Date of publication JUN-30, 2025, date of current version APR-11, 2025.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.24.2.4016

Convolutional Neural Network based Di-Strategy Cheetah Optimization Algorithm for Automatic Diabetes Prediction

PANKAJ KUMAR[1], MERIN THOMAS[2], MANOHAR MANUR[3], ALOK KUMAR PANI[4]
[1]Department of Computer Science and Engineering, Government Engineering College, Vaishali, India
[2]School of Computer science, RV University, Bengaluru, India
[3]Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bengaluru, India
[4]Birla School of Applied Sciences, Birla Global University, Bhubaneswar, India

Corresponding author: Pankaj Kumar (e-mail: paku9780@gmail.com).

ABSTRACT Diabetes is a chronic metabolic disease characterized by elevated blood sugar levels. Diabetes prediction leverages patient data to assess the risk of developing the condition, facilitating early diagnosis and intervention. However, existing models struggle to capture the complex interactions between risk factors due to limited feature representation, leading to inaccurate predictions. This research proposes a Convolutional Neural Network-based Di-Strategy Cheetah Optimization Algorithm (CNN-DS-COA) for automatic diabetes prediction using patient data. The COA is enhanced with tent chaotic mapping and an adaptive search agent, which improves population diversity distribution and convergence speed. Initially, the Pima Indians Diabetes Database (PIMA) and Germany datasets are employed to evaluate the performance of CNN-DS-COA. Min-max normalization is applied to scale the data within a uniform range while preserving relationships among values. The CNN is then used for automatic diabetes prediction, with DS-COA fine-tuning the CNN's parameter values effectively using two strategies. The proposed CNN-DS-COA achieves superior accuracy, with 99.90% and 99.72% on the PIMA and Frankfurt Hospital, Germany datasets, respectively, outperforming existing methods such as stacked ensemble approaches and statistical predictive models.

KEYWORDS adaptive search agent; di-strategy cheetah optimization algorithm; convolutional neural network; min-max normalization; tent chaotic mapping.

I. INTRODUCTION

Diabetes is a severe metabolic disorder characterized by elevated sugar levels in the body. It arises when the insulin hormone produced by the pancreas is either insufficient or not produced at all. Insulin's role is to transport glucose from the blood to tissues, cells, and organs [1]. Diabetes is caused by numerous factors, including chemical or toxic substances in food, poor diet, bacterial or viral infections, environmental pollution, lifestyle changes, eating habits, and obesity. It can lead to renal and retinal failure, pathological damage, coma, and weight loss [2]. There are four types of diabetes commonly seen in clinical settings: Type 1, Type 2, prediabetes, and gestational diabetes. Gestational diabetes occurs during pregnancy and typically resolves after childbirth. Type 1 diabetes, also known as juvenile diabetes, is an autoimmune disorder where the body's immune system mistakenly targets and destroys the insulin-producing cells in the pancreas. Insulin is a key hormone responsible for regulating blood sugar levels [3, 4]. Type 2 diabetes, on the other hand, is a metabolic disorder caused by a combination of genetic and lifestyle factors, such as a lack of physical activity [5, 6]. It also tends to precede gestational diabetes in certain ethnic populations and women, affecting both children and adolescents [7].

Insulin is used to effectively manage diabetes symptoms by regulating blood sugar levels. This treatment helps maintain overall health and prevent complications associated with the condition [8, 9]. Various factors such as Body Mass Index (BMI), age, blood pressure, and glucose levels are incorporated in clinical settings and research studies to accurately predict diabetes at an early stage [10, 11]. By assessing these factors, predictive approaches can identify individuals at high risk of developing diabetes, enabling early intervention and lifestyle modification. In this research, benchmark datasets like the Pima Indian Diabetes Dataset (PIMA) and Frankfurt Hospital in Germany are utilized, both of which are available in the University of California at Irvine (UCI) repository [12, 13]. Deep Learning (DL) classifiers outperform Machine Learning (ML) classifiers due to their ability to automatically learn and extract complex features from large datasets, as well as model intricate patterns through neural networks [14, 15]. However, existing models struggle to capture the complex interactions between risk factors due to limited feature representation, resulting in inaccurate predictions. To address these challenges, the CNN-DS-COA model is proposed to automatically predict diabetes from patient data. The proposed approach improves accuracy by leveraging CNN to automatically learn and integrate complex features from patient data, with CNN parameters optimized using DS-COA, which enhances prediction performance.

The primary contribution of this research is described as follows:

- CNN leverages its ability to automatically learn features from intricate diabetes data, enhancing predictive accuracy
- COA is improved using two strategies, namely: tent chaotic mapping helps to initialize the population and provide a more uniform distribution; an adaptive search agent is used to calculate the search agents, balancing both global and local abilities
- DS-COA optimizes the CNN parameter values and enhances prediction performance effectively

This research paper is organized as follows: Section 2 provides the literature survey of existing approaches; Section 3 explains a detailed description of the proposed methodology. Section 4 provides the experimental results and the conclusion of this research paper is given in Section 5.

