electronics

MDPI

Article

Utilizing Attention-Enhanced Deep Neural Networks for Large-Scale Preliminary Diabetes Screening in Population Health Data

Hongwei Hu [1,2,†], Wenbo Dong [3,†], Jianming Yu [1,*], Shiyan Guan [1] and Xiaofei Zhu [1]

[1] Electronics Equipment Manufacturing Engineering Technology Research and Development Center in Jiangsu Province, Huaian 223003, China; hongweihu_hit20170706@alu.hit.edu.cn (H.H.); 044213@jsei.edu.cn (S.G.); 230852@jsei.edu.cn (X.Z.)

[2] Jiangsu Vocational College of Electronics and Information, Huaian 223003, China

[3] Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR, China; dongwbo@connect.hku.hk

[*] Correspondence: 042102@jsei.edu.cn; Tel.: +86-0517-88241

[†] These authors contributed equally to this work.

Abstract: Early screening for diabetes can promptly identify potential early stage patients, possibly delaying complications and reducing mortality rates. This paper presents a novel technique for early diabetes screening and prediction, called the Attention-Enhanced Deep Neural Network (AEDNN). The proposed AEDNN model incorporates an Attention-based Feature Weighting Layer combined with deep neural network layers to achieve precise diabetes prediction. In this study, we utilized the Diabetes-NHANES dataset and the Pima Indians Diabetes dataset. To handle significant missing values and outliers, group median imputation was applied. Oversampling techniques were used to balance the diabetes and non-diabetes groups. The data were processed through an Attention-based Feature Weighting Layer for feature extraction, producing a feature matrix. This matrix was subjected to Hadamard product operations with the raw data to obtain weighted data, which were subsequently input into deep neural network layers for training. The parameters were fine-tuned and the L2 regularization and dropout layers were added to enhance the generalization performance of the model. The model's reliability was thoroughly assessed through various metrics, including the accuracy, precision, recall, F1 score, mean squared error (MSE), and R2 score, as well as the ROC and AUC curves. The proposed model achieved a prediction accuracy of 98.4% in the Pima Indians Diabetes dataset. When the test dataset was expanded to the large-scale Diabetes-NHANES dataset, which contains 52,390 samples, the test precision of the model improved further to 99.82%, with an AUC of 0.9995. A comparative analysis was conducted using multiple models, including logistic regression with L1 regularization, support vector machine (SVM), random forest, K-nearest neighbors (KNNs), AdaBoost, XGBoost, and the latest semi-supervised XGBoost. The feature extraction method using attention mechanisms was compared with the classical feature selection methods, Lasso and Ridge. The experiments were performed on the same dataset, and the conclusion was that the Attention-based Ensemble Deep Neural Network (AEDNN) outperformed all the aforementioned methods. These results indicate that the model not only performs well on smaller datasets but also fully leverages its advantages on larger datasets, demonstrating strong generalization ability and robustness. The proposed model can effectively assist clinicians in the early screening of diabetes patients. This is particularly beneficial for the preliminary screening of high-risk individuals in large-scale, extensive healthcare datasets, followed by detailed examination and diagnosis. Compared to the existing methods, our AEDNN model showed an overall performance improvement of 1.75%.

Keywords: diabetes prediction; attention-enhanced deep neural network (AEDNN); attention-based feature weighting layer; Pima Indians diabetes dataset

check for updates

Citation: Hu, H.; Dong, W.; Yu, J.; Guan, S.; Zhu, X. Utilizing Attention-Enhanced Deep Neural Networks for Large-Scale Preliminary Diabetes Screening in Population Health Data. *Electronics* **2024**, *13*, 4177. https://doi.org/10.3390/electronics13214177

Academic Editor: Antoni Morell

Received: 10 September 2024
Revised: 14 October 2024
Accepted: 18 October 2024
Published: 24 October 2024

