DOI: 10.37943/24JNSS7017

Indira Uvaliyeva
PhD, Professor of the School of Digital Technology and Intelligent Systems
indirauvalieva@gmail.com, orcid.org/0000-0002-2117-5390
D. Serikbayev East Kazakhstan Technical University, Kazakhstan

Bayan Assanova
Candidate of Medical Sciences, Founder and Director of the company
info@aksel.kz, orcid.org/0009-0003-4508-5634
LLC «A-Medical», Kazakhstan

Zarina Khassenova
PhD, Dean of the School of Digital Technology and Intelligent Systems
zthasenova@mail.ru, orcid.org/0009-0001-1417-7793
D. Serikbayev East Kazakhstan Technical University, Kazakhstan

Roza Mukasheva
Candidate of Technical Sciences, Associate Professor in the Department of Higher Mathematics
mukashevaru@edu.ektu.kz, orcid.org/0000-0002-4722-4576
D. Serikbayev East Kazakhstan Technical University, Kazakhstan

Bekzat Karimkyzy
Master of Management, Project manager
bekzatastana@gmail.com, orcid.org/0009-0004-4696-7927
LLC «A-Medical», Kazakhstan

Kristina Karassenko
Bachelor of Science in Chemistry, Production Manager
karasenko.kristina@gmail.com, orcid.org/0009-0001-3673-2874
LLC «A-Medical», Kazakhstan

PREDICTING DIABETES PROGRESSION USING AN ENSEMBLE OF CNN, RNN, AND LSTM MODELS

Abstract: This article presents an integrated approach to predicting diabetes progression based on an ensemble of multiple deep neural network architectures. To enhance diagnostic accuracy and reliability, convolutional neural networks (CNN), recurrent neural networks (RNN), and long short-term memory (LSTM) models are jointly utilized within a clinical decision support framework. The optimal combination of their predictions is achieved through the Dirichlet ensemble method, which adaptively distributes weights among individual models according to their validation performance. Hyperparameter optimization using the Grid Search algorithm allows systematic selection of training parameters, network depth, activation functions, and regularization techniques, ensuring better convergence and reducing overfitting risks. The study involves a comprehensive data preprocessing pipeline, including normalization, balancing, and One-Hot Encoding of categorical features, to manage heterogeneous medical datasets and minimize the effect of missing or noisy information. Experimental evaluation demonstrates that the proposed ensemble model significantly outperforms individual CNN, RNN, and LSTM architectures in terms of accuracy, sensitivity, and stability, achieving improved generalization ability and robustness to data variability. This research emphasizes the potential of ensemble deep learning approaches to strengthen modern clinical decision support

Received: 16.06.2025 Accepted: 20.10.2025 Published: 30.10.2025

systems (CDSS). The developed framework enables more precise and interpretable diagnostic predictions, contributing to early diabetes detection and prevention strategies. Furthermore, the proposed methodology can be extended to other medical classification problems, providing a flexible and adaptive analytical tool for healthcare applications. The findings confirm that adaptive ensemble methods based on the Dirichlet distribution can serve as a foundation for reliable, transparent, and intelligent clinical decision-making in future healthcare systems.

Keywords: Neural network ensembles; Dirichlet distribution; CNN; RNN; LSTM; diabetes prediction; clinical decision support; deep learning.

Introduction

Diabetes is one of the most serious challenges in modern healthcare, characterized by high prevalence, diagnostic complexity, and significant socio-economic consequences [1]. According to the World Health Organization, the number of diabetes patients continues to rise steadily, making timely predictions and early diagnosis critically important tasks [2]. Modern clinical decision support systems (CDSS) increasingly integrate artificial intelligence (AI) methods, which can enhance the accuracy and reliability of predictions while reducing the risk of diagnostic errors [3].

Although traditional machine learning methods have demonstrated effectiveness in various medical applications, they have limitations when handling large, multidimensional, and heterogeneous datasets. These limitations become especially evident in the presence of high variability in medical data, noise, missing values, and interrelated features. Consequently, deep learning has emerged as a promising approach for developing intelligent medical diagnostic systems [4].

Deep neural networks (DNNs) enable the automation of informative feature extraction and the construction of complex models to uncover hidden patterns in data. In recent years, convolutional neural networks (CNNs) and recurrent neural networks (RNNs, LSTMs) have gained particular popularity in medicine due to their ability to effectively process both tabular and sequential data while accounting for spatial and temporal dependencies.