II. LITERATURE SURVEY

The different existing methods for diabetes prediction were presented in detail along with their advantages and disadvantages

Huma Naz and Sachin Ahuja [16] presented a Synthetic Minority Oversampling Technique (SMOTE) with Sequential Minimal Optimization (SMO) to predict diabetes. Initially, the pre-processing of data was performed by utilizing a SMOTE approach and then an SMO classifier was performed. The pre-processing output was provided to SMO which enhances the classifier performance. However, SMO struggled with complex and non-linear relationships because it was based on linear decision boundaries which led to suboptimal performance.

Sivashankari R. et al. [17] implemented a Stacked ensemble approach, namely, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), and boosting for diabetes prediction. The implemented approach was based on the automated progress of early prediction. Each approach predicted the probability for each class successfully which results in an effective set of predicted probabilities. Nevertheless, the implemented approach suffers from overfitting because of combining multiple approaches that reduces generalization and leads to a complex ensemble model.

Usama Ahmed et al. [18] developed a fused Machine Learning (ML) that contains SVM and Artificial Neural Network (ANN) to predict diabetes. Two generally employed ML approaches were combined in the developed model by utilizing fuzzy logic. The output from these approaches serves as input membership for the fuzzy model while fuzzy logic evaluates whether a diabetes diagnosis was negative or positive. However, the developed approach faces constraints due to the challenges of effectively tuning the hyperparameters of both models which leads to suboptimal performance.

Mahendra Kumar Gourisaria et al. [19] introduced an ensemble model, namely, SVM, NB, Random Forest (RF) and ANN to predict diabetes. The Principle Component Analysis (PCA) and Linear Discriminative Analysis (LDA) were used for dimensionality reduction which enhances the performance and minimizes the computational cost. The significance of every feature of each dataset and their effect on training was assessed which enhances the model performance. Nevertheless, the introduced approach struggled with diabetes prediction because of constraints in managing complex and non-linear relationships in data.

Kebira Azbeg et al. [20] suggested a statistical predictive approach to predict diabetes. Initially, the data was filtered and scaled by utilizing a noise-invariant data expansion. Once the data was gathered and stored, the suggested approach was analyzed and determined to obtain appropriate features. Then, an adaptive RF was established to train the data and classify the diabetes using the suggested approach. However, the suggested approach struggled with managing interactions among multiple variables which suffer from poor performance when applied to different patient populations.

The overall analysis discovered that existing methods faced limitations like difficulties in managing complex interactions, non-linear relationships, issues with generalization and overfitting, and limited feature representation which leads to inaccurate performance. To solve this issue, CNN-DS-COA is proposed to automatically predict diabetes by effectively capturing complex patterns and optimizing the model parameters. This approach enhances prediction accuracy and generalization.

III. PROPOSED METHODOLOGY

In this research, the CNN-DS-COA is used for automatic diabetic prediction. Initially, the PIMA and Frankfurt Hospital, Germany datasets are utilized to evaluate the proposed approach performance. Min-max normalization is employed to normalize the data in a uniform range. Then, the CNN is performed with the help of hyperparameter optimization DS-COA. Fig. 1 shows a block diagram of the overall process.

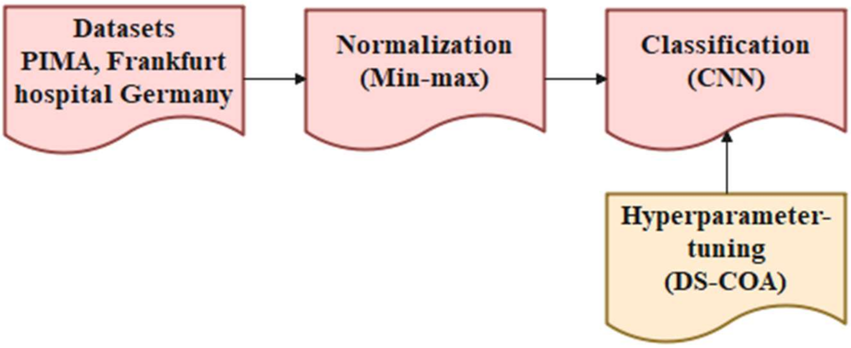

Figure 1. Block diagram of the overall process

A. DATASET

In this research, the PIMA [21] and Frankfurt Hospital, Germany datasets [22] are used to evaluate the model performance. The PIMA dataset contains 768 instances which is combination of 500 diabetic and 268 non-diabetic instances. It has 9 columns, and the final column is an output class which contains 0 and 1; 0 represents diabetic negative and 1 denotes positive. The other dataset is gathered from the hospital in Frankfurt, Germany and it involves 9 attributes and 2000 records. Both datasets have the same predictor variables such as Body Mass Index (BMI), pregnancies, skin thickness, age, insulin level, diabetes pedigree function, glucose, blood pressure, and outcome. Table 1 shows a description of both datasets. Then, the obtained data is fed into the pre-processing stage to normalize in a uniform range.