1. Introduction

Diabetes is a serious metabolic disorder [1]. Insulin levels directly influence blood glucose, and insufficient insulin secretion by the pancreas leads to elevated blood sugar levels, which over time results in diabetes [2]. According to the International Diabetes Federation's "IDF Diabetes Atlas 2021", 10.5% of the global population has diabetes, with more than 75% of patients residing in low- and middle-income countries. The number of diabetes cases is estimated to increase to 643 million by 2030 and 783 million by 2045. Additionally, the IDF reports that India ranks second worldwide in the number of diabetes patients, after China, with approximately 77 million individuals affected [3]. According to data from the World Health Organization [4], the mortality rate associated with diabetes and its complications is alarmingly high, with one person dying every five seconds due to diabetes or its related complications [5]. Diabetic nephropathy, retinopathy, and diabetic foot are leading causes of kidney failure, blindness, and amputations. In 2019, diabetes directly caused about 1.5 million deaths, and in 2021, 6.7 million adults died from diabetes [6].

Gestational diabetes is a specific type of diabetes that occurs during pregnancy. Unlike type 1 diabetes, gestational diabetes is usually temporary, with blood glucose levels usually returning to normal after delivery. It results from hormonal changes during pregnancy that affect the body's ability to use insulin, leading to elevated blood sugar levels. In contrast, children and adolescents are more susceptible to type 1 diabetes, a condition caused by pancreatic failure to produce insulin, requiring lifelong insulin therapy. Type 2 diabetes primarily affects middle-aged, elderly, and obese individuals and is characterized by pancreatic dysfunction and insulin resistance [7]. Early screening and targeted treatment of prediabetic patients are essential, significantly delaying the onset of complications and improving quality of life [8]. Artificial intelligence, particularly deep learning, plays an increasingly important role in disease screening assistance.

The main contributions of this study are as follows:

1. This paper proposes an Attention-based Feature Weighting Layer for dynamically analyzing the weights of physical examination data and adaptively adjusting the dimensions, thus establishing a predictive relationship between the physical examination data and diabetes.
2. This paper presents a universally applicable algorithm for processing physical examination data, which maintains data integrity under the conditions of outliers, missing values, and human recording errors, thus enhancing data robustness. In addition, this algorithm employs the SMOTE method to ensure data diversity.
3. This paper introduces a high-quality dataset, the Diabetes-NHANES dataset, derived from the NHANES data spanning 1999 to 2020, comprising 134,516 samples. In collaboration with endocrinologists from Xinyang Central Hospital in Henan Province, we meticulously selected 41 features from an initial pool of 5154, based on their direct and indirect relevance to diabetes. A mean imputation method was used to adjust for measurement discrepancies, and a rigorous sample exclusion strategy was developed: samples with more than 20% missing data in directly related features or less than 50% completeness in indirectly related features were excluded. Ultimately, 52,391 clean, well-organized, and ready-to-use diabetes-related samples were obtained, forming the Diabetes-NHANES dataset. This dataset is publicly available at https://github.com/hongweihaha/Diabetes-NHANESDataset.git (accessed on 17 October 2024).
4. In practical experiments, this data model exhibited superior performance compared to the existing models. The model achieved an accuracy of 98.4% on real-world datasets.

Section 2 explores the application of the relevant literature in the prediction of diabetes. Section 3 provides a detailed explanation of the materials and data preprocessing methods. Section 4 presents the proposed model architecture. Sections 5 and 6 introduce the model's training and optimization process, followed by its testing and evaluation.

2. Related Work

Many researchers have investigated various approaches for the prediction of diabetes. Shojaee-Mend et al. utilized a variety of algorithms, including artificial neural networks [9], to predict fasting blood glucose levels in adults from Tehran, Iran. The CatBoost model demonstrated the best performance with an AUC of 0.737[10]. Kumar et al. developed an integrated machine learning framework called "iDP", employing six techniques: random forest, Decision Tree, neural networks, AdaBoost, support vector machine, and XGBoost. The iDP framework achieved impressive results, with an accuracy of 95.26% and an AUC of 91.15% [11].

