Cluster Computing (2022) 25:523–536
https://doi.org/10.1007/s10586-021-03403-3

Adaptive learning style prediction in e-learning environment using levy flight distribution based CNN model

Sami Alshmrany[1]

Received: 21 December 2020 / Revised: 9 August 2021 / Accepted: 28 August 2021 / Published online: 30 September 2021

Abstract
A learning style that focuses on individual learning is one of the most important aspects of any learning environment. Each learner has a unique manner of understanding, retaining, processing, and interpreting new information based on their learning styles. The ability of an e-learning system to automatically determine a student's learning style has become more essential. For learning events, the evolution of e-learning platforms provides students with higher opportunities online. In this paper, we proposed a Convolutional Neural Network-based Levy Flight Distribution (CNN-LFD) algorithm for learning style prediction. An adaptive e-learning system is divided into two sections: automatic learning style prediction and classification based on the number of learning styles included. Initially, the student logs in with their user ID, and the data is saved in the database. The features such as questionnaire score, login credentials (session ID, learner ID, and course ID), and login time (location, session ID) are extracted along with the sequence of the user's learning behavior. After that, the CNN-LFD algorithm predicts the learning styles of the learners namely Active/reflective, Sensing/intuitive, visual/verbal, sequential/global based on the extracted features. The dataset information are gathered from a Massive Open Online Course (MOOC), and the proposed model is built in JAVA software. The experimental results demonstrate higher classification accuracy during learning style prediction. The proposed CNN-LFD algorithm accomplishes 97.09% accuracy, 94.76% specificity, 92.12% sensitivity, and 97.56%, precision values than other methods.

Keywords E-learning · Learning style prediction · Felder-Silverman Learning Style Model · Convolutional neural network · Levy flight distribution algorithm

1 Introduction

In order to achieve the education goal for everyone, e-learning is a new area that aims to augment conventional learning approaches using technology. The use of internet technology adopts the concept of e-learning to boost the learning of students every time. Chen et al. [1] defined the convenient learning methodology named E-Learning which denotes the knowledge and information delivery to all, anywhere, and anytime to minimize the cost, effort, and time. A five-fold target is set for the continuous evaluation of the learner's progress in e-learning, intelligent tutoring framework, knowledge establishment, personalized learning, and versatility. The achievement or degradation of the e-learning system [2] is determined by factors including the influence of learning styles, performance assessment, knowledge management, related information processing, distribution of learning objects, and learning objects.

The process of acquiring skills, values, and knowledge via teaching, experience, and learning is called learning. The learners spend extra time on information filtering and browsing that suits their requirements better in case of spending time on learning the materials and the domain of interests because of more amount of information available on the Net [3]. Nevertheless, the e-learning scenario is entirely varied because of improved self-efficacy, effective motivation, constant progress monitoring, body language understanding, and nullified face-to-face interactions. According to several attributes, the two significant factors such as static and dynamic factors are differed in learning

✉ Sami Alshmrany
s.alshmrany@iu.edu.sa

[1] Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia

styles. Further, the learning style models and learning theory are concurrent depending upon the dimensions [4]. Accurate e-contents are recommended for learners on the e-learning server, and the defined techniques should be utilized to determine individual learning styles [5–7].

Several applications and strategies for predicting learners' e-learning styles have been found in recent decades, including profile information disclosure, interviews, and questionnaires. The traditional classroom setting environments are identified with the help of these techniques [8–19]. In addition, each learner is compatible with their skills and interests. The learning style characterization is a major issue regarding various outputs and descriptors [20, 21]. Particularly, the learning objects are linked with the descriptors in terms of self-assignments, quizzes, outlines, contents, and forums. Previously, the most prevalent difficulties in learning style classification were several outputs with multiple descriptors. These descriptors might occur from databases, questionnaires, and logs. The learning styles such as self-assignments, quizzes, outlines, contents, and forums are associated with the descriptors. The results define whether the student is categorized into sequence/global, visual/verbal, sensing/intuitional, and active/reflective based on the current method, procedure, and stored information. In this paper, we proposed a Convolutional Neural Network-based Levy Flight Distribution (CNN-LFD) algorithm for appropriate learning style prediction. The major contribution of this paper is summarized as below:

- The development of the Felder Silverman Learning Styles Model (FSLSM) begins with the collection of learning style models, analyzing data sources, learning style attributes, and classification method selection.
- The LFD algorithm is used in this paper to tune the different hyperparameters associated with CNN such as the number of epochs, learning rate, etc. and this process increases the accuracy of the CNN classifier and prevents it from overfitting.
- The user learning behavior is formed into a sequence of learning objects and fed into the CNN-LFD classifier to predict the appropriate learning style for the user.
- Four different learning styles are predicted by the CNN-LFD classifier namely Active/reflective, Sensing/intuitive, visual/verbal, and sequential/global.
- The efficiency of the proposed CNN-LFD classifier is evaluated in terms of accuracy, specificity, sensitivity, computation time, and confusion matrix by comparing it with different state-of-art techniques.

The rest of the paper is organized as follows: Sect. 2 explains the related work and Sect. 3 describes the proposed CNN-LFD algorithms for the prediction of learning styles. The experimental study with state-of-the-art comparison is illustrated in Sect. 4. The paper is eventually concluded in Sect. 5.

2 Review of related works

For the classification of learning styles, Azzi et al. [22] introduced a robust classifier. The modified feature of a certain subject indicates the learners' learning behavior. Further, the learners' behaviors are captured through web usage mining and the class of Felder-Silverman Learning Style Model (FSLSM) maps the learning styles. With the assistance of the Fuzzy C Means (FCM) algorithm, the captured learning behavioral data is clustered into the FSLSM classes. Convergence time differences are also addressed, and the accuracy of the two tests is used to evaluate the clustering algorithm's efficiency. According to a given learning style model, learning styles were discovered automatically using a general approach through web mining algorithms, as provided by El Aissaoui et al. [23]. Learning sequences are separated from learners' log files employing web usage mining techniques. The extracted learners' sequences are then classified using a clustering technique and a specific learning style model. The use of real-world data was used to test positive outcomes. The data uncertainty is effectively managed via the FCM algorithm; nevertheless, the learning style detection accuracy is poor. Deborah et al.[24] also have used the Gaussian membership function-based fuzzy logic to predict the learning styles of diverse learners. However, the fuzzy logic is mainly based on assumptions and the accuracy obtained is not always acceptable.

The fuzzy decision tree was suggested by Crockett et al. [25] to construct the predictive model sequences. During tutoring, the Conversational Intelligent Tutoring System (CITS) named OSCAR uses Natural Language Processing(NLP) to simulate a person's learning style. The fuzzy models have shown the outputs via live data that enhance OSCAR-CITS prediction accuracy. With impersonalized learning experiences, the experimental results showed greater prediction accuracy and superior adaption decisions. The novel recommendation model was proposed by Madani et al. [26] to suggest the exact courses to learners. To develop an effective course, the student should suggest and understand this recommendation approach, which combines collaborative and social filtering. Further, the learning quality improvements, as well as optimal learning path findings, are carried out via novel reinforcement learning. The filtering technique of social collaboration leverages the social content of learners in terms of Facebook and Twitter comments and likes. When there is a cold start, sparsity, or scaling issue, the recommendation quality improves, but the error-susceptibility increases significantly.

The machine learning algorithms with Felder and Silverman learning style model (FSLSM) were proposed by El Aissaoui et al. [27] to automatically detect the learning styles. According to the FSLSM, an unsupervised algorithm such as K-means is used to extract the sequences of learners from the log file. The learning styles of new learners are predicted with the usage of supervised algorithms such as Naive Bayes. The confusion matrix technique was evaluated the performance of FSLSM. The classification performance based on naive Bayes is lower with higher computational complexity.