One of the key strategies for improving predictive model performance is ensembling, a method that combines multiple neural network architectures to compensate for individual shortcomings and enhance prediction accuracy. This study focuses on the development and optimization of an ensemble model for predicting the likelihood of diabetes progression, incorporating CNN, RNN, and LSTM architectures, with the Dirichlet Ensemble method employed for optimal prediction combination.

The proposed approach aims to increase the model's robustness to noise and variability in medical data, improve generalization capability, and reduce the probability of critical diagnostic errors. The conducted research includes data preprocessing, model construction, hyperparameter tuning using Grid Search, and comprehensive analysis of results based on standard classification metrics.

Literature review

The shift to deep learning for diabetes prediction is motivated by the need to enhance the accuracy and reliability of medical prognostic models. After data preparation, including balancing and preprocessing using traditional methods, the next stage involved the application of more powerful deep learning models.

Deep learning represents one of the most advanced directions in AI, effectively addressing both supervised and unsupervised tasks. It overcomes the limitations of classical machine learning, particularly when dealing with large and multidimensional datasets. Neural network training is performed using the backpropagation method combined with gradient descent

(GDM) to minimize error or categorical cross-entropy [5], [6], [7]. Modern architectures with optimized activation functions and layer structures have made deep neural networks successful across numerous domains [8], [9], [10], [11], including healthcare, where robust and interpretable models are crucial for clinical decision support.

In this study, three architectures were implemented:

- CNN (Convolutional Neural Network) – automatically extracts features and transforms input data into compact representations, which is particularly useful for heterogeneous medical datasets.
- LSTM (Long Short-Term Memory) – a recurrent network with memory for handling temporal dependencies and capturing long-term patterns, suitable for sequential medical records.
- RNN (Recurrent Neural Network) – a classical recurrent architecture capable of modeling sequential relationships, serving as a baseline for more advanced recurrent models.

To overcome the limitations of individual architectures and to leverage their complementary strengths, an ensemble approach was employed. Specifically, the Dirichlet Ensemble method was applied to optimally combine predictions from CNN, RNN, and LSTM, ensuring adaptive weighting based on individual model performance. Furthermore, hyperparameter optimization via Grid Search was integrated to fine-tune training parameters, activation functions, and network depth, thus improving generalization capability and reducing overfitting.

This integrated methodology aims to provide a more accurate, robust, and clinically reliable framework for diabetes prediction, ultimately supporting early detection and effective decision-making in healthcare practice.

The aim and objectives of the study

Relevance of the study. The increase in diabetes incidence and the complexity of its timely diagnosis make prediction and early detection one of the key tasks in modern healthcare. The use of deep learning and ensemble methods increases the robustness of models to noise, provides better generalization ability, and reduces the risk of diagnostic errors, which is of great importance for medical practice.

The object of the study is the processes of diabetes prediction and diagnosis using artificial intelligence and deep learning methods.

The aim of the study is to develop and optimize an ensemble model based on CNN, RNN, and LSTM for predicting the probability of developing diabetes using the Dirichlet Ensemble method and searching for optimal hyperparameters (Grid Search).

Scientific novelty of the research:

1) An ensemble model combining different deep neural network architectures (CNN, RNN, LSTM) is proposed to improve the accuracy and stability of prediction.

2) The Dirichlet Ensemble method is used for optimal combination of predictions, which has rarely been used in medical diagnostics before.

3) A comprehensive approach to the preparation and balancing of medical data using One-Hot Encoding and Dropout regularization has been implemented.

4) The use of he_normal initialization and ELU/SELU activation functions in medical classification tasks has been justified.

Practical significance of the research. The developed model can be integrated into clinical decision support systems (CDSS), helping doctors minimize the risk of errors. Improving the accuracy of predictions will allow diabetes to be detected at an early stage, which contributes to the timely start of treatment and reduces the socio-economic consequences of the disease. The hyperparameter selection algorithm (Grid Search) and the proposed regularization methods can be used for other medical analytics and diagnostic tasks.

Materials and methods

This study proposes an algorithm for applying the Grid Search method for clinical decision support tasks based on a CNN classification model, consisting of several stages as illustrated in Figure 1.

