Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 8, No.1, January 2026, pp. 02601024-01 ~ 02601024-010
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v8i1.2283

Optimized Deep Learning Framework for Clinical Data Classification Using Firefly-Enhanced Stacked Sparse Autoencoders

Yousif Samer Mudhafar[1,2]*, Ramy Riad Al-Fatlawy[2], Ali Ahmed Al-Fatlawy[3], Aboothar Mahmood Shakir[4]

[1]Department of Computer Science, Faculty of Education, University of Kufa, Najaf, Iraq

[2]Department of Computer Techniques Engineering, Faculty of Technical Engineering, The Islamic University, Najaf, Iraq

[3]Department of Computer Engineering, Faculty of Engineering, Islamic Azad University, Isfahan, Iran

[4]Computer Techniques Engineering Department, Faculty of Information Technology, Imam Ja'afar Al-Sadiq University, Baghdad 54001, Iraq

*yousifs.mudhafar@uokufa.edu.iq

Abstract. Diabetes is a chronic metabolic disorder characterized by sustained high blood sugar levels, which frequently cause complications, including neuropathy and cardiovascular disease. Due to the complex and nonlinear nature of clinical data, accurate and timely prediction is challenging. Traditional approaches struggle to generalize or extract rich features from low-resolution datasets. In this paper, a hybrid deep learning model (FA-SSAE: Firefly Algorithm-based Stacked Sparse Autoencoder) is proposed to improve diabetes classification using the Pima Indians Diabetes dataset. Data is synthesized using Variational Autoencoder (VAE) developed data augmentation and deep features are extracted using SSAE. The model achieved 91.67% accuracy, 96.38% precision, and 98.75% recall; results that significantly outperformed several state-of-the-art methods. The results demonstrate the robustness and reliability of the proposed approach. Its lightweight architecture can be deployed in resource-limited environments, providing value for mobile or embedded systems used in remote clinics. This research advances the development of scalable and accessible tools for diagnostic detection of diabetes in the earliest possible stages to aid in unsupervised clinical care.

Keywords: Diabetes prediction, healthcare informatics, metaheuristic optimization, biomedical classification, diabetes disease, firefly algorithm, stacked sparse autoencoder.

(Received 2025-06-30, Revised 2025-10-21, Accepted 2025-10-22, Available Online by 2026-01-30)

1. Introduction

Data mining exposes hidden patterns in data that are often missed by traditional techniques[1][2]. While general uses in marketing and finance are well-known, developments in applying data mining to healthcare—specifically disease prediction—have quickly gained momentum. With a plethora of medical records at their disposal, data mining can assist in clinical administration in early diagnoses. For diabetes especially Types 1 and 2, predictive modeling aids in early detection and better management reducing late-stage diagnostic complications[3].

With prolonged blood sugar elevation and impaired metabolism of carbohydrates, fats, and proteins mainly due to insulin dysfunction, diabetes is considered a chronic and metabolic disorder[4]. Type 2 Diabetes Mellitus (T2DM) accounts for 90–95% of all cases in the world and is on the rise. Although the exact cause is still unknown, genetic and lifestyle factors are highly implicated. Even though it is not curable, T2DM can be treated with medications in combination with lifestyle changes, thus, an early diagnosis is very important to avoid major complications such as cardiac issues and neuropathy [5].

Technically, this situation should have been avoided by the adoption of automated diagnostics. However, in many of the healthcare systems in developing countries, they rely still on commonsensical yet error-prone methods of diagnosing which disregard any hidden patterns in the given data [6]. The growing burden of diabetes on a global scale has called for instruments that are useable at scale and made accessible. Thus, AI models deployed on the cheaper Raspberry Pi-type platforms can enhance early detection in place-starved regions[7]. Most machine learning and deep learning techniques work especially well for finding hidden, complex patterns in clinical data [8][9]. In the study here, we present an enhanced form of the Stacked Sparse Autoencoder (SSAE) architecture enhanced by the Firefly Algorithm (FA) to maximize the accuracy and capability for generalization of diabetes prediction models.

