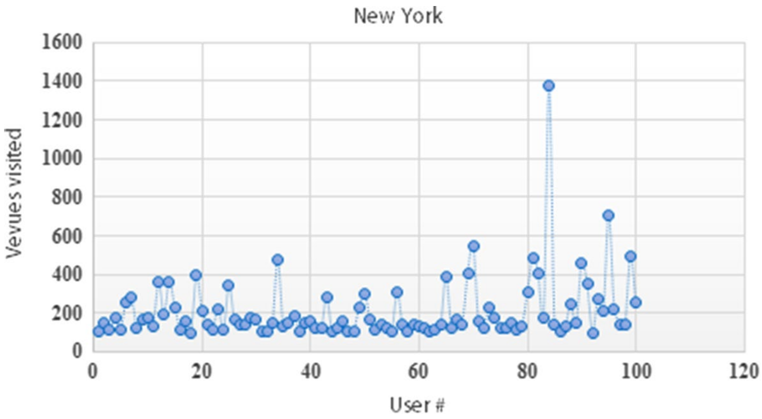

Fig. 5 Social venues visited by 100 users in New York

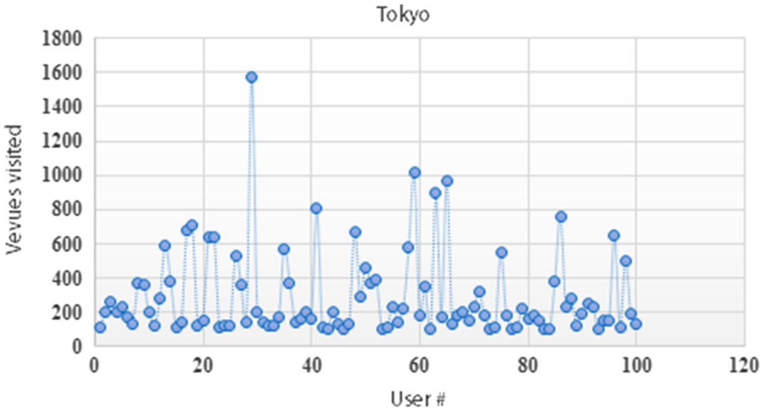

Fig. 6 Social venues visited by 100 users in Tokyo

3 Simulation Results and Discussion

3.1 Data Source

Well-known Foursquare LBSN dataset [46] is used for the experimentation. The data for two cities, New York and Tokyo, is taken into consideration. For New York dataset, check-in records of 100 social-users with 20,980 data entries and 228 venues visited are considered. Similarly, for the Tokyo dataset, 100 social-users with 29,015 data entries and 200 venues visited are considered. We show the distribution of social venues visited by the 100 users in each dataset in Figs. 5 and 6. In New York dataset, user 84 has the highest number of venues visited (1376) and users 18 and 92 have the least visits (100). Similarly, in Tokyo dataset, user 29 has highest visits (1569) to Foursquare venues and user 43 and 93 have the least visits (100) among the first 100 users. For improving the classification performance, best features are selected using various pre-processing methods used for model creation. Desired features chosen provide advantages like noise removal, computational cost reduction, easy model updation and better understanding of the final model. This improves the learning speed and overall accuracy of the classification model. We extracted 9 best attributes related to the social-user check-ins for experimentation that include user-ids, geographical coordinates and time-zone of the place, exact check-in time, day, date, etc.

3.2 Experimental Setup

In this paper, all the experiments are performed on a 64-bit machine with an Intel (R) Core i7 processor with 3.60 GHz speed and 8 GB RAM. The BPSO algorithm for classifier selection has been programmed with Matlab R2015a. The DMLEC model has been implemented using Scikit-learn library for the Python programming language.

It is desirable to do problem specific parameter tuning to achieve good performance, since parameters like population (PoP), cognitive acceleration coefficient (c_1), social acceleration coefficient (c_2), inertia weight (w) and transfer function, affect the performance of BPSO. In order to perform fine tuning (finding most suitable values) of these parameters for each dataset, we have to carry out some experimenting, which is explained in the following Table 1. We carried out experiments using the best performing machine learning algorithms that include SVM, MNB, LDA, LR, DTC, KNC and RFC over the datasets. The proposed DMLEC works on the various combinations and placements of these base classifiers taken together.