In cloud computing, an efficient personalized trust-based hybrid recommendation (TBHR) mechanism was introduced by Bhaskaran et al. [28] for the e-learning system. The learners' learning styles are tested to anticipate their behavior and different patterns of learning styles. The noise is removed by the re-rating method, and the K-means with hybrid fireflies clustered the learners according to their learning styles. The performance measures such as speed, accuracy, and mean absolute error (MAE) with comparison results yielded better performances in terms of the TBHR mechanism. The set of tags is skipped so that the expertise level of the student and the interest of the student in various fields are recognized. Wafaa Sayed et al.[29] designed a personalized e-learning platform for school students. Their proposed system is developed using VARK, learning styles, and Bloom taxonomy. Using the moodle learning management system they have designed their e-learning system. For adaptation, an Artificial Intelligence Module is developed using multitask deep Q-learning and trained using ε-greedy policy.

Fareeha Rasheed et al. [30] presented an e-learning system using machine learning to predict the learning styles. The main aim of their work is to personalize the users learning experiences by identifying the attributes of different learners and scaling down the attributes identified earlier. Rajkumar et al. [31] presented a bio-inspired learning style chatbot using a brain computing interface to enhance the efficiency of the e-Learning technique. The main aim of the work is to identify the interrelationship between the introvert and extrovert personality types and learning styles selected. The Visual, Auditory, Read and write, and Kinesthetic(VARK) questionnaire is deployed into the chatbot to identify the behavior of the introvert and the extrovert learners. The techniques used are namely Naïve Bayes, N-48, and Canopy.

Based on the Massive online open course (MOOC) platform, Hmedna et al.[32] proposed a predictive model for learning styles identification. This study adopts the most commonly used models of the Felder Silverman learning style model (FSLSM). For all three dimensions, the decision tree performed better results and higher accuracy thereby minimizing overfitting risk. According to the Bi-LSTM-CRF neural network, the profile attributes extraction (PAE-NN) model was proposed by Lin et al. [33]. The end-to-end training and Recurrent Neural Network extract the contextual representations and characteristics. The long-term dependencies on extracting entities are discovered via the characteristics of the long-memory series LSTM network. The PAE-NN model outperformed the existing models in terms of F1-score, recall, and precision. Zhang et al. [34] proposed a study on learning effect prediction models in MOOCs based on principal component analysis. The nine measurable performance indexes are analyzed because of the correlation between many behavior indexes. The comprehensibility is increased and also the computational dimension was minimized. The establishment of predictive models enhances the passing rate of the course thereby providing better prediction results.

3 Proposed approach

The incorporation of learning styles into an adaptive e-learning system is divided into two major areas, such as the application of this developed model to an adaptive e-learning system and the creation of online data for the prediction model of learning styles or the classification of online learning styles. For example, the Felder Silverman Learning Styles Model (FSLSM) development starts with learning style selection models thereby evaluating sources of data, attributes of learning styles, and the selection of classification methods. For a particular factor of the adaptive learning system, the relevant classification models with their results are accepted. The proposed workflow diagram is depicted in Fig. 1.

First, collect and analyze the dataset using the computational intelligence algorithm. In this experiment, the behavior of the student is obtained from the Learning Management System (LMS). Examples, navigation, questions, forums, quizzes, exercises, self-assessment, outlines, and content are used as learning objects. Table 1 explains the collected behavior. 150 students who graduated in computer science and registered the program of postgraduate in project management and computer sciences with 20 descriptors are obtained from the literature[35]. These descriptors were grouped into nine learning objects, according to the LMS course. The Boolean, count, and time are the important measures of datasets and the explanations of each measure are explained as below:

Boolean   Students' outcomes while responding to questions on a quiz.

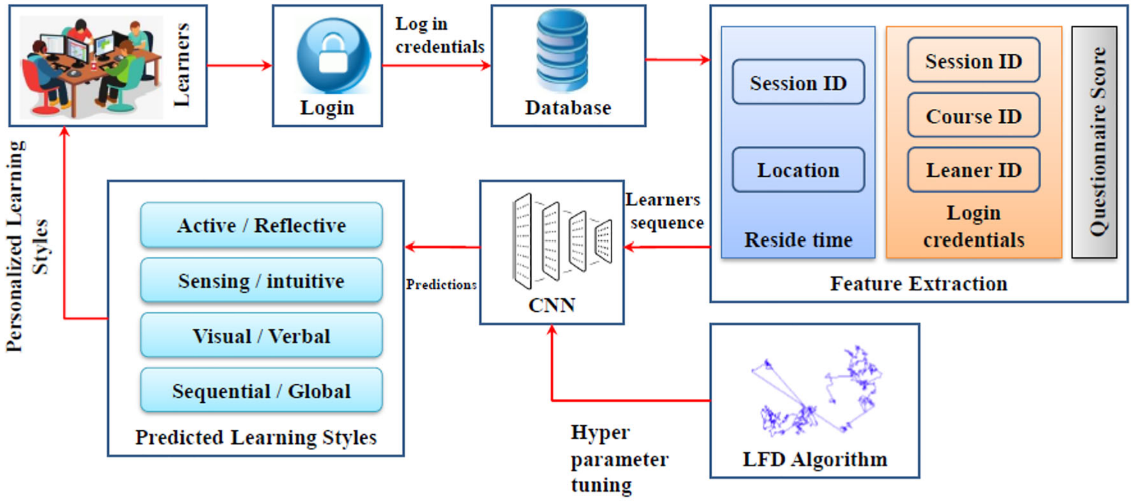

Fig. 1 Proposed workflow diagram

Table 1 Massive open online course (MOOC) [48] based behavior description

Learning point	Behavior	Assessment	Explanation
Example	example_stay	Time	Time spent at examples
Navigation	navigation_overview_visit	Time	Time taken for navigation
	navigation_overview_stay	Count	Number of time taken to visit the navigation
	navigation_skip	Count	Number of time taken to skip navigation
Questions	ques_interpret	Time	Time taken to understand the questions
	ques_overview	Time	Time spent at overview questions
	ques_text	Time	Time spent at questions from type text
	ques_graphics	Time	Time spent at questions from type graphics
	ques_develop	Time	Time spent at questions from type text
	ques_concepts	Time	Time spent at questions from type concepts
	ques_facts	Time	Time spent at questions from type facts
	ques_detail	Time	Time spent at question details
Forum	forum_stay	Time	Time spent at the forum
	forum_post	Count	Amount of times the student posts a forum message
	forum_visit	Count	Sum of the visit made by the student to the forum
Quiz	quiz_revisions	Count	Number of times the student takes the quiz again
	quiz_stay_results	Time	Time taken to achieve the quiz result
Exercise	exercise_stay	Time	Time spent to implement it
	exercise_visit	Count	Number of visits to an exercise by students
Self-assessment	selfass_two_wrong	Boolean	If the student checks for self-evaluation
	selfass_stay	Time	Time for self-evaluation
	selfass_twice_wrong	Count	Amount of visits to self-evaluation by students
Outline	outline_visit	Count	Amount of visits by students to an outline
	outline_stay	Time	Time spent on the outline
Content	content _visit	Time	Spending time on content
	content _stay	Count	Number of times a student views a material

Time The time that the students spent studying the artifacts.

Count How many times students visit a learning object.

Students' log average represents each descriptor and these records were received for 15 days to recapitulate students' results. The questions on the self-assessment quizzes are classified based on the overview knowledge which includes analyzing solutions, address building, charts or texts, and graphics with concepts or facts.