El-Bashbishy et al. focused on predicting pediatric diabetes using the Mansoura University Children's Hospital Diabetes dataset (MUCHD) through a deep neural network Multilayer Perceptron algorithm. Their model, consisting of ten hidden layers and automated hyperparameter tuning, achieved an impressive prediction accuracy of 99.8% [12]. Chen et al. began integrating deep neural networks with ensemble learning algorithms to enhance the interpretability of the models. For instance, by incorporating the modified random forest incremental interpretation (MRFII) algorithm, the model not only diagnoses the presence of diabetes but also provides a certain level of interpretability. This approach improved diagnostic accuracy by 11% compared to traditional models [13].

Zhang et al. utilized a Backpropagation Neural Network (BPNN) with batch normalization for non-invasive diabetes diagnosis, achieving high accuracy rates across several datasets [14]. In the realm of disease prediction, attention models have proven invaluable. An et al. introduced DeepRisk, an attention mechanism-based model that improves cardiovascular disease prediction accuracy by effectively integrating heterogeneous and time-ordered medical data [15]. Djenouri et al. combined multiple deep learning architectures (VGG16, RESNET, and DenseNet) with ensemble learning and attention mechanisms, achieving high detection accuracy in medical and plant disease datasets [16]. Zou et al. explored the XGBoost algorithm for predicting diabetes progression in prediabetic intervention treatments, demonstrating significant risk reduction in high-risk groups through specific interventions [17].

Despite these advancements, current methods have limitations in handling missing values and outliers effectively. This paper introduces a novel technique, the Attention-Enhanced Deep Neural Network (AEDNN), which leverages multi-head attention mechanisms and deep neural networks to enhance diabetes prediction accuracy. By addressing the Pima Indians Diabetes dataset's missing values and outliers through group median imputation and employing oversampling techniques, the AEDNN model offers a significant improvement in early diabetes screening and prediction. The proposed model, with meticulous parameter tuning and robust evaluation metrics, demonstrates an overall performance enhancement of 0.31% compared to the existing methods.

3. Materials and Data Preprocessing

The original medical data of patients contain a wealth of untapped information, which objectively assesses the relationship between patients and specific diseases. Different physiological parameters represent distinct features. Supervised learning, utilizing labeled data for model training, is widely adopted [18]. However, data quality is crucial for effective model learning. Missing values, outliers, and imbalanced data pose significant challenges to model generalization, leading to decreased classification performance and potentially erroneous predictions. Therefore, data preprocessing is imperative. The boxplot outlier detection algorithm is employed to identify anomalous data points, and then medians for each feature are calculated separately for diabetic and non-diabetic groups. These medians are then used to address missing and outlier values [19]. Oversampling techniques are also utilized to address the issue of data imbalance. Furthermore, data normalization eliminates the scale differences among indicators, scaling the data to a unified range. These preprocessing methods offer more accurate and higher-quality data for models, thereby enhancing their robustness and generalization capabilities.

This study introduces the innovative incorporation of the Attention-based Feature Weighting Layer, as shown in Figure 1. The model comprises four independent layers: the Attention-based Feature Weighting Layer, input layer, hidden layers, and output layer. Specifically, the Attention-based Feature Weighting Layer employs four attention heads to automatically adjust the attention weights of each feature based on a weight matrix. This aggregated attention is then fed into a fully connected neural network model. The input layer receives eight weighted features and the data propagate through three hidden layers. Each neuron in a hidden layer is connected to the output of the previous layer. The output of each layer, generated through activation functions, is then connected to the next layer. The output of each layer incorporates the abstract computational results of the previous layer. Fixing the initial random weights and biases ensures the reproducibility and stability of the model, while optimization of the weights and biases occurs during the backpropagation process. By combining the advantages of the Attention-based Feature Weighting Layer and employing three hidden layers, the model achieves robust and efficient classification of diabetes.