3.3 Experimental Results

For evaluating the performance of the proposed DMLEC using BPSO, the results obtained using single base classifiers are compared with different ensemble classifiers in the results section. The results have been obtained on the Foursquare LBSN datasets for New York and Tokyo cities. Single classifier classification accuracy results for k-fold cross validation (k = 10, 5, 4) are shown in Table 2. It is followed by the Table 3, which gives the weight assignment for each classifier. Further, accuracy results of 3-ensemble classifiers and 5-ensemble classifiers are shown in Tables 4 and 5, respectively. Moreover, the detailed results for the various combinations of multi-layered approach using dynamically chosen classifiers at different layers are given in Tables 6, 7, 8 and 9. Finally a comparison of the proposed DMLEC model and other state-of-the-art methods is presented in Table 10.

Table 1 Fine tuned parameters for BPSO

Parameter	Value	Description
PoP	50	No. of particles
c_1	1.5	Cognitive acceleration coefficient
c_2	0.5	Social acceleration coefficient
w	0.9	Inertia weight
$[V_{max}, V_{min}]$	[−0.5, 0.5]	Range of each particle velocity
Transfer function	$T(v_i^k(t)) = \frac{2}{\pi} arc\tan(\frac{\pi x}{2})$	Transfer function

Table 2 Single classifier results

Base learners	New York			Tokyo		
	10F	5F	4F	10F	5F	4F
DecisionTreeClassifier	39.79	36.65	37.02	47.58	45.5	44.8
DecisionTreeRegressor	30.78	28.26	28.56	36.60	35.2	34.0
KNeighborsClassifier	17.84	15.56	15.46	31.87	29.9	29.4
KNeighborsRegressor	0.88	0.56	0.67	4.25	3.95	3.47
LDA_MLPClassifier	10.68	10.42	10.58	35.70	35.5	35.6
LM_SGDClassifier	4.54	3.36	7.42	6.87	7.65	9.78
NB_Bernoulli	8.67	7.05	7.07	9.17	8.46	8.58
NB_Gaussian	1.53	1.47	1.44	3.86	3.68	3.78
NB_Multinomial	14.07	14.07	14.25	3.21	7.33	8.93
NearestCentroid	0.29	0.18	0.22	0.15	0.08	0.12
Perceptron	5.67	1.67	7.42	2.55	5.46	2.70
SVM_SVC	19.99	18.05	17.16	35.37	35.3	35.5
RFC	32.46	28.26	27.8	57.82	48.8	45.8

Table 3 Classifier weight assignment

Classifier	New York		Tokyo	
	Accuracy	Weights assigned	Accuracy	Weights assigned
DTC	40	1.79	47.6	1.83
KNC	18	1.62	32	1.75
LR	5	1.35	7	1.43
LDA	11	1.52	36	1.78
MNB	14	1.73	4	1.31
SVM	20	1.65	36	1.77
RFC	32	1.75	56	1.89

Table 4 3-ensemble classifier results

Classifiers	Weight	New York				Tokyo			
		k = 10 (best/avg)		k = 5 (best/avg)		k = 10 (best/avg)		k = 5 (best/ avg)	
LR, RFC, DT	U	87.9	86.5	90.21	89.0	87.6	85.0	88.0	87.3
LR, RFC, KNC	U	57.7	51.5	60.4	58.7	61.3	57.0	78.0	77.7
LR, RFC, SVM	U	70.6	69.0	89.8	88.8	86.0	85.1	80.1	78.8
LR, RFC, SVM	W	89.8	88.0	89.5	89.0	88.7	87.5	84.9	84.5
LR, RFC, KNC	W	89.7	88.0	90.8	89.1	86.7	85.5	85.6	85.4
LR, RFC, LDA	W	89.1	87.7	89.6	88.5	86.0	85.3	89.2	89.0