3.1 Feature extraction

In this work, we are extracting the features such as questionnaire score, login credentials (session ID, learner ID, and course ID), and login time (location, session ID). The finding of a resource or session ID is a straightforward technique performed within the administrator interface with several reports. It does not have the capability of displaying the lesson ID for lessons that are included in a course. The following technique is used for obtaining questionnaire scores.

The essential characteristics that represent the user's behavior are user product experience and extensive knowledge. The knowledge in a particular domain and awareness indicators are sampled with questionnaires and user discussions. The corresponding assessment questionnaire is then derived from the data [36]. The set of keywords is equivalent to the user's questionnaire. The learner selects the terms because of target domain clarification and important business value [27, 37]. The relevance questionnaire score is measured by using the numerical terms tf-idf or frequency-inverse document frequency that is the collection of learner keywords. Equation (1) calculates the $tf\!-\!idf$ scores (q_k, P).

$$\text{Equivalent questionnaire} = \text{Score}(q_k, P) = \sum_{(k) \in q_k, P} tf(k, P) \cdot \ln \frac{|R|}{df(k, R)}, \tag{1}$$

where the selected leaner is represented as q_k and the questionnaire collection size is $|R|$. The questionnaire number in R is a questionnaire that contains the term k, and the number of times k appears in the questionnaire is denoted as $tf(k, R)$. Equation (2) calculates the equal questionnaire score (q_k, R_M).

$$\text{Questionnaire Score} = \frac{\sum_{P \in R_M} (\text{Score}(q_k, R))}{|R_M|}. \tag{2}$$

3.2 Learning style prediction

During learning style prediction, we discuss the parameter tuning operation of the Convolutional Neural Network (CNN) using Levy Flight Distribution (LFD). The next section provides a detailed explanation of CNN-based LFD and how it can be used to predict learning styles using the CNN-LFD algorithm.

3.2.1 CNN overview

One of the important subtypes of Artificial Neural Network (ANN) is a Convolutional Neural Network (CNN) for grid-like structures such as an image with data processing[38]. The underlying mechanism of CNN is inspired by the visual cortex of an animal's organization, which is compressed with one or more fully connected convolutionary layers and pooling layers. The key concepts such as local communication, weight sharing, and sub-sampling present in CNN effectively learn substantial patterns from the input data [39]. The neurons of the lower layer are connected to the local receptive field of the higher layer which is known as the local connectivity of CNN. The convolution kernels move over an entire input data thereby the sparse interaction with the computational complexity is minimized.

CNN consists of biases and weights like other ANN models, which are updated in the learning process. Equivalent weight and biases are applied to relevant layers according to CNN that determines the similar features in various fields. For the learning process, the spatial connections among the neurons are essential to minimize the number of parameters. The standard back-propagation algorithm improves performance. After each layer separation, the sub-sampling denotes the sample extraction process. Further, overfitting and computational complexity are minimized by reducing the number of features. The CNN layers are explained in the following sub-section.

(i) *Convolutional layer* The central building block of a CNN is the convolutional layer. The parameters of the layer consist of a series of learnable filters (or kernels) that have a limited receptive field but extend through the maximum input volume depth. When the input data is transferred at a regular interval, the kernel or filter extracts the features in the convolution layer. The key role in CNN is the convolutional layer with the list of mathematical operations including linear and convolution operations [40]. The kernel is fed to the weight matrix of the conventional ANN. The input data convolves the trainable kernel and establishes the feature map in the convolutional layer at all possible offsets. Convolutional layer training kernels must be utilized in all convolutional layers, according to CNN. The kernels of all convolutional layer training are the basic requirement of CNN.

(ii) *Pooling layer* The most relevant samples in the pooling layer are derived from the convolution layer. Average pooling and max-pooling are sampling techniques used. Sampling is used to obtain each interval of the average or maximum value [41]. For the learning process, the essential information is extracted and the pooling layer rejects the unwanted information details. When the noisy data influence is minimized, the robustness improves.

(iii) *Fully connected layer* The neurons created in the convolution layer are connected to each higher layer of neurons via a fully connected layer and a pooling layer. The fully connected layers provide classification, recognition, or prediction results.

3.2.2 Formulation of the Levy Flight Distribution (LFD) algorithm

In this section, we discuss the parameter tuning operation of the Convolutional Neural Network (CNN) using Levy Flight Distribution (LFD). The detailed explanation of CNN-based LFD is delineated in the below section.

(i) *Levy Flight Distribution* Levy Flight Distribution (LFD) combines Levy Flight (LF) and an e-learning environment for learners [41]. The LF enhances resource search efficiency in uncertain environments. Several physical inspired and natural inspired phenomena in the environment inspire the Levy Flight. For the LF inspiration, the fluorescent molecule diffusions are inspired by physically inspired phenomena. Cooling and noise behavior show the LF properties under different light conditions. While comparing to Brownian random walks, the LFs are more effective in exploring unknown large search spaces [42].

(ii) *Mathematical representation* The e-learning environment is the simulation environment for mathematical modeling. From this, each of the two learners' positions with their corresponding Euclidean distance (ED) is calculated. The levy flight algorithm determines whether the learner remains in their original position or advances to the next position based on the calculated ED. Implementing LFs requires two fundamental characteristics: step length of the walk and uniform distribution direction. Here, the step length L is calculated according to Mantegna's algorithm [43]. Equation (3) explains the step length L.

$$L = \frac{V}{|U|^{1/\alpha}}. \tag{3}$$

From Eq. (1), the levy distribution α tends to the interval $0 < \alpha \leq 2$. Where, V and U is given as below:

$$V \sim M(0, \sigma_V^2),\ U \sim M(0, \sigma_U^2). \tag{4}$$

Equation (5) defines the standard deviations σ_V and σ_U.

$$\sigma_V = \left\{ \frac{\Gamma(1+\alpha) \times \sin(\pi\alpha/2)}{\Gamma[(1+\alpha)/2] \times \alpha \times 2^{(\alpha-1)/\alpha}} \right\}, \tag{5}$$

$$\sigma_U = 1. \tag{6}$$

For an integer y, the gamma function is denoted using Eq. (7).

$$\Gamma(y) = \int_0^\infty k^{y-1} e^{-k} dk. \tag{7}$$

The ED among the initial two adjacent agents such as Y_m and Y_n are calculated using the LFD algorithm.

$$ED(Y_m, Y_n) = \sqrt{(Y_m - Y_n)^2 + (Z_m - Z_n)^2}. \tag{8}$$

Here, the position coordinate of Y_m is (Y_m, Z_m) and the position coordinate of Y_n is (Y_n, Z_n). A comparison is then made between the specific number of iterations, the given threshold, and the Euclidian distance. When the resultant distance is smaller than the threshold, the algorithm begins by modifying the agent's positions using Eq. (9).

$$Y_n(k+1) = LF(Y_m(k), Y_{\text{Leader}}, U_B, L_B). \tag{9}$$

The index for the number of iterations is k. The LF function executes the direction and step length of levy flights. According to the 2D search space dimensions, the lowest and highest boundary values are denoted as L_B and U_B. The agent position is Y_{Leader}. Using Eq. (10), the agent Y_n moves towards the agent position based on the least number of neighbors.

$$Y_n(k+1) = L_B + (U_B - L_B)\text{random}(). \tag{10}$$

The random integer generates the uniform distribution interval 0 to 1. The exploration stage of LFD is improved using Eq. (10). If there are no other agents, it updates Y_n the position to a new search space region.