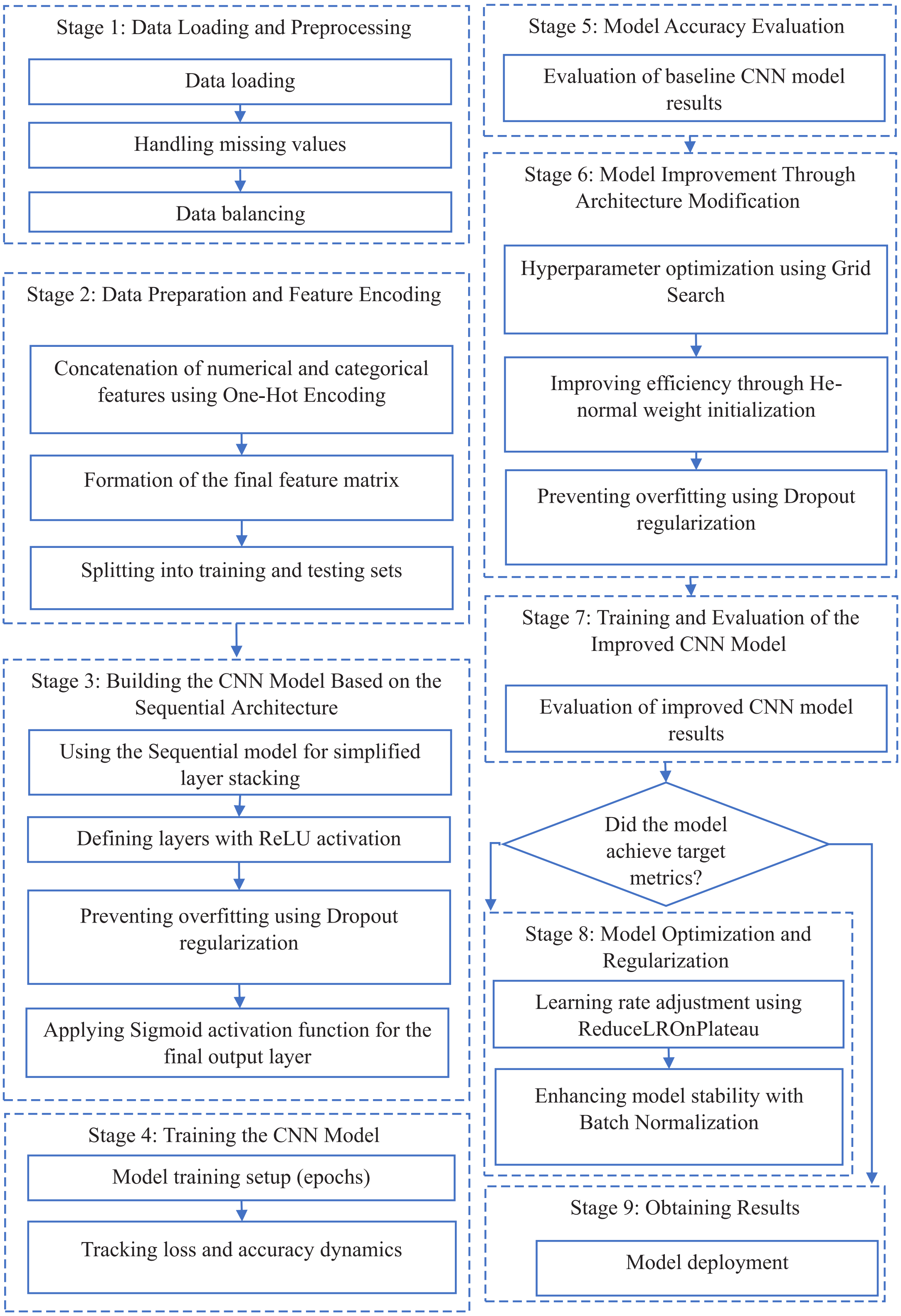
Stage 1: Data Loading and Preprocessing
Data loading
Handling missing values
Data balancing
Stage 2: Data Preparation and Feature Encoding
Concatenation of numerical and categorical features using One-Hot Encoding
Formation of the final feature matrix
Splitting into training and testing sets
Stage 3: Building the CNN Model Based on the Sequential Architecture
Using the Sequential model for simplified layer stacking
Defining layers with ReLU activation
Preventing overfitting using Dropout regularization
Applying Sigmoid activation function for the final output layer
Stage 4: Training the CNN Model
Model training setup (epochs)
Tracking loss and accuracy dynamics
Stage 5: Model Accuracy Evaluation
Evaluation of baseline CNN model results
Stage 6: Model Improvement Through Architecture Modification
Hyperparameter optimization using Grid Search
Improving efficiency through He-normal weight initialization
Preventing overfitting using Dropout regularization
Stage 7: Training and Evaluation of the Improved CNN Model
Evaluation of improved CNN model results
Did the model achieve target metrics?
Stage 8: Model Optimization and Regularization
Learning rate adjustment using ReduceLROnPlateau
Enhancing model stability with Batch Normalization
Stage 9: Obtaining Results
Model deployment

