



Churn prediction using optimized deep learning classifier on huge telecom data

Bharathi Garimella¹ · G. V. S. N. R. V. Prasad² · M. H. M. Krishna Prasad³

Received: 3 November 2020 / Accepted: 26 July 2021 / Published online: 29 November 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

With the increasing number of telecom providers and services, churn prediction gains tremendous interest in the current decade. The prediction models based on machine learning are the greatest fortune for customer retention campaigns as they pave the way to predict potential churners. Though the prediction is accurate by applying the machine learning classifiers, there are huge concerns imputing complexity in the prediction process. In most cases, the churn prediction becomes complex due to the telecom data's inconsistency, sparsity, and hugeness. An effective churn prediction model is proposed to tackle such kinds of issues. The proposed churn prediction is based on the optimized deep learning classifier built in the spark architecture to handle the hugeness of the telecom data. The optimized deep learning classifier is established through the optimal training of the deep convolutional neural network (DCNN) using the proposed Firefly-Spider Optimization (FSO), which is the integration of Spider Monkey Optimization (SMO) and firefly optimization algorithm (FA). The proposed prediction model's effectiveness is analyzed using the Churn in Telecom's dataset based on the performance measures. The proposed prediction model acquired the maximal dice coefficient, accuracy, and Jaccard coefficient of 94.61%, 94.76%, and 94.80%.

Keywords Churn prediction · Deep learning · Telecom data · Spark framework · Optimization

1 Introduction

Churn plays an important role in the telecommunications industry. The churn customers are the customers, who decides to escape from the service provider and prepare to shift other competitors in the market. The conventional techniques revealed that the customer churns (Lazarov and Capota 2007) are divided based on three aspects: active churners, passive churners, and rotational churners. The active churners are the customers who desire to quit the indenture and proceeds with the other service provider. Whenever the company discontinues the service with the customer then, they are referred to as passive churners. The rotational churners are the customers who terminate

the contract without any prior knowledge to both parties. The party is responsible for terminating the contract without any intimation. The active churners and the passive churners are easier to predict using the conventional churn prediction approaches based on the Boolean class value. Still, the rotational churners are complex to predict as there is no chance to predict whether the customer might churn in the future or not. The churners change the network due to the connectivity problem (Zema NR et al. 2017; Trotta A et al. 2013, 2015). The customer churns due to several reasons, which are either unknown or complicated to predict. The decision-maker aims to minimize the churn ratio, and it is an eminent fact that the existing customers are the most important for the companies than acquiring new ones (Hadden et al. 2007).

The behavior of churn customers affects the company's performance and provides negative impacts concerning the company's performance, and causes fewer sales due to short-term customers and less service. Moreover, the chunner helps the competitors expand the displeased customers with promotions in business, but leads to the loss of revenue, puts a pessimistic impact on the long-term customers, and maximizes the uncertainties that minimize the ratio of possible customers. Furthermore, the attraction of new customers is

✉ Bharathi Garimella
bgarimella42@gmail.com

¹ Jawaharlal Nehru Technological University Kakinada, Kakinada, Andhra Pradesh, India

² Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India

³ UCEK, JNTUK, Kakinada, Andhra Pradesh, India

complex than retaining the existing customers, and the risk of company image leads to loss of customer base (Amin et al. 2017). Thus, those companies need to know the churners beforehand to predict the customers, who may switch from their current service to the competitor's service. The customer-initiated churn is complex, and the factors responsible for the churn vary for each churn customer. Concerning the telecom industries, both data service and voice customers pose the capability to choose a service provider from a variety of companies and pose the sovereignty to switch their rights from one service provider to another service provider whom they feel to be better. Due to increasing performance, the customer demands customized products and the best services with lower prices. Several telecom companies deployed the retention mechanisms for synchronizing the services for keeping the customers for a long tenure. Thus, the churn prediction proves to be a crucial business goal for supporting telecommunications companies to reduce the churners. Thus, churn reduction needs to predict high-risk customers and evaluate the churn time for preventing churning (Arora and Singh 2013).

Churn prediction is applicable in several areas, such as banking, life insurance, and other sectors. This kind of prediction facilitates the company to know that customers are not happy with their company services to maintain their customers in advance (Ahmed and linen 2017). Customer churns can be addressed from two different angles in the existing research. The researcher concentrates on enhancing the customer churn prediction models in which more complicated models are being devised and tend to enhance the predictive performance (Verbeke et al. 2011). Moreover, the researchers tend to understand the churners and try to provide customer satisfaction (Gustafsson et al. 2005). The prediction of churn customers is considered a supervisory problem, given by the customer's individual choice. Thus, the customer churn prediction models are considered a key problem amongst researchers and assist the managers in understanding the churners for making improved decisions in combating customer churn (Verhoef 2003). Several techniques tend to point out the managerial value of customer segmentation (Athanasopoulos 2000; Chan 2008; Seret et al. 2012). To predict the customer churns, logistic regression (LR) and decision trees (DT) play a crucial role in evaluating the churn probability as they integrate the improved predictive performance with good comprehensibility. Both techniques are useful and pose their own merits and demerits. DT poses the ability to interact between variables and poses complexities while handling linear relations between the variables. On the other hand, LR can deal with the linear relations between variables but does not determine the interaction effects between the variables (Caigny et al. 2018). Numerous techniques are devised for churn prediction, and most of the techniques utilize machine learning

techniques. Here, learning occurs in two ways: supervised way and unsupervised way. Supervised learning tends to train a model with known input and output data. The data is labeled, and this set of labels utilizes the data to predict the future outputs using new data. Unsupervised learning is used for unlabeled data. This technique determines the hidden patterns on the input data based on statistical means for predicting the churns.

The research aims to devise a technique for detecting the churners using a newly devised hybrid optimization technique named FSO algorithm-based DCNN. The complete process is carried out using the spark architecture. The proposed approach comprises two steps, where the churners are detected using the input telecom data for effective churn prediction. At first, the telecom data are subjected to the feature selection phase for selecting the essential features based on the RFE technique for classification. After selecting the features, a hybrid optimization algorithm named FSO is adapted by combining the FA and SMO by selecting the optimal weights obtained by training the DCNN using the proposed FSO algorithm.

1.1 Proposed FSO-based DCNN for churn prediction

The proposed FSO is employed for selecting the optimal weights in the DCNN and is designed by integrating the update equation of the SMO algorithm and FA.

The paper is arranged in the following manner. Section 1 introduces the paper with the churn prediction, and Sect. 2 elaborates the conventional techniques related to churn prediction and the challenges. The proposed method of the churn prediction model is illustrated in Sect. 3. The churn prediction techniques are elaborated in Sect. 4, and then Sect. 5 provides the summary.

2 Motivation

The section discusses the literature survey of the customer churn prediction techniques and the disadvantages of the methods. At the end of the section, the challenges of the existing methods are deliberated.

2.1 Literature survey

The eight existing techniques based on customer churn prediction were illustrated in this section.

Höppner et al. (2018) developed a Prof Tree algorithm for selecting the most profitable churn model. In this model, a classifier was devised for integrating the EMPC metric for constructing the model. This technique was named ProfTree that used an evolutionary algorithm for training

the profit-driven decision trees. This method attained more profit for numerous telecommunication service providers.

Ahmed and linen (2017) used a hybridized firefly algorithm for estimating the survival of customers for predicting the churners. The method's goal was to identify the customer at the churning stage and the time they churn. This method summarized the churn prediction techniques for illustrating the churners. The method showed accurate churn prediction and improved the services for overturning the churn decision, but false-positive rates were not reduced.

Amin et al. (2017) devised a rough set approach for extracting the significant decision rules based on customer churn and non-churn. The method performed a churn and non-churn customers classification and the ratio of customers who were more likely to churn soon. However, the profiles of predicted customer chuns were not considered in this technique.

Amina et al. (2019) developed a customer churn prediction technique based on classifier certainty evaluation using distance and robust feature selection methods. The data were categorized into two sets with different factors. The first factor was data with more certainty, and the next was data with less certainty to predict the customer having Churn and Non-churn behavior.

Bi et al. (2016) developed a Semantic Driven Subtractive Clustering Method (SDSCM) to predict different customer chuners' behavior. The method increased the accuracy of clustering but minimized the profit maximization.

Nijhawan et al. (2018) designed a Classification And Regression Trees (CART) for the customer churn prediction. The method had more accuracy rate, while the chuners reduced the churning rate.

Strippling et al. (2018) designed a classifier named ProfLogit, which tended to maximize the Expected Maximum Profit Measure for Customer churn (EMPC) in the training step. Here, ProfLogit's interior model used threshold-independent recall and precision measures for maximizing the profit fraction in predicting the churn customers. This method posed the highest recall and accuracy values but failed to deal with complicated data structures.

Caigny et al. (2018) developed a logit leaf model (LLM) for classifying the churn and non-churn customers. The LLM aimed to construct the segments of the data while maintaining the unambiguousness of the models. The LLM contained two phases, named the segmentation phase and a prediction phase. In the segmentation phase, the customer segments were identified using the decision rules, and in the prediction phase, a model was created for each tree. The model employed Logistic regression and decision trees to improve performance for categorizing the churn and non-churn customers. However, the method failed to consider the training time as a metric in the performance evaluation.

The challenges faced by the existing system are high false-positive rates, prediction of churn and non-churn customers, accuracy, failure to deal with complicated data structures, and high training time. The proposed FSO-based DCNN churn prediction overcame these issues.

2.2 Challenges

The major challenges of the research are listed below:

- Customer churn is considered a major challenge as it affects the business and industries meticulously in the telecommunication sector. Thus, it is a challenging task to distinguish churn from non-churn customers (Amin et al. 2017).
- The genetic techniques yield high profits, but they cannot easily cope using complicated data structures, like nonlinearities (Strippling et al. 2018).
- Even though logistic regression is robust, the count of customers in the leaves must be high for obtaining a logistic regression model. This model uses complicated rules for adapting the number of leaves and leaf sizes (Caigny et al. 2018).
- Area Under the receiver operating characteristics curve (AUC) and Top Decile Lift (TDL) are utilized for assessing the performance of the prediction model, but they lacked direct profit criterion (Coussement et al. 2017).
- The Maxim profit (MP) criterion is considered as an advanced performance metric, which uses profit. The evaluation metric needs additional information for computing the customer lifetime value, which was impossible in gathering all datasets and benchmark studies (Verbeke et al. 2012).

3 Proposed churn prediction model based on the proposed FSO-based DCNN dependent spark architecture

In this section, the spark architecture predicts the chuns using DCNN (Tu et al. 2017). Figure 1 illustrates the customer churn prediction model using the proposed FSO-based DCNN. At first, the customers' demographic data are analyzed, and the data, such as network logs, customer data, and International Mobile Equipment Identity (IMEI) data, are gathered for evaluating the chuners. Here, the users are connected to the system over the internet, where the data analysis is made. Furthermore, the spark architecture is employed for predicting the chuners. Spark (Kusuma et al. 2016) is a framework used to process huge data processing tasks using a general-purpose programming language on the customer data. The spark supports several interactive data analysis

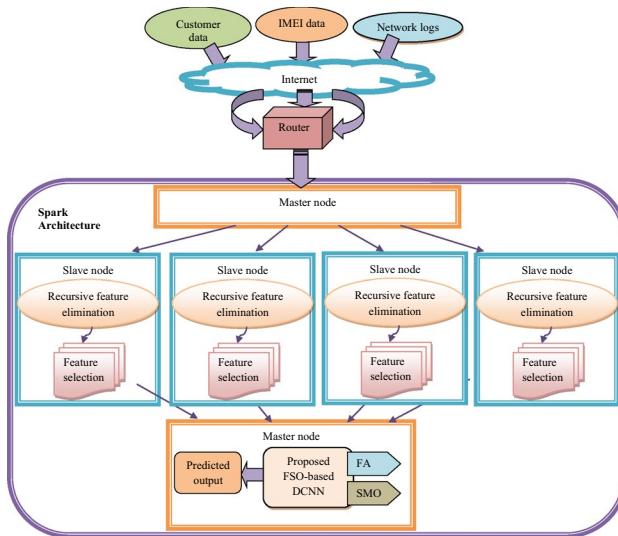


Fig. 1 Schematic diagram of a proposed Churn prediction model using FSO-based DCNN

tools and machine learning algorithms and reuses the data simultaneously while maintaining scalability.

The spark architecture possesses two main modules, namely master node and slave node. The master node is responsible for managing and distributing the task obtained from the user data by partitioning the obtained tasks into different subtasks for each slave node. These subtasks are processed by the slave nodes for processing the data. In this model, assume the master node's size is divided into four slave nodes, each of which is of size $[i \times j]$. In each slave node, the feature selection process is carried out using the RFE that selects the optimal features. Each extracted feature is of size $[k \times l]$. These features are combined for initiating the classification of churners, which possesses the size of $[m \times n]$. The classification is carried out using the proposed FSO-based DCNN in which the proposed FSO algorithm is used for training the DCNN. The proposed FSO algorithm is designed by integrating FA (Arora and Singh 2013) and SMO (Bansal et al. 2014). The classification is done in the master node and works by merging the data into the same subset of size $[e \times f]$. Thus, the churn prediction is performed on Apache Spark. Thus, the proposed Churn prediction model involves two processes, such as feature selection and classification, which will be performed in the initial cluster nodes of Spark architecture in a parallel manner.