Table 1. Description of both datasets

Predictors	Range	Missing values	Data type
BMI	0-67.1	0	Real
Pregnancies	0-17	0	Integer
Skin thickness	0-99	0	Integer
Age	21-81	0	Integer
Insulin Level	0-846	0	Integer
Diabetes pedigree function	0.078-2.42	0	Real
Glucose	0-199	0	Integer
Blood pressure	0-122	0	Integer
Outcome	N/A	(Non-diabetes, 500 and diabetes, 268)	Boolean

B. PRE-PROCESSING

After obtaining the data, normalization is applied in this stage because predictors such as insulin, BMI, and glucose have varying scales. Min-max normalization [23] scales every feature to the same range [0,1] which assists in ensuring that no single feature controls the others due to its scale. It preserves the relationships among values and maintains the proportion of the original data. Initially, this approach subtracts the minimum value from point of data X and divides by its range. The mathematical formula for the min-max normalization is expressed in equation (1).

$$X_{norm} = (X - X_{min})/(X_{max} - X_{min}), \quad (1)$$

where X represents the original value of the feature (predictor), X_{min} denotes the minimum value of the feature, X_{max} indicates the maximum value of the feature, and X_{norm} determines the normalized value of X respectively. It conserves the relationships between data points by sustaining the original distribution. Also, it provides a faster convergence during training by ensuring all features equally contribute to learning neural networks.

C. CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN [24] is used to predict diabetes because it automatically learns intricate patterns and feature interactions in the data. CNN captures spatial relationships and hierarchies which assists in understanding the correlations among various medical features. CNN is identical to the Multi-Layer Perceptron (MLP) which has different convolutional layers, pooling layers, batch normalization layers, and Fully Connected (FC) layers.

Convolutional Layers

In CNN, the most significant layer is the convolutional layer, and it has a collection of kernels and filters. A convolution operator is used between the data and filter to identify the features. It employs various kernels to convert the data and intermediate feature maps in the structure of convolutional layers to establish various feature maps. The 2-D discrete convolutional process for input data and filter k is represented in equation (2).

$$F(i,j) = (f * k)(i,j) = \sum_m \sum_n k(m,n) f(i-m, j-n), \quad (2)$$

where $F(i,j)$ indicates output of convolutional operator at position (i,j), $(f * k)$ represents convolutional process among two functions or matrix f and k, and k denotes convolution filter or kernel.

Activation Function

It is a mathematical function utilized in neural networks to introduce non-linearity, allowing the model to learn complex patterns by transforming the input signal of each neuron. Rectified Linear Unit (ReLU) allows for fasting training by establishing non-linearity though avoiding a vanishing gradient for the positive input value. The mathematical equation for ReLU function is expressed in equation (3).

$$F(x)_{ReLU} = max\ (0, x). \quad (3)$$

ReLU is a monotonous function that outputs 0 for negative inputs and yields the input value x for positive inputs. It generates an output range between 0 and infinity.

Batch Normalization (BN) layer

It is used to enhance the stability of the neural network by normalizing each layer's input. It is processed by shifting and scaling the input to have zero mean and unit variance in a min-batch. It assists in minimizing internal covariance shifts and enables higher learning rates. Also, it produces trainable parameters for shifting and scaling which enhances the model performance.

Pooling layer

The purpose of the pooling layer is to reduce the feature map size which is established. Here, the average pooling is used which enhances the number of convolution layers and maximizes exponentially with network parameters. Average pooling estimates the patch's average value on the map. It is utilized to establish a map for pooling and includes a small translation invariance.

FC layer

It is used for the classification portion and the features produced in an extraction procedure are converted into a 1D vector before passing through the FC layer. It is created by numerous nodes or neurons and each neuron is associated with the next layer. Each FC layer has a non-linear activation like ReLU and sigmoid is used for classification which is formulated in equation (4).

$$Sigmoid(x) = \frac{1}{1+e^{-x}}. \tag{4}$$

Also, the dropout layer is utilized to avoid divergence which enhances generalization by minimizing the dependency on certain neurons. The spatial feature enables to capture the intricate patterns which leads to reliable and accurate prediction of diabetes. Then, the COA is performed to effectively optimize the parameter values.