$$R = \text{random}(), C_{\text{om}}SV = 0.5. \tag{11}$$

Hence, each update for the position Y_n is R while the comparative scalar value is $C_{\text{om}}SV$. Each iteration validates the R-value in Eq. (11) then updates the sensor node position Y_n. By modifying the algorithm solutions, the exploration ability is enhanced with improved performance.

$$Y_m(k+1) = T_P + b_1 \times T_{F_{\text{neighbours}}} + \text{random}() \times b_2 \times (T_P + b_3 Y_{\text{Leader}})/2 - Y_m(t)). \tag{12}$$

$$Y_m^{\text{New}}(k+1) = LF(Y_m(k+1), T_P, U_B, L_B). \tag{13}$$

Equation (12) calculates the new position Y_n and Eq. (13) computes the final position Y_n. The excellent fitness value is achieved using the solution position T_P, which is termed as the target location. Where the random numbers become $0 < \beta_1, \beta_2, \beta_3 \le 10$. Where $Y_m(k)$ around the entire neighbor target fitness is $T_{F_{\text{neighbours}}}$.

$$T_{F_{\text{neighbours}}} = \sum_{j=1}^{MM} \frac{E(j) \times Y_j}{MM}. \tag{14}$$

According to Eq. (14), the neighbor position of $Y_m(k)$ is Y_j and the neighboring index is j. The whole number $Y_m(k)$ is MM, and each neighbor's fitness level is $E(j)$.

$$E(j) = \frac{\partial_1 (U - \text{Min}(U))}{\text{Max}(U) - \text{Min}(U)} + \partial_2, \tag{15}$$

$$U = \frac{F_{it}(Y_m(k))}{F_{it}(Y_m(k)}, 0 < \partial_1 \text{ and } \partial_2 \le 1. \tag{16}$$

In the levy flight distribution algorithm, the above-mentioned equations are repeated for each iteration. The LFD algorithm features a fast convergence time, a good balance of exploitation and exploration, better exploration and exploitation abilities, and the ability to escape local optima.

3.2.3 CNN-LFD for learning style prediction

To form an adaptive e-learning system we are integrating the different learning styles for personalization. In this way, each learner is taught in a style they prefer. The efficiency of a robust e-learning system is based on its accuracy. To overcome this problem, in this article, we have proposed a novel Convolutional Neural Network-based Levy Flight Distribution (CNN-LFD) algorithm for approximate learning style prediction. The CNN-LFD offers higher accuracy since it is trained with adequate data and based on this data the learning style is predicted. Initially, the courses for a specific subject are identified in the first phase, and later using the acquired data, we can find learning styles related to the courses to construct a sequence for the learning style prediction process. These sequences help to predict the learning style of the learner.

The CNN is an advanced extension of the standard deep-learning algorithm. Nevertheless, the CNN can offer increased performance when their parameters are optimized and consumes less time for decision-making. CNN can handle various structures with complex data. However, when it comes to predicting learning styles, the conventional CNN has encountered some challenges such as poor classification performance, overfitting concerns, and Coordinate Frame. If the settings of CNN designs are optimized with a specific problem in mind, they can provide better results. As a result, we employed the LFD algorithm to tune the CNN's hyperparameters, resulting in faster convergence, exploration, and exploitation [44–46]. The features are extracted for learning style prediction such as residence time, credentials, questionnaire scores, etc., and are fed as the input to the CNN to predict the exceptional learning styles. CNN is trained with the help of the LFD algorithm by utilizing effective levy distribution based on levy flight (LF) and e-learning environments.

To reduce the overfitting problem to some extent, the backpropagation algorithm is used and CNN is trained accordingly. The main objective is to increase the learning style prediction efficiency. The optimal parameters are defined as the kernel during training. Smaller numbers of arrays such as kernels are applied across the feature extracted inputs. The kernels and weights in CNN are effectively optimized with the help of the CNN-LFD algorithm in which the steps involved in the LFD algorithm are explained in Sect. 3.2.2.

For each subject, the user exhibits a learning behavior that demonstrates whether he is drawn to or distracted by the course. These behaviors are taken and converted into learning sequences. The different learning components/courses selected by the learner helps to identify his learning sequence from which his learning style can be easily predicted. To identify a learning type, different learning behaviors for a specific subject are identified and it is transformed into a learning sequence. The various learning concepts are thought to be related to different courses that deal with the same subject.

The sequences also describe the time spent by the learner to learn each course and it comprises user_id, course_id, sequence_id, and the sequence of learning objects (LO). The sequence estimates are not directly converted into LO and for every course, a specific sequence is designed. To accomplish this objective, m-tuples of the session are designed such as $(S_{C1}, S_{C2}, \ldots, S_{Cm})$, where S_{C1} is the course in each sequence and m is the total number of sequences. The next parameter to identify the sequence is the minimum amount of time spent by the learner in a LO which is represented as ATS.

The sequence that is given as the input to the CNN-LFD classifier is sequence_id, m-tuple_sessionid, ATS, and a sequence of LO. The m-tuple session-id for different sessions associated with every course is described as

$(S_{C1}id, S_{C2}id, \ldots, S_{Cm}id)$. The average time spent by the user for each LO is computed as shown below:

$$ATS = \sum_{j=1}^{m} \frac{AC}{m}, \tag{17}$$

$$AC = \sum_{j=1}^{n} \frac{AT}{n}, \tag{18}$$

$$AT = \sum_{j=1}^{K} \frac{LTS_{LO}}{K}, \tag{19}$$

where k is the number of LO for each subject theme, m is the total number of courses selected by the learner, n is the subject theme for each course selected by the learner, AC is the minimum time spent for each course, AT is the minimum time spent for each theme, and LTS_{LO} is the time spent by the learner on a single LO. Finally, the CNN-LFD algorithm classifies the learning styles such as Active/reflective, Sensing/intuitive, visual/verbal, sequential/global. The model of CNN-LFD for learning style prediction is illustrated in Fig. 2.

4 Experimental results and discussions

The performance of the proposed CNN-LFD for learning style prediction is investigated in this section. The proposed CNN-LFD for learning style prediction is evaluated on a desktop computer with an Intel® Core 3.91 GHz CPU and 8 GB of RAM, and the algorithms are implemented in Java. The parameter settings of the proposed CNN-LFD algorithm for learning style prediction are tabulated in Table 2.

The experimental data were acquired from the University of Stanford's 'Statistical Learning' course. In general, the datasets are provided by the Centre for Advanced Research via Online Learning (CAROL), ensuring that the learner's privacy is safeguarded. The dataset details used in this work are summarized in Table 3. From this, 70% of the data was used for training, and the remaining 30% of the data was used for validation.