3.1 Selection of features based on RFE using parallel slave nodes

The aim of parallelizing the techniques is to address huge-scale data in solving optimization issues and analytics for effective feature selection. The motivation behind

parallelizing the algorithm is derived based on the selected features. The process of selecting the best features is paralleled using the Spark architecture with master and slave nodes. Consider G be the database having a number of rows with b attributes and is represented as

$$G = \{G_{a,b}\}; 0 \leq a \leq A \text{ and } 0 \leq b \leq B \quad (1)$$

where $G_{a,b}$ represents ath row and bth attributes and is splitted into a finite number, equal to the number of slave nodes. In the slave node, the feature selection process is carried out based on RFE, and the steps for the feature selection is illustrated in algorithm 1 and is given as follows:

Algorithm 1 Parallel feature selection.

Algorithm1. Parallel feature selection

Procedure Parallel_Feature selection (Solution S)

{

Master:

Call RFE to select the optimal feature

Each block acquires the block of features of the dataset

Slave (Parallel):

Perform feature selection using each and every data of the customer using RFE.

Master:

S = Merging the selected features from all slaves and transferring the selected features to DCNN.

Return S

}

3.1.1 RFE-based feature selection process in the spark

Recursive Feature Elimination (RFE; Guyon et al. 2002) is a technique used to remove ineffective and weakest features until the specific feature set is attained for smoother processing. The RFE aims to choose the features by recursively considering lesser feature sets. The significance of the RFE is obtained through a coefficient attribute or a feature attribute. Then, the less important features are eliminated from the current set of features. This process is iteratively done to acquire the required features. It is complex to eliminate many features at a time and affects the performance of the model. This issue is overcome by using the RFE, which utilizes cost function $PL(U)$ and weights $(\omega_U)^2$ for eliminating one feature at a time.

Steps involved in the RFE are as follows:

- (i) Initially, the weights of the classifier are optimized based on the FSO technique.

(ii) Then, the ranking criterion for each set of features is computed using cost function and weights. Based on the cost function, the altered cost function is given by,

$$PL(U) = \left(\frac{1}{2}\right) \frac{\partial^2 L}{\partial \omega_U^2} (P\omega_U)^2 \quad (2)$$

where $PL(U)$ represents the change in the cost function, U denotes the feature, L denotes the cost function, ω_U is a weight of U^{th} feature, and $P\omega_U$ represents the change in weights. Based on weights, the change in weights is given by,

$$\omega_U = (\mu_U(+) - \mu_U(-)) / (\sigma_U(+) + \sigma_U(-)) \quad (3)$$

where μ and σ represents the mean and standard deviation of the feature values of the feature U . The high positive ω_U values represent a strong correlation with class (+), and large negative ω_U values indicate a strong correlation with the class (-).

(iii) In this step, the features with the least ranking criterion are required to be eliminated.

Finally, the features selected using the RFE technique is given by,

$$F = \{F_1, F_2, \dots, F_I, \dots, F_Z\} \quad (4)$$

where Z specifies the total number of selected features and F_I specifies the I^{th} feature. The selected features are given as an input to the DCNN for the churn prediction.

3.2 Proposed FSO-based DCNN for predicting the churners

This section illustrates the proposed churn prediction model using optimization-based DCNN, carried out using two steps. The first step is the feature selection, and the second step is the churn prediction using the proposed churn prediction model. This step is performed using the spark framework, which contains initial and final nodes. In other words, feature selection is progressed in the initial nodes, and the churn prediction is performed in the final nodes of spark. When the input churn data is subjected to the initial nodes of spark, the feature selection process is performed using the Recursive feature elimination procedure. Thus, the initial nodes' output in the selected features from the input churn data and the selected features are fed to the final nodes of spark. In the final nodes, the DCNN is employed to process the selected features to predict the churn data. DCNN is trained using the optimization algorithm, which is developed newly and named as Firefly-Spider Optimization (FSO) algorithm. The developed FSO is the integration of the FA (Arora and Singh 2013) and SMO (Bansal et al. 2014). The FA and SMO are selected due to their fascinating benefits. FA is an algorithm developed based on the behavior of fireflies. The

FA algorithm provides a better tradeoff between the exploration phase and the exploitation phase. SMO is a nature-inspired algorithm inspired by the characteristics of spider monkeys. The SMO is flexible in the group of swarm intelligence-based algorithms. Moreover, the SMO provides reliable, flexible, and efficient results. Thus, the inclusion of SMO in the update process of FA enhances the convergence and thereby improves the algorithm's performance to obtain a global optimal solution. Finally, the predicted data is acquired using the FSO-based DCNN.

3.2.1 Architecture of the DCNN

The FSO-based DCNN is illustrated in this section, along with its architecture in Fig. 2. The DCNN (Tu et al. 2017) comprises convolutional (Conv) layers, pooling (POOL) layers, and Fully Connected (FC) layers; each is responsible for performing a specific task. The goal of the conv layers is to form the feature maps based on the feature vector F obtained from the input telecom data, and the developed feature maps are sub-sampled using POOL layers, which is the second layer in DCNN. The main role of the third layer is the FC layer, which is utilized for leading the churn prediction process. Thus, the conv layers are organized as a multilayer loop of input maps, kernel weights, and output maps. The number of the input and the output data reduces with the successive Conv layers compared with the first conv layer. It is noted that the accuracy of prediction is based on the number of layers in DCNN (Tu et al. 2017).

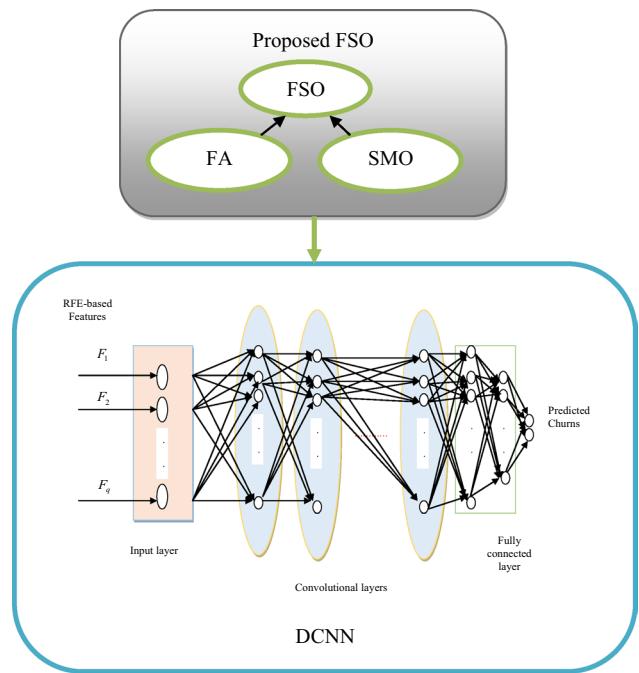


Fig. 2 Architecture of the FSO-based DCNN

3.2.1.1 Convolutional layers The Conv layers use the selected features to facilitate the prediction of the churners using the feature vector obtained from the input telecom data. The conv layer is the interconnection of the neurons that form the feature map, and these conv neurons from the receptive fields link the neurons of the previous and the next conv layer with the help of the trainable weights. In the conv layer, the convolution of the input data and the trained weights are formed to generate new feature maps. The outcome of the convolution is transmitted through the non-linear activation function. The input data is features selected from the input telecom data, and the number of the conv layers in DCNN is,

$$F = \{F_1, F_2, \dots, F_p, \dots, F_q\} \quad (5)$$

where q specifies the total number of the conv layers in DCNN and F_p specifies the p th conv layer in DCNN. The units located at (u, v) derive the output is denoted as,

$$(R_x^p)_{u,v} = (Q_x^p)_{u,v} + \sum_{y=1}^{M_1^{p-1}} \sum_{c=-t_1^s}^{t_1^s} \sum_{d=-t_2^s}^{t_2^s} (W_{x,y}^p)_{c,d} * (R_y^{p-1})_{u+c, v+d} \quad (6)$$

where $*$ denote the convolutional operator, $(R_y^{p-1})_{u+c, v+d}$ indicates the fixed feature maps, and M_1^{p-1} represents total feature maps from the previous conv layer. $(W_{x,y}^p)_{c,d}$ represents the network weights that are trained using the proposed FSO, which forms the weights of the p th conv layer.

3.2.1.2 Pooling (POOL) layer The POOL layer uses the activation function to ensure simplicity and effectiveness while dealing with large networks. The output of the POOL layer is given to the feature maps and is given as,

$$R_x^p = \text{fun}(R_x^{p-1}) \quad (7)$$

where $\text{fun}()$ denote the activation function in the layer p . Moreover, POOL layers reduce the complexity.

3.2.1.3 Fully connected layers The prediction result generated using the pooling, and the conv layers form the input to the fully connected layers subjected to high-level reasoning. The output from the fully-connected layers is given as,

$$O_x^p = T(X_x^p) \text{ with } X_x^p = \sum_{y=1}^{M_1^{p-1}} \sum_{c=t_1^s}^{M_2^{p-1}} \sum_{d=t_2^s}^{M_3^{p-1}} (Y_{x,y,u,v}^p) \cdot (R_x^{p-1})_{u,v} \quad (8)$$

where $Y_{x,y,u,v}^p$ denote the weight connecting (u, v) in y th feature map of layer $p - 1$ and x th unit in layer p . The weights that are to be tuned optimally using the proposed FSO algorithm and are given as, $K \in \{Q, W, Y\}$.

3.2.2 Training algorithm for DCNN

The training of DCNN is performed using the proposed FSO algorithm that aims at determining the optimal weights to tune the DCNN predicting the churners. The optimal weights derived from the proposed FSO algorithm tunes the DCNN for deriving the optimal prediction result. The churn prediction using the proposed FSO-based DCNN effectively predicts the data by deriving the optimal prediction results and can deal with the new data attributes arriving from the distributed sources.

3.2.3 Algorithmic steps of the proposed FSO algorithm

This section illustrates the algorithmic steps of the proposed FSO for training the DCNN to obtain the optimal weights. Here, the integration of FA (Arora and Singh 2013) in the SMO (Bansal et al. 2014) algorithm inherits both SMO and FA algorithm's advantages for tuning the optimal weights of DCNN. The FA algorithm is a bio-inspired optimization technique motivated by the behavior of fireflies and is highly robust, simple, and efficient for solving complex optimization issues. The demerits of FA are that it possesses lower convergence and is highly sensitive to hyperparameters. The demerits of FA are overcome by using SMO that offers a better convergence rate while obtaining a global optimal solution. SMO is a swarm intelligence-based optimization algorithm duly based on the foraging mechanism of spider monkeys. This method is highly reliable and efficient in solving optimization issues. The steps involved in the training algorithm are discussed below:

Step 1. Initializing random solutions: the first step is the initialization of the solutions and is given by,

$$D = \{D_1, D_2, \dots, D_r, \dots, D_E\} \quad (9)$$

where D_r represents the r th solution and E is the total number of solutions.

Step 2. Error estimation: the fitness is a minimization function computed to obtain a better solution. The fitness with minimum error value is considered as the best solution that is computed as,

$$MS_{err} = \frac{1}{A} \times \sum_{a=1}^A (H_a - H_a^*)^2 \quad (10)$$

where MS_{err} denotes the means square error, A is the total samples and H_a refers the obtained output and the H_a^* is the predicted output.