D. HYPERPARAMETER TUNING

After performing CNN, the hyperparameter tuning is established using DS-COA to optimize the model performance by determining the best set of parameters like batch size, learning rate, and a number of layers. Appropriate tuning increases the CNN's ability to generalize from training data which enhances accuracy and minimizes overfitting issues. While optimizing the hyperparameters, overfitting is minimized which ensures optimal generalization. The dropout rate 0.3 avoids excessive dependency on specific neurons that minimizes memorization. Learning rate 0.02 and AdaGrad optimizer provide stable training and prevent large weight updates. Batch size 64 manages a balance among stability and noise whereas filter size 3 × 3 and the number of layers of 4 ensure that the model learns appropriate patterns without complexity. Hence, DS-COA determines the optimal configuration which enhances model robustness and avoids overfitting. This progress enables the CNN to function effectively across different patients which results in reliable diabetes prediction. COA [25] is a meta-heuristic optimization method that attracts inspiration from cheetah-hunting strategies. It provides various benefits like minimal parameter adjustments, simplifying calculations, and attaining rapid convergence. Searching, attacking, and sitting and waiting are three common stages present in the COA. The primary hunting strategies used in COA involve a search strategy that provides a random search by updating the cheetah's new position based on its present location using equation (5).

$$X_{i,j}^{t+1} = X_{i,j}^{t} + \widehat{\gamma_{i,j}^{-1}}.\alpha_{i,j}^{t} \ , \tag{5}$$

where $X_{i,j}^{t+1}$ and $X_{i,j}^{t}$ represent the next and present cheetah's position in i,j, t indicates present hunting time, T denotes hunting time's maximum duration, $\alpha_{i,j}^{t}$ and $\widehat{\gamma_{i,j}^{-1}}$ determines a step length and randomization parameter. Then, the cheetah remains inactive and waits for prey to approach which is formulated in equation (6).

$$X_{i,j}^{t+1} = X_{i,j}^{t}, \tag{6}$$

where $X_{i,j}^{t+1}$ and $X_{i,j}^{t}$ denote updated and present cheetah's position. It selectively modifies the cheetah's position in each group to enhance the hunting success rate which helps to avoid premature convergence. The attacking strategy is expressed in equation (7).

$$X_{i,j}^{t+1} = X_{B,j}^{t} + \breve{\gamma_{i,j}}.\beta_{i,j}^{t}, \tag{7}$$

where $X_{B,j}^{t}$ represents present prey's position in permutation, $\breve{\gamma_{i,j}}$ and $\beta_{i,j}^{t}$ depict tuning factor and interaction factor related to i^{th} and j^{th}permutation cheetah. However, a random approach by generating initialization leads to poor diversity and uneven distribution and it has local optima issue. To solve this issue, tent chaotic mapping and adaptive search agent strategy is used which is discussed as follows.

Improved Tent Chaotic Mapping

In the traditional CSO, the cheetah's positions are initialized randomly which leads to an uneven distribution of population positions and minimizes solution accuracy. Hence, an improved tent chaotic mapping is used to initialize the cheetah's population. It does not alter the randomness but instead enhances the diversity distribution of the population and ensures even traversal during particle search. Chaotic sequence exhibits ergodicity, randomness, and regularity and by utilizing these features, the optimization search effectively preserves population diversity, avoids local optima, and enhances global search abilities. A tent chaotic map has better ergodicity, rapid iteration speed, and greater uniformity than the logistic chaotic map and is appropriate to generate a chaotic sequence. In tent chaotic mapping, an adaptive factor $\frac{\gamma}{N}$ is included which helps to enhance traversability among [0,1] and provide more uniform distribution. The traditional tent chaotic mapping and improved tent chaotic mapping are expressed in equations (8) and (9).

$$x_{i+1} = \begin{cases} 2x_i, 0 \leq x_i < 0.5 \\ 2(1-x_i), 0.5 \leq x_i < 1 \end{cases} i = 0,1,2, \quad (8)$$

$$x_{i+1} = \begin{cases} 2x_i + \frac{\gamma}{N}, 0 \leq x_i < 0.5 \\ 2(1-x_i) + \frac{\gamma}{N}, 0.5 \leq x_i < 1 \end{cases} i = 0,1,2. \quad (9)$$

It enhances the collective search ability and addresses the minimized population diversity, search accuracy, and local optima while approaching the optimal solution.

Adaptive Search Agent Strategy

In COA, the number of search agents remains fixed throughout the entire iterative process. When the number of search agents is small, the model exhibits poor global search ability. While it is high, the convergence speed becomes slow. To balance both local and global search ability, an adaptive search agent formula is used to estimate the search agents with iteration count which is expressed in equation (10).

$$m = (m_{max} - m_{min}) cos\left(\frac{\pi.it}{2Max}\right), \quad (10)$$

where m_{max} and m_{min} represent the convergence factor's maximum and minimum values, $MaxIt$ denotes a maximum number of iterations and it determines iteration. This approach helps to effectively balance the local and global abilities by using an adaptive search agent strategy. By implementing this process, the hyperparameter optimizes the values of CNN effectively which helps to increase the diabetes prediction accuracy. Table 2 determines the hyperparameter with its range.