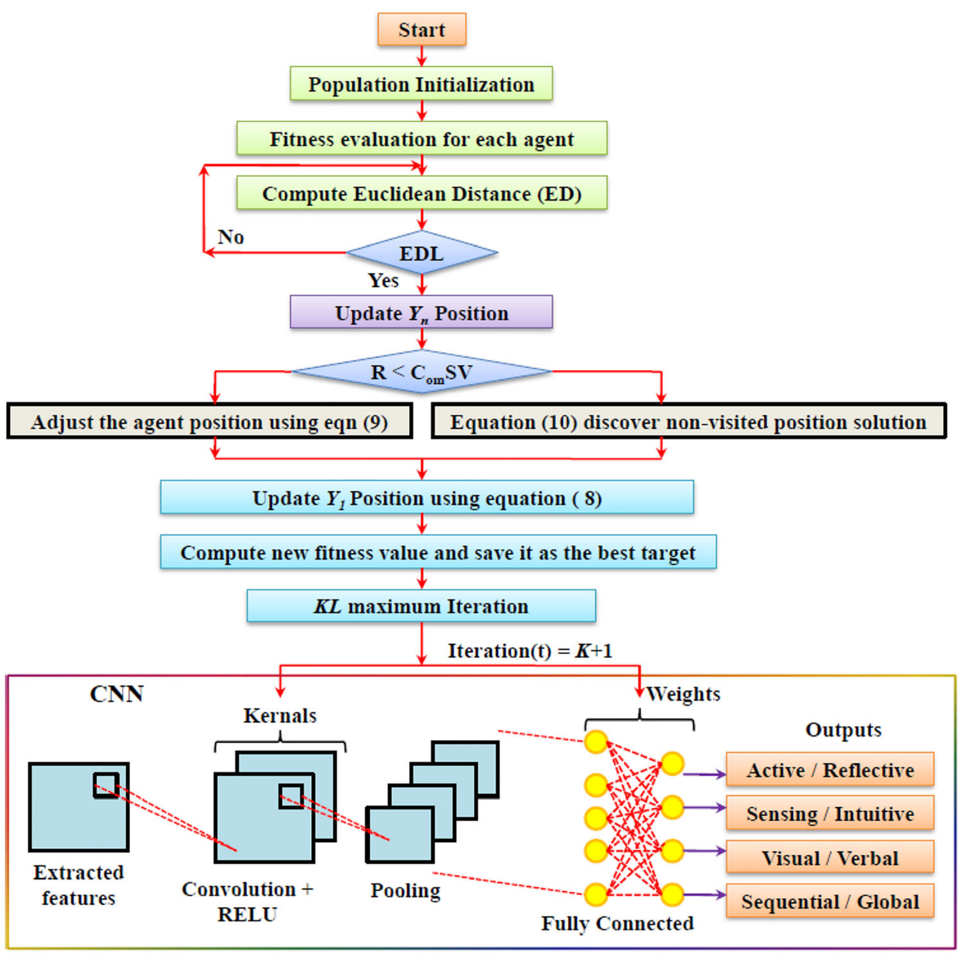

Fig. 2 CNN-LFD model for learning style classification

Table 2 Parameter settings of proposed CNN-LFD algorithm

Parameters	Ranges
Number of search agents	43
∂_1	0.9
∂_2	0.1
$C_{om}SV$	0.5
Threshold	2
b_1	10
b_2	0.0005
b_3	0.005
Number of the convolution layer	2
Number of the pooling layer	4
Momentum	0.9
Learn rate drop period	5
Learn rate drop factor	0.2
Initial learn rate	0.05
Max epochs	50
Mini batch sizes	22
Dropout rate	0.243

Table 3 The MOOC dataset details

	Sessions	
	Winter 2016	Winter 2017
#Learners	32,904	20,892
#Events	19, 862,678	11,812,190
Launch date	19-01-2016	12-01-2017
End date	02-04-2016	06-04-2017

4.1 Evaluation criterion

The evaluation criterions such as accuracy (Acc), specificity (Spec), sensitivity (Sen), and precision (Pre) are used to calculate the performance of the proposed work. The mathematical expression that belongs to each evaluation measure is illustrated from Eq. (20) to Eq. (23).

$$\mathrm{Acc} = \frac{T_{\mathrm{Neg}} + T_{\mathrm{Pos}}}{(T_{\mathrm{Neg}} + T_{\mathrm{Pos}} + F_{\mathrm{Neg}} + F_{\mathrm{Pos}})}, \tag{20}$$

$$\mathrm{Spec} = \frac{T_{\mathrm{Neg}}}{T_{\mathrm{Neg}} + F_{\mathrm{Pos}}}, \tag{21}$$

$$\mathrm{Sen} = \frac{T_{\mathrm{Pos}}}{T_{\mathrm{Pos}} + F_{\mathrm{Neg}}}, \tag{22}$$

$$\mathrm{Pre} = \frac{T_{\mathrm{Pos}}}{T_{\mathrm{Pos}} + F_{\mathrm{Pos}}}, \tag{23}$$

where T_{Pos} is the positive number of sequences accurately predicted. T_{Neg} is the negative number of sequences accurately predicted. F_{Pos} is the positive number of sequences incorrectly predicted. F_{Neg} is the negative number of sequences incorrectly predicted.

The experimental results of the predicted learning style are represented in Table 4. This experiment is conducted depending upon the initial and secondary tests with predicted learning styles. The eight groups of FSLSM such as Sequential, Global, Visual, Verbal, Sensing, Intuitive, Active, and Reflective are used for both two tests. For the first and second tests, 210 and 220 iterations with learning styles are obtained, respectively. Table 4 explains the predicted learning styles. We evaluate the few sequences that are given multiple labels, based on the featured values of FSLSM learning objects. The whole number of the predicted sequence is more than the whole amount of sequences in the output for two experiments. We discovered multiple classes in the predicted variance from the first test to the second test.

Figure 3 illustrates the CNN-LFD model for learning style classification. Here, we discussed both training and validation accuracy during learning style prediction concerning the different number of epochs. There are 0 to 50 epochs that have been chosen with various kinds of accuracy levels. The proposed CNN-LFD model is comprised of kernels in each convolution layer. Increased validation accuracy is achieved by increasing test accuracy. The training accuracy is higher than that of validation accuracy as demonstrated in Fig. 3.

The state-of-art comparison of classification accuracy concerning different techniques is illustrated in Fig. 4. The state-of-art methods including Fuzzy C Means (FCM) [22], fuzzy decision tree [25], Trust-based hybrid recommendation (TBHR) [27], and CNN [27 are selected. The proposed CNN-LFD algorithm yields higher classification accuracy in terms of predicted learning styles. The learning style prediction based on the confusion matrix with respect to accuracy is formulated in Table 5. The pink color in the diagonal represents the accuracy values. The confusion matrix is used to assess the performance of the proposed CNN-LFD algorithm for learning style prediction. The confusion matrix is the table form of output obtained for both actual and predicted classes. The columns represent the predicted values and rows represent the actual values. Every diagonal cell of the table is associated with a single fault class and it stands for the accurate prediction made. The numbers of wrong classification made are denoted using remaining cells or off-diagonal cells.

The state-of-art comparison of accuracy with respect to different iterations is depicted in Fig. 5. We have selected a different number of iterations in this experiment. The

Table 4 Number of sequences taken from different learning styles

Name of the predicted styles	Number of sequences	
	Initial test (depending upon 2 courses)	Secondary test (depending upon 3 courses)
Sequential	115	112
Global	124	126
Visual	228	225
Verbal	99	104
Sensing	107	105
Intuitive	131	129
Active	138	141
Reflective	119	114