If diabetes is not managed well, it can result in complications such as cardiovascular diseases, neuropathy, nephropathy, and strokes. Traditional methods for diagnosing patients by means of empirical and laboratory tests are not only time-consuming but require a large amount of resources. Despite several developments of predictive models, many of these models suffer from a variety of limitations, including working on small datasets or ineffective feature selection. To address these problems, researchers have undertaken studies based on big data analytics and machine learning (ML) applications for more accurate early detection. The present study aims at creating a novel, improved feature extraction model using a Firefly Algorithm (FA) optimized Stacked Sparse Autoencoder (SSAE), thereby improving prediction accuracy levels by enhancing the quality and depth of the features learned.

This research is motivated by three main questions which together, strive to advance intelligent medical diagnosis for diabetes. First, whether the classification performance of a Firefly Algorithm-based Stacked Sparse Autoencoder (FA-SSAE) is superior to traditional machine learning and existing deep learning methods when trained on clinical data. Second, how much an augmenting data process utilizing a Variational Autoencoder (VAE) improves the robustness and generalizability of the model when working with limited or imbalanced medical datasets. Finally, the research explores the practicality of deploying the proposed FA-SSAE framework in real-world healthcare settings, especially within resource-constrained or embedded environments, to determine whether the model maintains its efficiency and reliability outside of experimental conditions.

1.1. Related Work

Recent progress in deep learning and machine learning has greatly improved diabetes prediction and classification. Several studies have explored using autoencoder-based architectures for modelling complex patterns in clinical data. For instance, Abdulaimma et al. [10] applied a deep stacked autoencoder to high-dimensional GWAS data for classifying Type 2 Diabetes and identifying epistatic interactions, although their model required further hyperparameter optimization. Katsuki et al. [11] create a stacked convolutional autoencoder to extract both local and global temporal features from the EHR lab sequences, specifically for diabetic nephropathy. To address the lack of data, Faruqui et al.

[12] proposed a person-specific glucose prediction model based on LSTM networks supplemented with transfer learning and Bayesian optimization. For instance, traditional models like that of [13], which applied stacked autoencoders combined with SoftMax classification on the Pima Indians Diabetes dataset, made moderate performances (i.e., achieved 86.26% accuracies). In the recent work of García-Ordás et al. [14], sparse and variational autoencoders were combined with CNN and obtained improved classification accuracy (92.31%), consistent with recent deep learning–based diabetes prediction studies [15]. According to Zhou et al. [16], the progressive stacked framework using SSAE has been proven for multi-class classification with respect to diabetes and medical conditions, attaining 92.94% accuracy. Deep learning applied to heart rate variability (HRV) signals, as per what was shown in [17], has attained 95.7% accuracy by combining CNN, LSTM, and SVM. Also, there are some emerging trends towards the ensemble models, Singh and Kumar [18] applied an NSGA-II-based stacking framework, optimizing multiple base learners for better predictive results. Additionally, Srivastava et al. [19] proposed a hybrid pipeline through data imputation with K-Means++, outlier detection using Artificial Bee Colony (ABC), and classification through LS-SVM. Most of these models enhance accuracy, however, limited scalability, interpretability, or inefficient hyperparameter tuning-several remain with these challenges. To fill these voids, our work presents an FA-optimized SSAE framework, which harnesses the potential of deep feature extraction, coupled with swarm intelligence, to enhance classification accuracy while keeping manual configuration at imperceptible levels.

2. Methods

This study presents a Stacked Sparse Autoencoder (SSAE) architecture, where each sparse autoencoder receives input from the hidden representation of the preceding layer, enforcing sparsity constraints to learn compact and discriminative features [20]. Following the unsupervised pretraining phase, the decoder components are discarded, preserving only the feature representations learned within the hidden layers. To the last hidden layer, a SoftMax classifier is attached to carry out supervised classification. This is what forms the complete SSAE model-train sparse autoencoders with a SoftMax output layer. In the fine-tuning step, the whole network is considered a model, and all parameters are optimized jointly through backpropagation using the labelled training dataset. Let $\{y_1, y_2, \dots, y_m\}$Note the set of target labels. The network's cost function is defined as:

$$E = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{N} 1\,\{y^{(i)} = j\}\,log\left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{N} e^{\theta_l^T x^{(i)}}}\right) \tag{1}$$

Where 1 $\{y^{(i)} = j\}$ represents the indicator function, i.e., 1 $\{y^{(i)} = j\}$ = 1 if y = 1, and 1 $\{y^{(i)} = j\}$ = 0 if y ≠ j, N denotes the number of classes, and θi denotes the weight matrix linking the ith output unit.

