RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt2514

A Diabetes Diagnosis Model using Optimized Long Short-Term Memory Based on Improved Particle Swarm Optimization

J. Revathy [a, *], S.K. Jayanthi [a]

[a] Department of Computer Science, Vellalar College for Women, Erode-638012, Tamil Nadu, India.

* Corresponding Author Email: revamsc12@gmail.com

DOI: https://doi.org/10.54392/irjmt2514

Received: 10-09-2024; Revised: 14-12-2024; Accepted: 19-12-2024; Published: 30-12-2024

Abstract: Diabetes, a chronic disease, arises when the body is either unable to utilize the insulin generated by the pancreas or is unable to create enough of it. It could be the deadliest if left undiagnosed and untreated. If diabetes is identified early enough, a person can receive the right care and live a healthy life. An automated system is required to identify diabetes from clinical and physical data when the traditional method is laborious. The paper proposed a new diabetes classifying model based on optimized long short-term memory (LSTM). The proposed method uses a new variant of particle swarm optimization (PSO) based on partial opposition-based learning (POBL) and a local search algorithm (LSA) approach called PLPSO for optimizing hyperparameters of LSTM (PLPSO-LSTM). PSO uses the POBL during the initialization phase to increase population diversity and the LSA during the updating position to increase exploitation. The proposed model has been tested using four diabetes datasets for analyzing its performance. These results show that optimized PLPSO-LSTM performs better than other state-of-the-art algorithms.

Keywords: Long short-term memory, Hyper parameter optimization, Particle swarm optimization, Partial opposition-based learning, Diabetic prediction, Local search algorithm

1. Introduction

As per the updates of WHO, it is estimated that over 77 million individuals aged 18 years and above are afflicted with type 2 diabetes, while an additional 25 million individuals are classified as pre-diabetic, indicating a heightened susceptibility to acquiring diabetes shortly in India. It may cause serious health problems if timely detection and intervention measures are not implemented, although more than half of the population is unaware that they have diabetes [1, 2]. Diabetes is a commonly encountered chronic condition characterized by elevated glucose levels in the bloodstream. The accurate identification of diabetes can significantly improve an individual's overall well-being, as failing to do so can lead to adverse health outcomes such as renal failure, cardiac complications, and impairment of blood vessels and nerves [2]. There exist two distinct classifications of diabetes, namely Type-1 and Type-2 Diabetes mellitus. Type 1 diabetes mellitus (T1DM), alternatively referred to as insulin-dependent diabetes, and is classified as an autoimmune disorder characterized by the destruction of pancreatic β cells, resulting in unreliable production of insulin. Type 2 diabetes mellitus (T2DM), alternatively referred to as noninsulin-dependent diabetes, and is characterized by the ability of β cells in the pancreas to make insulin. However, the cells within the body exhibit insulin resistance, resulting in impaired normal functioning [3, 4].

Currently, there are various techniques and tools in the literature for diabetes classification and prediction; they are used by physicians and clinicians to enhance the diabetes diagnosing process. Machine learning (ML) and deep learning (DL) approaches are most often utilized in this field [5]. Research has shown that when it comes to diagnosing diabetes, DL approaches beat the conventional approach. Recurrent neural networks (RNNs) are among the several DL approaches that have been widely used in diagnosing diabetes recently [6]. The RNN framework learns about the information that is already out there and in what order. On the other hand, over time, RNN's ability to learn even from distant input is lost when a vanishing gradient issue arises. LSTM, which resolves the vanishing gradient issue and keeps the benefits of RNN, is regarded as an improved form of RNN. A long-term relationship's reliance is obtained by LSTM. During training, the LSTM's idea of memory aids in capturing the underlying data pattern and long-term dependencies among the data [7]. Memory cells in LSTM allow it to extract deep characteristics from a limited number of samples. It has good application effects in numerous fields and is appropriate for processing time series. The LSTM networks are useful for diabetes prediction

because of their capacity to identify sequential patterns and temporal connections in data. Time-series data, including blood glucose levels, lifestyle factors (like diet and exercise), and other health indicators (like blood pressure and insulin levels) that change over time are frequently used in diabetes prediction. LSTMs are especially well-suited for diabetes prediction tasks because of their unique structure for processing sequential data, which allows them to simulate these temporal patterns efficiently. However, since experience typically determines the LSTM model's hyperparameters, subjectivity is significant and will have an impact on the model's ability to fit data. The random hyperparameters have the potential to produce absurd outcomes, which is the most significant limitation of the LSTM algorithm [8].

The collective behavior of decentralized, self-organizing systems, which are typically modeled after biological systems like fish schools, ant colonies, and bird flocks, is referred to as "swarm intelligence". One of the most well-known swarm intelligence algorithms in the literature is the PSO algorithm, which when it was first developed demonstrated good qualities including quick convergence and high solution efficiency. But as we approach the most ideal particle, particle population swarms grow more "convergent," unable to fully utilize the knowledge discovered during the search and more vulnerable to local optimality. Consequently, to achieve optimum outcomes, the link between general exploration and local exploitation must be dynamically coordinated. The foundation of applying SI algorithms to issue-solving is understanding how to strike a balance between exploration and development. Therefore, a variety of viewpoints and research backgrounds are used in the effort to improve the PSO algorithm. The remarkable shortcomings of PSO are trapped in the local optimal solution and population diversity [9].

Tizhoosh *et al.* [10] have developed an intelligent technology called OBL. It involves the notion that to improve learning, one should take into account, not just a potential answer but also it's opposite option. The basic idea is to enable a more thorough exploration of the solution space by taking into account the opposing solution. This could result in learning and optimization that is possibly faster and more effective. Hence, the various OBL schemes are developed to enhance the performance of PSO and overwrite the shortcomings. However, the OBL algorithms compute the complete opposite vector [11]. Hence, the computing effort can be doubled by evaluating both the contender and its opposite, and also becomes increasingly difficult to define and compute opposites [12]. In POBL, the reverse of OBL is generated by reflecting only a portion of the candidate solution's dimensions. By choosing which dimensions to reflect, only a portion of the opposite vector-rather than the entire vector-is computed. With the use of partial opposition learning, an optimization algorithm will be able to strike a balance between exploration and exploitation. By choosing the best populations in a subset of solution dimensions rather than the entire solution, the POBL technique outperforms conventional OBL schemes when it comes to search space exploration. Additionally, it will raise the robustness of the model, which will result in an improvement in the convergence of the model as well as enhance population diversity.

On the other hand, OBL and POBL were utilized in several research for population update in SI approaches with each iteration. Because they find the opposing alternatives for every answer and choose the best solutions, this will make the search algorithm more complex in each iteration. Hence, LSA and PSO are combined to solve its local optimal problem and improve the current optimal solution. However, PSO usually searches for optimal solutions using particle locations, but in some situations-particularly when the solution space is big or complicated-it may converge slowly. By concentrating on promising regions of the solution space, LSA aid in accelerating convergence by rapidly honing in on potential solutions. In PSO variants that incorporate local search algorithms, the trade-off between diversification and intensification is managed by combining the swarm's global search capability with local optimization techniques. On the other hand, POBL is a PSO version that dynamically introduces "oppositional" solutions to improve the balance between intensity and diversification. The swarm can examine alternate solutions on the other side of the search space since POBL produces solutions that are partially opposed to the particles that are now present.

To improve early detection, precise forecasts, individualized treatment regimens, and overall diabetes management, optimized LSTM is being used for diabetes diagnosis. This is done by taking use of the networks' capacity to comprehend and predict temporal patterns in patient data. The objectives are to provide a thorough, high-performing diagnostic tool that is efficient, comprehensible, and useful in practical situations in order to improve diabetes management and prevention. The contributions of the proposed PLPSO-LSTM are:

- A new variant of PSO is proposed for optimizing the hyperparameters of the LSTM network
- The proposed PSO used the POBL algorithm for enhancing population diversity and LSA is used to avoid local optima
- The PLPSO is used to analyze the hyperparameters for LSTM, to increase the global convergence rate and prediction accuracy.
- By choosing ideal parameters that preserve a balance between training accuracy and generalization, improved PSO can assist the LSTM model in avoiding overfitting and produce a more reliable diagnostic model.

Table 1. Related research articles on diabetes classification

S.No	Author & year	Model/ Objective	Dataset	Feature Selection	Noise Removal	Accuracy (%)	Pros	Cons
1.	Kandhasamy, *et al,* 2015 [13]	J48, DT, KNN, RF, SVM	Pima	No	Yes	86.46	• Comparative analysis among the state-of-art models • Highlights the necessity of noise removal	• Lack of result quality • KNN and RF overfit the data • K=1 for KNN may not be a feasible model and overfits the data • Missing values imputation caused model overfitting
2.	Erkaymaz, *et al,* [14], 2016	Small World Feed Forward Neural network (SW-FFNN)	Pima	No	No	91.66	• Brilliant rewiring approach to handle complex data • Relevant features can be focused	• Complex model to tune • Increase computational overhead • Experimented small dataset
3.	Yu, *et al,* 2017 [15]	Linear Discriminant analysis (LDA), Minimax Probability Machine (MPM)	Pima	No	No	68.70	• Better results are obtained with an alpha value of 90% • 5 non-diabetic datasets are used • MPM reached maximal results	• MPM overfits the Pima data • Lack of model tuning for robust model
4.	Chen *et al,* 2017 [16]	Hybrid Kmeans with Decision Tree	Pima	Yes	No	90.04	• Used K-Means for Data Reduction • Improved the accuracy rate significantly	• Increases the computational overhead on large datasets • probably overfits the data
5.	Dadgar, *et al,* 2017 [17]	Hybrid Multi-Layer Perceptron with Genetic Algorithm (MLP-GA)	Pima	Yes	No	87.46	• Reduced number of learning iterations • Used UTA approach for feature selection	• Probably converge at the local optimal • Parameter tuning has to be considered
6.	Hashi, *et al,* 2017 [18]	Sequential Forward selection algorithm	Pima	No	No	87.01	• Used 10-fold cross-validation • SVM is used for this algorithm	• Lack of model tuning for better performance
7.	Zhang *et al,* 2018 [19]	Feed Forward Neural network (FFNN)	Pima	No	No	82	• Trained FFNN for diabetes diagnosis	• Lack of quality

Table 1. Related research articles on diabetes classification

S.No	Author & year	Model/ Objective	Dataset	Feature Selection	Noise Removal	Accuracy (%)	Pros	Cons
							• Can deal with complex patterns	• The latest approaches have not been studied
8.	Haritha, *et al,* 2018 [20]	Hybrid Firefly and cuckoo search	Pima	No	No	81 %	• Used bio-inspired optimization for type1 and type2 diabetes classification	• Increase computational overhead • Lack of quality • Highly probable to local optimal convergence
9.	Srivastava, *et al*, 2019 [21]	Artificial Neural Network (ANN)	Pima	No	Yes	92	• Obtained a higher accuracy rate • Handled the missing values	• comparative analysis is missing • Missing values are imputed with mean value which is insignificant for medical datasets.
10.	Kannadasan, *et al*, 2019 [22]	Deep learning: Stacked Auto-encoders	Pima	No	No	86.26	• Applied deep learning to achieve précised diagnostic results	• More complex to adapt to newer datasets • Experimented with one dataset • Lack of comparisons with recent approaches
11.	Prema, *et al,* 2019 [23]	Ensemble of SVM	Pima	No	No	80.52	• SVM performs well in an ensemble approach • 10-fold cross-validation is used	• Disease classification requires a more accurate rate of diagnosis • Lack of current approaches
12.	Dinh, *et al,* 2019 [24]	eXtreme Gradient Boost (XGBoost)	BRFSS	Yes	No	95	• Attained maximal accuracy • Experimented with the CDC dataset	• Complex parameter tuning is required • More probably overfits the data and produces biased results as it is an unbalanced dataset
13.	Pradhan, *et al,* [25], 2020	ANN	Pima	No	No	85.09	• Improved Accuracy rate than conventional approaches • Can understand the non-linear patterns in the dataset	• Required appropriate tuning of hyper-parameters of ANN • Insufficient Accuracy rate for disease diagnosis
14.	Lakhwani, *et al,* 2020 [26]	Three-layered FFNN,	Pima	No	No	88.8	• Versatile analysis proves the model's performance	• No Parameter tuning • Lack of comparative study