Figure 1. Algorithm of Applying the Grid Search Method
for Clinical Decision Support Tasks Based on a CNN Model

After loading and balancing the dataset, the second stage involves encoding categorical data into a numerical format. For this purpose, the One-Hot Encoding (OHE) method is applied [12]. Medical datasets often contain categorical features such as diagnoses, symptoms, patient gender, etc. The One-Hot Encoding method converts these categorical variables into a numerical format suitable for machine learning models. Each unique class is represented as a separate binary feature, eliminating issues related to ordering or hierarchy in the data.

Let C denote the set of categories for a given categorical variable. For example, let C={c1,c2,…,ck} represent k unique categories. For each category ci, a binary vector of length k is created, as shown in Equation (1) [13]:

$$\mathrm{OHE}(c_i) = \begin{cases} 1, if \ \mathrm{c} = c_i \\ 0, if \ \mathrm{c} \neq c_i \end{cases} \tag{1}$$

where c – is the value of the encoded variable.

The application of One-Hot Encoding in medical data is an important step toward improving the quality and accuracy of machine learning models. This method transforms categorical data into a format suitable for analysis and ensures better interpretability of the results.

At the third stage of the algorithm, a CNN model is constructed based on Sequential architecture. This architecture is specifically designed for sequential neural networks, where each layer passes its output directly to the next layer without complex branching [14]. It is an ideal option for simple architecture and rapid prototyping of models.

The mathematical model of a convolutional neural network can be expressed by the following equation, Equation (2) [15]:

$$n^{[l]} = \left[\frac{n^{[l-1]}+2p^{[l]}-f^{[l]}}{s^{[l]}} + 1\right] \ n_c^{[l-1]} \tag{2}$$

where $n^{[l]}$ – is the depth (number of channels) of the output volume at layer l; $n^{[l-1]}$ – is the depth of the input volume at the previous layer $(l-1)$; $p^{[l]}$ – is the padding applied to the input volume at layer l; $f^{[l]}$ – is the size of the receptive field (filter/kernel) at layer l; $s^{[l]}$ – is the stride used in the convolution operation at layer l; $n_c^{[l-1]}$ – is the height or width of the input volume at layer $l-1$ (assuming a square input).

The mathematical model presented in equation (3) is used to calculate the size of the output volume, which is determined by the height, width, and depth of the output volume in the current layer:

$$output = \ n_h^{[l]} \times n_w^{[l]} \times n_c^{[l]} \tag{3}$$

where $n_h^{[l]}$ denotes the height of the output volume at layer l; $n_w^{[l]}$ denotes the width of the output volume at layer l; $n_c^{[l]}$ denotes the depth of the output volume at layer l.

Suppose that the architecture consists of L layers. Each layer l applies a certain nonlinear or linear operation to the input data, and the output of layer l is used as the input for layer $lx+1$. Sequentially, such layers can be described using functions f_l, where l is the index of the layer.

Let the input data x_0 represent the input features or vectors for the model.

Each layer l is represented by a function f_l, which may include a linear operation (e.g., matrix multiplication with weights W_l) and a nonlinear activation (e.g., ReLU, sigmoid, or softmax activation), as expressed in equation (4):

$$x_l = f_l(x_{l-1}) = \sigma(W_l x_{l-1} + b_l) \tag{4}$$

where: W_l – the weight matrix of layer l, b_l – the bias vector; σ– the activation function (e.g., ReLU, sigmoid, etc.); x_{l-1} – the output from the previous layer.

The final layer f_L produces the final output of the model, equation (5):

$$\hat{y} = f_L(f_{L-1}(\dots f_1(x_0)\dots)) \tag{5}$$

The sequence of applying functions $f_1, f_2, \dots, f_l$ corresponds to the sequence of layers in the model.

Thus, the Sequential model is formalized as the consecutive application of several linear and nonlinear functions that transform the input data into the output [16].

When constructing a CNN model, the Dropout regularization method is applied to prevent overfitting. This regularization technique consists of randomly "dropping out" (setting to zero) a certain percentage of neurons at each training step, which makes the network more robust to overfitting and increases its ability to generalize to new data.