Table 2. Hyperparameter process with its range

Hyperparameter	Range
Batch size	64
Learning rate	0.02
Dropout rate	0.3
Filter size	3 × 3
Optimizer	AdaGrad
Number of layers	4

IV. RESULTS

The proposed CNN-DS-COA is simulated by using a Python 3.4 environment, an Intel i7 processor, and 128 GB RAM with Tensorflow version of 1.5.0 and Keras version of 2.1.5. The accuracy, recall, precision, and f1-score are used to evaluate the performance of the CNN-DS-COA method which is determined in equations (11 - 14).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \quad (11)$$

$$Recall = \frac{TP}{TP+FN}, \quad (12)$$

$$Precision = \frac{TP}{TP+FP}, \quad (13)$$

$$F1 - Score = \frac{2TP}{2TP+FP+FN}, \quad (14)$$

where TP indicates True Positive, TN represents True Negative, FN denotes False Negative, and FP depicts False Positive.

A. PERFORMANCE ANALYSIS

Table 3 presents a performance analysis of different hyperparameter methods. The existing methods like Dung Beetle Optimization (DBO), White Shark Optimization (WSO), and COA are compared with the proposed DS-COA. When compared to the existing methods, the proposed DS-COA achieves a better accuracy of 99.90% and 99.72% using PIMA and Germany datasets due to its enhanced exploration-exploitation balance to optimize the parameter values which enables it to efficiently search the solution space and avoid local optima.

Fig. 2 indicates a graphical representation of different classification methods. The proposed CNN achieves a better performance of 99.90% and 99.72% using both datasets compared to existing methods like ResNet, VGG, Inception, Swin transformer, and Vision Transformer (ViT). The proposed CNN is capable of learning and capturing complex features and patterns of data through its convolutional layers. Also, it effectively manages spatial hierarchies in data which helps to predict the data effectively.

Table 3. Performance analysis of different hyperparameter methods

Datasets	Methods	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
PIMA	DBO	79.65	76.12	75.65	76.32
	WSA	82.32	81.76	79.62	80.76
	COA	84.21	82.98	79.64	81.98
	DS-COA	99.90	99.25	98.25	98.64
Frankfurt Hospital, Germany	DBO	81.20	80.08	79.27	80.87
	WSA	84.65	82.32	80.35	82.37
	COA	85.37	83.56	82.36	83.21
	DS-COA	99.72	99.47	98.65	98.72

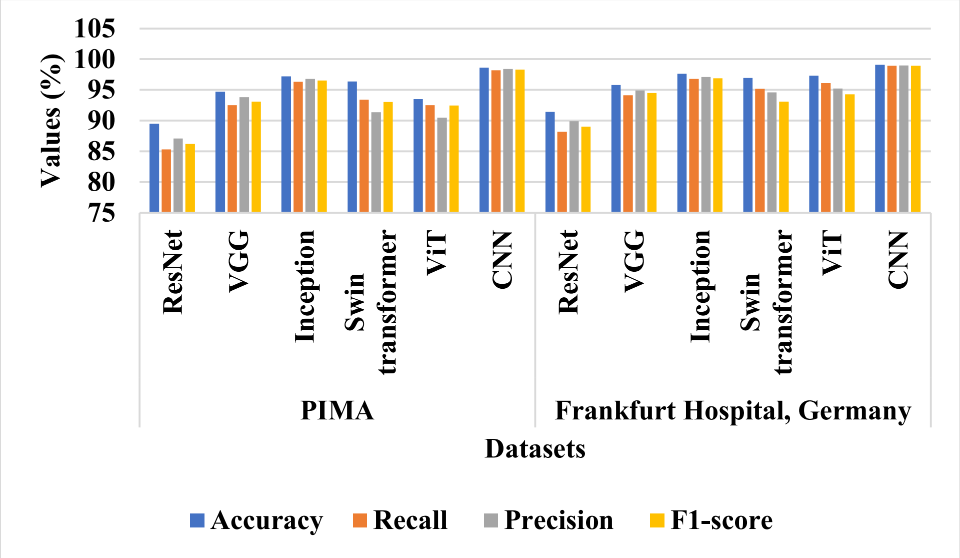

Figure 2. Graphical representation of different classification methods

Table 4 determines a performance analysis of k-fold analysis. It provides a balanced trade-off between variance and bias. It divides the data into five subsets and enables the model to be trained and validated on various portions which minimizes overfitting and enhances generalization. Also, it ensures that every data point is utilized for both validation and training which makes the model more reliable.

Table 4. Performance analysis of k-fold validation

Datasets	k-fold	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
PIMA	3.00	89.43	89.46	87.38	87.79
	5.00	99.90	99.25	98.25	98.64
	7.00	90.57	90.37	86.39	89.54
	9.00	94.95	92.46	89.37	87.43
Frankfurt Hospital, Germany	3.00	92.07	86.35	88.56	94.28
	5.00	99.72	99.47	98.65	98.72
	7.00	93.47	90.79	89.54	92.53
	9.00	96.34	94.67	94.37	95.88

Table 5 represents the performance analysis of different classifiers with training on PIMA and testing on Frankfurt Hospital, Germany datasets.