The Firefly Algorithm (FA) is used to enhance the training procedure by optimizing the weights and bias parameters of SSAE. Choosing suitable values of bias and weights is paramount in the development of rigid neural networks. Conventionally, this optimization depends on manual hyperparameter tuning or grid search, which are mostly error-prone, cumbersome, and computationally expensive. The FA presents an efficient alternative, whereby an optimal configuration is searched through the hyperparameter space. A flowchart illustrating the complete methodology for constructing FA-optimized SSAE is presented in Figure 1.

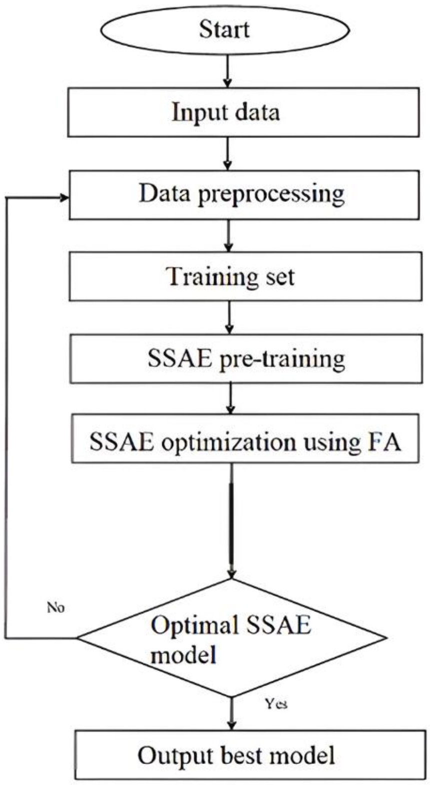

Figure 1. Flowchart of proposed method

2.1. Firefly algorithm

The Firefly algorithm (FA) is a swarm-based optimization algorithm that is inspired by the light attraction mechanism displayed by fireflies, originally proposed by Xin-She Yang [21] in 2008. Brighter fireflies attract others, thereby guiding the search toward optimal solutions. Fireflies will then move randomly if there is no brighter one nearby. The balance between exploration and exploitation is maintained by three major parameters: attractiveness, randomness, and light absorption. A flowchart that shows how the technique of the firefly algorithm works is shown in the following figure. Although the firefly algorithm is strong enough to address global optimization, it was found to be really ineffective in terms of performance in high-dimensional problems because of its premature convergence and accuracy deterioration. Therefore, this work improves on the Firefly Algorithm by utilizing self-adaptive logarithmic inertia weights and dynamically changing step sizes. Today, Lévy flight is efficient at improving exploration, but this is not enough in many cases. Following the work done by the researchers [22], the suggested enhancements augment convergence stability and accuracy in high-dimensional spaces. this study incorporates the step adjustment mechanism influenced by the following exponential decay formula:

$$c = \theta^{D} \cdot T \cdot e^{(-t/T)} \tag{2}$$

where T denotes the maximum number of iterations, t is the current iteration, D represents the dimensionality of the firefly (i.e., the number of decision variables), and $\theta \in [0,1]$ is a scaling factor. For this study, θ has been set empirically to 0.1. This formulation ensures that for random step sizes produced by various fireflies, it decreases progressively with an increase in the dimension of the search space or the number of iterations. Such regulations ensure a more stable and localized exploration in high-dimensional environments, which is more towards convergence by avoiding increased noise because of non-productive exploration.