Let x be the input vector for a neural network layer, and W the weight matrix of this layer. During standard forward propagation (without Dropout), the output of the layer can be described by equation (6) [17]:

$$y{=}Wx{+}b \tag{6}$$

where W – is the weight matrix, b – is the bias vector, and x – is the input vector.

When applying Dropout during the training phase, a fraction of neurons is randomly deactivated. Let ρ - be the probability that a neuron is retained (i.e., not dropped). Then, Dropout is introduced using a mask m, which consists of random values taking 1 (the neuron is active) with probability p and 0 (the neuron is "dropped") with probability 1−p.

Mathematically, the mask can be expressed by equation (7):

$$m_i{\sim}\mathrm{Bernoulli}(p) \tag{7}$$

where m_i – is an element of the mask m, which takes the value 1 with probability p and 0 with probability $1{-}\mathrm{p}$. For each input neuron such a mask is generated.

During training, the output of the layer with Dropout can be described as follows, given in equation (8):

$$\tilde{x} = m \odot x \tag{8}$$

m — the Dropout mask, $\odot$ — element-wise multiplication (Hadamard product), x — the input vector. This means that some elements of vector *x* are multiplied by 0 (deactivated), while the remaining ones remain unchanged.

After applying Dropout to the input data, the output of the layer can be expressed as shown in equation (9):

$$y = W\tilde{x} + b = W(m \odot x) + b \tag{9}$$

The vector x is modified by the mask mmm, which randomly nullifies its elements.

Thus, the Dropout regularization method helps to reduce the interdependence of neurons and thereby decreases the risk of overfitting, ensuring more robust model training.

At the sixth stage of the algorithm, optimal hyperparameters are selected using the Grid Search method. Grid Search is one of the approaches to hyperparameter optimization in machine learning, allowing the exhaustive enumeration of all possible combinations of hyperparameters from a specified range of values and the selection of those that yield the best results based on a chosen quality metric [18].

Let there be a set of hyperparameters, each taking different values, equation (10):

$$\begin{gathered}H_1 \in \{h_{1,1}, h_{1,2}, \dots, h_{1,m_1}\} \\ H_2 \in \{h_{2,1}, h_{2,2}, \dots, h_{2,m_2}\} \\ H_k \in \{h_{k,1}, h_{k,2}, \dots, h_{k,m_k}\}\end{gathered} \tag{10}$$

where: $H_1, H_2,...,H_k$ – are the hyperparameters, $h_{i,j}$ – are the possible values of hyperparameter H_i, m_i – is the number of values of hyperparameter H_i.

Ultimately, the task of Grid Search is to enumerate all possible combinations, equation (11):

$$\{(h_{1,i_1}, h_{2,i_2}, \dots, h_{k,i_k}) | i_1 \in [1, m_1], i_2 \in [1, m_2], \dots, i_k \in [1, m_k]|\} \tag{11}$$

Thus, for each combination, the model can be evaluated using cross-validation or on a validation dataset.

When training neural networks, proper weight initialization plays a key role in ensuring efficient gradient propagation. Incorrect initialization may lead to vanishing or exploding gradients, which is particularly critical for deep architectures. To improve the efficiency of CNNs, this study applies the *he_normal* initializer [19], recommended for networks with nonlinear activation functions such as ReLU and its variations (e.g., Leaky ReLU).

As the activation function, ELU (Exponential Linear Unit) is employed—a modified version of ReLU that can accelerate training and mitigate the "dead neuron" problem. Similar to ReLU, ELU returns the input value for positive arguments, but for non-positive values it applies an exponential function, driving the result closer to zero and thereby improving training stability.

The *he_normal* initialization prevents issues typical of ReLU by scaling the initial weights depending on the number of neuron inputs, thus enhancing convergence [20].

If the accuracy of the improved model remains insufficient, regularization and optimization with the *ReduceLROnPlateau* method are applied. This method automatically decreases the learning rate when no improvement is observed over a predefined number of epochs. Implemented in Keras and PyTorch, it enables the model to escape local minima.

Model quality is evaluated using the metrics *Loss, Accuracy, Precision, Recall,* and *F1-score,* while the confusion matrix is employed for a visual analysis of the areas where the model makes errors.

Results

The experimental study of the Grid Search algorithm for clinical decision support tasks based on the CNN model begins with importing the necessary libraries such as Keras and TensorFlow. These libraries enable the construction of complex neural network architectures and optimization methods.