Table 1. Related research articles on diabetes classification

S.No	Author & year	Model/ Objective	Dataset	Feature Selection	Noise Removal	Accuracy (%)	Pros	Cons
		Logistic activation function, Quasi-Newton training algorithm					• Can understand the non-linear patterns in the dataset	• A significant outcome is missing when compared to other existing approaches
15.	Abedini, *et al,* 2020 [27]	Ensemble of Decision tree, Logistic regression, and ANN	Pima	No	No	83.08	• An effective approach to improving the diagnostic quality • The robustness of the model will increase with ensembling	• Complex to handle ensemble approaches • Need manual intervention for tuning all the models • May probably collapse the diagnosis decision
16.	Naz *et al,* 2020 [28]	Deep Learning	Pima	No	No	94.07	• Maximal accuracy obtained • Effectively handles the non-linear patterns in the dataset	• Experimented with one dataset and may chance to over-fit the data • Require parameter tuning to adapt different datasets
17.	Pradhan, *et al*, 2020 [29]	ANN with 4 hidden layers	Pima	No	No	88.71	• ReLU activation function yields better results • Compared with various conventional approaches	• More hidden layers may complicate the model and may collapse the model • Lack of parameter optimization for better results
18.	Guldogan, *et al,* 2020 [30]	MLP and RBF	Pima	No	No	78.1, & 76.8	• Comparative analysis among ANN models is carried out • The positive and negative predictive values of both models are comparable	• Lack of quality • Lack of tuning hyperparameters
19.	Bukhari, *et al,* [31], 2021	Artificial back-propagation scaled conjugate gradient neural network (ABP-SCGNN)	Pima	No	No	93	• Accuracy improved • adaptable for complex data	• Recent approaches are not considered • Complex model but experimented with a small dataset
20.	Nadeem, *et al,* 2021 [32]	Hybrid SVM-ANN	Pima, CDC	No	No	94.67	• Fusion-based machine learning approach	• High probability of model overfitting on data

Table 1. Related research articles on diabetes classification

S.No	Author & year	Model/ Objective	Dataset	Feature Selection	Noise Removal	Accuracy (%)	Pros	Cons
								• Require a comprehensive understanding of data in tuning the model to a newer dataset
21.	Chowdary, *et al,* 2021 [33]	Convolution based Long-Short Term Memory (CLSTM)	Pima	No	No	95.6	• Used Convolution-based LSTM for better performance	• Lack of hyperparameter tuning • Overfits to the data
22.	Sivasankari, *et al,* 2022 [34]	MLP	Pima	No	No	86.06	• Comparative analysis carried out with conventional approaches	• Lack of parameter tuning
23.	Pujari, 2022 [35]	SVM with Polynomial Kernel	Pima	No	Yes	96	• Detailed analysis of SVM kernels	• Generalization of the models is not justified with one dataset
24.	Chang*, et al,* 2023 [36]	E-diagnosis system with ML algorithms	Pima	Yes	Yes	79.57	• PCA for feature selection • Random Forest obtained better results	• Maximal accuracy rate obtained in various existing literature • Less accuracy rate even with feature selection
25.	Mousa, *et al,* 2023 [37]	Comparative study on LSTM, RF, CNN	Pima	No	Yes	85	• LSTM performed well in the classification	• Experimented with only one dataset • Generalization of the model not justified
26.	H. Shao *et. al*. 2024 [38]	LightGBM-Optuna	Kaggle	No	No	97.11	• High accuracy and handing imbalance	• High computational
27.	M. Rahman *et al.* 2020 [39]	Conv-LSTM	Pima	Yes	No	97.26	• Find the optimal parameters	• Low accuracy

- The optimized LSTM is used for diabetic diagnosis and experimented with the four standard diabetes datasets namely: the PIMA Indian diabetes dataset, CDC diabetes health indicators dataset, Mendeley diabetes dataset (MDD), and diabetes type dataset (DTD).
- Furthermore, a performance comparison of the PLPSO-LSTM with other diagnosis methods for performance analysis.

The upcoming sections of this article are: section 2 discusses the recent related research works in the literature; sections 3 and 4 deal with the problem formulation and the research methods; Section 5 elaborates the proposed model PLPSO-LSTM; Section 6 explores the comparative analysis of the experimental results and discussion; and finally, section 7 summarizes this research work.

2. Related Works

In the field of disease diagnosis problems, supervised machine learning models play a vital role in assisting physicians. The various diagnostic approaches used for diabetes classification and prediction are listed in Table 1. This table highlights the ML method proposed, the pros and cons of that approach, the dataset experimented with, the accuracy level obtained by that model, and the usage of feature selection and noise handling methods in that research work. This literature study covered about 25 related research articles in recent times. The various ML and DL approaches are used for diabetes diagnosis. In all these research works, the benchmark PIMA Indian diabetes dataset is used. All hybridization, ensemble, and parameter tuning seen in these research articles are experimented with and tuned only for the PIMA dataset. The rigidity of the model will be questionable if it is attempted with only one dataset. Training the model with more than one dataset will justify the performance of the prediction model and like generalizability, rigidity, and reliability of the model. Conventional swarm intelligence techniques are useful for global optimization, but they have drawbacks like parameter-dependent behavior, high sensitivity to parameters, scalability problems, slow convergence in later stages, noise sensitivity, stagnation in large populations, and premature convergence. Modern variations and hybrid systems frequently use mechanisms like opposition-based learning, adaptive parameters, and local search modifications to overcome these drawbacks and increase performance and robustness in challenging optimization tasks.

3. Problem Definitions

The prediction of diabetes is a supervised binary classification task that involves categorizing patient clinical trials into two distinct groups: diabetic and non-diabetic. The primary aim of this study is to utilize a supervised prediction model to discern the initial phase of diabetes based on the medical data of patients. Let the training data set consist of N samples, each sample $\overrightarrow{S_i} = \langle \overrightarrow{X_i}, Y_i \rangle$ is a vector consisting of conditional variables ($\overrightarrow{X_i}$) and a decision or target variable Y_i with N samples, $X_i \in R^m$ represents the m-dimensional conditional feature space, Y_i represents the target class, where $i = 1, 2, \dots, N$.

4. Research Methods

This section provides comprehensive discussions about the used research methodologies like LSTM, PSO, POBL, LSA, and PLPSO.

4.1 Long Short-Term Memory (LSTM)

The LSTM is a variant of the RNN architecture that incorporates a memory unit capable of retaining information over extended periods [40]. This characteristic enables the LSTM to effectively capture dependencies within sequential data, making it particularly suitable for analyzing sequential patterns. The inclusion of the memory unit addresses the challenges of vanishing or exploding gradients that commonly arise in RNNs. Consequently, LSTM has found applications in diverse domains such as speech recognition, natural language processing, sequential data analysis, time series prediction, language modeling, and machine translation. The LSTM model is comprised of specialized units known as cells and is equipped with three gates. The network's general structure is comprised of layers that are systematically organized as shown in Figure 1. The essential elements of the LSTM model consist of the cell state, hidden state, input gate, forget gate, cell update, and output gate.

The **Cell State (C_t)** might be likened to a pipeline or conveyor that traverses the network, facilitating the transmission of information to each cell within the network. Gate procedures are utilized to either retain or delete the information stored in individual cells. It is alternatively referred to as the long-term memory component of the LSTM model expressed as follows,

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t} \quad (1)$$

The **Hidden State (h_t)** is determined by the current input, the prior hidden state, and the current cell state. The output of the LSTM cell is utilized as input for the subsequent time step and for making predictions. The network's short-term memory is responsible for retaining information temporarily. It is expressed as follows,

$$h_t = o_t \odot \tanh(C_t) \quad (2)$$

The **Input Gate (igt)** is responsible for regulating the flow of information inside the cell state. The process determines whether to update or add values from the input to the cell state. It is given as follows,

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \quad (3)$$

The **Forget Gate (ft)** is responsible for determining the relevance of the information to be discarded, utilizing both the prior concealed state and the present input as follows,

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \quad (4)$$

Cell Update ($\widetilde{C_t}$) refers to the incorporation of fresh information into the existing state of a cell. The composition of the input state in this context involves incorporating possible values derived from the preceding hidden state, the present input, and the determinations made on the input state as follows,

$$\widetilde{C_t} = tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \quad (5)$$

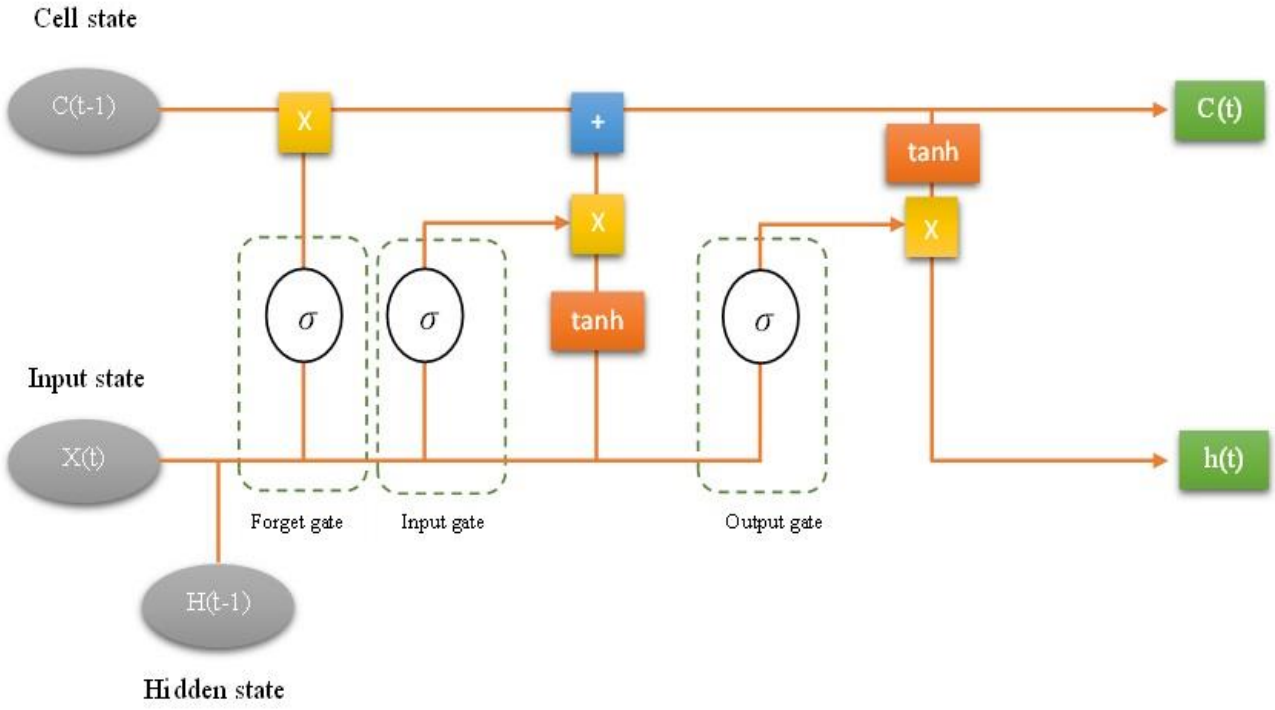

Figure 1. The LSTM architecture

The **Output Gate (O_t)** is responsible for determining the subsequent hidden state by considering the present input and cell state. It is expressed as follows,

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \quad (6)$$

In these equations W and b represent the weights and bias vectors respectively, $\odot$ represents the element-wise multiplication. The effectiveness of this approach has been demonstrated in a range of applications, particularly in cases where the ability to capture and model long-term dependencies is of utmost importance.