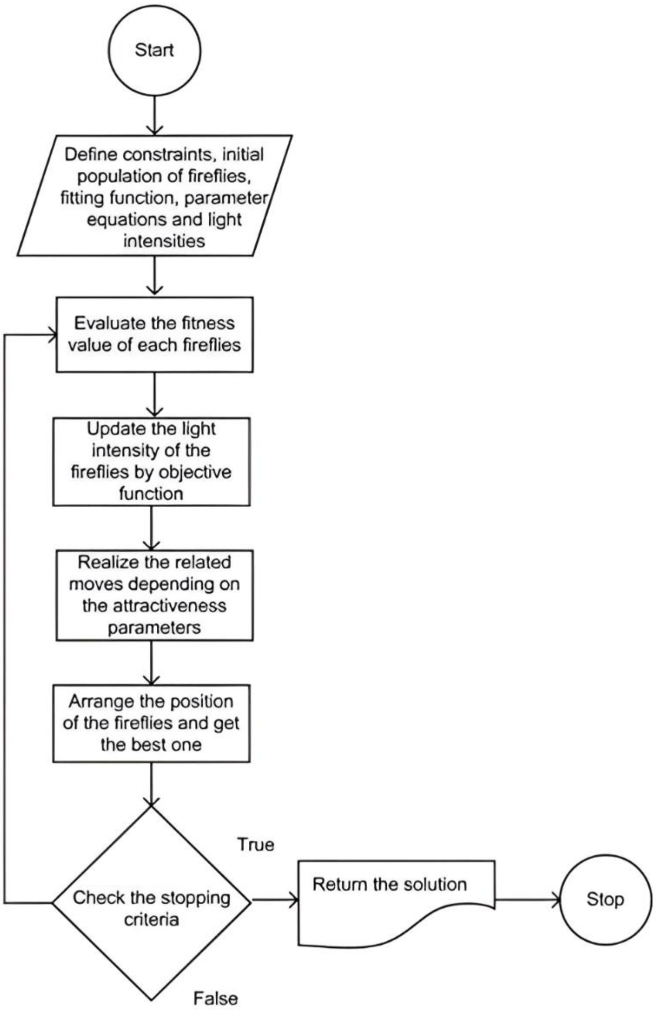

Figure 2. Flowchart of the Firefly Algorithm Technique

In a training scheme for SSAE, the Firefly Algorithm is used that efficiently optimize the initial weights and biases. The setting of the firefly algorithm, including a self-adaptive logarithmic inertia weight and dynamically adjusted step-size, enhances the balance of exploration versus exploitation, which helps to avoid premature convergence to local optima in high-dimensional optimization problems, thereby achieving better convergence stability and classification accuracy.

To lend reliability and reproducibility to this study, all models were developed and evaluated in a controlled environment. To that end, MATLAB R2023b was used for computations on a 64-bit Windows 10 machine. To ensure stable and reproducible results across experiments, a fixed random seed (seed = 42) was applied to the data partitioning, neural network weights initialization process, the VAE sampling, and any operations on the Firefly Algorithm.

The Variational Autoencoder (VAE) was deployed for data augmentation as described in Step 1 of the methods section, developed according to the following architecture [23]:

- **Encoder:** an input layer containing 8 neurons (for the attributes of the dataset) connected to a hidden layer of 6 neurons using a ReLU activation function, yielding output in a 2D latent space.
- **Decoder:** The decoder maps the 2D latent vector back symmetrically through a hidden layer of 6 neurons using a ReLU activation function and generates the 8 features input by means of a Sigmoid output layer.

The VAE was trained for 100 epochs using the Adam optimizer (learning rate = 0.001, batch size = 32). The loss function combined the rate of reconstruction error based on binary cross-entropy and the Kullback-Leibler (KL) divergence to establish strong generative ability. This implementation, coupled with the parameters in Table 2 for the Firefly Algorithm, allows for full reproducibility of the proposed FA-SSAE framework.

The development of the FA-optimized Stacked Sparse Autoencoder (SSAE) followed these steps:

- Step 1: Preprocess the data and augment features through a Variational Autoencoder (VAE) to achieve compressed, abstract representations of the data.
- Step 1: Create a three-layer Stacked Sparse Autoencoder (SSAE) and feed the output from every layer to the next layer.
- Step 3: Add a Softmax to the output of the last layer to form a classifier and train the SSAE using backpropagation for classification.
- Step 4: Change the weights and biases of the SSAE to the Firefly Algorithm and modify surcharge parameters (swarm size and step size) to reach convergence faster and yield a more accurate representation of the features.

3. Results and Discussion

The proposed model was evaluated using the commonly used Pima Indians Diabetes data set taken from the UCI Machine Learning Repository [19]. There are a total of 768 records in the data set and each record is represented by eight numerical features and a binary label indicating whether the patient has diabetes or not. The attributes of the dataset are summarized in Table 1. Of these samples, 268 (34.9 percent) have diabetes, and the remaining 500 (65.1 percent) do not.