Once all required libraries are loaded, the dataset is imported, and the final data frame is created. During this stage, missing values are removed, the data is balanced, merged, and randomly shuffled, and indices are reset to eliminate any prior ordering. As a result, the dataset contains:

– number of patients without diabetes: 552336
– number of patients with diabetes: 15969
– size of the final balanced DataFrame: 31938 rows × 13 columns

Data preparation for subsequent machine learning model training is then performed. The process begins with excluding the target variable Diabetes from the dataset while storing its values in the label vector *Y* for training. For features involved in modeling, categorical variables are transformed into numerical format using the One-Hot Encoding method, which converts categorical attributes into a machine-learning-compatible representation. At the same time, selected numerical features are extracted directly from the original DataFrame.

The result of these operations is the final feature matrix X, which combines both numerical and encoded categorical data, ready for use in machine learning algorithms. The concatenation of numerical and processed categorical features into a unified matrix ensures a comprehensive representation of the data required for effective model training. A final verification of the dimensions of matrices X and Y confirms their consistency, with 31938 samples and 38 features, indicating successful data preparation for the modeling phase. Next, the dataset is split into training and test sets.

Hyperparameter optimization of the model using the Grid Search method provides several advantages:

- Selection of activation function and weight initializer. The SELU activation function automatically normalizes layer outputs, maintaining a mean of 0 and standard deviation of 1, thereby reducing the risk of vanishing or exploding gradients.
- Optimizer selection. Among the tested options, Nadam – a modification of Adam with adaptive moment estimation – demonstrated the best performance.
- Learning rate tuning. This parameter is critical for training efficiency. GridSearchCV identified the optimal value as 0.0005.
- Optimization of epochs and batch size. GridSearchCV also assisted in selecting these parameters, taking into account the specifics of the dataset and model.
- Model construction. The create_model function generates sequential architecture with multiple hidden layers and Dropout for overfitting prevention. The output layer employs sigmoid activation for binary classification.

The optimization process can be either simultaneous or sequential, including adjustments to the number of layers, neurons, learning rate, and other parameters. The model is trained across all hyperparameter combinations in the grid, with each combination undergoing 1000 epochs of training with a batch size of 100 and validation on 20% of the data.

After training, accuracy and loss plots for both training and validation sets are constructed (Figure 2), allowing for visual evaluation of the training dynamics and the model's generalization ability.