Table 5. Performance analysis of different classifiers with training on PIMA and testing on Frankfurt Hospital, Germany datasets

Methods	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
ResNet	85.21	83.69	82.54	83.11
VGG	88.36	87.26	81.52	84.29
Inception	89.64	85.69	82.69	84.16
Swin transformer	91.23	89.21	87.42	88.30
ViT	90.38	86.98	89.02	87.98
CNN	97.58	96.35	97.28	96.81

Table 6 demonstrates different classifiers with training on Frankfurt Hospital, Germany and testing on PIMA datasets. While compared to existing methods like ResNet, VGG, Inception, swin transformer, and ViT, CNN achieves a high accuracy of 97.58% and 95.36% due to capturing intricate patterns from medical data. Unlike transformer requires large datasets to generalize well but CNN performs effectively with small medical data like PIMA and Frankfurt Hospital, Germany datasets. Its hierarchical structure allows learning both low-level and high-level features, which enhances classification. Spatial dependencies in patient data are effectively captured and make them appropriate for prediction which leads to high accuracy.

Table 6. Performance analysis of different classifiers with training on Frankfurt Hospital, Germany datasets and testing on PIMA

Methods	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
ResNet	82.69	80.36	81.25	80.80
VGG	81.65	85.69	82.30	83.96
Inception	78.54	86.39	84.69	85.53
Swin transformer	83.49	88.43	85.39	86.88
ViT	89.02	89.01	88.20	88.60
CNN	95.36	87.25	86.39	86.81

Table 7 indicates a performance analysis of computational complexity for both datasets. While compared to existing methods like CNN-DBO, CNN-WSA, and CNN-COA, the proposed CNN-DSCOA achieves a less time complexity of 21.65s and 25.98s for one iteration. This is because of efficient hyperparameter tuning which rapidly converges to optimal values and prevents unwanted computations. DSCOA's exploration and exploitation avoids excessive iterations and effectively fine-tunes CNN parameters. Therefore, these results in rapid training are compared to existing methods.

Table 7. Performance analysis of computational complexity

Methods	Datasets	Time complexity (s)
CNN-DBO	PIMA	42.15
CNN-WSA		39.87
CNN-COA		35.98
CNN-DSCOA		21.65
CNN-DBO	Frankfurt Hospital, Germany	41.28
CNN-WSA		36.22
CNN-COA		32.78
CNN-DSCOA		25.98

B. COMPARATIVE ANALYSIS

Table 8 demonstrates a comparative analysis of existing methods. The existing methods like SMOTE-SMO [16], Stacking [17], Fused ML [18], ANN [19], and Statistical predictive model [20] are compared with the proposed CNN-DS-COA. The value from the existing method [19] is presented in decimal form and is converted into percentages based on the values from the proposed method. The proposed approach achieves a better accuracy of 99.90% and 99.72% using both PIMA and Frankfurt Hospital, Germany datasets. This improvement is due to the CNN approach optimizing hyperparameters with COA which enhances the model convergence and performance.

Table 8. Comparative Analysis of Existing Methods

Methods	Datasets	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
SMOTE-SMO [16]	PIMA	99.07	98.24	96.23	97.71
Stacking [17]		93.1	83.9	84	83.5
Fused ML [18]		94.87	N/A	N/A	N/A
Proposed CNN-DS-COA		99.90	99.25	98.25	98.64
ANN [19]	Frankfurt Hospital, Germany	95.8	95	95	95
Statistical predictive model [20]		99.5	N/A	N/A	N/A
Proposed CNN-DS-COA		99.72	99.47	98.65	98.72

C. DISCUSSION

The advantages of the proposed CNN-DS-COA and the disadvantages of existing techniques are discussed in this section. The limitations of the existing techniques are as follows: SMO [16] struggles with complex and non-linear relationships because it is based on linear decision boundaries, leading to suboptimal performance. The stacked automated approach [17] suffers from overfitting due to the combination of multiple approaches, which reduces generalization and results in a complex ensemble model. Fused ML [18] faces challenges in effectively tuning the hyperparameters of both models, which also leads to suboptimal performance. ANN [19] has difficulty predicting diabetes because it struggles to manage complex and non-linear relationships in the data. The statistical predictive model [20] struggles to manage interactions among multiple variables, resulting in poor performance when applied to different patient populations. The proposed CNN-DS-COA addresses these limitations by leveraging advanced hyperparameter optimization and CNN, which enhances generalization and accuracy. CNN effectively captures intricate features in the data, while COA efficiently fine-tunes hyperparameters. This approach improves convergence speed and accuracy by optimizing the parameter space and increasing the model's ability to generalize. Additionally, it balances exploitation and exploration, leading to more accurate and reliable diabetes predictions.