The FA-SSAE parameters were fixed according to past researches [12, 17] to allow for fair and reliable evaluation. A maximum of 100 training epochs were fixed for the network of 16 layers. The experiments were conducted using MATLAB on a 64-bit Windows 10 machine. To ensure statistical rigor, each experiment was conducted 100 times on the Pima dataset. Main parameters of the Firefly Algorithm are light absorption coefficient (γ), attraction coefficient (β_0), mutation coefficient (α), and mutation damping ratio (α_damp). The generalization of the model was evaluated on the dataset used for testing. The hyperparameters of the FA-SSAE are summarized in Table 2.

Table 1. Attribute information of the Pima Indians Diabetes dataset

Attribute number	Attribute name	Minimum	Maximum
1	Number of times pregnant	0	17
2	Plasma glucose concentration a 2 h in an oral glucose tolerance test	0	199
3	Diastolic blood pressure	0	122
4	Triceps skin fold thickness	0	99
5	2-Hour serum insulin	0	846
6	Body mass index	0	67.1
7	Diabetes pedigree function	0.078	2.42
8	Age	21	81

Table 2. Initial values for hyperparameters in FA–SSAE

Parameters	Values
L2WeightRegularization	0.004
SparsityRegularization	0
SparsityProportion	0
ScaleData	false
Epoch number	100
Number of search agents	20
Max_iter	100
lb	2
ub	2
gamma	1
beta0 (β_0)	2
alpha (α)	0.2
alpha_damp (α_{damp})	0.98

One method to evaluate classification accuracy is by calculating the accuracy of the output rules as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (3)$$

In this context, a true positive (TP) means the model correctly identifies a diabetic patient, while a true negative (TN) correctly identifies a healthy individual. A false positive (FP) occurs when a healthy person is incorrectly classified as diabetic, and a false negative (FN) is when the model fails to detect diabetes in an affected patient. These four outcomes collectively determine the model's overall accuracy. The ability of a test to identify healthy items is a specificity parameter. Specificity is calculated as the ratio of true negatives to the sum of true negatives and false positives. The following formula can be used to represent this ratio:

$$Specificity = \frac{TN}{TN+FP} \qquad (4)$$

The sensitivity criterion is defined as follows:

$$Sensitivity = \frac{TP}{TP+FN} \qquad (5)$$

The precision parameter indicates the probability of the patient being predicted and the accuracy of that prediction. The higher this value, the more the rules have a high power in diagnosing diseases.

$$Precision = \frac{TP}{TP+FP} \qquad (6)$$

The F score criterion is based on the combination of accuracy and sensitivity criteria as below. It is defined as:

$$F - Score = \frac{2\times(Precision\times Recall)}{Precision+Recall} \qquad (7)$$

The hyperparameters of the FA-SSAE model were determined by thorough experimentation on the PIMA dataset in this part of the paper. This hyperparameter tuning process was followed by an evaluation of the model's predictive powers with a focus on essential classification metrics. In addition, further evaluation of the feasibility and efficacy of the proposed approach was carried out, and comparative results with related works were made [11, 15].

The next step was to evaluate the performance of the proposed FA-SSAE model on the PIMA dataset, which was divided into two parts: a training set to construct the classifier and a test set for evaluating its generalization ability. In its turn, the effectiveness of the model in binary classification was assessed through two-class experiments (diabetic vs. non-diabetic). Each run of our experimental evaluation maintained a one-hundred-epoch training of the model for uniformity between evaluations.

The performance of the proposed FA-SSAE model is compared with the models of [12] and [17] using accuracy, sensitivity, specificity, precision, and F-score, as shown in Table 3. The FA-SSAE model outperformed others in each metric with a total classification accuracy of 90.53% highlighting its great generalization capacity. From these results, it is confirmed that the proposed method very significantly surpasses baseline models on the Pima dataset, gauging its efficacy for diabetes prediction. Figure 3 shows a comparison of the proposed method's accuracy compared to that accuracy of the previous method mentioned in the previous papers.