Step 3. Solution update using SMO: the SMO algorithm (Bansal et al. 2014) utilizes group members, local leaders, and all group members' self-experience to generate the solutions. The updated solution obtained using the global leader phase of SMO is expressed as,

$$D_{rz}(o+1) = D_{rz}(o) + J(0, 1) \times (N_{\ell z} - D_{rz}(o)) + J(-1, 1) \times (D_{wz}(o) - D_{rz}(o)) \quad (11)$$

where o is the current iteration, D_{rz} denotes the z th dimension of r th solution in the current iteration, $N_{\ell z}$ specifies the ℓ th solution of z th dimension and D_{wz} represents the randomly generated w th solution in the z th dimension, and $J(0, 1)$ is the random number ranging between 0 and 1 and $J(-1, 1)$ indicates the random number ranging between -1 and 1. After rearranging the above equation, the equation obtained is represented as,

$$D_{rz}(o+1) = D_{rz}(o)[1 - J(0, 1) - J(-1, 1)] + J(0, 1)N_{\ell z} + J(-1, 1)D_{wz}(o) \quad (12)$$

Step 4. Solution update using FA: the FA (Arora and Singh 2013) employs the light absorption coefficients and light intensity to generate the optimal solution. The updated solution obtained using FA is expressed as,

$$D_{rz}(o+1) = D_{rz}(o) + \alpha_o e^{-\mu Z_{rw}^2} (D_{wz} - D_{rz}) + \eta \vartheta \quad (13)$$

where α_o indicates the attractiveness, μ specifies the absorption coefficient, Z_{rw}^2 is a random number, and ϑ represents a vector of the random number, the randomization parameter is indicated as η .

$$D_{rz}(o+1) = D_{rz}(o) + \alpha_o e^{-\mu Z_{rw}^2} D_{wz}(o) - \alpha_o e^{-\mu Z_{rw}^2} D_{rz}(o) + \eta \vartheta \quad (14)$$

$$D_{rz}(o+1) = D_{rz}(o) \left[1 - \alpha_o e^{-\mu Z_{rw}^2} \right] + \alpha_o e^{-\mu Z_{rw}^2} D_{wz}(o) + \eta \vartheta \quad (15)$$

$$D_{rz}(o) \left[1 - \alpha_o e^{-\mu Z_{rw}^2} \right] = D_{rz}(o+1) - \alpha_o e^{-\mu Z_{rw}^2} D_{wz}(o) - \eta \vartheta \quad (16)$$

After rearranging the above equations,

$$D_{rz}(o) = \frac{1}{\left[1 - \alpha_o e^{-\mu Z_{rw}^2} \right]} \left[D_{rz}(o+1) - \alpha_o e^{-\mu Z_{rw}^2} D_{wz}(o) - \eta \vartheta \right] \quad (17)$$

After substituting Eq. (17) in the update Eq. (12) of SMO is given by,

$$D_{rz}(o+1) = \frac{D_{rz}(o+1) - \alpha_o e^{-\mu Z_{rw}^2} D_{wz}(o) - \eta \vartheta}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] + J(0, 1)N_{\ell z} + J(-1, 1)D_{wz}(o) \quad (18)$$

$$D_{rz}(o+1) = \frac{D_{rz}(o+1)}{1 - \alpha_o e^{-\mu Z_{rw}^2}} - \frac{\alpha_o e^{-\mu Z_{rw}^2} D_{wz}(o)}{1 - \alpha_o e^{-\mu Z_{rw}^2}} - \frac{\eta \vartheta}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] + J(0, 1)N_{\ell z} + J(-1, 1)D_{wz}(o) \quad (19)$$

$$\begin{aligned} D_{rz}(o+1) &= \frac{D_{rz}(o+1)}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] - \\ &\quad \frac{\alpha_o e^{-\mu Z_{rw}^2} D_{wz}(o)}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] - \frac{\eta \vartheta}{1 - \alpha_o e^{-\mu Z_{rw}^2}} \\ &\quad [1 - J(0, 1) - J(-1, 1)] + J(0, 1)N_{\ell z} + J(-1, 1)D_{wz}(o) \end{aligned} \quad (20)$$

$$\begin{aligned} D_{rz}(o+1) &- \frac{D_{rz}(o+1)}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] \\ &= \frac{-\alpha_o e^{-\mu Z_{rw}^2} D_{wz}(o)}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] \\ &\quad - \frac{\eta \vartheta}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] \\ &\quad + J(0, 1)N_{\ell z} + J(-1, 1)D_{wz}(o) \end{aligned} \quad (21)$$

$$\begin{aligned} D_{rz}(o+1) &\left[1 - \frac{1 - J(0, 1) - J(-1, 1)}{1 - \alpha_o e^{-\mu Z_{rw}^2}} \right] \\ &= D_{wz}(o) \left[J(-1, 1) - \frac{\alpha_o e^{-\mu Z_{rw}^2}}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] \right] \\ &\quad - \frac{\eta \vartheta}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] + J(0, 1)N_{\ell z} \end{aligned} \quad (22)$$

The final equation forms the position update equation of the proposed FSO algorithm, which improves the performance of the algorithm in churn prediction and is formulated as,

$$\begin{aligned} D_{rz}(o+1) &\left[\frac{-\alpha_o e^{-\mu Z_{rw}^2} + J(0, 1) + J(-1, 1)}{1 - \alpha_o e^{-\mu Z_{rw}^2}} \right] \\ &= D_{wz}(o) \left[J(-1, 1) - \frac{\alpha_o e^{-\mu Z_{rw}^2}}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] \right] \\ &\quad - \frac{\eta \vartheta}{1 - \alpha_o e^{-\mu Z_{rw}^2}} [1 - J(0, 1) - J(-1, 1)] + J(0, 1)N_{\ell z} \end{aligned} \quad (23)$$

Step 5. Evaluation of the error: the error value is computed by applying the weights using the proposed FSO to the DCNN classifier. The optimal solution for training the DCNN is updated based on the minimum error value. Thus,

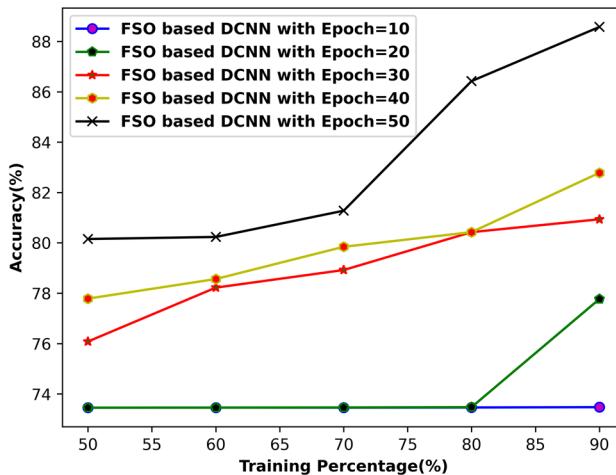


Fig. 3 Performance analysis of the proposed algorithm based on Training data percentage using Accuracy measure

the solution obtained using the proposed FSO is applied to the DCNN classifier for tuning the optimal weights, which is defined in Eq. (10).

Step 6. Determination of a feasible solution: the error value computed is fed into the DCNN classifier to find the best solution. The minimum error value is determined as the optimal solution.

Step 7. Termination: the training process continues until the best solution is obtained. Once the optimal solution is determined, the process gets terminated.

4 Results and discussion

In this section, the analysis of results is performed based on training data percentage and selected features. The analysis is carried out by varying the Epoch values from 10 to 50. The metrics utilized for the analysis are the Dice coefficient, Jaccard coefficient, and accuracy.

4.1 Evaluation metrics

The performance measures utilized for evaluating the performance are the Dice coefficient, Jaccard coefficient, and accuracy.

4.1.1 Jaccard coefficient

The Jaccard coefficient is employed to compute the similarity between the two sets of data and is given by,

$$J_c = \frac{|X \cap Y|}{|X \cup Y|} \quad (24)$$

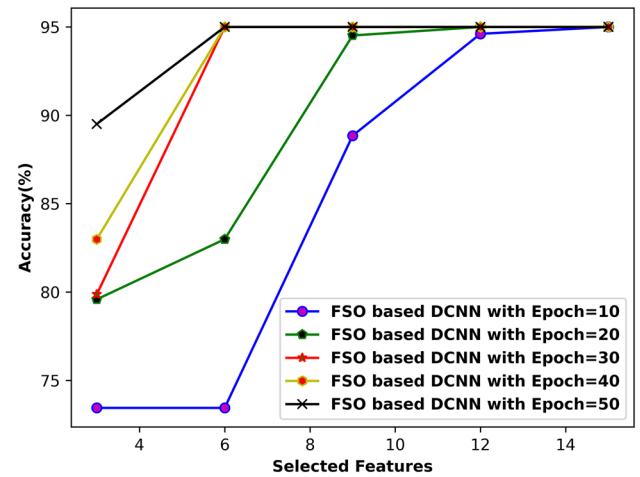


Fig. 4 Performance analysis of the proposed algorithm based on varied number of features using Accuracy measure

where X , and Y are two sets of data.

4.1.2 Dice coefficient

The dice coefficient is another metric used to measure the similarity between the two data sets. It is a type of statistical validation that can be utilized for evaluating performance. The dice coefficient is more perceptive than the Jaccard coefficient because the dice coefficient poses the capability to compute the percentage overlap between the two data sets and is formulated as,

$$D_c = \frac{2|X \cap Y|}{|X| + |Y|} \quad (25)$$

where X and Y depicts the total number of elements present in the data.

4.1.3 Accuracy

The accuracy shows the accurate detection process that is computed as,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (26)$$

where TP is true positive, TN refers true negative, FP indicates false positive, and FN is false negative, respectively.

4.2 Dataset description

4.2.1 Dataset-1 churn in telecom

The Churn in Telecom's dataset (Churn in Telecom's dataset (2019). <https://www.kaggle.com/becksddf/churn-in-telec>

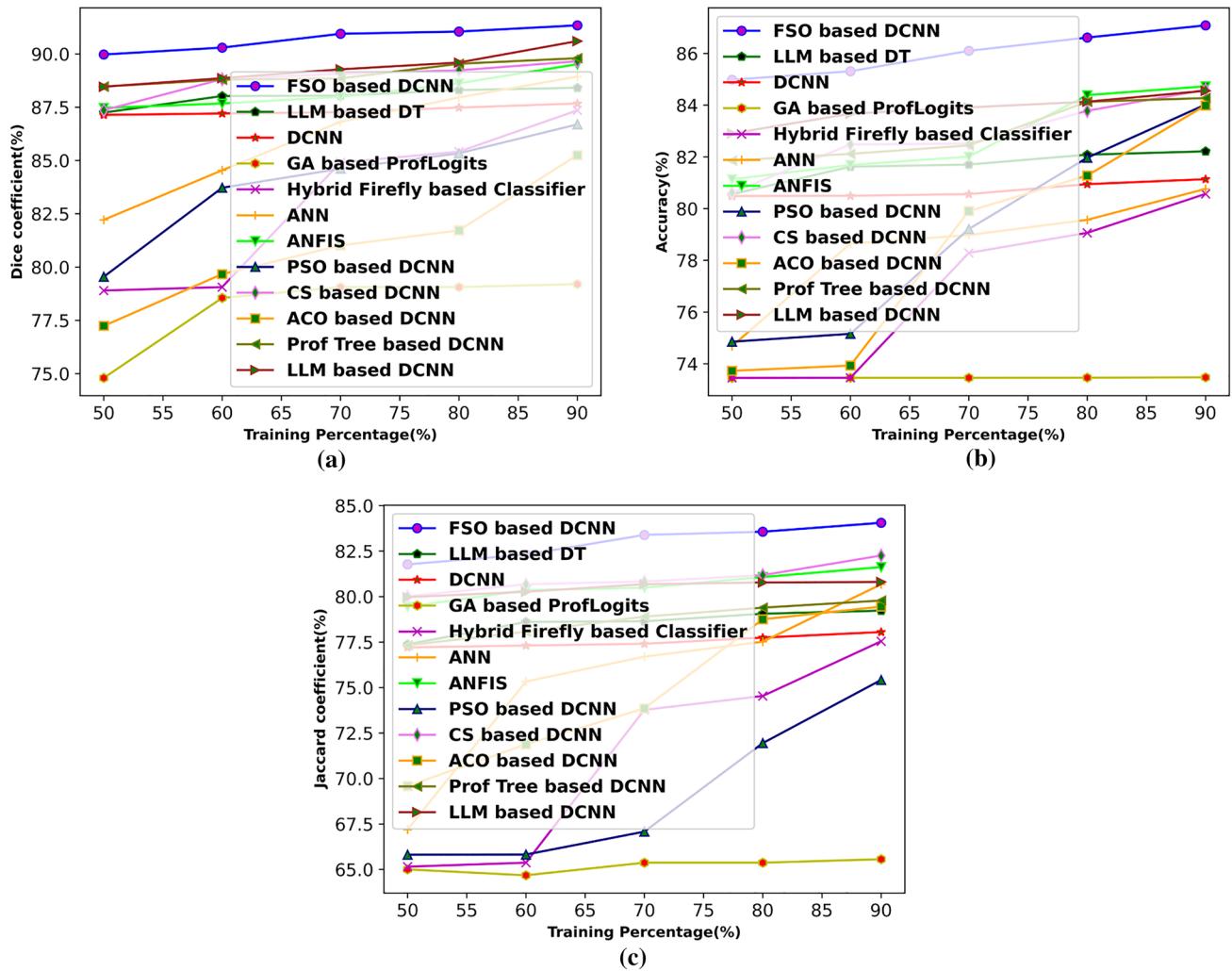


Fig. 5 Comparative analysis based on training data percentage in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

oms-dataset) is used for the experimental evaluation and analysis. The dataset was updated by David Beck and consisted of several attributes that facilitate the analysis of the churn prediction. The dataset contains several attributes: state, account length, area code, phone number, and total customer service calls.

4.2.2 Dataset-2 telecom churn (cell2cell)

Cell2Cell dataset (Telecom churn (cell2cell) (2019).<https://www.kaggle.com/jpacse/datasets-for-churn-telecom>) is pre-processed, and a balanced version is provided for analyzing the process. It consists of 71,047 instances and 58 attributes.

4.2.3 Dataset-3 telco customer churn

Telco customer churn (Telco customer churn (2019).<https://www.kaggle.com/bblastchar/telco-customer-churn>) contains

7043 rows and (customers) and 21 columns (features). Each row represents a customer; each column contains the customer's attributes described on the column Metadata. Here, the "Churn" column is our target.