Table 5 5-ensemble classifier results

Classifiers	Weight	New York				Tokyo			
		k = 10 (best/avg)		k = 5 (best/avg)		k = 10 (best/avg)		k = 5 (best/avg)	
LR, RFC, DT, KNC, SVM	U	86.6	85.0	90.0	89.4	86.6	85.0	86.8	86.5
LR, RFC, DT, KNC, MNB	U	91.5	89.8	90.2	89.5	85.6	87.0	87.7	87.4
LR, RFC, DT, KNC, LDA	U	92.6	91.0	89.7	89.3	86.4	87.5	85.3	85.0
LR, RFC, DT, KNC, SVM	W	88.0	86.5	92.3	91.0	88.0	86.5	88.0	86.8
LR, RFC, DT, KNC, MNB	W	92.6	90.4	90.5	89.7	88.8	77.6	88.0	87.5
LR, RFC, DT, KNC, LDA	W	92.6	91.5	90.3	89.7	88.5	85.0	89.2	88.1

Table 6 5-fold DMLEC with repetition results

Layer 1	Layer 2	New York		Tokyo	
CC1	CC2	Best	Avg.	Best	Avg.
Unweighted	Unweighted	95.9	90.6	86.1	82.0
Unweighted	Weighted	96.1	90.8	89.5	86.0
Weighted	Unweighted	96.0	90.8	90.0	86.3
Weighted	Weighted	96.1	90.9	90.3	87.6

Table 7 5-fold DMLEC with non-repetition results

Layer-1	Layer-2	New York		Tokyo	
CC1	CC2	Best	Avg.	Best	Avg.
Unweighted	Unweighted	95.0	90.2	85.1	74.4
Unweighted	Weighted	96.0	91.0	86.4	79.2
Weighted	Unweighted	96.1	91.3	86.8	81.8
Weighted	Weighted	96.1	92.2	87.2	84.1

Table 8 10-fold DMLEC with repetition results

Layer 1	Layer 2	New York		Tokyo	
CC1	CC2	Best	Avg.	Best	Avg.
Unweighted	Unweighted	96.6	90.6	90.1	86.5
Unweighted	Weighted	97.0	91.1	91.4	88.1
Weighted	Unweighted	96.8	91.0	91.5	88.0
Weighted	Weighted	97.1	91.4	92.8	89.9

3.3.1 Single Classifier Results

Table 2 shows the results of classification accuracy obtained using the various machine learning classifiers on Foursquare New York and Tokyo datasets, for *k*-fold (*k* = 10, 5, 4) cross validation. The classifiers that are diverse and give the optimum results, are selected

Table 9 10-fold DMLEC with non-repetition results

Layer 1	Layer 2	New York		Tokyo	
CC1	CC2	Best	Avg.	Best	Avg.
Unweighted	Unweighted	75.4	66.1	64.4	56.9
Unweighted	Weighted	78.1	73.0	77.6	67.4
Weighted	Unweighted	86.1	76.6	74.7	64.7
Weighted	Weighted	90.8	82.1	88.7	81.5

Table 10 Comparison of best accuracy of proposed DMLEC model with state-of-the-art ensemble methods

Classifier-ensemble	New York	Tokyo
Meta-Adaboost	36.7	35.3
Meta-Bagging	68.5	76.1
Random Subspace	68.8	75.5
J48 Tree	64.6	75.5
Proposed DMLEC	97.1	92.8

for training the proposed DMLEC model. RFC and DTC give the best classification accuracy results among all the techniques.

3.3.2 Classifier Weight Assignment

Weights are assigned to the selected base classifiers based on their accuracy using Eq. 7 described in Sect. 2.3.4 (weight assignment method). The adjusted weights obtained for the best classifiers for New York and Tokyo datasets, are given in Table 3.

3.3.3 3-Ensemble Classifier Results

The classifiers selected for the ensemble, based on their diversity from others, include DTC, KNC, LDA, LR, MNB, SVM and RFC. For k-fold cross validation (k = 10, 5), Table 4 presents the best and average case of results generated from the combination of efficient classifiers selected dynamically using BPSO. For generating the results, un-weighted and weighted majority vote aggregator is used for classification. Weights are assigned to respecticve classifiers as per the function described in 2.3.4. It is observed that the weighted 3-ensemble classifiers results are better than the un-weighted ones. The ensemble of LR and RFC along with LDA, SVM or DTC is dynamically selected by the model, gives the best results.