Table 3. Comparison results of the proposed method and the other papers

	Accuracy	**Sensitivity**	**Specificity**	**Precision**	**F-Score**
Paper [12]	86.26	87.92	83.41	90.66	89.27
Paper [17]	83.8	96.1	79.9	-	88.5
Proposed Method	91.67	98.75	88.95	96.38	99.39

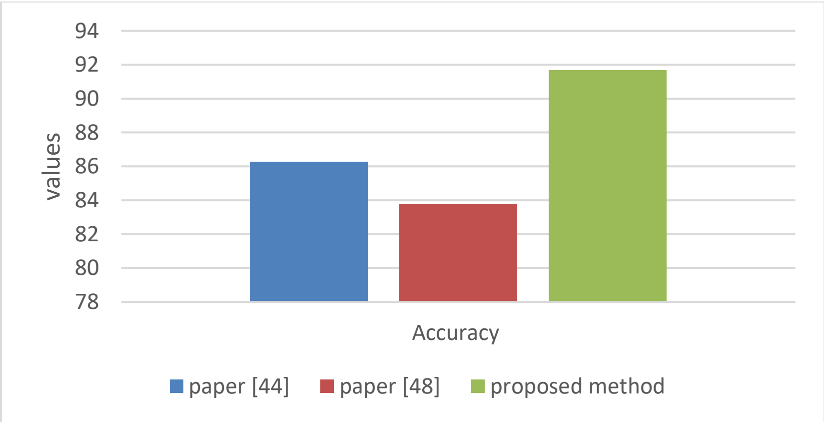

Figure 3. Comparison results of the proposed method and the other papers

4. Conclusion

The FA-SSAE framework proposed in this study has shown great promise towards diabetes classification by achieving an accuracy of 91.67% with high precision and recall, leading to good classification performance over traditional and state of the art methods on the Pima Indians Diabetes dataset. This demonstrates that the framework was effective in distinguishing patients diagnosed with diabetes. Moreover, the lightweight nature of the structural design suggests future deployment on energy-efficient FPGA-based platforms for embedded, resource-constrained healthcare systems such as the one in [25]. Further exploration is warranted at the sake of confirming generalizability and robust applicability in clinical practice, as the results are from a single data source, and further evaluation on more diverse clinical datasets will strengthen confidence in the performance of the proposed FA-SSAE framework in real-world healthcare contexts.

References

[1] T. Sharma and M. Shah, "A comprehensive review of machine learning techniques on diabetes detection," *Vis. Comput. Ind. Biomed. Art*, vol. 4, p. 30, 2021, doi: 10.1186/s42492-021-00097-7.

[2] A. S. Chauhan, M. S. Varre, K. Izuora, M. B. Trabia, and J. S. Dufek, "Prediction of Diabetes Mellitus Progression Using Supervised Machine Learning," *Sensors*, vol. 23, no. 10, p. 4658, 2023, doi: 10.3390/s23104658.

[3] K. Arunkumar, P. Manikandan, and S. Sundaram, "A Hybrid Machine Learning Model for Diabetes Prediction Using Feature Selection and Outlier Detection," *Comput. Biol. Med.*, vol. 143, p. 105263, 2022, doi: 10.1016/j.compbiomed.2022.105263.

[4] S. I. Ayon and M. Islam, "Diabetes prediction: A deep learning approach," *International Journal of Information Engineering & Electronic Business*, vol. 11, no. 2, 2019.

[5] A. Kumar, P. Sharma, and S. Gupta, "Machine learning methods for diabetes prediction: A systematic review and meta-analysis," *Comput. Biol. Med.*, vol. 145, p. 105430, 2022, doi: 10.1016/j.compbiomed.2022.105430.

[6] L. Yao, "Improved Models for Diabetes Prediction by Integrating PCA Technique," *Advances in Sustainable Science, Engineering and Technology*, vol. 1, no. 1, pp. 12–25, 2023, doi: 10.1234/aset.v1i1.2023.

[7] A. M. A. Al-muqarm and others, "Low-Cost Smart Learning with Moodle-Based Raspberry Pi 4 for University Students," in *2023 6th International Conference on Engineering Technology and its Applications (IICETA)*, Al-Najaf, Iraq, 2023, pp. 603–608. doi: 10.1109/IICETA57613.2023.10351266.

[8] H. Gupta, H. Varshney, T. K. Sharma, N. Pachauri, and O. P. Verma, "Comparative performance analysis of quantum machine learning with deep learning for diabetes prediction," *Complex & Intelligent Systems*, pp. 1–15, 2021.