4.2.4 Dataset-4 telecom customer

The Telecom customer dataset (Telecom customer. (2019).<https://www.kaggle.com/abhinav89/telecom-customer>) consists of 100 variables and approx 100 thousand records. This data set contains different variables explaining the attributes of the telecom industry and various factors considered important while dealing with customers of the telecom industry. The target variable here is churn which explains whether the customer will churn or not.

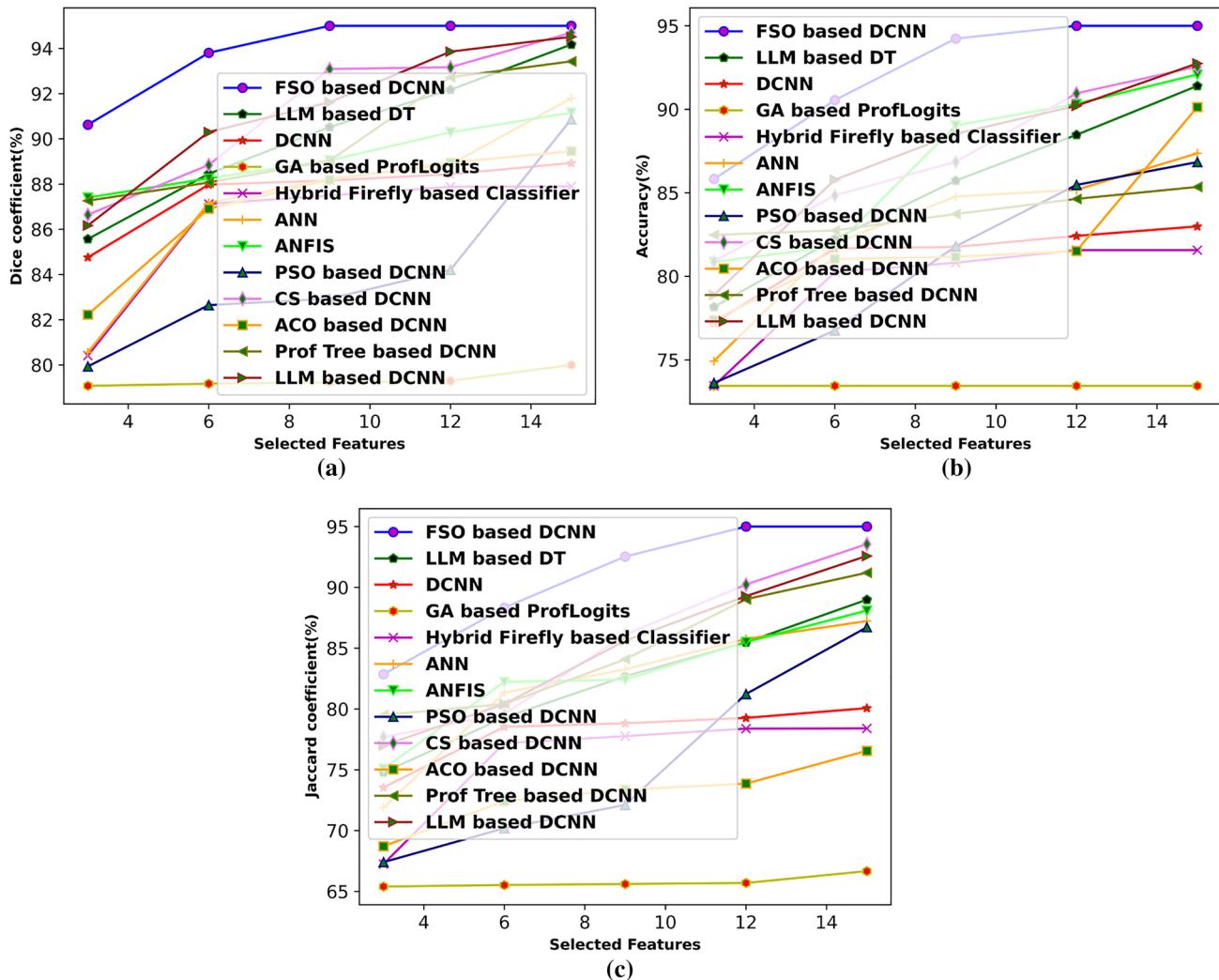


Fig. 6 Comparative analysis based on selected features in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

4.3 Comparative methods

The analysis of classifiers, named FSO-based DCNN, logit leaf model-based decision tree (LLM-based DT; Caigny et al. 2018), DCNN (Tu et al. 2017), Genetic algorithm-based ProfLogits (GA-based ProfLogits; Strippling et al. 2018), hybrid firefly based classifier (Ahmed and linen 2017) ANN Dudzik M, Stręk AM (2020), ANFIS Kaur R et al. (2020), PSO Eberhart and Shi Y (2001) based DCNN, CS Rajabioun R (2011) based DCNN, ACO Dorigo M et al. (2006) based DCNN, Prof Tree (Höppner et al. 2018) based DCNN, and LLM (Caigny et al. 2018) based DCNN are taken for the evaluation.

4.4 Performance analysis

Performance of the proposed algorithm shown in the Fig. 3 with different training data percentages and with different feature dimensions. This analysis is carried out with different epochs of 10, 20, 30, 40 and 50.

4.4.1 Analysis with various training percentages

The analysis in terms of Accuracy is portrayed in Figure 3. It shows that the accuracy of the proposed FA- based DCCN is constant or increasing by increasing training percentage from 50 to 90 at different epochs 10, 20, 30, 40 and 50.

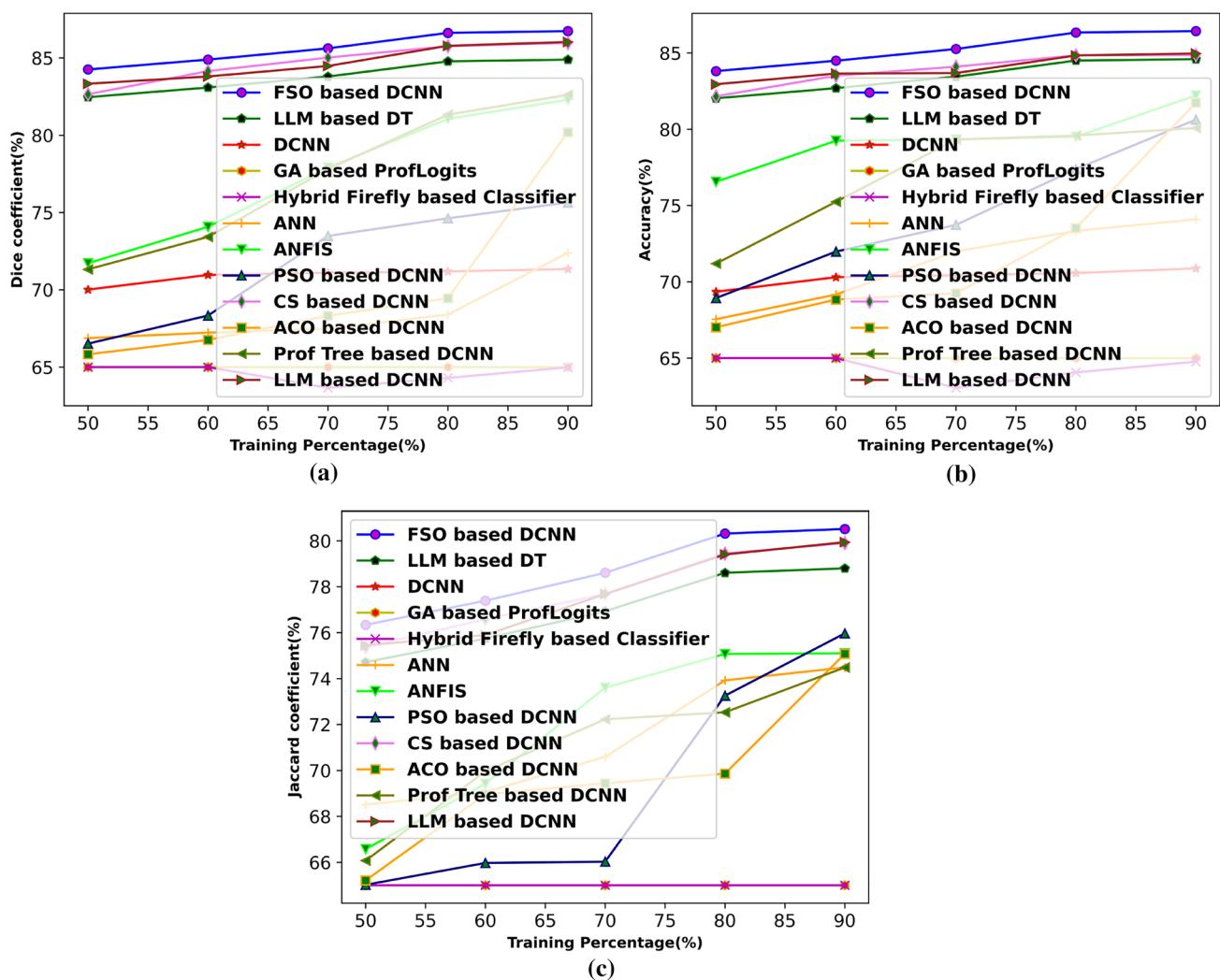


Fig. 7 Comparative analysis based on Training percentage in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

But in general by increasing training percentage at certain training percentage the accuracy will not increase even there is the possibility of decrease, which is not the case in the proposed FSO-based DCNN, which shows goodness of the algorithm.

4.4.2 Analysis with varied feature dimensions:

Figure 4 illustrates the performance of proposed FSO based DCNN with different feature dimensions. This analysis is carried out with different epochs of 10, 20, 30, 40 and 50.

The analysis is done based on the accuracy with varied number of features like 3, 6, 9, 12 and 15 given by the RFE feature selection algorithm. Start with top three ranked features. Figure 4 shows that for three features, the corresponding accuracy values computed by proposed FA-based DCNN with epochs 10, 20, 30, 40 and 50 are 73.4%, 79.6%, 79.9%,

0.83%, and 89.5%, respectively. Similarly, for 15 features, the corresponding accuracy values computed by proposed FSO-based DCNN with epochs 10, 20, 30, 40 and 50 are 93.1%, 93.1%, 93.1%, 93.1%, and 93.1%, respectively. Up to certain number of features the accuracy of FSO based DCNN is increased then onwards accuracy is constant by increasing number of features at various epochs, which shows the stability of the proposed algorithm with increase in number of features.

4.5 Comparative analysis

The comparative analysis of methods is carried out based on varying training data percentage and feature dimensions regarding dice coefficient, Jaccard coefficient, and Accuracy parameter.

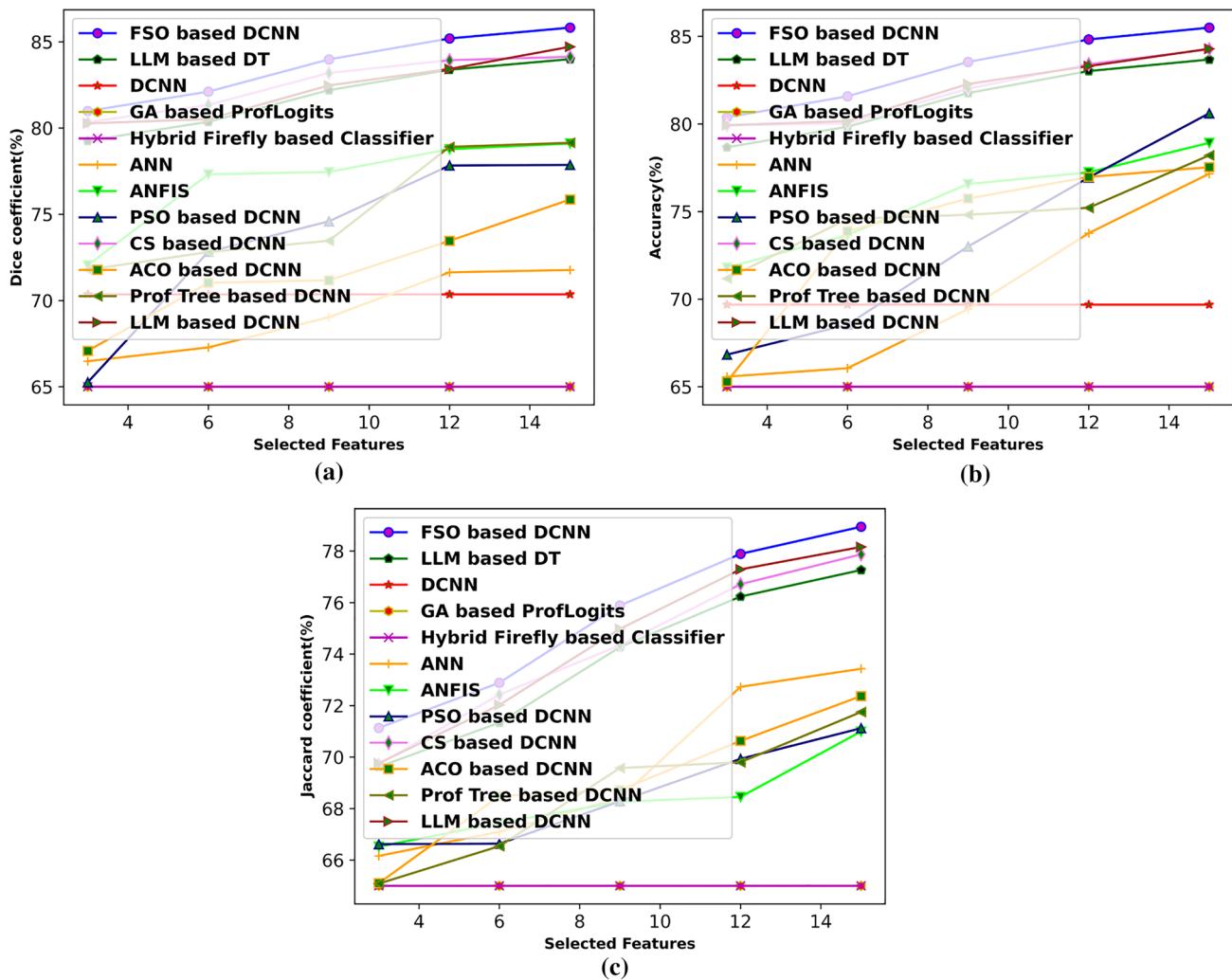


Fig. 8 Comparative analysis based on selected features in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