3.3.4 5-Ensemble Classifier Results

The best and average case accuracy results of the 5-ensemble classifiers are depicted in Table 5. In case of 5-ensemble classifiers, best results are obtained using combination of LR, RFC, DT and KNC along with one classifier among SVM, LDA or MNB, selected dynamically by the proposed model. Results are obtained using the majority vote technique (un-weighted and weighted). Again, better results are obtained with the application of the proposed weight-assignment function as compared to the un-weighted combinations.

3.3.5 Proposed DMLEC Simulation Results

The proposed DMLEC model uses the combination of base classifiers divided into 2 layers. In the first layer, 3 classifiers are dynamically selected using BPSO for testing and in the second layer, 4 classifiers and the result from the previous layer i.e. output of Aggregator-1 are used. The classifier selection and placement is done with the help of the proposed encoding scheme, discussed in Sect. 2.3.3. The results are obtained after setting up the model in two ways, i.e. when classifiers in the layer 1 are allowed for repetition in layer 2 and when no repetition of classifiers is allowed. First, we show the results obtained using weighted and un-weighted classifiers on the 5-fold cross validation of the proposed DMLEC model for both cases (repetition and non-repetition), given in Tables 6 and 7. Similarly, we present the results obtained for both cases on 10-fold cross validation of proposed DMLEC model in Tables 8 and 9.

3.3.5.1 Results of 5-fold DMLEC with Repetition Table 6 gives the 5-fold classification accuracy results of the proposed DMLEC approach, when repetition of classifiers is allowed. The model dynamically chooses the best combination of classifiers possible, with RFC and DTC getting repeated in both the layers. Layer 1 classifier combination (CC_1) includes RFC, DTC and KNC and layer 2 (CC_2) includes LR, RFC, DTC, SVM and the result from the layer 1 (Aggregator-1). The best accuracy is obtained for the weighted New York datasets (96.1%). For the Tokyo datasets, we obtain a best accuracy of 90.3%.

3.3.5.2 Results of 5-fold DMLEC with Non-repetition 5-fold accuracy results for the proposed DMLEC model with no repetition of base classifiers is shown in Table 7. Classifier combination (CC_1) used in layer 1 is RFC, LDA and MNB, whereas LR, DTC, KNC, SVM and the output of layer 1 (Aggregator-1) constitute the classifier combination (CC_2) of second layer of the model. The highest classification accuracy obtained is 96.1% for the weighted classifiers, that use the proposed weight-assignment function for calculating final output.

3.3.5.3 Results of 10-fold DMLEC with Repetition The results of classification accuracy (10-fold) using the proposed DMLEC approach, when repetition of classifiers in the 2 layers is allowed dynamically, are given in Table 8. Classifiers combination in layer 1 (CC_1) includes RFC, DTC and KNC. Similarly, the classifiers combining in the layer 2 (CC_2) are LR, RFC, DTC, SVM and the result from the layer 1 (Aggregator-1). Here, RFC and DTC get repeated in both layers whereas other classifiers are used once. Important observations are that the best accuracy (97.1%) is achieved for the New York datasets. In addition, the results are best for weighted classifiers, as claimed.

3.3.5.4 Results of 10-fold DMLEC with Non-repetition In the following Table 9, accuracy results are shown when both layers of the proposed DMLEC model consist of different base classifiers. In the first layer, the classifier combination (CC_1) used is RFC, DTC and LDA whereas LR, MNB, KNC, SVM and the output of layer 1 (Aggregator-1) are dynamically used in layer 2 as the classifier combination (CC_2). 10-fold cross validation results are obtained for both weighted and un-weighted classifier ensembles, with the best being 90.8% for New York dataset, using the proposed weight-assignment method.

3.3.5.5 Comparative Analysis The proposed DMLEC model results are compared with the other state-of-the-art ensemble techniques, for classifying the given LBSN datasets in Table 10. The proposed approach gives much more improved classification accuracy results. It is due to the application of the proposed weight-assignment function used in the aggregators and the dynamically chosen set of base classifiers for calculating final output.