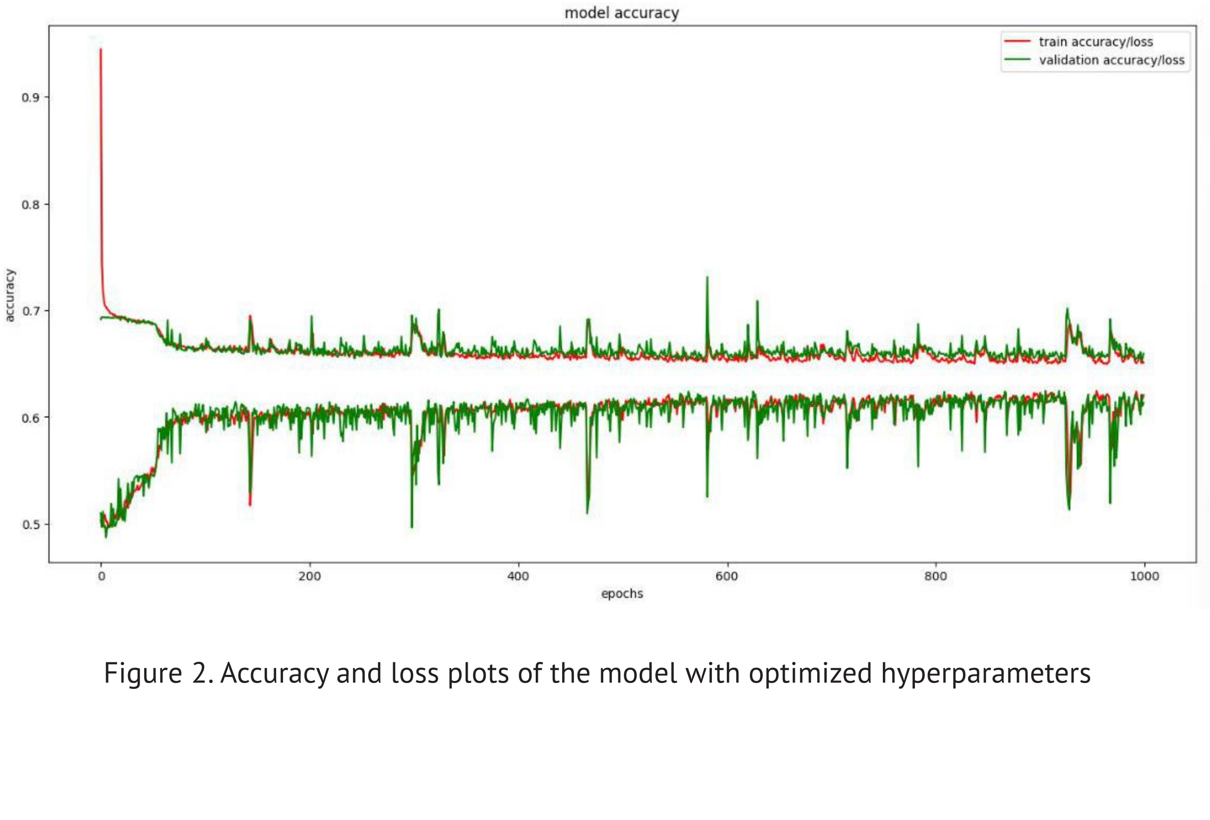

Figure 2. Accuracy and loss plots of the model with optimized hyperparameters

Discussion of results

Based on the analysis of the accuracy and loss plots (Figure 2), the following observations can be made:

- Initial training phase (first few dozen epochs): there is a sharp increase in training accuracy and a simultaneous decrease in the loss function, indicating active weight adaptation to the training set.
- Stabilization phase: training accuracy stabilizes around 0.7, while validation accuracy fluctuates within the same range.
- Generalization: the small gap between training and validation accuracy curves suggests the absence of pronounced overfitting; the stable dynamics indicate that the model has reached its performance limit for the given task.
- Validation variability: fluctuations in validation metrics reflect the model's sensitivity to data variability, which may be due to data heterogeneity or reduced representativeness.
- Final stage of training: a slight increase in the gap between training and validation accuracy is observed, which may indicate the onset of mild overfitting.

Conclusion

This study presents an approach to predicting diabetes risk based on an ensemble of CNN, RNN, and LSTM models using a Dirichlet Ensemble framework and hyperparameter optimization via Grid Search. The proposed method accounts for both spatial and temporal dependencies in the data, enhances robustness to noise, and improves the model's generalization capability. Hyperparameter optimization enabled the selection of optimal activation functions, weight initializers, optimizers, learning rate, number of epochs, and batch size, which contributed to increased accuracy and reduced risk of overfitting. The model demonstrated stable performance with an accuracy of approximately 0.7 on both the training and validation datasets, with a minimal gap between them. These results confirm that the proposed method can be effectively integrated into clinical decision support systems for early diabetes detection, thereby improving the accuracy and reliability of diagnosis.

Acknowledgment

The work was carried out with the support of grant funding for scientific and (or) scientific and technical projects for 2023-2025 of the Ministry of Science and Higher Education of the Republic of Kazakhstan (grant №AP19679525).

References

[1] Mangoulia, P., Milionis, C., Vlachou, E., & Ilias, I. (2024). The Interrelationship between Diabetes Mellitus and Emotional Well-Being: Current Concepts and Future Prospects. Healthcare, 12(14), 1457. https://doi.org/10.3390/healthcare12141457

[2] Hossain, M. J., Al-Mamun, M., & Islam, M. R. (2024). Diabetes mellitus, the fastest growing global public health concern: Early detection should be focused. Health Science Reports, 7(3), e2004. https://doi.org/10.1002/hsr2.2004

[3] Ouanes, K., & Farhah, N. (2024). Effectiveness of artificial intelligence (AI) in clinical decision support systems and care delivery. Journal of medical systems, 48(1), 74. https://doi.org/10.1007/s10916-024-02098-4

[4] Asif, S., Wenhui, Y., ur-Rehman, S. et al. (2025). Advancements and Prospects of Machine Learning in Medical Diagnostics: Unveiling the Future of Diagnostic Precision. Arch Computat Methods Eng 32, 853–883. https://doi.org/10.1007/s11831-024-10148-w