4.5.1 Analysis based on training percentage using dataset 1

4.5.1.1 Analysis based on training percentage The analysis based on dice coefficient, accuracy, and Jaccard coefficient concerning varying training data percentage is depicted in Fig. 5. The analysis in terms of dice coefficient is portrayed in Fig. 5a. When the training data percentage is 50, then the corresponding dice coefficient values computed by proposed FSO-based DCNN and existing LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN are 89.97%, 87.24%, 87.13%, 74.79%, 78.89%, 82.2%, 87.46%, 79.54%, 87.35%, 77.23%, 88.46%, and 88.46%, respectively. Figure 5b represents the accuracy, and Fig. 5c represents the Jaccard coefficient graph. For both the metrics, the proposed system has improved performance

compared to the existing techniques. With the increase in training percentage, the performance of all the methods was improved.

4.5.1.2 Analysis based on feature dimension The analysis based on dice coefficient, accuracy, and Jaccard coefficient concerning varying training data percentage is depicted in Fig. 6. The analysis in terms of accuracy is portrayed in Fig. 6b. When the selected feature is 4, then the corresponding dice coefficient values computed by proposed FSO-based DCNN and existing LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN is 85.82%, 78.16%, 77.22%, 73.44%, 73.44%, 74.92%, 80.86%, 73.60%, 80.88%, 77.37%, 82.47%, 78.88%, respectively. Figure 6a represents the dice coefficient, and Fig. 6c represents the Jaccard coefficient graph. For both the met-

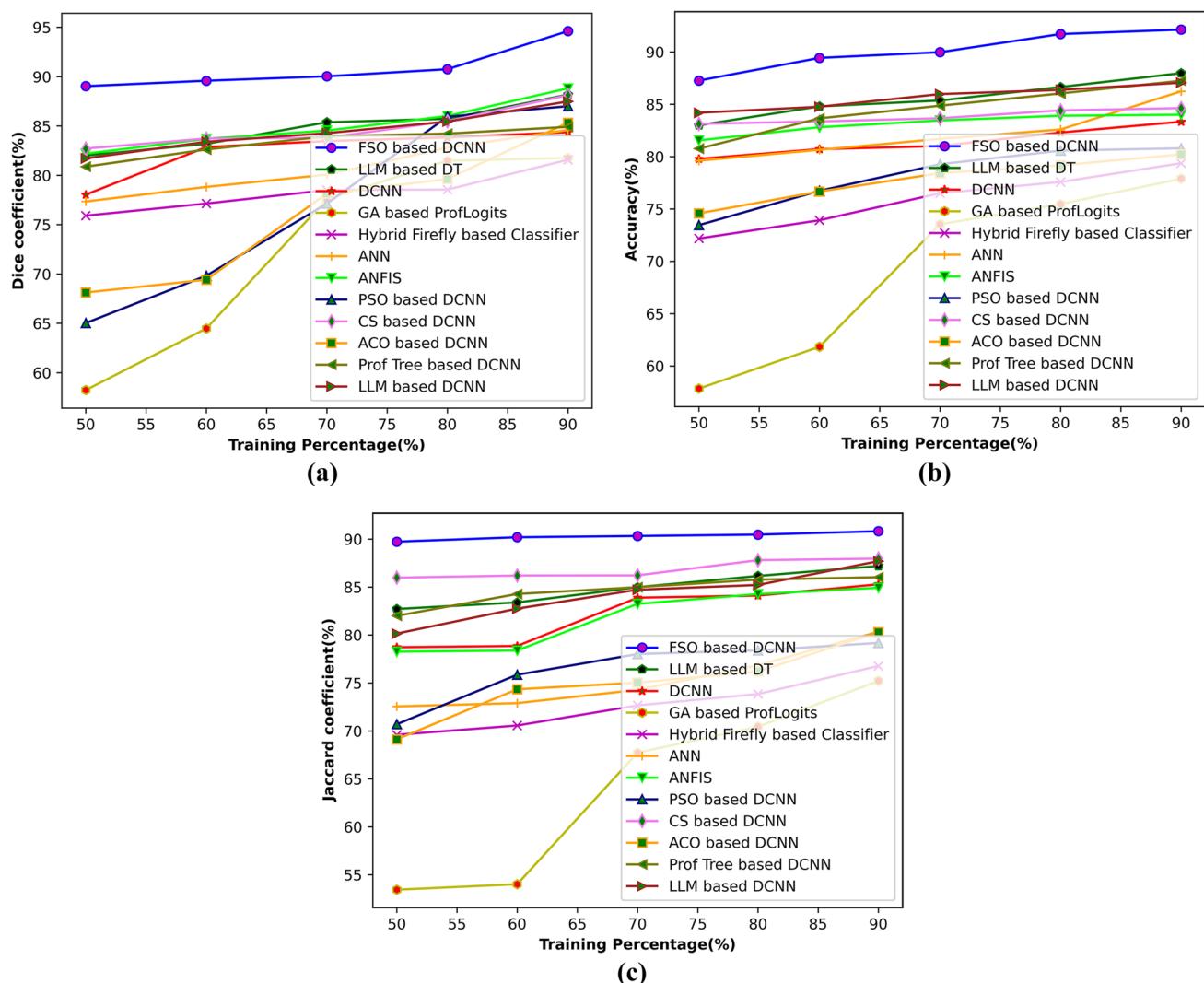


Fig. 9 Comparative analyses based on Training percentage in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

rics, the proposed system has improved performance compared to the existing techniques.

4.5.2 Analysis based on using dataset 2

4.5.2.1 Analysis based on training percentage Figure 7 depicts the analysis of the metrics, like the Dice coefficient, accuracy, and Jaccard coefficient based on the training percentage. Figure 7c shows the analysis of the Jaccard coefficient based on the training percentage. When the training percentage is 90, the Jaccard coefficient attained by the proposed FSO-based DCNN is 80.5%. The existing methods, like LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN, are 78.79%, 65%, 65%, 65%, 74.48%, 75.10%, 75.96%, 79.90%, 75.07%, 74.48%,

and 79.93%, respectively. Figure 7a represents the dice coefficient, and Fig. 7b represents the accuracy graph. For both the metrics, the proposed system has improved performance compared to the existing techniques.

4.5.2.2 Analysis based on feature selection This section provides the feature dimension analysis results as presented in Fig. 8 by varying selected features. Figure 8a show the result of the analysis based on dice coefficient; when the selected features is 14, the values attained from the existing methods like GA LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN, are 83.99%, 70.35%, 65%, 65%, 71.76%, 79.10%, 77.85%, 84.12%, 75.85%, 79.14%, and 84.71% and the proposed FSO-based DCNN is 85.81%. Figure 8b represents the accuracy, and

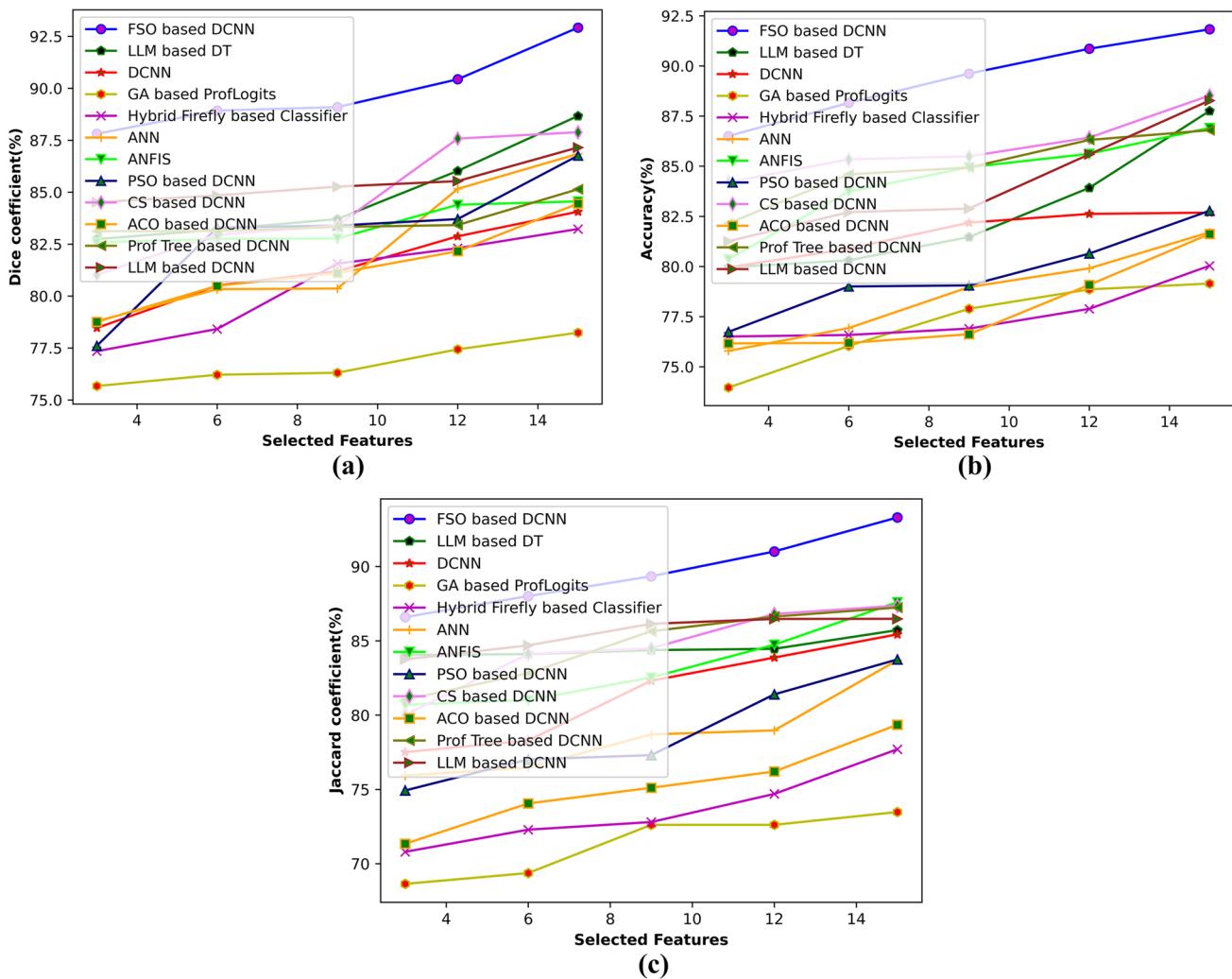


Fig. 10 Comparative analyses based on selected features in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

Fig. 8c represents the Jaccard coefficient graph. For both the metrics, the proposed system has improved performance compared to the existing techniques.

4.5.3 Analysis based on using dataset 3

4.5.3.1 Analysis based on training percentage In this section, the analysis of the proposed FSO-based DCNN is carried out by varying the training percentage. Figure 9 shows the comparative performance of the proposed FSO-based DCNN classifier based on the different training percentage values. Figure 9a shows the comparative analysis of the proposed FSO-based DCNN based on the Dice coefficient for varying training percentages. When training percentage is 70, the existing models, like LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN

have the Dice coefficient as 85.37%, 83.46%, 77.57%, 78.50%, 80.06%, 84.52%, 77.12%, 83.79%, 78.15%, 83.95%, and 84.27% respectively. In contrast, the proposed FSO-based DCNN has a dice coefficient of 90.03%. Figure 9b represents the accuracy, and Fig. 9c represents the Jaccard coefficient graph. For both the metrics, the proposed system has improved performance compared to the existing techniques.