[9] S. S. Islam, R. Ahmed, and Md. H. Rahman, "Deep Learning Applications in Healthcare: A Comprehensive Survey," *Artif. Intell. Med.*, vol. 137, p. 102472, 2023, doi: 10.1016/j.artmed.2023.102472.

[10] L. Xie, Y. Zhang, and H. Wang, "Machine learning and deep learning approaches for predicting diabetes progression: A comparative analysis," *Electronics (Basel).*, vol. 14, no. 13, p. 2583, 2023, doi: 10.3390/electronics14132583.

[11] T. Katsuki and others, "Feature extraction from electronic health records of diabetic nephropathy patients with convolutional autoencoder," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] S. H. A. Faruqui and others, "Development of a deep learning model for dynamic forecasting of blood glucose level for type 2 diabetes mellitus: secondary analysis of a randomized controlled trial," *JMIR Mhealth Uhealth*, vol. 7, no. 11, p. e14452, 2019.

[13] K. Kannadasan, D. R. Edla, and V. Kuppili, "Type 2 diabetes data classification using stacked autoencoders in deep neural networks," *Clin. Epidemiol. Glob. Health*, vol. 7, no. 4, pp. 530–535, 2019.

[14] M. T. García-Ordás, C. Benavides, J. A. Benítez-Andrades, H. Alaiz-Moretón, and I. García-Rodríguez, "Diabetes detection using deep learning techniques with oversampling and feature augmentation," *Comput. Methods Programs Biomed.*, vol. 202, p. 105968, 2021.

[15] Md. A. Rahman, Md. S. Hossain, and K. Andersson, "A Deep Learning-Based Framework for Diabetes Prediction Using Clinical Data," *Sensors*, vol. 23, no. 7, p. 3521, 2023, doi: 10.3390/s23073521.

[16] J. Zhou, Q. Zhang, and B. Zhang, "A Progressive Stack Face-based Network for Detecting Diabetes Mellitus and Breast Cancer," in *Proc. 2020 IEEE International Joint Conference on Biometrics (IJCB)*, 2020, pp. 1–9.

[17] G. Swapna, R. Vinayakumar, and K. P. Soman, "Diabetes detection using deep learning algorithms," *ICT Express*, vol. 4, no. 4, pp. 243–246, 2018.

[18] N. Singh and P. Singh, “Stacking-based multi-objective evolutionary ensemble framework for prediction of diabetes mellitus,” *Biocybern. Biomed. Eng.*, vol. 40, no. 1, pp. 1–22, 2020.
[19] A. K. Srivastava, Y. Kumar, and P. K. Singh, “Hybrid diabetes disease prediction framework based on data imputation and outlier detection techniques,” *Expert Syst.*, p. e12785, 2021.
[20] J. Li, Z. Wang, and X. Chen, “Sparse Autoencoder-Based Feature Learning for Medical Diagnosis,” *Comput. Biol. Med.*, vol. 152, p. 106325, 2023, doi: 10.1016/j.compbiomed.2022.106325.
[21] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, “Recent Advances in Metaheuristic Algorithms: Applications and Foundations,” *IEEE Access*, vol. 10, pp. 11932–11957, 2022, doi: 10.1109/ACCESS.2022.3142216.
[22] Y. Li, Y. Zhao, Y. Shang, and J. Liu, “An improved firefly algorithm with dynamic self-adaptive adjustment,” *PLoS One*, vol. 16, no. 10, p. e0255951, 2021.
[23] H. Xu, Y. Zhang, and J. Liu, “Variational Autoencoder-Based Data Augmentation for Medical Data Classification,” *IEEE Access*, vol. 11, pp. 35641–35653, 2023, doi: 10.1109/ACCESS.2023.3264218.
[24] C. L. Blake and C. J. Merz, “UCI Repository of Machine Learning Databases,” 1998.
[25] S. H. Abdulnabi, Y. S. Mudhafar, A. A. Kadhim, M. B. Mahdi, and H. H. Sojar, “Neural Network-based System Identification: A Comprehensive FPGA Design and Implementation,” in *2024 IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, Bandung, Indonesia, 2024, pp. 1–7. doi: 10.1109/AIMS61812.2024.10512531.