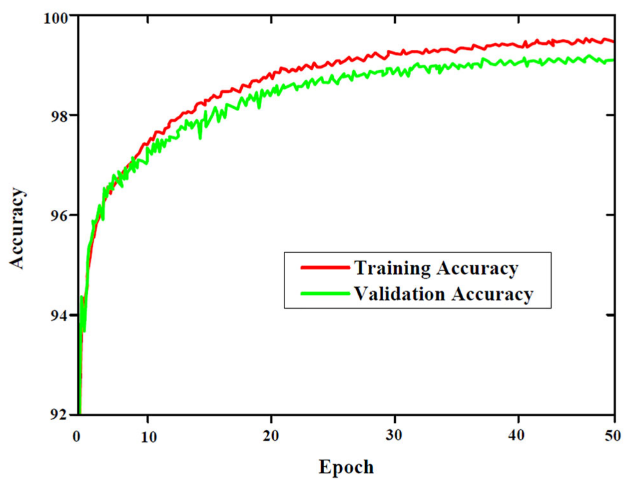

Fig. 3 CNN-LFD model for learning style classification

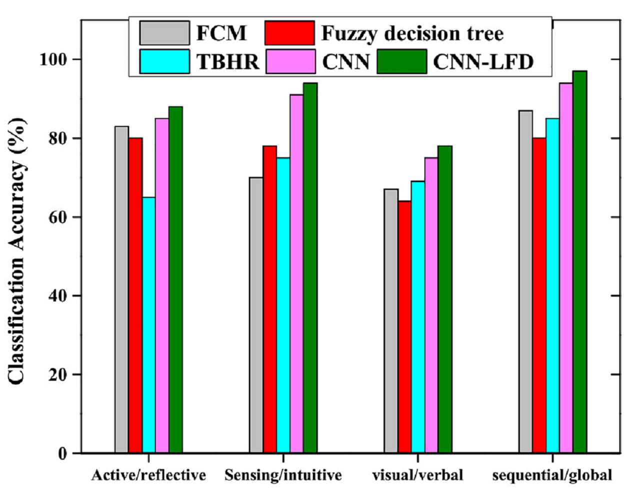

Fig. 4 State-of-art comparison of classification accuracy based on different learning styles

accuracy performance is evaluated using various types of state-of-art methods such as Fuzzy C Means (FCM) [22], fuzzy decision tree [25], Trust-based hybrid recommendation (TBHR) [27], CNN [47], and proposed CNN-LFD. The FCM, TBHR, Fuzzy decision tree, CNN and proposed method demonstrated 83.78%, 81.21%, 80.56%, 90.03% and 94.56% classification accuracy. Finally, the proposed CNN-LFD accomplished higher accuracy during learning style prediction.

The specificity results based on a different number of iterations are as shown in Fig. 6. In this experiment, we have chosen Fuzzy C Means (FCM) [22], fuzzy decision tree [25], Trust-based hybrid recommendation (TBHR) [27], CNN[47], and proposed CNN-LFD with specificity performance results. The FCM, TBHR, Fuzzy decision tree, CNN, and proposed CNN-LFD methods provided 85.67%, 86.54%, 77.34%, 87.67%, and 97.13% specificity values. When compared to other methods, the proposed CNN-LFD achieves better specificity results in the case of learning style prediction.

The state-of-art comparison of sensitivity when evaluated with the MOOC dataset is depicted in Fig. 7. The specificity performance percentages with state-of-art methods including Fuzzy C Means (FCM) [22], fuzzy decision tree [25], Trust-based hybrid recommendation (TBHR) [27], CNN [47], and proposed CNN-LFD is chosen. According to the state-of-art methods, the FCM, TBHR, Fuzzy decision tree, CNN, and proposed CNN-LFD provide 82.45%, 90.67%, 85.73%, 79.89%, and 97.09% sensitivity outcomes. In any case, the proposed CNN-LFD algorithm outperforms existing state-of-the-art approaches in terms of sensitivity.

Figure 8 depicts the state-of-art comparison of precision with respect to different iterations. The precision performances with various state-of-art methods are included in this experiment. Especially, Fuzzy C Means (FCM) [22], fuzzy decision tree (FDT) [25], Trust-based hybrid recommendation (TBHR) [27], CNN [47], and proposed CNN-LFD are used. The proposed CNN-LFD method

Table 5 Confusion matrix analysis based on predicted learning style accuracy

Predicted values	Actual values			
	Active/reflective	Sensing/intuitive	Visual/verbal	Sequential/global
Active/reflective	89.34%	6.20%	0	4.46
Sensing/intuitive	3.67	94.71%	1.62	0
Visual/verbal	9.82%	0%	79.04%	11.14%
Sequential/global	0.48%	0%	1.18%	98.34%

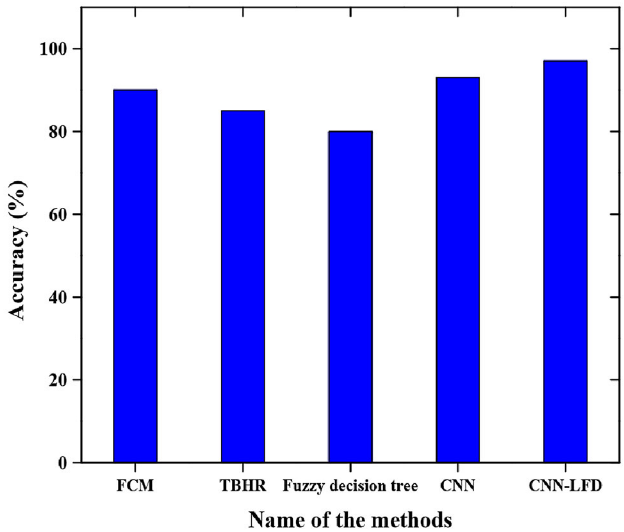

Fig. 5 Prediction results obtained for different techniques in terms of accuracy using the MOOC dataset

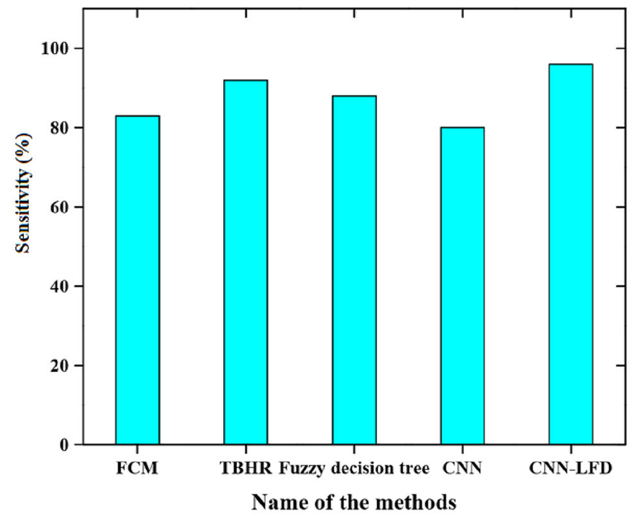

Fig. 7 Performance evaluation in terms of sensitivity for different techniques using the MOOC dataset

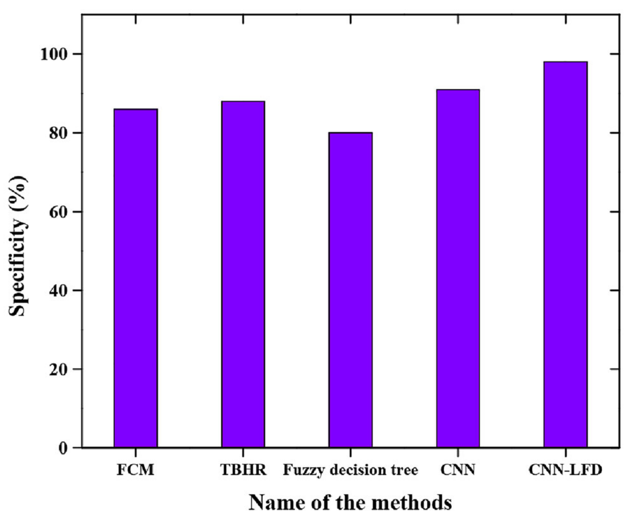

Fig. 6 Performance evaluation in terms of specificity for different techniques using the MOOC dataset

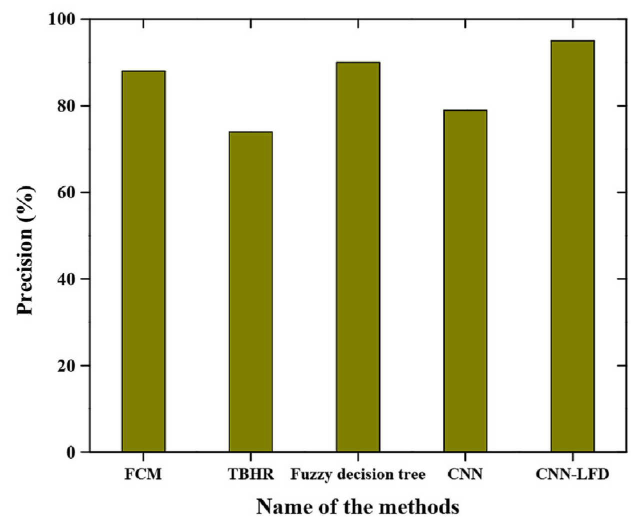

Fig. 8 Performance evaluation of different techniques using precision for the MOOC dataset

accomplishes 97.56% precision results are obtained. Finally, the proposed CNN-LFD method achieves higher precision rates depending upon appropriate learning style prediction.