4.5.3.2 Analysis based on feature selection To evaluate the performance of the proposed classifier, a comparison is made with that of several existing methods based on varying the selected features. Figure 10 depicts the analysis of selected features based on the metrics like Dice coefficient, Accuracy, Jaccard coefficient. Figure 10b when the selected feature is 12, the corresponding accuracy values of the proposed FSO-based DCNN is 90.85%. The values of the existing method are 83.9%, 82.62%, 78.85%, 77.88%,

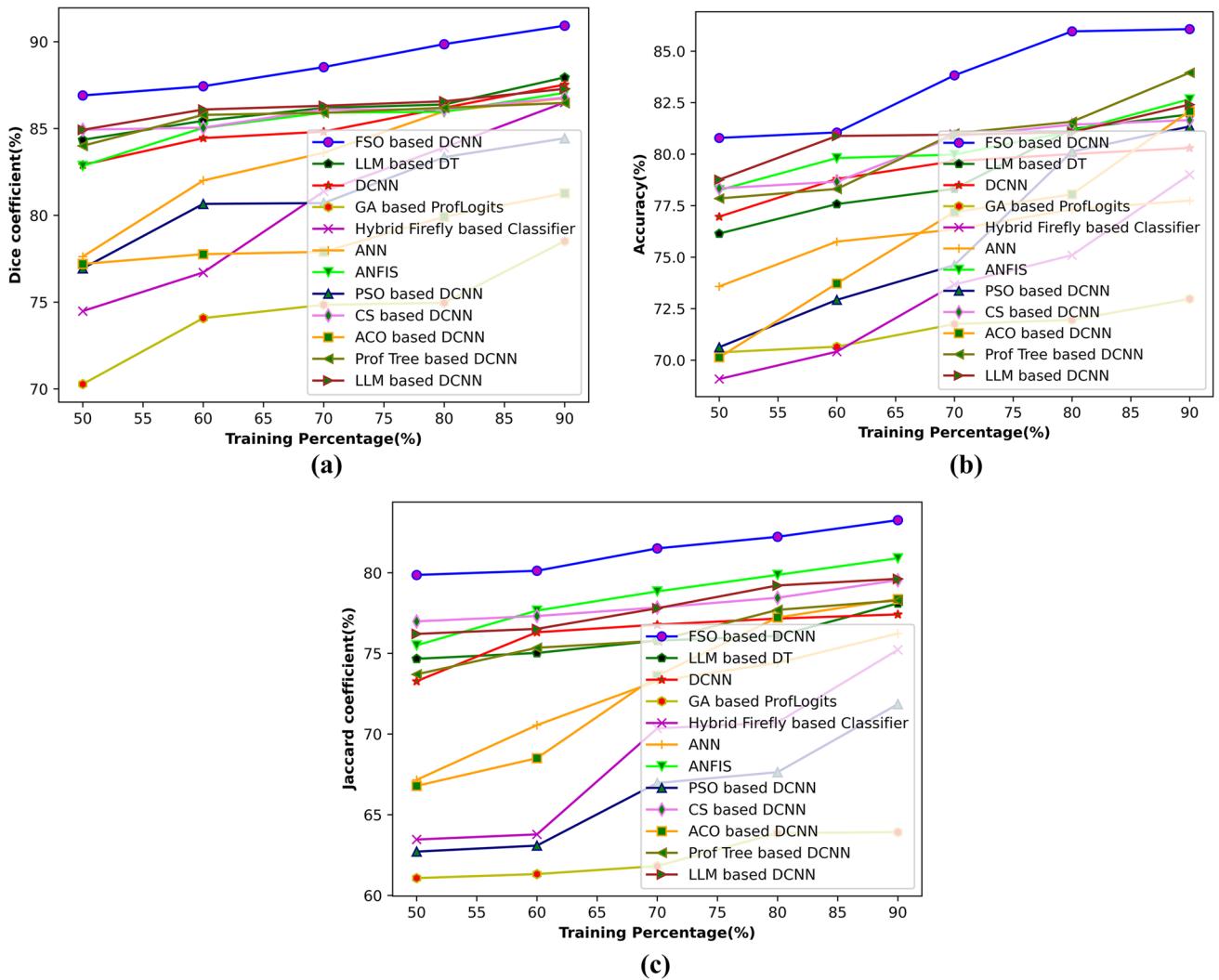


Fig. 11 Comparative analyses based on Training percentage in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

79.90%, 85.61%, 80.64%, 86.42%, 79.08%, 86.30%, and 85.59% for the LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN respectively. Figure 10a represents the dice coefficient, and Fig. 10c represents the Jaccard coefficient graph. For both the metrics the proposed system has improved performance compared to the existing techniques.

4.5.4 Analysis using dataset 4

4.5.4.1 Analysis based on training percentage This section provides the analysis of the proposed FSO-based DCNN is based on varying the training percentage. Figure 11 shows the comparative performance of the proposed FSO-based DCNN classifier based on the different training percentage

values. Figure 11c shows the comparative analysis of the proposed FSO-based DCNN based on the Jaccard coefficient for varying training percentages. When training percentage is 80, the existing models, like LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN have the Jaccard coefficient as 76.07%, 77.15%, 63.85%, 70.71%, 74.41%, 79.86%, 67.63%, 78.44%, 77.22%, 77.69%, and 79.20%. In contrast, the proposed FSO-based DCNN has the dice coefficient of 82.2%. Figure 11a represents the dice coefficient, and Fig. 11b represents the accuracy graph. For both the metrics, the proposed system has improved performance compared to the existing techniques.

4.5.4.2 Analysis based on selected feature Figure 14 depicts selected features based on the metrics like Dice

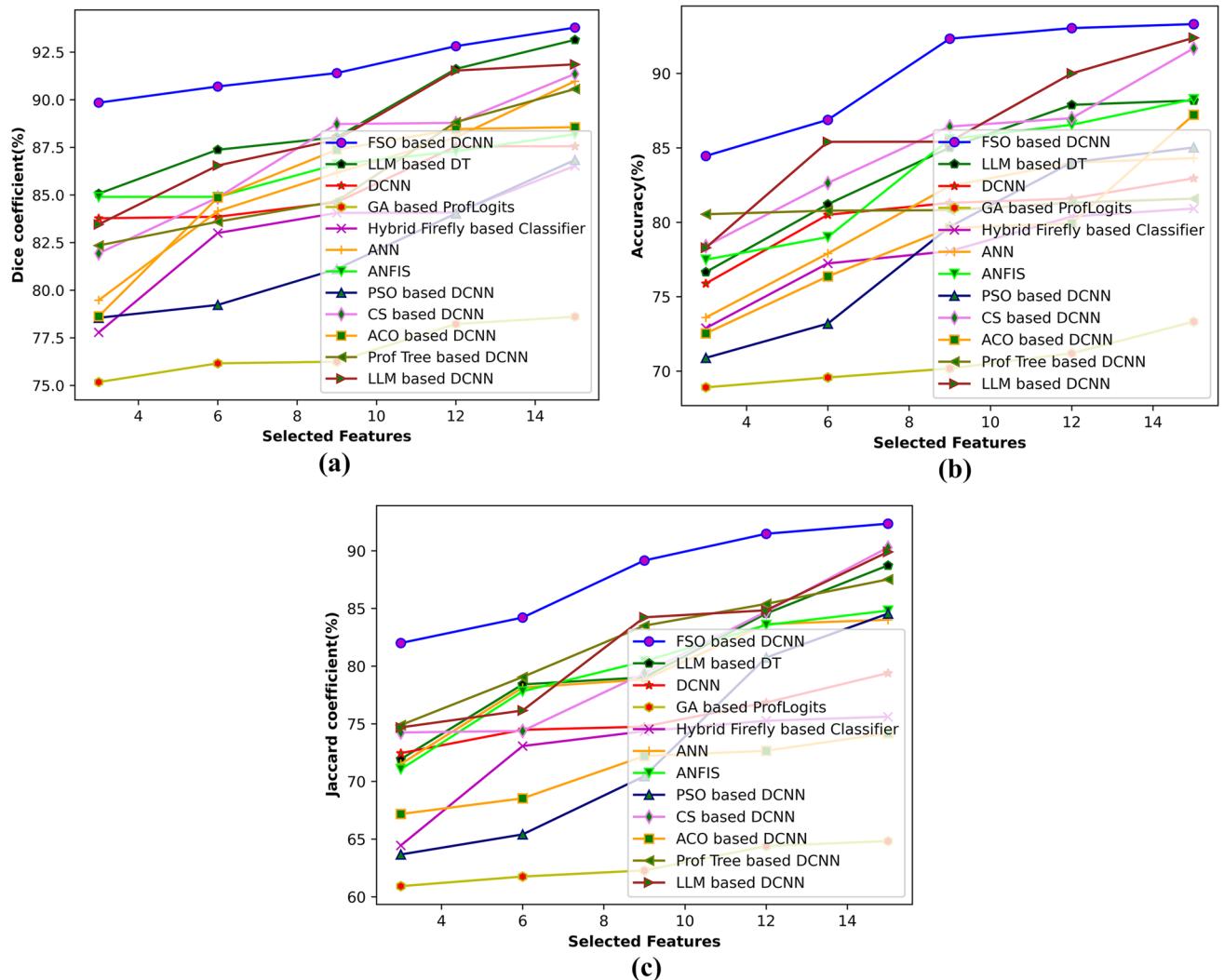


Fig. 12 Comparative analyses based on selected features in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

coefficient, Accuracy, Jaccard coefficient. In Fig. 12a, when the selected feature is 14, the corresponding Dice coefficient values of the proposed FSO-based DCNN is 93.8%. The values of the existing method are 93.1%, 87.6%, 78.6%, 86.5%, 91%, 88.2%, 86.8%, 91.4%, 88.6%, 90.6%, and 91.9% for the LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN respectively. Figure 12b represents the accuracy, and Fig. 12c represents the Jaccard coefficient graph. For both the metrics the proposed system has improved performance compared to the existing techniques.

4.6 Analysis using *k*-fold validation

4.6.1 Analysis using dataset 1

Figure 13 depicts the analysis of *k*-fold validation using dataset 1 based on the metrics like Dice coefficient, Accuracy, Jaccard coefficient. In Fig. 13a, when $k=7$, the corresponding Dice coefficient values of the proposed FSO-based DCNN is 86.50%, and the values of the existing method are 80.77%, 83.64%, 74.37%, 78.50%, 82.49%, 84.37%, 80.10%, 85.88%, 75.33%, 83.75%, and 85.71% for the LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN,

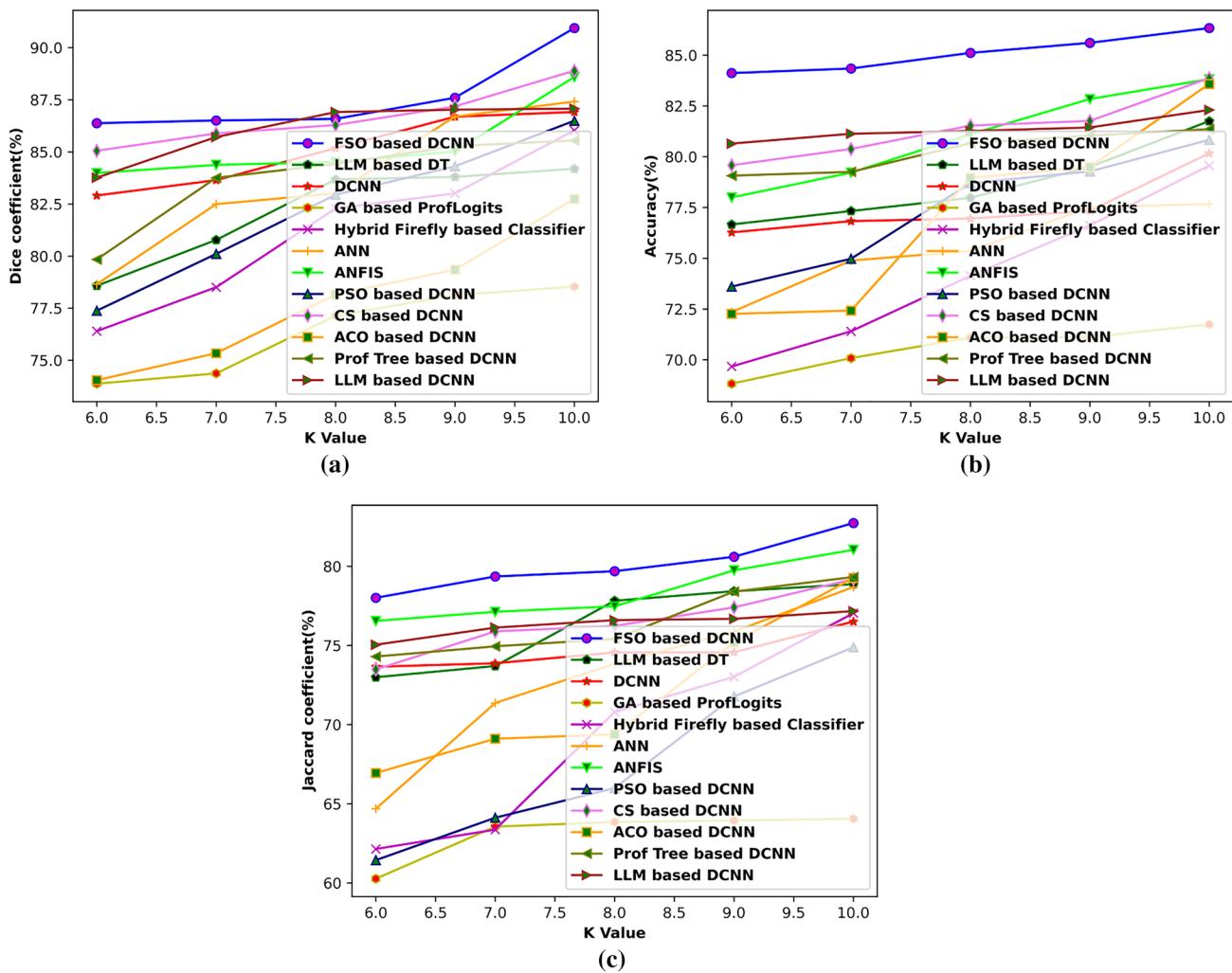


Fig. 13 Analysis based on k -fold in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN respectively. The analysis based on accuracy is shown in Fig. 13b, and the Jaccard coefficient is shown in Fig. 13c, respectively.