[5] Shanthi, K., Manimekalai, S. (2024). Cervical Cancer Detection Using Ensemble Neural Network Algorithm with Stochastic Gradient Descent (SGD) Optimizer. SN COMPUT. SCI. 5, 1151. https://doi.org/10.1007/s42979-024-03365-4

[6] Verma R. and Muhuri S. (2025). Predicting Gestational Diabetes and PCOS Susceptibility: A Machine Learning-Based Cross-Framework Analysis," 2025 3rd International Conference on Data Science and Information System (ICDSIS), Hassan, India, pp. 1-6, doi: 10.1109/ICDSIS65355.2025.11071152

[7] Ismukhamedova, A., Uvaliyeva, I., & Belginova, S. (2024). Integrating machine learning in electronic health passport based on WHO study and healthcare resources. Informatics in Medicine Unlocked, 44, 101428. https://doi.org/10.1016/j.imu.2023.101428

[8] Rane, C., Tyagi, K., Kline, A., Chugh, T., & Manry, M. (2024). Optimizing performance of feedforward and convolutional neural networks through dynamic activation functions. *Evolutionary Intelligence*, *17*(5), 4083-4093. https://doi.org/10.1007/s12065-024-00973-0

[9] Wassan, S., Dongyan, H., Suhail, B., Jhanjhi, N. Z., Xiao, G., Ahmed, S., & Murugesan, R. K. (2024). Deep convolutional neural network and IoT technology for healthcare. *Digital health*, *10*, 20552076231220123.

[10] Cantemir, E., & Kandemir, O. (2024). Use of artificial neural networks in architecture: determining the architectural style of a building with a convolutional neural networks. *Neural Computing and Applications*, *36*(11), 6195-6207. https://doi.org/10.1007/s00521-023-09395-y

[11] Ayad, J. (2024). Survey on neural networks in networking: Applications and advancements. *Babylonian Journal of Networking*, *2024*, 135-147. https://doi.org/10.58496/BJN/2024/014

[12] Rezvan, M. R., Sorkhi, A. G., Pirgazi, J., & Kallehbasti, M. M. P. (2024). AdvanceSplice: Integrating N-gram one-hot encoding and ensemble modeling for enhanced accuracy. *Biomedical Signal Processing and Control*, *92*, 106017. https://doi.org/10.1016/j.bspc.2024.106017

[13] Al-Shehari, T., & Alsowail, R. A. (2021). An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. *Entropy*, *23*(10), 1258. https://doi.org/10.3390/e23101258

[14] Gu, B., & Sung, Y. (2021). Enhanced Reinforcement Learning Method Combining One-Hot Encoding-Based Vectors for CNN-Based Alternative High-Level Decisions. Applied Sciences, 11(3), 1291. https://doi.org/10.3390/app11031291

[15] Nguyen-Ngoc, L., Nguyen-Huu, Q., De Roeck, G., Bui-Tien, T., & Abdel-Wahab, M. (2024). Deep neural network and evolved optimization algorithm for damage assessment in a truss bridge. *Mathematics*, *12*(15), 2300. https://doi.org/10.3390/math12152300

[16] Li, G., Qian, G., Delgadillo, I. C., Muller, M., Thabet, A., & Ghanem, B. (2020). Sgas: Sequential greedy architecture search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1620-1630). https://openaccess.thecvf.com/content_CVPR_2020/papers/Li_SGAS_Sequential_Greedy_Architecture_Search_CVPR_2020_paper.pdf

[17] Garbin, C., Zhu, X., & Marques, O. (2020). Dropout vs. batch normalization: an empirical study of their impact to deep learning. Multimedia tools and applications, 79(19), 12777-12815. (2020). https://doi.org/10.1007/s11042-019-08453-9

[18] Rimal, Y., Sharma, N., & Alsadoon, A. (2024). The accuracy of machine learning models relies on hyperparameter tuning: student result classification using random forest, randomized search, grid search, bayesian, genetic, and optuna algorithms. Multimedia Tools and Applications, 83(30), 74349-74364. 74349–74364 (2024). https://doi.org/10.1007/s11042-024-18426-2

[19] Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D., & Traore, D. (2022). Explainable deep convolutional neural networks for insect pest recognition. Journal of Cleaner Production, 371, 133638. https://doi.org/10.1016/j.jclepro.2022.133638

[20] Narkhede, M. V., Bartakke, P. P., & Sutaone, M. S. (2022). A review on weight initialization strategies for neural networks. Artificial intelligence review, 55(1), 291-322. https://doi.org/10.1007/s10462-021-10033-z