V. CONCLUSION

This research proposes CNN-DS-COA to automatically predict diabetes. CNN captures spatial relationships and hierarchies, helping to understand the correlations among medical features. COA is used to optimize the CNN parameters, increasing model performance by efficiently exploring the parameter space, enhancing convergence speed, and avoiding local minima. COA is further improved by employing di-strategies to enhance traversability and provide a more uniform distribution. An adaptive search mechanism is implemented to balance both local and global exploration, optimizing the search agents. Min-max normalization is applied to normalize the data, preserving the relationships among values and maintaining the original data's proportions. As a result, the proposed CNN-DS-COA achieves an accuracy of 99.90% and 99.72% on the PIMA and Frankfurt Hospital, Germany datasets, respectively. In the future, attention mechanism will be integrated with DL to enhance feature learning. This will assist in early detection, risk assessment, and personalized treatment recommendations for diabetes management that can improve model performance like robustness.

References

[1] A. Doğru, S. Buyrukoğlu, M. Arı, "A hybrid super ensemble learning model for the early-stage prediction of diabetes risk," *Medical & Biological Engineering & Computing*, vol. 61, issue 3, pp. 785-797, 2023. https://doi.org/10.1007/s11517-022-02749-z.

[2] F. A. Khaleel, A. M. Al-Bakry, "Diagnosis of diabetes using machine learning algorithms," *Materials Today: Proceedings*, vol. 80, no. 3, pp. 3200-3203, 2023. https://doi.org/10.1016/j.matpr.2021.07.196.

[3] S. M Ganie, P. K. D. Pramanik, M. B. Malik, S. Mallik, H. Qin, "An ensemble learning approach for diabetes prediction using boosting techniques," *Frontiers in Genetics,* vol. 14, p. 1252159, 2023. https://doi.org/10.3389/fgene.2023.1252159.

[4] S. K. Chinnababu, A. Jayachandra, "Diabetes prediction and classification using self-adaptive evolutionary algorithm with convolutional neural network," *International Journal of Intelligent Engineering & Systems*, vol. 17, issue 3, pp. 460-472, 2024. https://doi.org/10.22266/ijies2024.0630.36.

[5] A. Naseem, R. Habib, T. Naz, M. Atif, M. Arif, and S. Allaoua Chelloug, "Novel Internet of Things based approach toward diabetes prediction using deep learning models," *Frontiers in Public Health*, vol. 10, p. 914106, 2022. https://doi.org/10.3389/fpubh.2022.914106.

[6] M. Ragab, A. S. A. M. Al-Ghamdi, B. Fakieh, H. Choudhry, R. F. Mansour, and D. Koundal, "Prediction of diabetes through retinal images using deep neural network," *Computational Intelligence and Neuroscience*, vol. 2022, issue 1, p.7887908, 2022. https://doi.org/10.1155/2022/7887908.

[7] J. J. Sonia, P. Jayachandran, A. Q. Md, S. Mohan, A. K. Sivaraman, K. F. Tee, "Machine-learning-based diabetes mellitus risk prediction using multi-layer neural network no-prop algorithm," *Diagnostics,* vol. 13, issue 4, p. 723, 2023. https://doi.org/10.3390/diagnostics13040723.

[8] V. Chang, M. A. Ganatra, K. Hall, L. Golightly, Q. A. Xu, "An assessment of machine learning models and algorithms for early prediction and diagnosis of diabetes using health indicators," *Healthcare Analytics,* vol. 2, p. 100118, 2022. https://doi.org/10.1016/j.health.2022.100118.

[9] C. C. Olisah, L. Smith, M. Smith, "Diabetes mellitus prediction and diagnosis from a data preprocessing and machine learning perspective," *Computer Methods and Programs in Biomedicine,* vol. 220, p. 106773, 2022. https://doi.org/10.1016/j.cmpb.2022.106773.

[10] A. Armghan, J. Logeshwaran, S. M. Sutharshan, K. Aliqab, M. Alsharari, S. K. Patel, "Design of biosensor for synchronized identification of diabetes using deep learning," *Results in Engineering,* vol. 20, p. 101382, 2023. https://doi.org/10.1016/j.rineng.2023.101382.

[11] Y. Xiong, L. Lin, Y. Chen, S. Salerno, Y. Li, X. Zeng, and H. Li, "Prediction of gestational diabetes mellitus in the first 19 weeks of pregnancy using machine learning techniques," *The Journal of Maternal-Fetal & Neonatal Medicine,* vol. 35, issue 13, pp. 2457-2463, 2022. https://doi.org/10.1080/14767058.2020.1786517.

[12] M. Atif, F. Anwer, and F. Talib, "An ensemble learning approach for effective prediction of diabetes mellitus using hard voting classifier," *Indian Journal of Science and Technology,* vol. 15, issue 39, pp. 1978-1986, 2022. https://doi.org/10.17485/IJST/v15i39.1520.

[13] I. Tasin, T. U. Nabil, S. Islam, and R. Khan, "Diabetes prediction using machine learning and explainable AI techniques," *Healthcare Technology Letters,* vol. 10, issue 1-2, pp. 1-10, 2023. https://doi.org/10.1049/htl2.12039.

[14] S. A. Alex, N. Z. Jhanjhi, M. Humayun, A. O. Ibrahim, and A. W. Abulfaraj, "Deep LSTM model for diabetes prediction with class balancing by SMOTE," *Electronics,* vol. 11, issue 17, p. 2737, 2022. https://doi.org/10.3390/electronics11172737.