4.6.2 Analysis using dataset 2

Figure 14 depicts the analysis of k -fold validation using dataset 2 based on the metrics like Dice coefficient, Accuracy, Jaccard coefficient. In Fig. 14b, when $k=9$, the corresponding accuracy values of the proposed FSO-based DCNNN is 86.49%. The values of the existing method are 83.08%, 68.32%, 62.74%, 63.73%, 69.47%, 75.32%, 73.61%, 83.7%, 69.18%, 75.68%, and 82.61%, for the LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN respectively. The analysis based on dice coefficient

is shown in Fig. 14a and the Jaccard coefficient is shown in Fig. 14c respectively.

4.6.3 Analysis using dataset 3

Figure 15 depicts the analysis of k -fold validation using dataset 3 based on the metrics like Dice coefficient, Accuracy, Jaccard coefficient. From Fig. 15c, when $k=10$, the corresponding Jaccard coefficient values of the proposed FSO-based DCNNNN is 94.80%, and the values of the existing method are 93.56%, 89.46%, 80.58%, 81.01%, 88.56%, 90.99%, 86.59%, 94.02%, 88.66%, 92.01%, and 93.2% for the LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN respectively. The analysis based on the dice coefficient is shown in Fig. 15a, and the accuracy is shown in Fig. 15b, respectively.

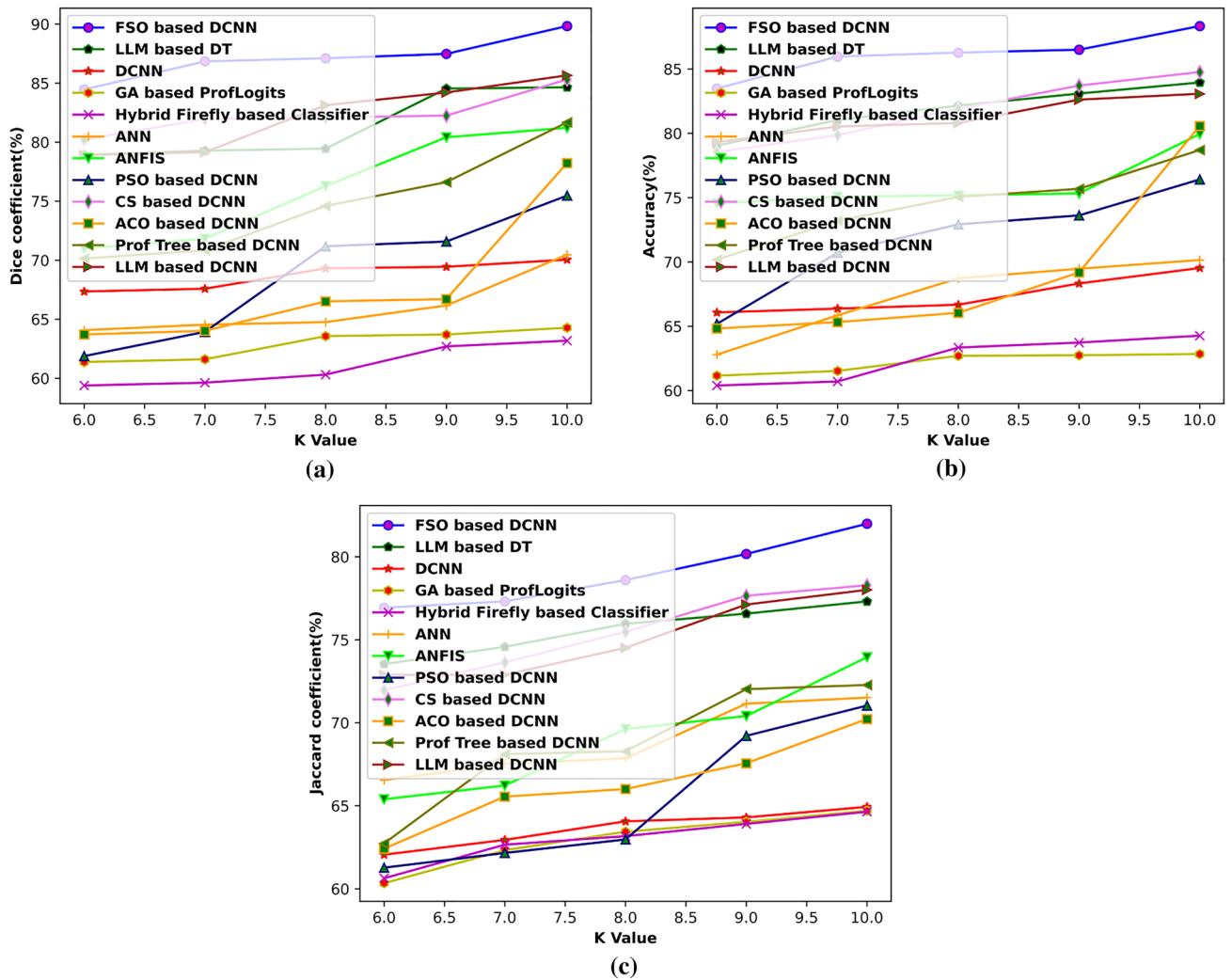


Fig. 14 Analysis based on k -fold in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

4.6.4 Analysis using dataset 4

Figure 16 depicts the analysis of k -fold validation using dataset 4 based on the metrics like Dice coefficient, Accuracy, Jaccard coefficient. In Fig. 16c, when $k=10$, the corresponding Jaccard coefficients values of the proposed FSO-based DCNN are 79.13%, and the values of the existing method are 76.93%, 75.29%, 62.95%, 74.49%, 75.39%, 77.19%, 70.13%, 76.31%, 75.08%, 78.13%, and 75.63% for the LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN respectively. The analysis based on the dice coefficient is shown in Fig. 16a, and the accuracy is shown in Fig. 16b, respectively.

4.7 Statistical analysis

The statistical analysis of the methods based on the Dice coefficient, accuracy, and Jaccard coefficient using four databases is deliberated in Table 1. The maximum Dice coefficient, accuracy, and Jaccard coefficient are obtained using the Telecom customer database. The proposed FSO-based DCNN produces the maximal Dice coefficient value is 94.61%. In contrast, the Dice coefficient of the existing LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN, are 88.16%, 84.36%, 81.76%, 81.57%, 84.59%, 88.81%, 87.00%, 88.12%, 85.32%, 84.90%, and 87.49%, respectively. The proposed FSO-based DCNN

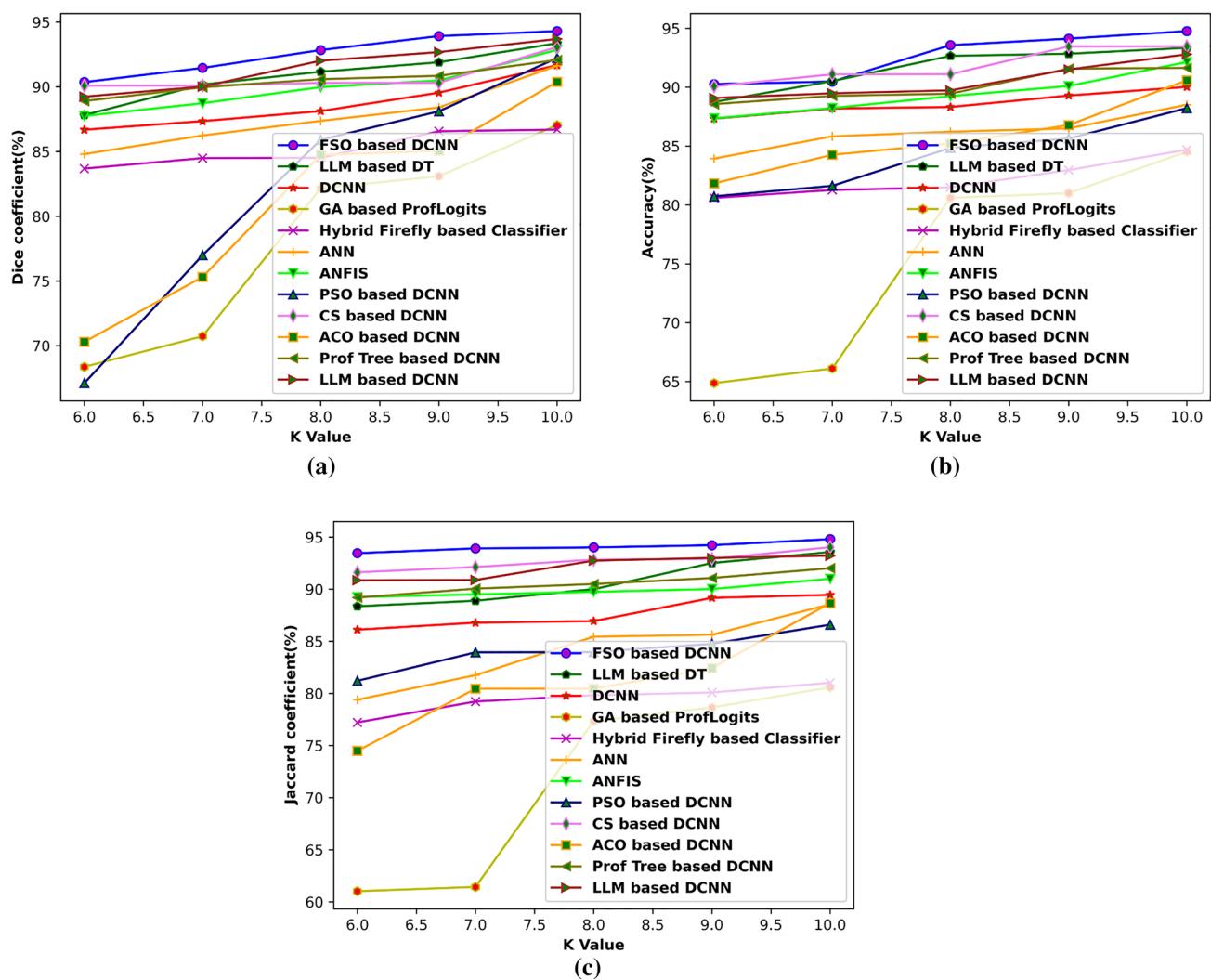


Fig. 15 Analysis based on k -fold in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

produces a maximal accuracy value of 94.76%. In contrast, the accuracy of the existing LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN is 93.34%, 90.02%, 84.52%, 84.69%, 88.51%, 92.14%, 88.22%, 93.47%, 90.59%, 91.64%, and 92.79%, respectively. The proposed FSO-based DCNN produces a maximal Jaccard coefficient value of 94.80%. In contrast, the Jaccard coefficient of the existing LLM-based DT, DCNN, GA-based ProfLogits, hybrid firefly based classifier, ANN, ANFIS, PSO based DCNN, CS-based DCNN, ACO based DCNN, Prof Tree-based DCNN, and LLM based DCNN are 93.57%, 89.46%, 80.58%, 81.02%, 88.56%, 91.00%, 86.60%, 94.03%, 88.67%, 92.01%, and 93.22%, respectively.

The proposed Churn prediction model using FSO-based DCNN, the feature selection is made by using the RFE method. It is easy to configure and effective in selecting the features for predicting the target variable. Then, the Spark architecture is used for predicting the churners. The Spark architecture is a platform for streaming the data in which the data processing is faster, and it is closer to real time processing. It also has faster memory computation. Then the classification is done by DCNN by the proposed FSO optimization algorithm. FSO has benefits like rapid convergence rate, global search capability, and multimodal optimization problems. Moreover, DCNN has high accuracy, which detects the important features without human supervision. It has the important feature of weight sharing for better classification.