Overall, it is clear from the results that the efficiency of the proposed DMLEC model gives better classification accuracy as compared to other ensemble classifiers. In addition, we compared the results obtained from the proposed DMLEC model, with the results of other state-of-the art techniques including meta-bagging, meta-adaboost, random subspace and also J48-Tree. The results prove that proposed DMLEC model is outperforming other methods.

3.4 Discussion

The results prove that proposed DMLEC model outperforms various state-of-the-art methods. In addition, proposed DMLEC model gives much better accuracy for venue prediction (97.1%) for Foursquare LBSNs, when compared to the works of Gu et al. [20] and Almallah et al. [21] who obtain accuracy of 92.1% and 66%, respectively. The improved classification accuracy results are due to the efficient use of BPSO as a classifier selector and proposed weight-assignment function as classifier aggregator. Moreover, the proposed model allows repetition of best classifiers and uses weighted and un-weighted voting schemes dynamically and interactively.

Compared to single layered ensemble of classifiers, the proposed DMLEC approach is very flexible in dynamically selecting different combinations of base learning classifiers at different levels with varying weights. Efficient use of BPSO helps in placement of right classifiers at right places in the multi-layered architecture, which in turn helps to greatly improve the accuracy of classification. The ability of dynamically changing classifiers at various levels makes the approach scalable and applicable to broader domains of classification. The refining of results can be easily accomplished by varying the use of classifier combinations and the weights associated for different types of big datasets. In addition, compared to the flat architecture, the layered architecture produces much better classification accuracy results, as the chance of errors in the top-down classification procedure are diminished.

4 Conclusion and Future Works

Classification accuracy plays a vital role in the field of social network security. Extracting information from the social-user check-ins and analysing certain trends, can be helpful in finding solutions to various security-related problems. In this paper, BPSO is used for dynamic classifier selection task for choosing the best classifier in right place in the DMLEC model. A new classifier weight-assignment method is also proposed. The main motive of our work is to develop an efficient DMLEC framework for venue-classification using social-user check-ins in LBSN datasets. For the identification of appropriate classifiers in the proposed DMLEC model as base learners, a dynamic classifier selection system is implemented. Based on the accuracy of classification, the proposed model dynamically chooses the best classifiers among the given ones using the proposed encoding scheme and

BPSO. This helps to assign the right classifiers at the right place that gives the model the capability of giving the best results. The proposed model is experimented on Foursquare LBSN datasets. The experimental results are compared with state-of-the-art ensemble frameworks in the literature. The results show that the proposed approach outperformed state-of-the-art methods. The classification model can be very important for the security purposes in various governmental organizations, wherein locations of social-users can be traced efficiently to help nab criminals. Also, commercial benefits can be obtained by incorporating the model in business domains for classifying malicious users from the legitimate ones. The high classification accuracy can be very beneficial for tracking criminals and other security purposes.

Future works include extending the layers of the model in an efficient manner that further improves the classification accuracy while keeping the complexity lower. Also, classification results can be further improved by incorporating the unstructured social-user information like user-tips and ratings along with the temporal characteristics of social-users.

Acknowledgements This research work is funded by SERB, MHRD, under Grant [EEQ/-2016/000413] for Secure and Efficient Communication inside Partitioned Social Overlay Networks project, currently going on at National Institute of Technology Goa, Ponda, India.