4.2 Particle Swarm Optimization (PSO)

PSO is a type of SI technique and it is a method of optimization that involves randomness and probability. This strategy is population-based and is built using the flocking behavior of birds as a basis. The selection of the initial population is determined according to the problem criteria. The population size is determined according to the problem's objective. The major two parameters that represent a particle are particle position and particle velocity. The population consists of a particle swarm, which refers to a collection of particles. The type and size of the population are determined according to the problem criteria. Velocity denotes the orientation of particle motion within the search space. The update process is determined by both the individual's best position and the global best position, which are utilized as follows,

$$v_{ij}(t+1) = wV_{ij}(t) + c_1 r_1 \left(pbest_{ij} - P_{ij}(t)\right) + c_2 r_2 \left(gbest_{ij} - P_{ij}(t)\right) \quad (7)$$

Where, c_1 and c_2 are constant values chosen randomly, and r_1 and r are random numbers between 0 and 1.

$$W = W_{max} - \frac{W_{max} - W_{min}}{iter(max)} - iter(P) \quad (8)$$

Algorithm 1: PSO

Step 1: *Generate a swarm with* ***N*** *particles*

Step 2: *Initialize particles with random position and velocity*

Step 3: *Compute the particle fitness*

Step 4: *Compute the pbest position and gbest position for each particle*

Step 5: *Repeat*

- ***a.*** *Update particle velocity and particle position*
- ***b.*** *Evaluate the particle fitness*
- ***c.*** *Update pbest position for each particle*
- ***d.*** *Update gbest particle*

Step 6: Until stopping criteria;

where the W_{max} is the initial weight and is the final weight of the particle. The updated particle velocity is used for position updating as follows,

$$P_{ij}(t+1) = P_{ij}(t) + v_{ij}(t+1) \quad (9)$$

The fitness of the particle decides the best particle which is defined based on the objective function of the problem. At each iteration, the particle evaluates its fitness and updates its position and velocity by considering its own best position attained so far and the best position achieved by any particle in the swarm. Algorithm 1 represents the pseudocode of the PSO algorithm.

4.3 Partial Opposition based Learning (POBL)

The opposition-based learning [10] is one of the machine learning techniques used for imputing stochastism in any learning problem and searching for a solution in the opposite way to increase the search space exploration and exploitation. For example: in D-dimensional space let $P = (p_1, p_2, p_3, \dots, p_D) \in \mathbb{R}^D$, where every $p_i \in [l_i, u_i]$. The opposite number can be computed as $p_i' = l_i + u_i - p_i$. This opposite number is for i^{th} particle; instead of computing the opposite value for all the particles only some particles undergo opposition and the remaining particles are restored is referred to as POBL. POBL is a novel OBL scheme that was created by Z. Hu et al. (2014). An innovative variant of the OBL method is implemented within the adaptive differential evolution (ADE) algorithm [41]. An entirely distinct point for OBL in a multidimensional search space may be represented as $\begin{bmatrix} p_1 & \acute{p}_2 & \acute{p}_3 \\ \acute{p}_1 & p_2 & \acute{p}_3 \end{bmatrix}$. This point exhibits opposite values to the original point in every dimension. The divergent viewpoints of a specific point, If X in a D-dimensional space can be defined as the set of points $o\acute{P}$ as follows,

$$o\acute{P}^1 = \begin{bmatrix} o\acute{P}_1^1 \\ o\acute{P}_2^1 \\ \cdots \\ o\acute{P}_d^1 \end{bmatrix}_{D\times 1} = \begin{bmatrix} p_1 & \acute{p}_2 & \acute{p} & \dots & \acute{p}_D \\ \acute{p}_1 & p_2 & \acute{p}_3 & \dots & \acute{p}_D \\ \vdots & \vdots & \vdots & \ddots & \dots \\ \acute{p}_1 & \acute{p}_2 & \acute{p}_3 & \dots & p_D \end{bmatrix}_{D\times D} \quad (10)$$

The level of partial opposition is denoted by the superscript 1 in $o\acute{P}^1$. Given that each point corresponds to a unique integer in a single dimension, the aforementioned partially opposed points can be classified as having a first-order or linear relationship.

4.4 Local search algorithm (LSA)

The LSA can enhance the system's exploitation capacity. By carefully exploring the immediate vicinity of the current solutions, LSA aids in solution refinement and ensures a more sophisticated convergence to the global optimum [42]. While LSAs are primarily concerned with local search, when utilized strategically, they can aid in escaping local optima by offering a variety of options, which, when paired with any SI, facilitates greater exploration of the search space. The swarm's solutions can be swiftly adjusted by LSAs, accelerating the convergence of low-quality solutions. Algorithm 2 represents the LSA approach,

Algorithm 2: LSA Algorithm

Step 1: *best = best_particle (search-agent)*

Step 2: *Temp=* gbest

Step 3: *Lt=1*

Step 4: While *(Lt < Maximum iteration)*

Randomly select three values from the Temp

If *selected-best == 1 (1 means the value is chosen and 0 means not chosen)*

Selected-best= 0

Else

Selected-best=1

End if

Compute the fitness value of Temp

If *f(Temp) < f(F)*

F=Temp

End if

Lt= Lt+1

End while

Step 4: *Return F*

4.5 PSO based on OBL and LSA

PSO uses straightforward principles for updating particle locations and velocities along with elementary mathematical operations, making it very easy to learn and apply. Because there aren't many parameters to adjust, the optimization process is made easier. The many studies that have been applied to different real-world scenarios are the motivations behind this. However, there two major shortcomings are discussed in PSO such as premature convergence and population diversity. PSO is prone to premature convergence to local optima, particularly in intricate fitness landscapes with several modes. Because particles tend to follow the best-performing particles, they may become stuck in local optima and not explore as much. When the diversity of the swarm decreases over time, particles may begin to concentrate around a less-than-ideal solution. The search for space exploration may be lessened by the natural mechanism of particles convergent towards the most well-known spots. To find potential regions in the search space fast during the early stages of optimization, OBL offers a broad exploratory strategy.

However, POBL offers a more consistent and regulated search process, which makes it appropriate for more complex optimization problems and fine-tuning. The particulars of the optimization problem at hand will determine which of OBL and POBL to choose, with POBL typically being more successful at striking a

balance between exploration and exploitation. On the other hand, the combination of the LSA algorithm and PSO enhances its utilization and prevents it from becoming trapped in local optima. The PLPSO algorithm consists of essential phases such as POBL-based population initialization and best particle finding based on LSA. Initially, PLPSO uses OBL to create a population of particles. Subsequently, it selects the n best-fit particles from both the starting and destination particle positions. Furthermore, F will be assigned to the best particle out of these n fittest particles. In addition, these n particles will have their positions updated by applying the main loop. Currently, LSA will be applied to the present to check and find a better solution than the best one presently found at the end of the PLPSO main loop. PLPSO will ultimately provide the optimal results.

4.5.1 POBL-based population initialization

The population of PLPSO is initialized by computing random values for each dimension of a particle. The population initialization includes fixing the size and nature of the population and defining the position and velocity for each particle. The positions can be initialized using equation (12), where each dimension is assigned a random value within the search space. The position of the j^{th} dimension of the i^{th} particle is initialized as follows,

$$P_{ij}(t)|_{(t=0)} = P_j^{min} + \left(P_j^{max} - P_j^{min}\right).rand_{ij}^{u}(t)|_{(t=0)} \quad (11)$$

where $rand_{ij}^{u}(t)|_{(t=0)}$ - uniformly distributed random values in the range [0,1]. Similarly, the velocity of each particle can be initialized using the equation (13). The j^{th} dimension of $V_{ij}(t)|_{(t=0)}$ of the i^{th} particle is initialized as follows:

$$V_{ij}(t)|_{(t=0)} = V_j^{min} + \left(V_j^{max} - V_j^{min}\right).rand_{ij}^{u}(t)|_{(t=0)} \quad (12)$$

After initialization of position $P_i(t)|_{(t=0)}$ and velocity $V_i(t)|_{(t=0)}$ of i^{th} particle, the opposite of position and velocity are considered. Finally, select the particles with a high probability ranking of occurrence from the original swarm and its opposite swarm.

4.5.2 Local search algorithm (LSA)

The best position found by any particle throughout the entire swarm is known as the global best position. By guaranteeing that every particle has access to the most well-known global solution and encouraging convergence in its direction, it offers a type of social learning. The ideal location aids in striking a balance between exploitation and exploration, resulting in efficient optimization. Nonetheless, the optimum best is chosen from all populations using the traditional PSO. As a result, poor convergence rates are generated, and obtaining globally optimal solutions requires longer calculation times. Additionally, the suggested work selects the best possible solution for each iteration using the LSA technique. The LSA algorithm selects the best option based on fitness values and uses a random process to extract the three values from the total population. After every PSO iteration, LSA will be invoked to improve the already optimal solution. When the PSO iteration ends, LSA initially stores the value of the best solution that was obtained by PSO. LSA makes several iterations to enhance global best value. Every time LSA runs, it randomly chooses the three gbest values from Temp. Depending on its parameters, LSA sets or resets the chosen gbest values. Additionally, LSA will calculate the new solution's fitness value. If it is higher than the previous solution's fitness value, it will be set to Temp; otherwise, it stays untouched.

5. Proposed PLPSO-LSTM

The proposed PLPSO-LSTM model uses the PLPSO for obtaining appropriate hyperparameters for LSTM. The new training algorithm PLPSO uses the POBL algorithm which will prevent premature convergence and improve the diversity of the swarm in the PSO algorithm. The PLPSO algorithm is utilized to optimize the hyperparameters of the LSTM model and mitigate the subjective influence of manually selected parameters. Algorithm 3 shows the step-by-step procedure and Figure 2 shows the workflow of the PLPSO-LSTM model. Initially, fix the parameters of PLPSO and LSTM; generate the PLPSO population, and compute the opposite population with their fitness. Then select the best k particles from the set $\{P, P'\}$.

Algorithm 3: Proposed PLPSO-LSTM

Step 1: *Fix PLPSO parameters: number of particles, particle size, $iter_{max}$, and other parameters*

Step 2: *Fix LSTM Parameters: input size, hidden units, and output unit as in Table 3.*

Step 3: *Initialize **PLPSO** population P and its fitness*

Step 4: *Compute the opposite swarm P' and its fitness*

Step 5: *Select k best particles from $\{P, P'\}$*

Step 6: *Evaluate P_{pbest} and P_{gbest}*

Step 7: *Repeat $(t \leq iter_{max})$ or until convergence*

a) If $rand(0,1) \leq prob(P_{pbest}(t))$

i. *Compute the $o\acute{P}^{a}(t)$ the partial opposition particles for $P(t)$*

ii. *Select k best particles from $\{P(t), \acute{P}(t), o\acute{P}^{a}(t)\}$*

End if

b) for $(i = 1{:}\,k)$

Update the velocity V_i and position P_i, compute fitness of i^{th} particle, and update $P_{pbest}(t)$ and $P_{gbest}(t)$

End for

c) *gbest = $\mathcal{F}$((search-agent)*

d) *Determine whether there is a better solution by applying **LSA** to $\mathcal{F}$. If one is found, update $\mathcal{F}$; if not, leave it untouched.*

e) *End while*

Step 8: *Initialization of LSTM cell state C_0 and hidden state h_0 to a small random number, and time step t=1*

Step 9: *Assign P_{gbest} to the LSTM weights and bias*

Step 10: *Repeat until convergence*

a. *Feed inputs x_t, $h_{(t-1)}$ and $C_{(t-1)}$ to LSTM*

b. *Perform three gating operations (input, forget, and output gates) of LSTM*

c. *Add new candidate values $\widetilde{C}_t$ to the current cell state*

d. *Update cell state C_t and hidden state h_t*

Step 11: *Return trained network*

These particles are the initial population of the PLPSO algorithm? These particles are iterated through several iterations. If the probability of the $P_{pbest}(t)$ particle is lesser than the random value then LSA is used to select the current best population is computed. Then k best particles of $\{P(t), \acute{P}(t), o\acute{P}^a(t)\}$ is selected for further processing. Further update the velocity, position, and fitness of the population and perform the LSA algorithm. After convergence, the optimal hyperparameters are obtained and assigned it to the LSTM network. Then train the network as shown in Figure 2 until convergence. High robustness for diabetes diagnosis is achieved by the optimized LSTM model with the combination of PSO and LSA within an improved PSO framework. This method successfully strikes a compromise between exploration and exploitation, guaranteeing that the model retains steady convergence and robustness to data fluctuation while also generalizing well to new data. All of these features work together to make the model a trustworthy resource for assisting with diabetes clinical diagnosis.

6. Experimental Analysis and Discussion

The proposed models are developed using MATLAB 2019b on a Windows 10 Professional (64-bit) PC with an Intel I7 processor and 8 GB of RAM. This section presents a comparison between the diabetes prediction results obtained using the proposed PLPSO-LSTM model and those obtained from traditional approaches. Table 2 provides the descriptions of the different diabetes datasets utilized in this investigation. The results are given based on quality and performance indicators, convergence analysis, ROC, and AUC analysis.