The state-of-art comparison of different techniques with computational time is delineated in Fig. 9. This graph is plotted between different techniques such as Fuzzy C Means (FCM) [22], fuzzy decision tree (FDT) [25], Trust-based hybrid recommendation (TBHR) [27], CNN [47], and proposed CNN-LFD with computational time in

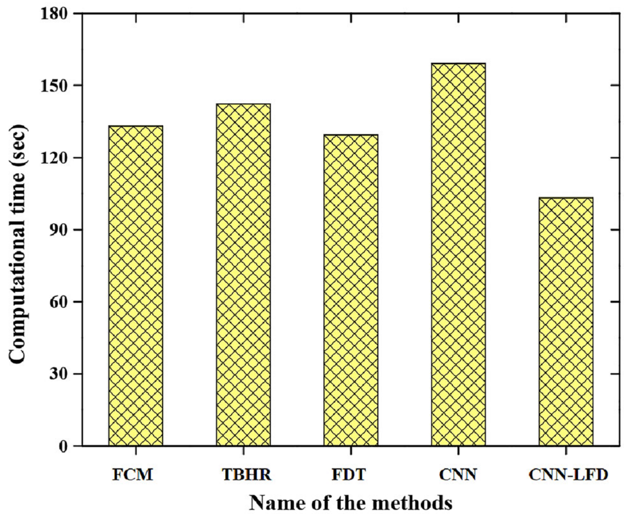

Fig. 9 Comparison results of different techniques using computational time

Table 6 Performance evaluation in terms of accuracy for different number of epochs

Initial test		Secondary test	
Number of epochs	Accuracy (%)	Number of epochs	Accuracy (%)
10	89.36	10	88.97
20	92.34	20	91.36
25	97.09	25	97.14
30	97.09	30	97.14
50	97.09	50	97.14

seconds (s). This FCM, FDT, TBHR, CNN, and CNN-LFD took a computational time of 133.13 s, 142.38 s, 129.47 s, 160. 26 s, and 109.67 s for learning style prediction. When compared to the existing methods, the proposed CNN-LFD method demonstrates less computational time.

The accuracy obtained for a different number of epochs for the proposed CNN-LFD approach is presented in Table 6. Table two provides the accuracy obtained by the CNN-LFD approach for both the initial and secondary tests conducted for 10, 20, 25, 30 and 50 epochs. After the 25th epoch, both tests' accuracy improves, and the second test's accuracy is even better than the initial test due to the presence of a huge number of features.

5 Conclusion

This article proposed a CNN-LFD algorithm for learning style prediction. The proposed work is implemented in JAVA software and the details of the dataset are obtained from Advanced Research via Online Learning (CAROL) platform. The initial and secondary test is used to evaluate the prediction accuracy in terms of different epochs. For both tests, the total number of predicted sequences is larger than the total number of sequences found in the output. The proposed methodology is evaluated using different performance evaluation metrics such as accuracy, specificity, sensitivity, precision, computational time, and confusion matrix. The comparative analysis conducted evaluates the efficiency of this scheme. Using the learner's learning sequence, the CNN-LFD classifier predicts different learning types including Active/Reflective, Sensing/Intuitive, Visual/Verbal, and Sequential/Global with an accuracy of 85.32%, 90.47%, 78.98%, and 90.37% respectively. The proposed CNN-LFD algorithm accomplished higher accuracy, specificity, sensitivity, and precision results when compared to the other state-of-art methods including FCM, fuzzy decision tree, TBHR, and CNN. In the future, we aim to improve this model to create a personalized e-learning system that caters to different types of autistic students depending on their preferred learning styles.

Data availability Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Human and animal rights This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Chen C, Duh L, and Liu C (2004) A personalized courseware recommendation system based on fuzzy item response theory. In: Proceedings of IEEE international conference on E-Technology, E-Commerce, EService, Washington, pp. 305–308
2. Liye, Ma., Sun, B.: Machine learning and AI in marketing–connecting computing power to human insights. Int. J. Res. Mark. **37**(3), 481–504 (2020)
3. Patrick, B., Doyle, E.: Individualising gamification: an investigation of the impact of learning styles and personality traits on the efficacy of gamification using a prediction market. Comput. Educ. **106**, 43–55 (2017)
4. Khamparia, A., Pandey, B.: Association of learning styles with different e-learning problems: a systematic review and classification. Educ. Inf. Technol. **25**(2), 1303–1331 (2020)
5. Sundararaj, V., Selvi, M.: Opposition grasshopper optimizer based multimedia data distribution using user evaluation strategy. Multimedia Tools and Applications, pp. 1-17 (2021)