References

1. Cao, X., Cong, G., & Jensen, C. S. (2010). Mining significant semantic locations from gps data. *Proceedings of the VLDB Endowment*, *3*(1–2), 1009–1020.
2. Foursquare Labs: Foursquare checkins. (2017). http://www.swarmapp.com.
3. Yang, J., & Olafsson, S. (2006). Optimization-based feature selection with adaptive instance sampling. *Computers & Operations Research*, *33*(11), 3088–3106.
4. Valentini, G. & Masulli, F. (2002). Ensembles of learning machines. In *Italian workshop on neural nets*, (pp. 3–20). Springer.
5. Zolfaghar, K., Verbiest, N., Agarwal, J., Meadem, N., Chin, S.-C., Roy, S. B., Teredesai, A. et al. (2013). Predicting risk-of-readmission for congestive heart failure patients: A multi-layer approach. arXiv preprint arXiv:1306.2094.
6. Virrantaus, K., Markkula, J., Garmash, A., Terziyan, V., Veijalainen, J., Katanosov, A., & Tirri, H. (2001). Developing GIS-supported location-based services. In *Proceedings of the 2nd international conference on web information systems engineering, 2001* (Vol. 2, pp. 66–75). IEEE.
7. Emersion, B. J. (2011). Using location-based services to get customers. *Franchising World*, *43*(7), 9.
8. Shugan, S. M. (2004). The impact of advancing technology on marketing and academic research. *Informs, 1*, 469–475.
9. CNET: Foursquare gives out unsolicited tips on iPhone. (2013). http://news.cnet.com/8301-10233-57606718-93/foursquare-gives-outunsolicitedtips-on-iphone.
10. Cramer, H., Rost, M., & Holmquist, L. E. (2011). Performing a check-in: Emerging practices, norms and 'conflicts' in location-sharing using foursquare. In *Proceedings of the 13th international conference on human computer interaction with mobile devices and services* (pp. 57–66). ACM.
11. Wang, D., Pedreschi, D., Song, C., Giannotti, F., & Barabasi, A.-L. (2011). Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1100–1108). ACM.
12. Chorley, M. J., Whitaker, R. M., & Allen, S. M. (2015). Personality and location-based social networks. *Computers in Human Behavior*, *46*, 45–56.
13. Lindqvist, J., Cranshaw, J., Wiese, J., Hong, J., & Zimmerman, J. (2011). I'm the mayor of my house: Examining why people use foursquare-a social-driven location sharing application. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 2409–2418). ACM.

14. Schwartz, R., & Halegoua, G. R. (2015). The spatial self: Location-based identity performance on social media. *New Media & Society*, *17*(10), 1643–1660.
15. Noulas, A., Scellato, S., Mascolo, C., & Pontil, M. (2011). An empirical study of geographic user activity patterns in foursquare. *ICwSM*, *11*, 70–573.
16. Getoor, L., & Diehl, C. P. (2005). Link mining: A survey. *ACM SIGKDD Explorations Newsletter*, *7*(2), 3–12.
17. Goswami, A., & Kumar, A. (2017). Challenges in the analysis of online social networks: A data collection tool perspective. *Wireless Personal Communications*, *97*(3), 4015–4061.
18. Bliss, C. A., Frank, M. R., Danforth, C. M., & Dodds, P. S. (2014). An evolutionary algorithm approach to link prediction in dynamic social networks. *Journal of Computational Science*, *5*(5), 750–764.
19. Torabi, N., Shakibian, H., & Charkari, N. M. (2016). An ensemble classifier for link prediction in location based social network. In *Proceedings of the 24th Iranian conference on electrical engineering (ICEE)* (pp. 529–532). IEEE.
20. Gu, Y., Yao, Y., Liu, W., & Song, J. (2016). We know where you are: Home location identification in location-based social networks. In *Proceedings of the 25th international conference on computer communication and networks (ICCCN)* (pp. 1–9). IEEE.
21. Almallah, O. F., & Albayrak, S. (2017). Predicting venues in location based social network. In *Proceedings of the 7th international conference on computer science, engineering and applications (CCSEA)* (pp. 11–21). CSIT.
22. Tian-ran, H., Luo, J., Kautz, H., & Sadilek, A. (2016). Home location inference from sparse and noisy data: models and applications. *Frontiers of Information Technology & Electronic Engineering*, *17*(5), 389–402.
23. Cho, E., Myers, S.A., & Leskovec, J. (2011). Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1082–1090). ACM.
24. Parvin, H., MirnabiBaboli, M., & Alinejad-Rokny, H. (2015). Proposing a classifier ensemble framework based on classifier selection and decision tree. *Engineering Applications of Artificial Intelligence*, *37*, 34–42.
25. Ghasemi, E., Kalhori, H., & Bagherpour, R. (2017). Stability assessment of hard rock pillars using two intelligent classification techniques: A comparative study. *Tunnelling and Underground Space Technology*, *68*, 32–37.
26. Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, *160*, 3–24.
27. Yeh, I.-C., & Lien, C. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, *36*(2), 2473–2480.
28. Ibarguren, I., M Pérez, J., Muguerza, J., Gurrutxaga, I., & Arbelaitz, O. (2015). Coverage-based resampling: Building robust consolidated decision trees. *Knowledge-Based Systems*, *79*, 51–67.
29. Datta, S., & Das, S. (2015). Near-Bayesian support vector machines for imbalanced data classification with equal or unequal misclassification costs. *Neural Networks*, *70*, 39–52.
30. Li, S., Zong, C., Wang, X. (2007). Sentiment classification through combining classifiers with multiple feature sets. In *International conference on natural language processing and knowledge engineering, NLP-KE 2007* (pp. 135–140). IEEE.
31. Dietterich, T. G., et al. (2000). Ensemble methods in machine learning. *Multiple classifier systems*, *1857*, 1–15.
32. Kuncheva, L. I. (2004). *Combining pattern classifiers: Methods and algorithms*. New York: Wiley.
33. Orrite, C., Rodríguez, M., Martínez, F., & Fairhurst, M. (2008). Classifier ensemble generation for the majority vote rule. In *Iberoamerican congress on pattern recognition* (pp. 340–347). Springer.
34. Wang, G., Hao, J., Ma, J., & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, *38*(1), 223–230.
35. Marqués, A. I., García, V., & Sánchez, J. S. (2012). Exploring the behaviour of base classifiers in credit scoring ensembles. *Expert Systems with Applications*, *39*(11), 10244–10250.
36. Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the 6th international symposium on micro machine and human science, MHS'95* (pp. 39–43). IEEE.