6.1 Datasets

The present research work considered four difference diabetes datasets such as PIMA, CDC, MDD, and DTD. The details of each dataset discuss as below,

- **PIMA:** the Pima Indian datasets contain nine columns that make up the dataset, and one output column contains a binary value that indicates whether a person has diabetes or not. There are 268 patients with diabetes out of the 768 rows, 500 of them are not diabetics. The dataset has nine feature columns and one class label.
- **CDC:** CDC Diabetes Health Indicators dataset contains 2, 53,680 data samples, 21 features with 3 classes such as Diabetes (213703), Pre-diabetes (4631), and Normal (35346).
- **MDD:** the MDD dataset contains 1000 data samples, 13 features with 3 classes. The patient's diabetes disease class may be Diabetic (844), Pre-Diabetic (53), and Normal (103).
- **DTD:** the DTD dataset contains 1009 data samples, and 8 features with 2 classes. The 632 normal class and 377 diabetes samples are collected in this dataset.

6.2 Hyperparameter Optimization

Hyper parameters impact the effectiveness, efficiency, and results of algorithms and are essential to both SI and DL. In both SI and DL, hyper parameters have a big impact on how algorithms behave and work. To get the best outcomes, these parameters must be properly adjusted, which calls for technique. The present research work uses the optimized LSTM for the diagnosis of diabetes. LSTM network performance and efficiency are greatly influenced by hyper parameter adjustment. LSTMs are quite sensitive to the selection of hyper parameters and their significant tasks. Selecting suitable hyper parameters can greatly increase the model's precision, rate of convergence and capacity for generalization. The present research work uses the PLPSO algorithm for selecting optimal hyper parameter optimization such as weights, biases, and learning rate. The strength of the connections between neurons in neighboring layers is represented by ***weights***. The amount of influence that a neuron's output will have on another neuron's input in the layer below is determined by each weight. The range of the weights is considered between 0.5 to -0.5. ***Biases*** provide the network with an extra degree of flexibility. By serving as offsets, they enable neurons' activation to change up or down, improving the network's ability to appropriately predict the data.

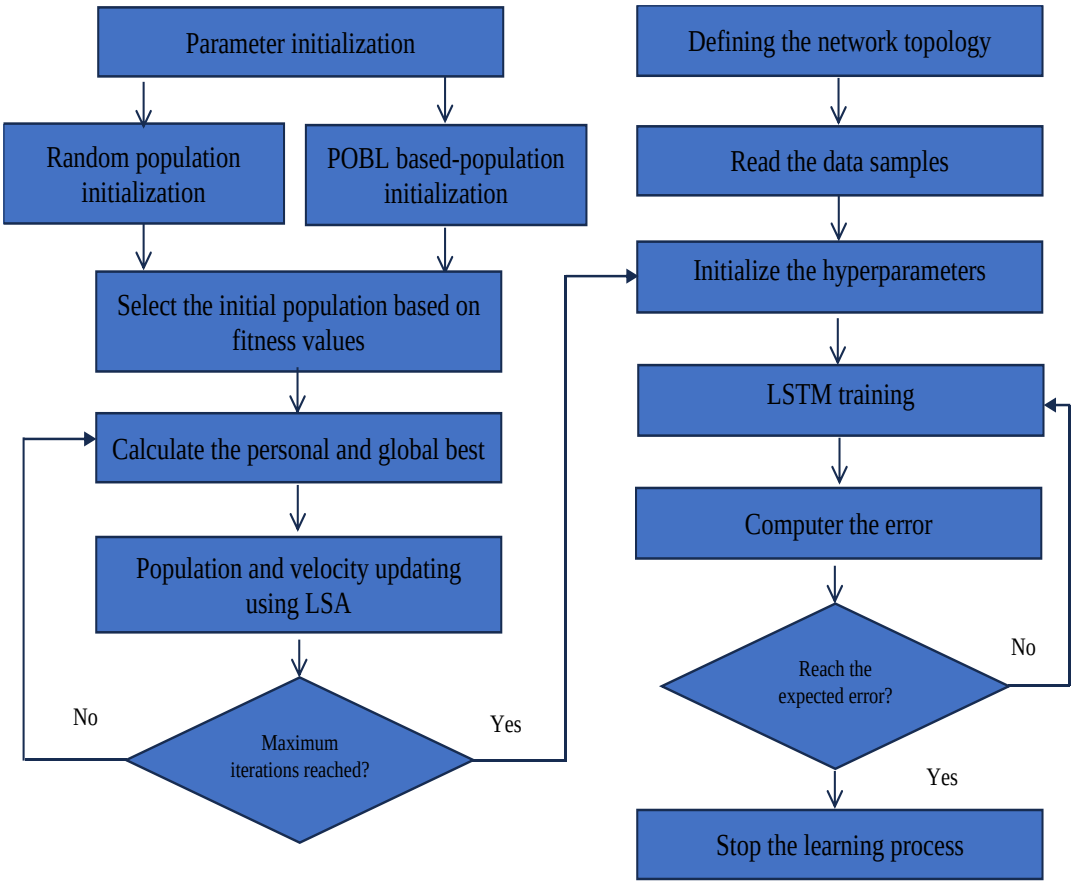

Figure 2. Workflow of PLPSO-LSTM

The network's capacity to fit the data would be constrained in the absence of biases. The ***learning rate*** controls the step size of the optimization process which ranges is between 0 to 1. While an overly low learning rate could impede the training process, an overly high learning rate could hasten the model's convergence to a subpar answer. Further, LSTM cells frequently activate the sigmoid, tanh functions, and softmax for input, hidden, and output layers. Depending on the issue, the number of hidden neurons in each layer can be adjusted as a hyperparameter.

Underfitting can result from having too few neurons, and overfitting can result from having too many. Fifty hidden neurons in the hidden layer are considered. "Epoch" denotes a single iteration of the learning process over the whole training dataset. One hyper parameter that determines how many times the learning algorithm will run through the training dataset is the number of epochs. The epochs are selected as 300 and optimal learning is selected as 0.05. The optimal parameters of PLPSO and LSTM are given in table 3.

6.3 Comparison Methods

The various conventional and existing methodologies used for the comparative analysis are discussed in detail. The SVM is one of the pioneer conventional supervised classification models used for most of the classification problems [13]. It is very efficient in learning complex problems is used for most real-time problems and is used as a benchmark approach for classification. It provides a very good approximation for non-linear classification problems. The BPNN is one of the artificial neural networks that use the backpropagation algorithm for training [43]. It can also handle non-linear problems efficiently with higher fault tolerance and universal approximation of complex problems.

Table 2. Datasets details

S. No	Datasets	Instances	Features	Classes
1	PIMA	768	8	2
2	CDC	2,53,680	21	3
3	MDD	1,000	14	3
4	DTD	1,009	8	2

Table 3. Parameter Settings for PLPSO-LSTM model

LSTM		PSO	
Parameters	**Values**	**Parameters**	**Values**
Learning rate	0.05	Population Size	20
LSTM layer	4	C_1	2.0
Epochs	300	C_2	2.0
Dropout rate	0.5	R_1	0.01
Batch size	20	R_2	0.09
Hidden neurons	50		
Loss function	MSE		

Even though it can analyze non-linear relationships, it is probably sensitive to hyper-parameters and owes to overfitting. It is used in vast applications in NLP, control systems, and finance applications. The Convolutional LSTM (CLSTM) is the hybrid model of the CNN with the LSTM model [33]. It adopted the Convolutional layer and yielded better performance in diabetes prediction. However, the model experimented with the PIMA dataset alone and the robustness and reliability of the model are questionable. The model may probably suffer from an overfitting issue. The next hybrid approach is the GA-based LSTM model, it is used for the Parkinson's disease diagnosis by Srivastava *et al.* (2021) [44, 45]. This enhancement causes better performance in that application by fixing the optimal weights of the LSTM model. The GA model still possesses a higher probability of local optimal convergence as well as a slow convergence rate. Zhang, *et al* (2019) [46] proposed PSO-LSTM for the stock price prediction problem. The PSO is used for the optimal weight and bias initialization of the LSTM. This enhancement produced a better result and this approach is used for diabetes prediction in this research work. Similarly, the APSO-based LSTM model was used for monkeypox diagnosis by Shobhana *et al.* (2021). They also used the adaptive PSO for fixing the optimal weights of LSTM and yield a significant result than the conventional approaches.

6.4 Preprocessing

To prepare the data for feeding into the DL algorithm, pre-processing is a crucial step that transforms the data into an effective and useful shape. The present research work focused on three kinds of pre-processing steps such as filling in missing values, label encoding, and data normalization. The median for each attribute served as a stand-in for all the missing values. Then, label encoding is performed for all datasets. Because the datasets have categorical and numerical values. Two class labels are present in the PIMA and DTD databases; these include yes for diabetes and no for normal. After encoding, the value is 1 for diabetes, whereas for normal samples, it is 0. The two remaining datasets, the CDC and the MDD, are divided into three classes: Normal, Pre-diabetes, and Diabetes. Hence, these classes are denoted by the numbers 0, 1, and 2, respectively. Data normalization is the final step in the pre-processing of data. Data transformation into a linear format is accomplished using this method. The data set is normalized through the application of the min-max transformation technique. As a result, the properties are normalized from one value range to another which is defined as follows,

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (13)$$

Where x represent the present value, x_{max} and x_{min} denote the maximum and minimum values in the input samples.

6.5 k-Fold Cross-Validation Techniques

The optimized LSTM framework's performance is evaluated by the application of the k -fold cross-validation technique. Ten equal portions are taken from the collected dataset using a k -fold cross-validation approach. If there are k groups in total, the classifier uses $k-1$ groups as its training dataset and the remaining part as its testing dataset. Because the technique is carried out to validate the dataset k times, it is known as k -fold validation. Based on the model, the k findings aid in estimating the classifier's performance. To improve accuracy, the testing dataset evaluates the model's performance at each stage. In this experiment, ten classes are created from the original dataset using the 10-fold cross-validation technique. Nine groups remain, which are used for model training. Each group is regarded as legitimate data on its own. The dataset contains ten repetitions of the entire operation. After the cycle, the average of the performance metrics is determined. Ten-fold cross-validation techniques were used to test the classifiers. The training dataset is divided into two halves using the 10-fold cross-validation

method: 90% of the data is used for training and the remaining 10% is utilized for testing.

6.6 Performance Metrics

Typically, performance indicators are utilized to justify the efficiency of any classification model. The comprehensive explanations of these measurements are shown below.

6.6.1 Accuracy

A general indicator of the model's correctness, accuracy gives the ratio of successfully predicted instances to all instances. The prediction's accuracy is indicated as follows,

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (14)$$

6.6.2 Precision

The number of positive forecasts that come true is known as precision. The proportion of accurately forecast positive observations to the total number of positive predictions is well-defined as follows,

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

6.6.3 Recall

The proportion of all observations made in the actual class to all positively anticipated observations. Recall quantifies the model's capacity to recognize every pertinent instance (true positives) which is defined as follows,

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

6.6.4 F-Measure

F-Measure is the harmonic mean of precision and recall, strikes a balance between the two, and is helpful when both false positives and false negatives are significant.

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (17)$$

6.6.5 Receiver operating characteristic (ROC) and Area under the curve (AUC)

The ROC curve is valuable for evaluating the True Positive Rate (TPR) and False Positive Rate (FPR) at various categorization thresholds. A probability graph illustrates the fluctuations in the classification thresholds, where the minimum threshold value might potentially raise both the false positive rate (FPR) and the true positive rate (TPR). The ROC curve refers to the spatial region enclosed by the ROC curve, which spans from the point (0,0) to the point (1,1). The AUC value ranges from 0 to 1. The maximum area (about 1) represents the highest level of separability in the classification model. The scale-invariant measure is used to assess the quality of forecasts without relying on absolute values. The measure is classification-threshold invariant, meaning it evaluates the quality of the model's predictions without considering the classification threshold. Here, True positive (TP) refers to the proportion of patients who are labeled as positive but are positive. The number of patients who are genuinely negative despite being projected to be negative is known as true negatives (TP). The number of patients who are categorized as positive but are negative is known as false positives (FP). Furthermore, the number of patients who are labeled as negative but are positive is known as false negatives (FN). To evaluate the models' quality of categorization, these parameters are frequently estimated.

6.6.6 Computation time

The computation time determines how long it takes the suggested framework to produce a prediction. Our dataset is used here to train various models. The 10-fold cross-validation dataset made a diagnosis about diabetes disease using only CPU time.