[15] H. Zhou, Y. Xin, and S. Li, "A diabetes prediction model based on Boruta feature selection and ensemble learning," *BMC Bioinformatics,* vol. 24, issue 1, p. 224, 2023. https://doi.org/10.1186/s12859-023-05300-5.

[16] H. Naz, and S. Ahuja, "SMOTE-SMO-based expert system for type II diabetes detection using PIMA dataset," *International Journal of Diabetes in Developing Countries*, vol. 42, issue 2, pp. 245-253, 2022. https://doi.org/10.1007/s13410-021-00969-x.

[17] R. Sivashankari, M. Sudha, M. K. Hasan, R. A. Saeed, S. A. Alsuhibany, S. Abdel-Khalek, "An empirical model to predict the diabetic positive using stacked ensemble approach," *Frontiers in Public Health,* vol. 9, p. 792124, 2022. https://doi.org/10.3389/fpubh.2021.792124.

[18] U. Ahmed, G. F. Issa, M. A. Khan, S. Aftab, M.F. Khan, R. A. Said, T. M. Ghazal, and M. Ahmad, "Prediction of diabetes empowered with fused machine learning," *IEEE Access,* vol. 10, pp. 8529-8538, 2022. https://doi.org/10.1109/ACCESS.2022.3142097.

[19] M.K. Gourisaria, G. Jee, G. M. Harshvardhan, V. Singh, P. K. Singh, T. C. Workneh, "Data science appositeness in diabetes mellitus diagnosis for healthcare systems of developing nations," *IET Communications*, *Special Issue: Deep learning-based intelligent communication systems: Using big data analytics*, vol. 16, issue 5, pp. 532-547, 2022. https://doi.org/10.1049/cmu2.12338.

[20] K. Azbeg, M. Boudhane, O. Ouchetto, S. J. Andaloussi, "Diabetes emergency cases identification based on a statistical predictive model," *Journal of Big Data,* vol. 9, issue 1, p. 31, 2022. https://doi.org/10.1186/s40537-022-00582-7.

[21] PIMA dataset. [Online]. Available at: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.

[22] Frankfurt Hospital, Germany dataset. [Online]. Available at: https://www.kaggle.com/code/linggarmaretva/frankfurt-hospital-diabetes-with-lgbmclassifier.

[23] A. Saha, A. R. Mishra, P. Rani, I. M. Hezam, and F. Cavallaro, "A q-rung orthopair fuzzy FUCOM double normalization-based multi-aggregation method for healthcare waste treatment method selection," *Sustainability,* vol. 14, issue 7, p. 4171, 2022. https://doi.org/10.1186/s40537-022-00582-7.

[24] J. P. Bharadiya, "Convolutional neural networks for image classification," *International Journal of Innovative Science and Research Technology,* vol. 8, issue 5, pp. 673-677, 2023.

[25] M. A. Akbari, M. Zare, R. Azizipanah-Abarghooee, S. Mirjalili, and M. Deriche, "The cheetah optimizer: A nature-inspired metaheuristic algorithm for large-scale optimization problems," *Scientific Reports,* vol. 12, p. 10953, 2022. https://doi.org/10.1038/s41598-022-14338-z.

PANKAJ KUMAR received his B Sc. Engineering in Information Technology from MIT Muzaffarpur and his M.Tech in Computer Science and Engineering from IIT(ISM) Dhanbad. He is currently working as an Assistant Professor with Department of Computer Science and Engineering at Government Engineering College, Vaishali. His current research interests include Cryptography, Information Security, and Database Management System.

MERIN THOMAS is an Associate Professor in the School of Computer science, RV University, Bengaluru, India. Her research specialization is in Natural Language Processing in the domain of Artificial Intelligence and Machine Learning.

M. Manohar was brought up in Udupi and currently he lives in Bangalore, Karnataka. He was qualified Bachelor in CSE in the year 1999 at Dr AIT, Bangalore. He has his Master in CSE in the year 2004 at Dr AIT, Bangalore. He completed his Ph.D in CSE in the area of data mining and big data in the year 2016 at Jain University, Bangalore. He is an Associate Professor in the Computer Science and Engineering Department at Faculty of engineering of CHRIST (Deemed to be University), Bengaluru. He is an educator by choice and vocation, with an experience of 15 years in Teaching. He has many publications in both conference and Journals. He is a member of Professional body IACSIT, IET.

Alok Kumar Pani has pursued his Bachelor of Engineering from Biju Patnaik University of Technology, Rourkela in 2006 and Master of Technology from Indian Institute of Technology (ISM), Dhanbad in 2014. He has completed his Ph.D in computer science and currently working as an Assistant Professor at Birla School of Applied Sciences in Birla Global University. He has 15 years of teaching experience and 1 year of Industry Experience. His primary research interests are in the fields of Databases, Algorithms and Computational Complexity.

•••