37. Eberhart, R. C., & Kennedy, J. (1995). Particle swarm optimization. In *Proceeding of IEEE international conference on neural network, Perth, Australia* (pp. 1942–1948). IEEE.
38. Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844.
39. Luh, G.-C., Lin, C.-Y., & Lin, Y.-S. (2011). A binary particle swarm optimization for continuum structural topology optimization. *Applied Soft Computing*, *11*(2), 2833–2844.
40. Pal, A., & Maiti, J. (2010). Development of a hybrid methodology for dimensionality reduction in mahalanobis-taguchi system using mahalanobis distance and binary particle swarm optimization. *Expert Systems with Applications*, *37*(2), 1286–1293.
41. Zeng, X.-P., Li, Y.-M., & Qin, J. (2009). A dynamic chain-like agent genetic algorithm for global numerical optimization and feature selection. *Neurocomputing*, *72*(4), 1214–1228.
42. Tasgetiren, M. F., Suganthan, P. N., & Pan, Q.-Q. (2007). A discrete particle swarm optimization algorithm for the generalized traveling salesman problem. In *Proceedings of the 9th annual conference on genetic and evolutionary computation* (pp. 158–167). ACM.
43. Mirjalili, S. A., & Hashim, S. Z. M. (2012). BMOA: Binary magnetic optimization algorithm. *International Journal of Machine Learning and Computing*, *2*(3), 204.
44. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2010). BGSA: Binary gravitational search algorithm. *Natural Computing*, *9*(3), 727–745.
45. Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, *9*, 1–14.
46. Yang, D., Zhang, D., & Bingqing, Q. (2016). Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *7*(3), 30.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ahsan Hussain is a Research Scholar in the Department of Computer Science and Engineering, National Institute of Technology Goa, India. His research interests are Data Mining, Computer Networks and Social Media Mining.

Bettahally N. Keshavamurthy Keshavamurthy is currently working as an Assistant Professor in the Department of Computer Science and Engineering, National Institute of Technology Goa, India. His research interests include Data Mining, Stream Data Mining, Privacy Preserving Data Mining and Social Media Mining.

Ramalingaswamy Cheruku is presently working as full-time research scholar in department of Computer Science and Engineering at National Institute of Technology Goa. He received B. Tech. Degree in Computer Science and Engineering from JNT University, Kakinada campus in 2008, M. Tech. Degree in Computer Science and Engineering from ABV-Indian Institute of Information Technology, Gwalior in 2011. He has served as developer in Tata Consultancy Services for 2 years. He has also published several papers in reputed journals and conferences.