7. Results and Analysis

The proposed model PLPSO-LSTM is experimented on four diabetes datasets and compared with seven existing models. This section further discusses the performance validation, convergence analysis, complexity analysis, and AUC-ROC analysis of the experimental results. The results of the proposed model on the PIMA dataset are compared with twenty-five approaches in the literature which is highlighted in Table 4. This shows that the proposed model has attained the maximum accuracy of 98.92 percent. Among these approaches, the proposed model proved to be the significant model in diabetes prediction.

The construction of DL models involves two essential stages: training and testing. The training split is utilized for the construction of the model, while the testing split is employed for the evaluation of the model. Each is essential to the model's ability to generate precise predictions and generalize well to fresh, untested data. The training and testing results are shown in Tables 5, 6, 7, and 8 based on performance measures such as accuracy, f-measure, precision, and recall.

According to the performance measures, the developed proposed diabetes prediction model produced higher performance in terms of both training and testing.

Table 4. Performance results comparison on various diagnostic methods using PIMA Dataset

Author(s) (Year)	Model	Accuracy	Author & year	Model	Accuracy
Kandhasamy *et al.* (2015) [13]	SVM	86.46 %	Lakhwani *et al.* (2020) [26]	FFNN	88.8 %
Erkaymaz *et al.* (2016) [14]	SW-FFNN	91.66 %	Abedini *et al.* (2020) [27]	Ensemble DT, LR, & ANN	83.08 %
Yu *et al*. (2017) [15]	LDA	68.70 %	Pradhan *et al.* (2020) [29]	ANN with 4 hidden layers	88.71 %
Chen *et al*. (2017) [16]	K-means DT	90.04 %	Zhou *et al.* (2020) [47]	Enhanced DNN	94.02 %
Dadgar *et al.* (2017) [17]	MLP-GA	87.46 %	Guldogan *et al* (2020) [39]	MLP and RBF	78.1 %
Hashi *et al.* (2017) [18]	SFSA	87.01 %	Bukhari *et al.* (2021) [31]	ABP-SCGNN	93 %
Zhang *et al.* (2018) [19]	FFNN	82 %	Nadeem *et al*. (2021) [32]	Hybrid SVM-ANN	94.67 %
Haritha *et al.* (2018) [20]	Firefly with Cuckoo Search	81 %	P. B. K. Chowdary *et al.* (2021) [33]	CLSTM	95.6 %
Srivastava *et al.* (2019) [21]	ANN	92 %	Sivasankari *et al.* (2022) [34]	MLP	86.06 %
Kannadasan *et al.* (2019) [22]	Stacked Auto-encoder	86.26 %	Pujari (2022) [35]	SVM with Polynomial Kernel	96 %
Prema *et al.* (2019) [23]	Ensemble of SVM	80.52 %	Chang *et al*. 2023 [36]	ML algorithms	79.57 %
Dinh *et al.* (2019) [24]	XGBoost	95 %	Mousa *et al.* 2023 [37]	LSTM, RF, CNN	85 %
Pradhan *et al.* (2020) [25]	ANN	85.09 %	**Proposed Model**	**PLPSO-LSTM**	**98.92 %**

Table 5. Performance results of diagnostic models on the PIMA Dataset

PIMA		PLPSO-LSTM	APSO-LSTM	PSO-LSTM	GA-LSTM	CLSTM	LSTM	BPNN	SVM
Accuracy	**Test**	**98.92**	96.60	95.21	93.78	91.82	88.80	85.00	86.46
	Train	**97.64**	96.68	95.33	94.68	92.55	88.25	81.37	88.08
F-Measure	**Test**	**98.70**	95.05	92.94	89.35	87.94	86.49	83.86	84.64
	Train	**96.68**	89.52	86.92	85.36	83.46	77.88	84.78	85.49
Precision	**Test**	**99.02**	96.22	95.96	92.97	92.35	88.87	82.79	85.40
	Train	**99.36**	97.27	95.23	93.92	92.89	87.5	85.92	86.38
Recall	**Test**	**99.39**	86.43	91.98	89.57	86.01	84.23	84.95	83.90
	Train	**99.85**	95.38	90.44	89.33	87.67	85.51	85.28	84.39

Table 6. Performance results of diagnostic models on the CDC Dataset

CDC		PLPSO-LSTM	APSO-LSTM	PSO-LSTM	GA-LSTM	CLSTM	LSTM	BPNN	SVM
Accuracy	Test	**89.39**	85.93	84.07	83.92	85.06	84.97	79.93	81.93
	Train	**93.67**	88.42	85.94	82.10	81.75	80.09	75.47	78.30
F-Measure	Test	**90.27**	78.17	75.95	60.69	65.63	61.99	62.82	57.71
	Train	**81.90**	78.44	76.54	61.14	65.71	57.31	63.21	52.67
Precision	Test	**95.08**	86.97	91.37	89.59	89.28	84.66	86.54	84.21
	Train	**92.03**	82.55	86.97	89.35	80.88	84.88	84.47	84.37
Recall	Test	**82.90**	70.98	64.98	45.89	51.89	48.89	49.31	43.90
	Train	**83.78**	71.84	65.30	43.99	48.24	41.08	45.99	42.00

Table 7. Performance results of diagnostic models on the MDD Dataset

MDD		PLPSO-LSTM	APSO-LSTM	PSO-LSTM	GA-LSTM	CLSTM	LSTM	BPNN	SVM
Accuracy	Test	**98.76**	96.53	92.69	91.56	89.29	90.49	89.95	90.79
	Train	**98.36**	96.58	87.00	84.66	82.59	82.46	86.87	91.14
F-Measure	Test	**94.56**	97.06	92.80	89.85	88.06	88.15	90.00	91.38
	Train	**95.34**	97.83	93.17	85.93	88.62	88.42	90.27	86.77
Precision	Test	**96.34**	87.41	94.44	89.97	89.89	87.03	83.93	76.26
	Train	**94.20**	81.88	91.25	84.73	89.81	87.23	84.51	76.44
Recall	Test	**94.71**	90.99	89.85	89.95	91.61	86.51	81.22	82.77
	Train	**87.02**	89.84	90.42	90.30	86.25	86.73	81.30	83.09

Table 8. Performance results of diagnostic models on DTD Dataset

DTD		PLPSO-LSTM	APSO-LSTM	PSO-LSTM	GA-LSTM	CLSTM	LSTM	BPNN	SVM
Accuracy	Test	**96.03**	90.81	93.90	91.94	89.09	93.89	91.08	92.87
	Train	**92.29**	91.32	89.20	82.99	89.69	94.38	91.85	93.86
F-Measure	Test	**91.56**	83.39	86.37	79.73	81.78	78.71	75.73	77.35
	Train	**92.06**	84.38	80.81	70.84	77.17	76.97	76.63	71.55
Precision	Test	**92.31**	85.90	87.92	80.43	82.95	74.08	71.06	74.92
	Train	**90.79**	85.93	88.51	80.53	77.78	74.68	71.24	69.41
Recall	Test	**90.82**	81.03	84.87	79.05	80.64	83.95	81.06	79.95
	Train	**84.01**	72.26	78.90	71.01	81.36	79.08	81.14	76.93

In the training phase, the PLPSO-LSTM model achieved a maximum accuracy of 97.64 %, an f-measure of 96.68 %, a precision of 96.39%, and a recall of 99.85% on the PIMA dataset. The PLPSO-LSTM model achieved a maximum accuracy of 93.67 %, an f-measure of 81.90 %, a precision of 92.03%, and a recall of 83.33% on the CDC dataset. The PLPSO-LSTM model achieved a maximum accuracy of 98.36 %, an f-measure of 95.34 %, a precision of 94.20%, and a recall of 87.02% on the MDD dataset.

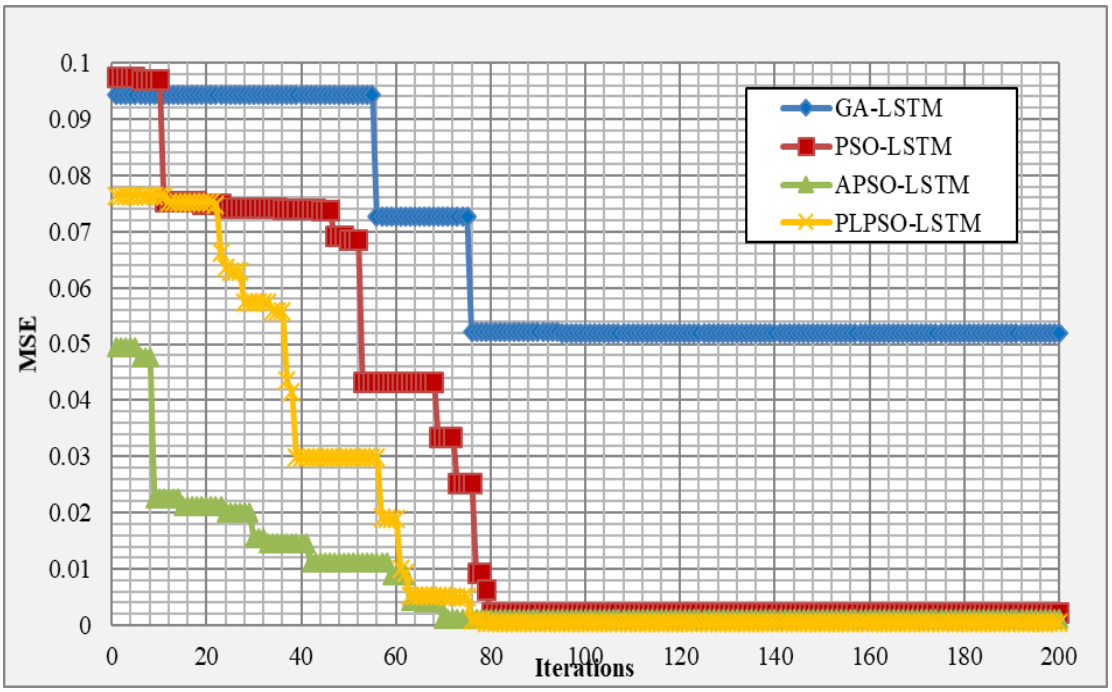

Figure 3. Convergence analysis of PLPSO-LSTM on PIMA Dataset

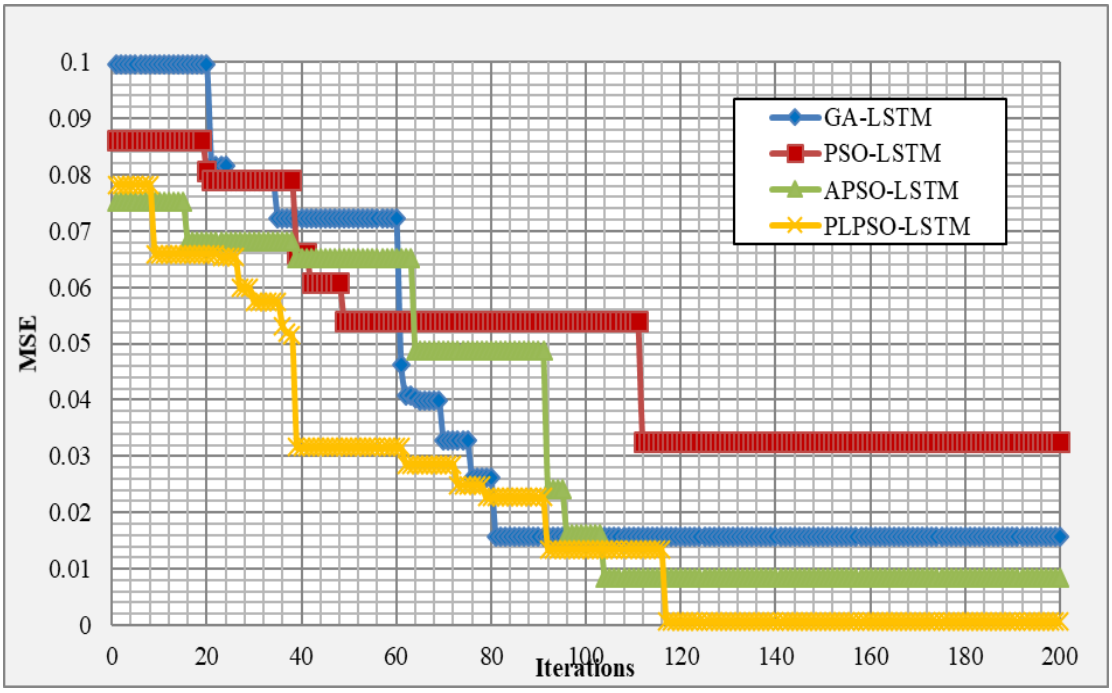

Figure 4. Convergence analysis of PLPSO-LSTM on CDC Dataset

The PLPSO-LSTM model achieved a maximum accuracy of 92.09 %, an f-measure of 92.06 %, a precision of 90.79%, and a recall of 84.01% on the MDD dataset.