6. Sivaranjani, J. and Madheswari, A.N., (2017) March. A novel technique of motif discovery for medical big data using hadoop. In 2017 Conference on Emerging Devices and Smart Systems (ICEDSS) (pp. 214-217). IEEE
7. Hari, V. and Madheswari, A.N., (2013) Improving Security in Digital Images through Watermarking Using Enhanced Histogram Modification. In Advances in Computing and Information Technology (pp. 175-180). Springer, Berlin, Heidelberg
8. Tarus John, K., Niu, Z., Yousif, A.: A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. Futur. Gener. Comput. Syst. **72**, 37–48 (2017)
9. Gowthul Alam, M.M., Baulkani, S.: Reformulated query-based document retrieval using optimised kernel fuzzy clustering algorithm. Int. J. Bus. Intell. Data Min. **12**(3), 299 (2017)
10. Sundararaj, V.: An efficient threshold prediction scheme for wavelet based ECG signal noise reduction using variable step size firefly algorithm. Int. J. Intell. Eng. Syst. **9**(3), 117–126 (2016)
11. Gowthul Alam, M.M., Baulkani, S.: Geometric structure information based multi-objective function to increase fuzzy clustering performance with artificial and real-life data. Soft. Comput. **23**(4), 1079–1098 (2019)
12. Sundararaj, V.: Optimised denoising scheme via opposition-based self-adaptive learning PSO algorithm for wavelet-based ECG signal noise reduction. Int. J. Biomed. Eng. Technol. **31**(4), 325 (2019)
13. Gowthul Alam, M.M., Baulkani, S.: Local and global characteristics-based kernel hybridization to increase optimal support vector machine performance for stock market prediction. Knowl. Inf. Syst. **60**(2), 971–1000 (2019)
14. Hassan, B.A., Rashid, T.A.: Datasets on statistical analysis and performance evaluation of backtracking search optimisation algorithm compared with its counterpart algorithms. Data Brief **28**, 105046 (2020)
15. Hassan, B.A.: CSCF: a chaotic sine cosine firefly algorithm for practical application problems. Neural Comput. Appl. **33**, 1–20 (2020)
16. Rejeesh, M.R.: Interest point based face recognition using adaptive neuro fuzzy inference system. Multimed. Tools Appl. **78**(16), 22691–22710 (2019)
17. Sundararaj, V., Muthukumar, S., Kumar, R.S.: An optimal cluster formation based energy efficient dynamic scheduling hybrid MAC protocol for heavy traffic load in wireless sensor networks. Comput. Secur. **77**, 277–288 (2018)
18. Sundararaj, V., Anoop, V., Dixit, P., Arjaria, A., Chourasia, U., Bhambri, P., Rejeesh, M.R., Sundararaj, R.: CCGPA-MPPT: cauchy preferential crossover-based global pollination algorithm for MPPT in photovoltaic system. Prog. Photovolt. Res. Appl. **28**(11), 1128–1145 (2020)
19. Vinu, S.: Optimal task assignment in mobile cloud computing by queue based ant-bee algorithm. Wirel. Pers. Commun. **104**(1), 173–197 (2019)
20. Fei, G., Li, Z., Jun, Yu., Junze, Yu., Huang, Q., Tian, Qi.: Style-adaptive photo aesthetic rating via convolutional neural networks and multi-task learning. Neurocomputing **395**, 247–254 (2020)
21. Baidada M., Mansouri, K., Poirier, F. (2019) Personalized E-learning recommender system to adjust learners' level. In: EdMedia+ innovate learning, pp. 1353–1357. Association for the Advancement of Computing in Education (AACE).
22. Ibtissam, A., Jeghal, A., Radouane, A., Yahyaouy, A., Tairi, H.: A robust classification to predict learning styles in adaptive e-learning systems. Educ. Inf. Technol. **25**(1), 437–448 (2020)
23. Ouafae, E.A., El Alami, Y., Madani, El., Oughdir, L., El Allioui, Y.: A fuzzy classification approach for learning style prediction based on web mining technique in e-learning environments. Educ. Inf. Technol. **24**(3), 1943–1959 (2019)
24. Deborah, L., Jegatha, R., Sathiyaseelan, S.A., Vijayakumar, P.: Fuzzy-logic based learning style prediction in e-learning using web interface information. Sadhana **40**(2), 379–394 (2015)
25. Crockett, K., Latham, A., Whitton, N.: On predicting learning styles in conversational intelligent tutoring systems using fuzzy decision trees. Int. J. Hum Comput Stud. **97**, 98–115 (2017)
26. Madani, Y., Ezzikouri, H., Erritali, M., Hssina, B. (2019) Finding optimal pedagogical content in an adaptive e-learning platform using a new recommendation approach and reinforcement learning. J Ambient Intell. Hum. Comput. pp: 1–16.
27. Ouafae, E.A., El Madani, Y., Alami, El., Oughdir, L., El Allioui, Y.: A hybrid machine learning approach to predict learning styles in adaptive E-learning system. In: International Conference on Advanced Intelligent Systems for Sustainable Development, pp. 772–786. Springer, Cham (2018)
28. Bhaskaran, S., Santhi, B.: An efficient personalized trust based hybrid recommendation (tbhr) strategy for e-learning system in cloud computing. Clust. Comput. **22**(1), 1137–1149 (2019)
29. Sayed, W.S., Mostafa, G., Moemen, A., El-Tantawy, S.: Towards a learning style and knowledge level-based adaptive personalized platform for an effective and advanced learning for school students. Recent Adv. Eng. Math. Phys. **42**, 261–273 (2020)
30. Rasheed, F., Wahid, A.: Learning style detection in E-learning systems using machine learning techniques. Expert Syst Appl **174**, 114774 (2021)
31. Rajkumar, R., Ganapathy, V.: Bio-inspiring learning style chatbot inventory using brain computing interface to increase the efficiency of E-learning. IEEE Access **8**, 67377–67395 (2020)
32. Hmedna, B., El-Mezouary, A., Baz, O.: A predictive model for the identification of learning styles in MOOC environments. Clust. Comput. **14**, 1–26 (2020)
33. Lin, W., Xu, H., Li, J., Wu, Z., Hu, Z., Chang, V., Wang, J.Z.: Deep-profiling: a deep neural network model for scholarly web user profiling. Clust. Comput. **14**, 1–14 (2021)
34. Zhang, W., Qin, S., Yi, B., Tian, P.: Study on learning effect prediction models based on principal component analysis in MOOCs. Clust. Comput. **22**(6), 15347–15356 (2019)
35. Bernard, J., Chang, T.W., Popescu, E., Graf, S.: Learning style Identifier: Improving the precision of learning style identification through computational intelligence algorithms. Expert Syst. Appl. **75**, 94–108 (2017)
36. Index of Learning Styles Questionnaire. Available online: https://www.webtools.ncsu.edu/learningstyles/. Accessed 29 February 2020.
37. Cook David, A., Thompson, W.G., Thomas, K.G.: The motivated strategies for learning questionnaire: score validity among medicine residents. Med. Educ. **45**(12), 1230–1240 (2011)
38. Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon,V.K., Soman, K.P. (2017) Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics (ICACCI), pp. 1643–1647. IEEE.
39. Zhan, M., Gan, J., Lu, G., Wan, Y (2020) Graph convolutional networks of reconstructed graph structure with constrained Laplacian rank. Multimedia Tools Appl. pp: 1–12.
40. Tateno, K., Tombari, F., Laina, I., Navab, N. (2017) Cnn-slam: real-time dense monocular slam with learned depth prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6243–6252).
41. Zhao, R., Wang, Y., Liu, C., Hu, P., Li, Y., Li, H., Yuan, C.: Selfish herd optimizer with levy-flight distribution strategy for global optimization problem. Physica A **538**, 122687 (2020)
42. Yang X.-S., Z. Cui, R. Xiao, A. H. Gandomi, M. Karamanoglu, eds. (2013) Swarm intelligence and bio-inspired computation: theory and applications. Newnes.

43. Marcin, M., Szczotka, W.: Quenched trap model for Lévy flights. Commun. Nonlinear Sci. Numer. Simul. **30**(1–3), 5–14 (2016)
44. Alagarsamy S., Govindaraj, V., Irfan, M., Swami, R., Kumar, N. M. (2020) Smart recognition of real time face using convolution neural network (CNN) Technique.
45. Mokeddem, D, Nasri, D (2020) A new levy flight trajectory-based grasshopper optimization algorithm for multi-objective optimization problems. In: 2020 Second International Conference on Embedded & Distributed Systems (EDiS) (pp. 76–81). IEEE.
46. Hailun, X., Zhang, Li., Lim, C.P.: Evolving CNN-LSTM Models for time series prediction using enhanced grey wolf optimizer. IEEE Access **8**, 161519–161541 (2020)
47. Shen, X., Yi, B., Zhang, Z., Shu, J., Liu, H. (2016) Automatic recommendation technology for learning resources with convolutional neural network. In:2016 International Symposium on Educational Technology (ISET), pp. 30–34. IEEE.
48. Everton, G., Miranda, R., de Barros, L., de Souza Mendes, : Use of deep multi-target prediction to identify learning styles. Appl. Sci. **10**(5), 1756 (2020)

Sami Alshmrany is currently working as Assistant Professor of Information Technology since October 2018. He finished his BSc in Computer from University of King Abdulaziz, Jeddah, Saudi Arabia in 2008. In 2009, he moved to Adelaide, Australia for further studies.He joined Flinders University, Adelaide, Australia and obtained his MSc in Information Technology in 2013. In 2018 he finished his PhD in Information Technology from Flinders University, Adelaide, Australia.His research area interests are in E-learning and Implementation of ICT on educational sectors.