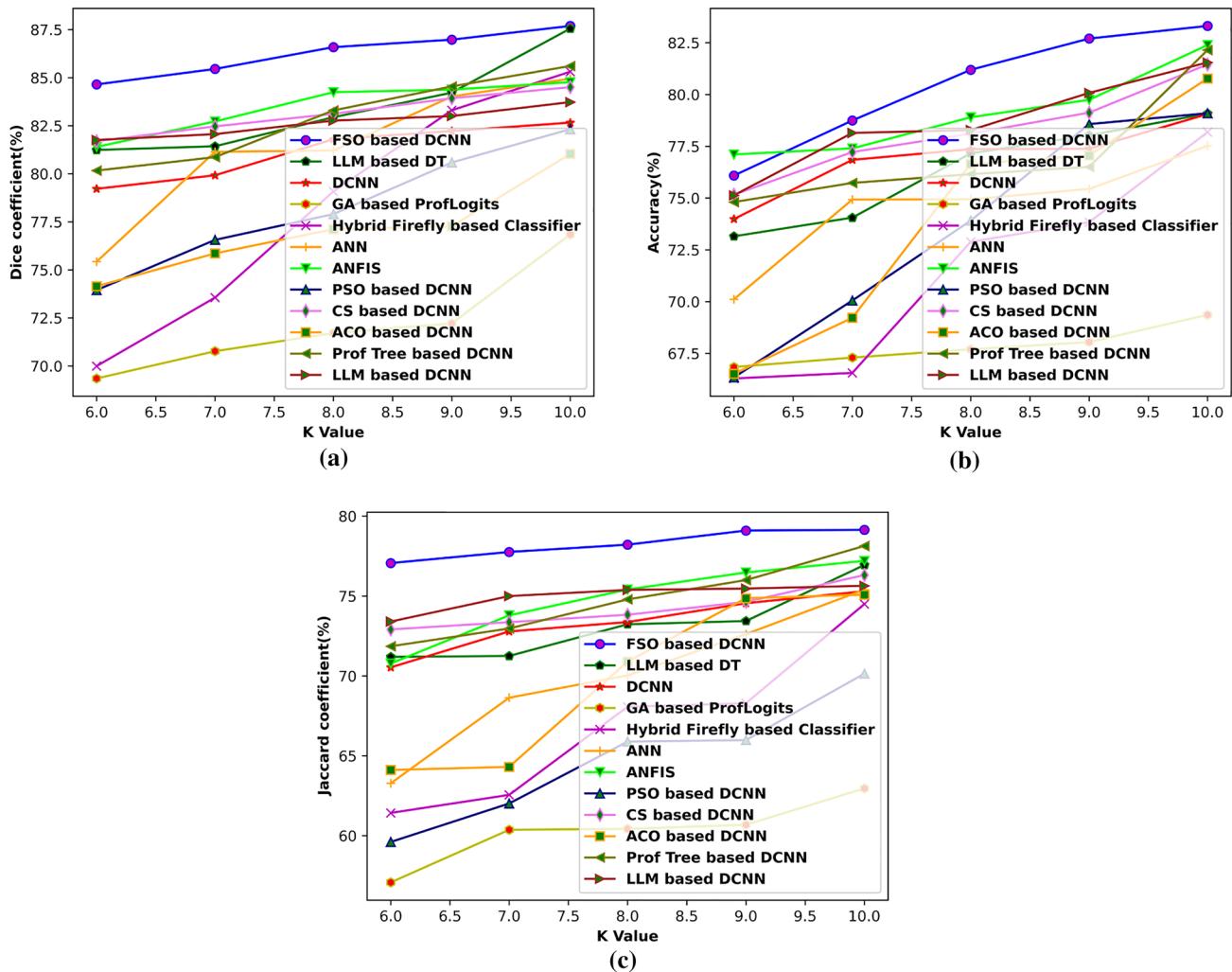


Fig. 16 Analysis based on k -fold in terms of a) Dice coefficient b) Accuracy c) Jaccard coefficient

5 Conclusion

This paper devises an effective churn prediction model using spark architecture for predicting the churners using the telecom data stored in the cloud. In this research, the spark architecture is employed to process a huge amount of telecom data and can handle the data in parallel. The proposed churn prediction model is processed in two stages, which involve feature selection and churn prediction. These steps are performed in the spark framework using initial and final nodes. When the data is provided to the initial node

of the spark then, the feature selection process is started. The feature selection is performed based on the RFE technique. After obtaining the selected features, the features are subjected to the DCNN for the churn prediction. Here, the DCNN is trained for tuning the optimal weights using a newly designed optimization algorithm named FSO algorithm, which is the integration of SMO and FA. The performance of the proposed FSO-based DCNN outperformed other existing methods in terms of dice coefficient, accuracy, Jaccard coefficient, with values 94.61%, 94.76%, and 94.80%, respectively.

Table 1 Statistical Analysis

Methods	Metrics	Proposed FSO-based DCNN	GA-based ProLogits based classifier	Hybrid firefly DCNN	LLM-based DT	ANN	ANFIS	PSO based DCNN	CS based DCNN	ACO based DCNN	Prof Tree based DCNN	LLM based DCNN
Dice coefficient	Training data	Best	94.61	88.16	84.36	81.76	81.57	84.59	88.81	87.00	88.12	85.32
	Mean	94.58	88.12	84.32	81.73	81.53	84.55	88.78	86.97	88.09	85.30	84.87
	Variance	0.03	0.04	0.04	0.03	0.04	0.04	0.03	0.03	0.03	0.02	0.03
	Best	92.13	87.97	83.33	77.89	79.36	86.23	84.00	80.79	84.63	80.21	87.24
	Mean	92.10	87.94	83.29	77.85	79.32	86.20	83.97	80.75	84.61	80.18	87.22
	Variance	0.03	0.03	0.04	0.04	0.04	0.03	0.03	0.04	0.02	0.03	0.02
k-fold	Best	94.30	93.37	91.67	87.04	86.69	91.62	92.84	92.23	93.07	90.38	92.07
	Mean	94.26	93.33	91.63	87.00	86.65	91.59	92.81	92.20	93.04	90.36	92.05
	Variance	0.04	0.04	0.04	0.04	0.04	0.03	0.03	0.03	0.03	0.02	0.02
	Best	90.82	87.20	85.28	75.22	76.77	80.33	84.93	79.17	87.97	80.29	86.03
	Mean	90.78	87.16	85.24	75.19	76.74	80.30	84.91	79.15	87.94	80.25	85.99
	Variance	0.04	0.04	0.04	0.03	0.03	0.03	0.02	0.02	0.03	0.04	0.04
Selected features	Best	92.92	88.67	84.06	78.24	83.23	86.84	80.00	86.76	87.88	84.45	85.14
	Mean	92.89	88.65	84.04	78.22	83.20	86.81	-0.04	86.73	87.84	84.42	85.11
	Variance	0.03	0.02	0.02	0.02	0.03	0.03	0.04	0.03	0.04	0.03	0.03
	Best	94.76	93.34	90.02	84.52	84.69	88.51	92.14	88.22	93.47	90.59	91.64
	Mean	94.73	93.31	89.99	84.49	84.66	88.48	92.10	88.18	93.43	90.55	91.61
	Variance	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.03
Jaccard coefficient	Training data	Best	91.83	87.75	82.69	79.15	80.03	81.70	86.91	82.78	88.52	81.62
	Mean	91.81	87.73	82.67	79.13	80.00	81.67	86.89	82.76	88.50	81.59	86.77
	Variance	0.02	0.02	0.02	0.03	0.02	0.03	0.02	0.02	0.02	0.03	0.02
	Best	93.30	85.72	85.43	73.48	77.70	83.69	87.59	83.74	87.36	79.35	87.24
	Mean	93.27	85.69	85.40	73.45	77.67	83.65	87.57	83.72	87.32	79.31	87.22
	Variance	0.03	0.03	0.03	0.03	0.03	0.04	0.02	0.02	0.04	0.04	0.02
k-fold	Best	94.80	93.57	89.46	80.58	81.02	88.56	91.00	86.60	94.03	88.67	92.01
	Mean	94.78	93.55	89.44	80.56	81.00	88.53	90.97	86.57	94.01	88.65	92.00
	Variance	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.02	0.02	0.01	0.01

References

- Ahmed AA, linen DM (2017) A review and analysis of churn prediction methods for customer retention in telecom industries. In: proceedings of 4th International Conference on Advanced Computing and Communication Systems, p 1–7
- Amin A et al (2017) Customer churn prediction in the telecommunication sector using a rough set approach. Neurocomputing 237:242–254
- Amina A et al (2019) Customer churn prediction in telecommunication industry using data certainty. J Bus Res 94:290–301
- Arora S, Singh S (2013) The firefly optimization algorithm convergence analysis and parameter selection. Int J Comput Appl 69(3):48
- Athanassopoulos AD (2000) Customer satisfaction cues to support market segmentation and explain switching behaviour. J Bus Res 47(3):191–207
- Bansal JC et al (2014) Spider monkey optimization algorithm for numerical optimization. Memetic Comput 6(1):31–47
- Bi W et al (2016) A big data clustering algorithm for mitigating the risk of customer churn. IEEE Trans Ind Inform 12(3):1270
- Caigny D et al (2018) A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. Eur J Oper Res 269(2):760–772
- Chan CCH (2008) Intelligent value-based customer segmentation method for campaign management: a case study of automobile retailer. Expert Syst Appl 34:2754–2762
- Churn in Telecom's dataset (2019). <https://www.kaggle.com/becksddf/churn-in-telecoms-dataset>. Accessed 12 Feb 2020
- Coussement, et al (2017) A comparative analysis of data preparation algorithms for customer churn prediction: a case study in the telecommunication. Decis Support Syst 95:27–36
- Dorigo M et al (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
- Dudzik M, Stręk AM (2020) ANN architecture specifications for modelling of open-cell aluminum under compression. Math Prob Eng. <https://doi.org/10.1155/2020/2834317>
- Eberhart, Shi Y (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, Korea (South) 2001, p 81–86
- Gustafsson A et al (2005) The effects of customer satisfaction, relationship commitment dimension, and triggers on customer retention. J Market 69(4):210–218
- Guyon I et al (2002) Gene selection for cancer classification using support vector machines. Mach Learn 4:389–422
- Hadden J et al (2007) Computer assisted customer churn management: State-of-the-art and future trends. Comput Oper Res 34(10):2902–2917
- Höppner S et al (2018) Profit driven decision trees for churn prediction. Eur J Operat Res 284:920–933
- Kaur R et al (2020) An improved and adaptive approach in ANFIS to predict knee diseases. Int J Healthc Inform Syst Inform 15(2):22–33
- Kusuma I et al. (2016) Design of intelligent k -means based on spark for big data clustering. In: proceedings of International Workshop on Big Data and Information Security, p 89–96
- Lazarov V, Capota M (2007) Churn prediction. Bus Anal Course TUM Comput Sci 33:34
- Nijhawan VK et al. (2018) An analytical implementation of CART using RStudio for churn prediction. In: information and communication technology for competitive strategies, p 109–120
- Rajabioun R (2011) Cuckoo optimization algorithm. Appl Soft Comput 11(8):5508–5518
- Seret A et al (2012) A new SOM-based method for profile generation: theory and application in direct marketing. Eur J Operat Res 220:199–209
- Singh M et al. (2018) Comparison of learning techniques for prediction of customer churn in telecommunication. In: proceedings of 28th International Telecommunication Networks and Applications Conference, p 1–5.
- Stripling E et al (2018) Profit maximizing logistic model for customer churn prediction2 using genetic algorithms. Swarm Evol Comput 40:116–130
- Telco customer churn (2019). <https://www.kaggle.com/blastchar/telco-customer-churn>. Accessed 12 Feb 2020
- Telecom churn (cell2cell) (2019). <https://www.kaggle.com/jpacse/datasets-for-churn-telecom>. Accessed 12 Feb 2020
- Telecom customer (2019). <https://www.kaggle.com/abhinav89/telecom-customer>. Accessed 12 Feb 2020
- Trotta A, et al. (2013) Re-establishing network connectivity in post-disaster scenarios through mobile cognitive radio networks. In: proceedings of 12th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), p 18–25
- Trotta A et al (2015) Connectivity recovery in post-disaster scenarios through cognitive radio swarms. Comput Netw 91:68–89
- Tu F et al (2017) Deep convolutional neural network architecture with reconfigurable computation patterns. IEEE Trans VLSI Syst 25(8):2220–2233
- Verbeke W et al (2011) Building comprehensible customer churn prediction models with advanced rule induction techniques. Expert Syst Appl 38:2354–2364
- Verbeke W et al (2012) New insights into churn prediction in the telecommunication sector: a profit driven data mining approach. Eur J Oper Res 218(1):211–229
- Verhoef PC (2003) Understanding the effects of customer relationship management efforts on customer retention and customer share development. J Market 67(4):30–45
- Zema NR, et al. (2017) CUSCUS: An integrated simulation architecture for distributed networked control systems. In: proceedings of 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), p 287–292

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

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”). Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com