In testing, the PLPSO-LSTM model achieved a maximum accuracy of 98.92%, an f-measure of 98.70 %, a precision of 99.02%, and a recall of 99.80% on the PIMA dataset. In terms of accuracy, the PLPSO model demonstrated a 4 to 7 percent enhancement compared to the hybrid LSTM models, and a 10 to 15 percent improvement compared to the conventional approaches. There is a remarkable improvement of about 5 to 17 percent in terms of precision, recall, and f-measure. The proposed model obtained maximal performance of 89.93%, 90.27%, 95.08%, and 82.90% of accuracy, f-measure, precision, and recall respectively for the CDC dataset. It results in an improvement ranging from 3 to 30 percent compared to the existing hybrid and conventional models. A maximum performance of 98.76 in terms of accuracy, 94.56 percent in terms of f-measure, 96.34 percent in terms of precision, and 94.71 percent in terms of recall was achieved by the proposed model for the MDD dataset.

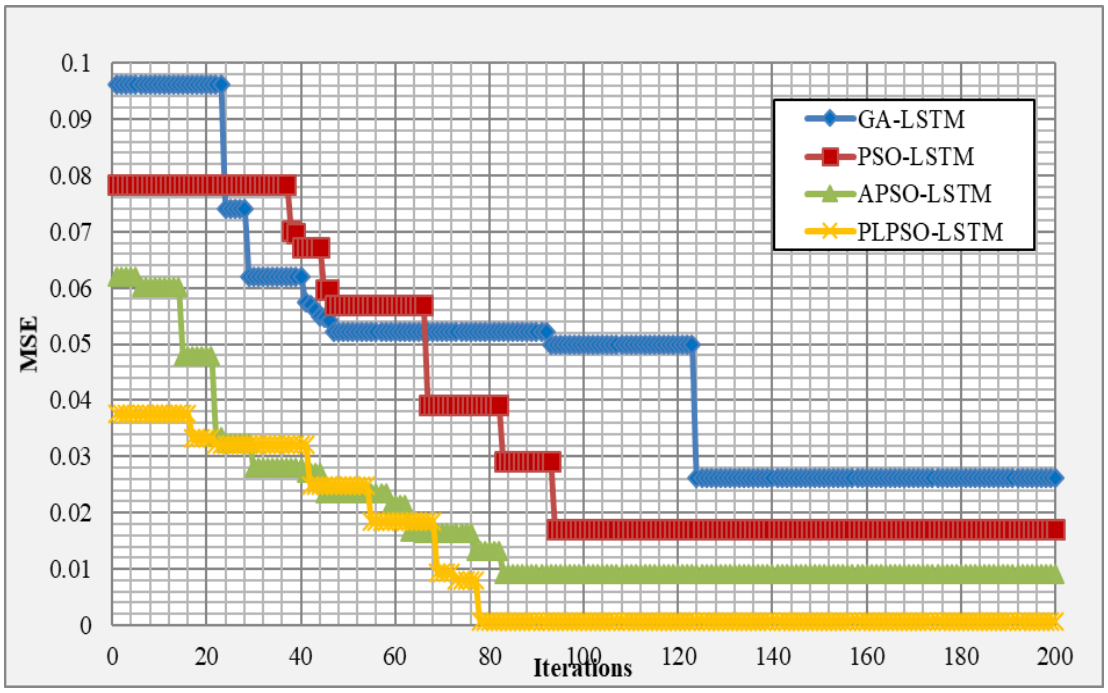

Figure 5. Convergence analysis of PLPSO-LSTM on MDD Dataset

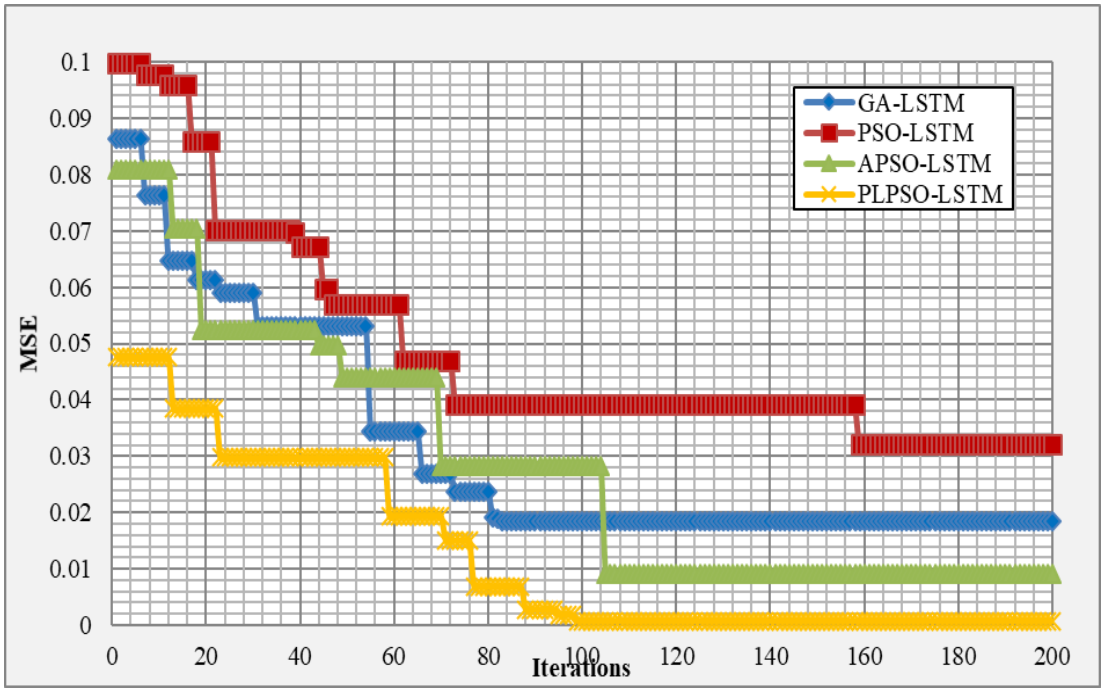

Figure 6. Convergence analysis of PLPSO-LSTM on DTD Dataset

When compared to the existing hybrid and traditional models, it results in an improvement that ranges from three to thirty-three percent. For the DTD dataset, the proposed PLPSO-LSTM model yields a maximal accuracy of 96.03, which is ≈5 percent higher than the existing models. Similarly obtained ≈an 8 to 15 percent hike for f-measure, precision, and recall. It justifies the generalization and pattern recognition ability of the proposed PLPSO-LSTM model.

Understanding and validating the learning process and predictive performance of the model is crucial when using convergent analysis with the proposed PLPSO-LSTM for diabetes prediction. If the LSTM model has stabilized to the point where more training does not appreciably enhance performance, convergence analysis can be used to assess this. The convergence analysis of GA-LSTM, PSO-LSTM, APSO-LSTM, and the suggested model PLPSO-LSTM, for four diabetes datasets, including PIMA, CDC, MDD, and DTD are described. The examination of convergence for these models is shown in Figures 3, 4, 5 and 6. The convergence curve uses the MSE value which is plotted against the number of iterations. The PLPSO-LSTM

model has reached the global optimum compared with other methods. The results indicate that combining LSTM with other techniques enhances the speed at which the model achieves convergence. Additionally, it is observed that achieving the lowest MSE requires finding the optimal hyperparameters. Moreover, the proposed model achieves this low MSE at an earlier stage compared to traditional models, across all datasets. The proposed model has achieved the highest level of accuracy with the fastest convergence when compared to both hybrid and conventional techniques.

The AUC and ROC curves are essential tools for assessing how well PLPSO-LSTM models predict diabetes. They guarantee the accuracy and dependability of predictions, aid in the selection of the best threshold, enable comparison analysis and offer a thorough grasp of the discriminative capability of the model—all of which are essential for successful diabetes management and intervention techniques. This analysis studies the efficiency of the diagnostic models at different decision thresholds. The ROC analysis on these four datasets is highlighted in Figures 7, 8, 9 and 10. For the PIMA dataset, the PLPSO-LSTM and APSO-LSTM approaches have yielded the highest AUC of 0.98 which is almost 3 to 5 percent higher than other approaches. It reached a maximum possible value of 1.00 with an FPR of 0.08, whereas it is around 0.40 for other existing approaches. This proves the outstanding capabilities of the proposed model.

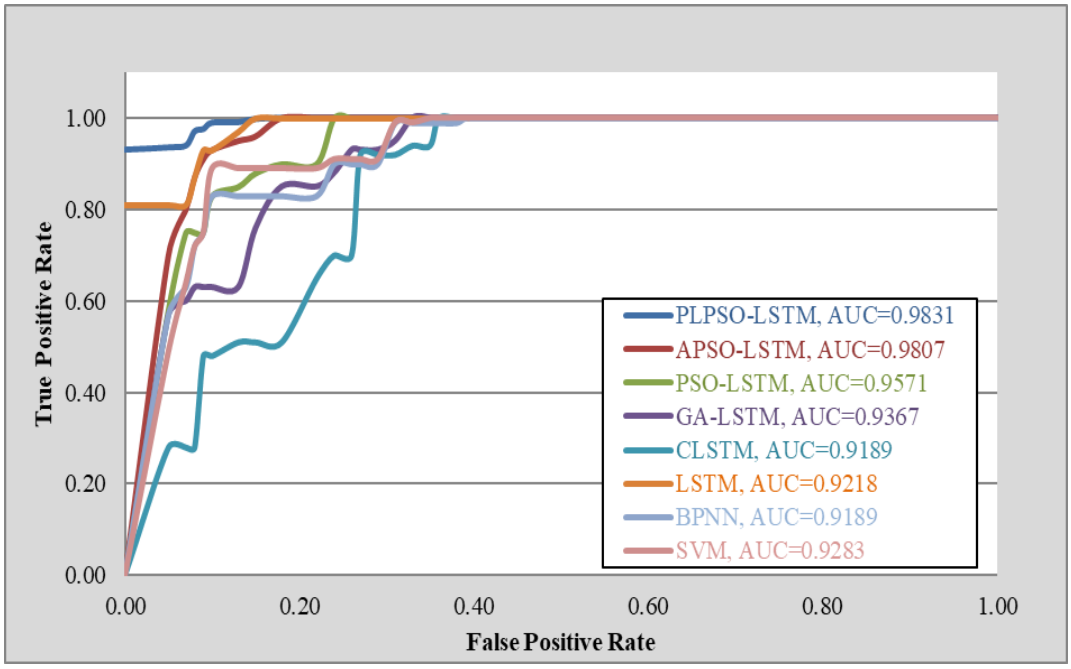

Figure 7. AUC and ROC analysis on the PIMA Dataset

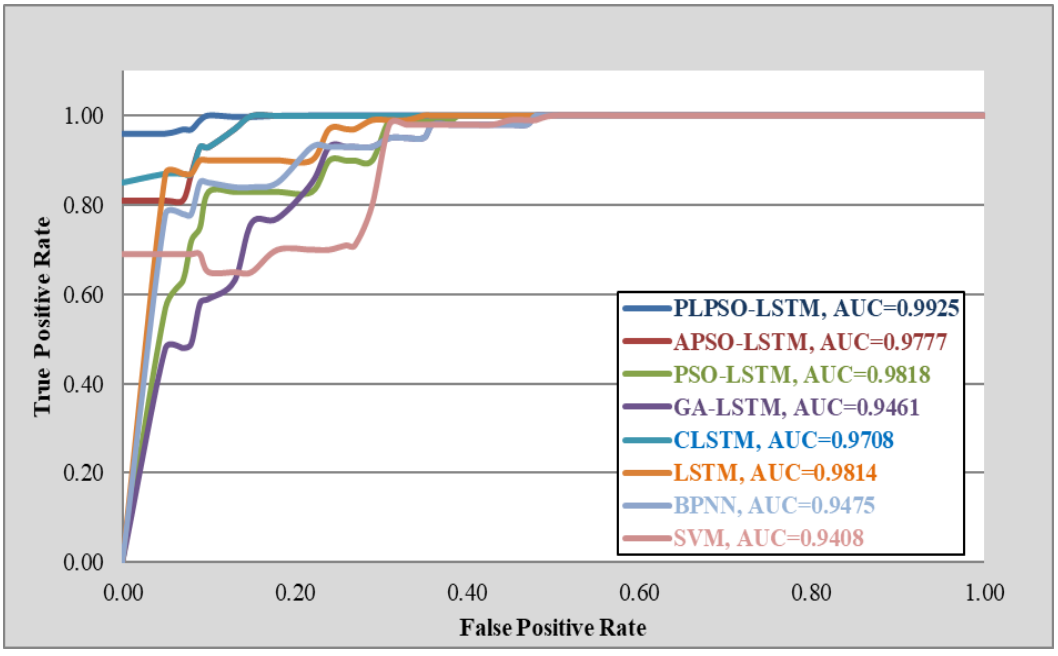

Figure 8. AUC and ROC analysis on the CDC Dataset

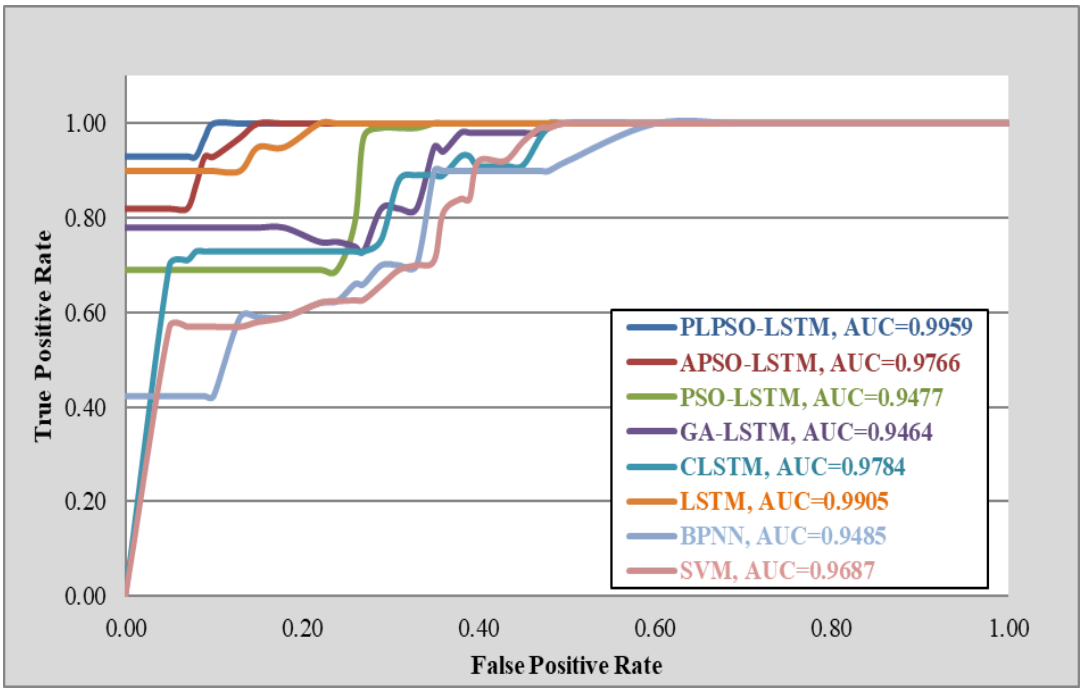

Figure 9. AUC and ROC analysis on the MDD Dataset

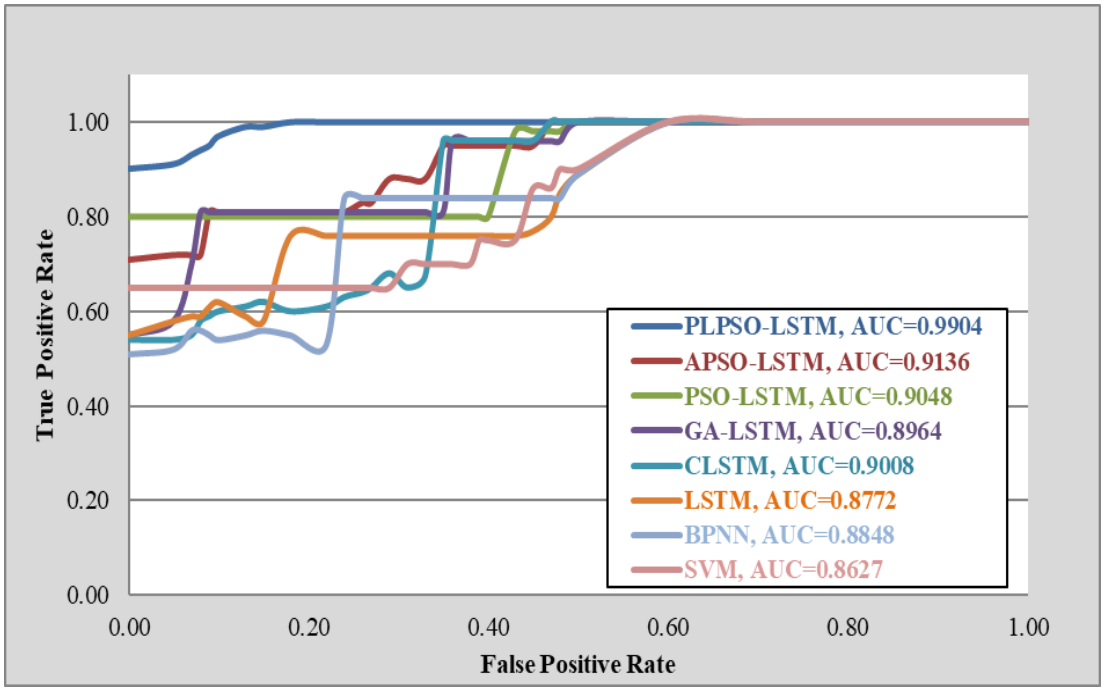

Figure 10. AUC and ROC analysis on the DTD Dataset

For the CDC and MDD datasets, the proposed approach has obtained the maximal AUC of 0.99 which is higher than the existing approaches. The maximal TPR of 1 is obtained with a very low FPR of 0.10 which shows the efficiency of the proposed approach even at the different decision thresholds; whereas other models reached it around the FPR of 0.50 to 0.60. It evidences that the proposed classifier can recognize the patterns with higher precision. For the DTD dataset, the proposed model reached the maximal TPR at 0.18, while the existing approaches attained it in the zone of 0.50 to 0.60. On the whole, it evidences the rigidity of the proposed prediction model on different datasets.

When it comes to the application and efficacy of PLPSO-LSTM models for diabetes prediction, computational time is crucial. It has an impact on user experience, resource management, model evaluation, real-time prediction capabilities, model training, and overall system efficiency.

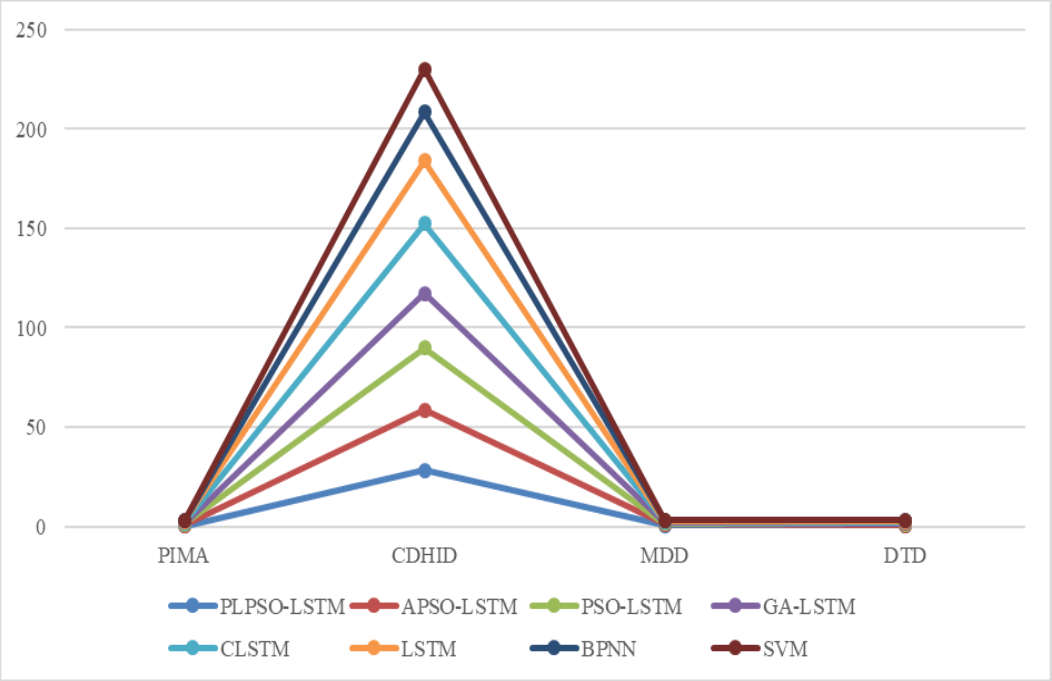

Figure 11. Performance comparison based on training computational

The practical application of LSTM models is improved when they are computationally efficient, increasing the likelihood of their broad usage in diabetes prediction and management systems. Figure 11 shows the computation time for each compared method which shows the SVM taking low computation time. However, the proposed PLPSO-LSTM achieved a second low computation time and also a very low computational time when compared with variants of LSTM methods.

Therefore, the efficiency of the proposed approach must be significant. It is justified by experimenting with four diabetes datasets and various analyses are carried out to showcase the generalization ability, rigidity, reliability, faster global convergence, and high accuracy level at minimal computational time. The convergence analysis showcased the efficiency and robustness of the model. The proposed PLPSO-LSTM has yielded the maximal results at minimal iterations without stuck in the local optimal traps.

The LSTM model training process can converge more quickly and efficiently when the precision of a local search algorithm is combined with PSO's global search capability. While the local search method offers fine-tuning to help prevent suboptimal solutions, PSO rapidly finds general regions that enhance performance. By cutting down on tuning time, this hybrid technique enables the LSTM model to reach a high-performance threshold more quickly. By eliminating the necessity for thorough searches throughout the solution space, the hybrid PSO-local search method uses less computing power for hyperparameter optimization. For sophisticated models like LSTM, which take a lot of time and money to train, this efficiency is especially beneficial. This proved the stability of the model over the local traps and took minimal time for solution search. The LSTM enhanced with PLPSO has shown promising results in accomplishing the objective. It demonstrated superior performance in comparison to the existing methodologies and the conventional approaches. Because the results of diabetes classification need to be precise and free of ambiguity, the PLPSO-LSTM that has been suggested is highly recommended for applications that are related to real-time diabetes diagnosis as well as other clinical prediction applications.

8. Conclusion

Diabetes is a frequently encountered chronic ailment characterized by high amounts of glucose in the bloodstream. Precisely identifying diabetes can greatly enhance an individual's health, otherwise results in unfavorable health consequences. The current research has proposed a new diabetes diagnosis model using PLPSO-LSTM. The hyperparameters of LSTM are optimized using a POBL and LSA-based PSO algorithm to enhance detection rate, and convergence and reduce the computation time. The two enhancements incorporated into the basic PSO are utilized to increase population diversity and prevent local optima from being trapped. To increase the diversity of the PSO population, the PLPSO makes use of the advantages of the POBL method. Moreover, the new LSA algorithm was implemented within PLPSO to prevent it from becoming trapped in local optima. The experimental results show that the developed model is a robust and unbiased diabetes prediction model and yields a reliable,

unambiguous diagnostic result when compared with existing approaches. In the future, the model can be experimented with larger datasets and can tune other hyper-parameters to understand newer datasets.

References

[1] W.H.O. WHO. (2023) Diabetes. WHO, India. https://www.who.int/india/health-topics/mobile-technology-for-preventing-ncds#:~:text=In%20India%2C%20there%20are%20estimated,developing%20diabetes%20in%20near%20future

[2] K.M. West, M.M.S. Ahuja, P.H. Bennett, A. Czyzyk, O.M. De Acosta, J.H. Fuller, B. Grab, V. Grabauskas,R.J. Jarrett, K. Kosaka, H. Keen, A.S. Krolewski, E. Miki, V. Schliack, A. Teuscher, P.J. Watkins, J.A. Stober, The role of circulating glucose and triglyceride concentrations and their interactions with other “risk factors” as determinants of arterial disease in nine diabetic population samples from the WHO multinational study. Diabetes care, 6(4), (1983) 361-369. https://doi.org/10.2337/diacare.6.4.361

[3] N. Sneha, T. Gangil, Analysis of diabetes mellitus for early prediction using optimal features selection. Journal of Big data, 6(1), (2019) 1-19. https://doi.org/10.1186/s40537-019-0175-6

[4] R. Kamalraj, S. Neelakandan, M. Ranjith Kumar, V. Chandra Shekhar Rao, R. Anand, H. Singh, Interpretable filter based convolutional neural network (IF-CNN) for glucose prediction and classification using PD-SS algorithm. Measurement, 183, (2021) 109804. https://doi.org/10.1016/j.measurement.2021.109804

[5] B.F. Wee, S. Sivakumar, K.H. Lim, W. Wong, F.H. Juwono, Diabetes detection based on machine learning and deep learning approaches. Multimedia Tools and Applications, 83(8), (2024) 24153-24185. https://doi.org/10.1007/s11042-023-16407-5

[6] R. Annamalai, R. Nedunchelian, Design of optimal bidirectional long short term memory based predictive analysis and severity estimation model for diabetes mellitus. International Journal of Information Technology, 15(1), (2023) 447-455. https://doi.org/10.1007/s41870-022-00933-w

[7] A. ElSaid, F. El Jamiy, J. Higgins, B. Wild, T. Desell, Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration. Applied Soft Computing, 73, (2018) 969-991. https://doi.org/10.1016/j.asoc.2018.09.013

[8] H. Chung, K.S. Shin, Genetic algorithm-optimized long short-term memory network for stock market prediction. Sustainability, 10(10), (2018) 3765. https://doi.org/10.3390/su10103765

[9] P. Pirozmand, H. Jalalinejad, A. A. R. Hosseinabadi, S. Mirkamali, Y. Li, An improved particle swarm optimization algorithm for task scheduling in cloud computing. Journal of Ambient Intelligence and Humanized Computing, 14(4), (2023) 4313-4327. https://doi.org/10.1007/s12652-023-04541-9

[10] H.R. Tizhoosh, (2005) Opposition-based learning: a new scheme for machine intelligence. International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), IEEE, Austria. https://doi.org/10.1109/CIMCA.2005.1631345

[11] W.H. Bangyal, K. Nisar, T.R. Soomro, A.A. Ag Ibrahim, G.A. Mallah, N.U. Hassan, N.U. Rehman, An improved particle swarm optimization algorithm for data classification. Applied Sciences, 13(1), (2022) 283. https://doi.org/10.3390/app13010283

[12] J. Xu, S. Xu, L. Zhang, C. Zhou, Z. Han, A particle swarm optimization algorithm based on diversity-driven fusion of opposing phase selection strategies. Complex & Intelligent Systems, 9(6), (2023) 6611-6643. https://doi.org/10.1007/s40747-023-01069-5

[13] J.P. Kandhasamy, S. Balamurali, Performance analysis of classifier models to predict diabetes mellitus. Procedia Computer Science, 47, (2015) 45-51. https://doi.org/10.1016/j.procs.2015.03.182

[14] O. Erkaymaz, M. Ozer, Impact of small-world network topology on the conventional artificial neural network for the diagnosis of diabetes. Chaos, Solitons & Fractals, 83, (2016) 178-185. https://doi.org/10.1016/j.chaos.2015.11.029

[15] S. Yu, Z. Cao, X. Jiang, (2017) Robust linear discriminant analysis with a Laplacian assumption on projection distribution. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, USA. https://doi.org/10.1109/ICASSP.2017.7952620

[16] W. Chen, S. Chen, H. Zhang, T. Wu, (2017) A hybrid prediction model for type 2 diabetes using K-means and decision tree. IEEE international conference on software engineering and service science (ICSESS), IEEE, China. https://doi.org/10.1109/ICSESS.2017.8342938

[17] S. M. H. Dadgar, M. Kaardaan, A hybrid method of feature selection and neural network with genetic algorithm to predict diabetes. International Journal of Mechatronics, Electrical and Computer Technology (IJMEC), 7, (2017) 3397-3404.

[18] E. Hashi, S.U. Zaman, R. Hasan, Developing diabetes disease classification model using sequential forward selection algorithm.

International Journal of Computer Applications, 180(5), (2017)1-6. https://doi.org/10.5120/ijca2017916018

[19] Y. Zhang, Z. Lin, Y. Kang, R. Ning, Y. Meng, A feed-forward neural network model for the accurate prediction of diabetes mellitus. International Journal of Scientific and Technology Research, 7(8), (2018) 151-155.

[20] R. Haritha, D. S. Babu, P. Sammulal, A hybrid approach for prediction of type-1 and type-2 diabetes using firefly and cuckoo search algorithms. International Journal of Applied Engineering Research, 13(2), (2018) 896-907.

[21] S. Srivastava, L. Sharma, V. Sharma, A. Kumar, H. Darbari, Prediction of diabetes using artificial neural network approach. Engineering Vibration, Communication and Information Processing: ICoEVCI 2018, Springer, 679-687. https://doi.org/10.1007/978-981-13-1642-5_59

[22] K. Kannadasan, D. R. Edla, V. Kuppili, Type 2 diabetes data classification using stacked autoencoders in deep neural networks. Clinical Epidemiology and Global Health, 7(4), (2019) 530-535. https://doi.org/10.1016/j.cegh.2018.12.004

[23] N.S. Prema, V. Varshith, J. Yogeswar, Prediction of diabetes using ensemble techniques. International Journal of Recent Technology and Engineering, 7(6), (2019) 203-205.

[24] A. Dinh, S. Miertschin, A. Young, and S. D. Mohanty, A data-driven approach to predicting diabetes and cardiovascular disease with machine learning. BMC medical informatics and decision making, 19(1), (2019) 1-15. https://doi.org/10.1186/s12911-019-0918-5

[25] N. Pradhan, G. Rani, V. S. Dhaka, R. C. Poonia, Diabetes prediction using artificial neural network. Deep Learning Techniques for Biomedical and Health Informatics, (2020) 327-339. https://doi.org/10.1016/B978-0-12-819061-6.00014-8

[26] K. Lakhwani, S. Bhargava, K. K. Hiran, M. M. Bundele, D. Somwanshi, (2020) Prediction of the onset of diabetes using artificial neural network and Pima Indians diabetes dataset. 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), IEEE, India. https://doi.org/10.1109/ICRAIE51050.2020.9358308

[27] M. Abedini, A. Bijari, T. Banirostam, Classification of Pima Indian diabetes dataset using ensemble of decision tree, logistic regression and neural network. International Journal of Advanced Research in Computer and Communication Engineering, 9(7), (2020) 7-10. https://doi.org/10.17148/IJARCCE.2020.9701

[28] H. Naz, S. Ahuja, Deep learning approach for diabetes prediction using PIMA Indian dataset. Journal of Diabetes & Metabolic Disorders, 19, (2020) 391-403. https://doi.org/10.1007/s40200-020-00520-5

[29] N. Pradhan, V.S. Dhaka, S.C. Kulhari, Experimental and comparison based study on diabetes prediction using artificial neural network. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 13(6), (2020) 1173-1179. https://doi.org/10.2174/2213275912666190801112119

[30] E. Güldoğan, T. Zeynep, A. Ayça, C. Colak, Performance evaluation of different artificial neural network models in the classification of type 2 diabetes mellitus. The Journal of Cognitive Systems, 5(1), (2020) 23-32.

[31] M.M. Bukhari, B.F. Alkhamees, S. Hussain, A. Gumaei, A. Assiri, S.S. Ullah, An improved artificial neural network model for effective diabetes prediction. Complexity, 2021(1), (2021) 5525271. https://doi.org/10.1155/2021/5525271

[32] M.W. Nadeem, H.G. Goh, V. Ponnusamy, I. Andonovic, M.A. Khan, M. Hussain, A fusion-based machine learning approach for the prediction of the onset of diabetes. In Healthcare, 9(10), (2021) 1393. https://doi.org/10.3390/healthcare9101393

[33] P.B.K. Chowdary, R.U. Kumar, An effective approach for detecting diabetes using deep learning techniques based on convolutional LSTM networks. International Journal of Advanced Computer Science and Applications, 12(4), (2021) 519-525. https://dx.doi.org/10.14569/IJACSA.2021.0120466

[34] S. Sivasankari, J. Surendiran, N. Yuvaraj, M. Ramkumar, C. Ravi, R. Vidhya, (2022) Classification of diabetes using multilayer perceptron. IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), IEEE, India. https://doi.org/10.1109/ICDCECE53908.2022.9793085

[35] P. Pujari, Classification of Pima Indian diabetes dataset using support vector machine with polynomial kernel. Deep Learning, Machine Learning and IoT in Biomedical and Health Informatics: CRC Press, (2022) 55-67. https://doi.org/10.1201/9780367548445-5

[36] V. Chang, J. Bailey, Q. A. Xu, Z. Sun, Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms. Neural Computing and Applications, 35(22), (2023) 16157-16173. https://doi.org/10.1007/s00521-022-07049-z

[37] A. Mousa, W. Mustafa, R.B. Marqas, S.H. Mohammed, A comparative study of diabetes detection using the Pima Indian diabetes

database. Journal of Duhok University, 26(2), (2023) 277-288. https://doi.org/10.26682/sjuod.2023.26.2.24

[38] H. Shao, X. Liu, D. Zong, Q. Song, Optimization of diabetes prediction methods based on combinatorial balancing algorithm. Nutrition & Diabetes, 14(1), (2024) 63. https://doi.org/10.1038/s41387-024-00324-z

[39] M. Rahman, D. Islam, R.J. Mukti, I. Saha, A deep learning approach based on convolutional LSTM for detecting diabetes. Computational Biology and Chemistry, 88, (2020) 107329. https://doi.org/10.1016/j.compbiolchem.2020.107329

[40] S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural computation, MIT Press, 9(8), (1997) 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[41] Z. Hu, Y. Bao, T. Xiong, (2014) Partial opposition-based adaptive differential evolution algorithms: Evaluation on the CEC 2014 benchmark set for real-parameter optimization. IEEE Congress on Evolutionary Computation (CEC), IEEE, China. https://doi.org/10.1109/CEC.2014.6900489

[42] M. Tubishat, N. Idris, L. Shuib, M.A. Abushariah, S. Mirjalili, Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. Expert Systems with Applications, 145, (2020) 113122. https://doi.org/10.1016/j.eswa.2019.113122

[43] S. Joshi, M. Borse, (2016) Detection and prediction of diabetes mellitus using back-propagation neural network. in 2016 International conference on micro-electronics and telecommunication engineering (ICMETE), IEEE, India. https://doi.org/10.1109/ICMETE.2016.11

[44] M.K. Suddle, M. Bashir, Metaheuristics based long short term memory optimization for sentiment analysis. Applied Soft Computing, 131, (2022) 109794. https://doi.org/10.1016/j.asoc.2022.109794

[45] S. Srivastava, (2021) Genetic Algorithm Optimized Deep Learning Model for Parkinson Disease Severity Detection. Dublin, National College of Ireland.

[46] Y. Zhang, S. Yang, (2019) Prediction on the highest price of the stock based on PSO-LSTM neural network. 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), IEEE, China. https://doi.org/10.1109/EITCE47263.2019.9094982

[47] H. Zhou, R. Myrzashova, R. Zheng, Diabetes prediction model based on an enhanced deep neural network. EURASIP Journal on Wireless Communications and Networking, 2020, (2020) 1-13. https://doi.org/10.1186/s13638-020-01765-7

Authors Contribution Statement

Both the authors equally contributed to the Conceptualization, Methodology, Investigation, Validation, Formal analysis, Data Curation, Writing - Original Draft and Writing - Review & Editing. The final manuscript has been read and approved by all authors.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The used datasets are available as follows

- PIMA: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
- CDHID: http://archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators
- MDD: https://data.mendeley.com/datasets/wj9rwkp9c2/1
- DTD: Https://data.world/eam2bi/idoctor/workspace/file?filename=Diabetestype.csv

Has this article screened for similarity?

Yes

About the License