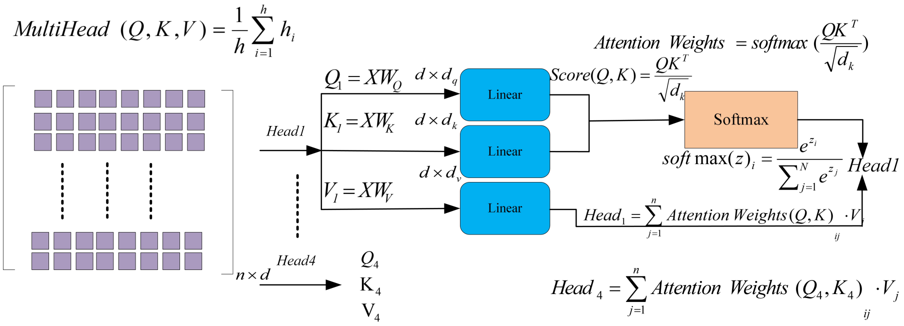

$MultiHead\ (Q,K,V) = \frac{1}{h}\sum_{i=1}^{h} h_i$
$Attention\ Weights\ = softmax\ (\frac{QK^T}{\sqrt{d_k}})$
$Q_1 = XW_Q$
$d \times d_q$
$Score(Q,K) = \frac{QK^T}{\sqrt{d_k}}$
Linear
Softmax
Head1
$K_1 = XW_K$
$d \times d_k$
Linear
$soft\max(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$
Head1
$d \times d_v$
$V_1 = XW_V$
Linear
$Head_1 = \sum_{j=1}^{n} Attention\ Weights(Q,K)_{ij} \cdot V_j$
Head4
$n \times d$
Q_4
K_4
V_4
$Head_4 = \sum_{j=1}^{n} Attention\ Weights\ (Q_4, K_4)_{ij} \cdot V_j$

Figure 1. Multi-head attention feature weighting module.

The system comprises several stages, including data collection, data preprocessing, model design and training, optimization, and model testing and evaluation, as shown in Figure 2. To address missing values and outliers in the dataset, we adopted median imputation. This method is suitable for continuous numerical features and is robust to outliers. In addition, we employ oversampling techniques, specifically the SMOTE algorithm, to address data imbalance. Subsequently, feature normalization is performed to ensure that different features have roughly equal influence on the model.

The multi-head attention mechanism empowers the model to autonomously learn from data by selectively focusing on key features. This approach enhances learning efficiency and enables the model to effectively extract relevant information. The weighted data are then fed into a deep learning model for training, where the model iteratively optimizes its parameters using backpropagation to minimize the loss function.

3.1. Raw Dataset

The self-constructed Diabetes-NHANES dataset used in this study is derived from the NHANES dataset, which includes a wide range of features that describe samples from multiple perspectives, such as demographic characteristics, medical history, nutritional status, physiological measurements, and laboratory test results. Due to its comprehensive coverage, this dataset is well suited for large-scale population health analyses. Additionally, the well-known Pima Indians Diabetes dataset was employed to further validate the performance of the models.

From the NHANES data (1999–2020), 134,516 samples were extracted and a proprietary dataset was formed specifically for diabetes prediction based on the relevance of the features for diabetes and the extent of missing data in each sample. Under the guidance of endocrinologists, 41 features directly or indirectly related to diabetes were selected from an initial 5154 features. Invalid survey responses, such as blood pressure data that could not be accurately filled out by respondents, were excluded, instead focusing on qualitative data, such as whether the individual was diagnosed with diabetes. The determination of diabetes status was based on three survey responses: “DID040: Age when first told you had diabetes”, “DID040G: Age when first told you had diabetes—questionnaire”, and “DID040Q: Age when first told you had diabetes—number”, which was further validated by “DIQ010: The doctor told you that you have diabetes.” Multiple data sources were linked by unique IDs to confirm each sample’s diabetes status.

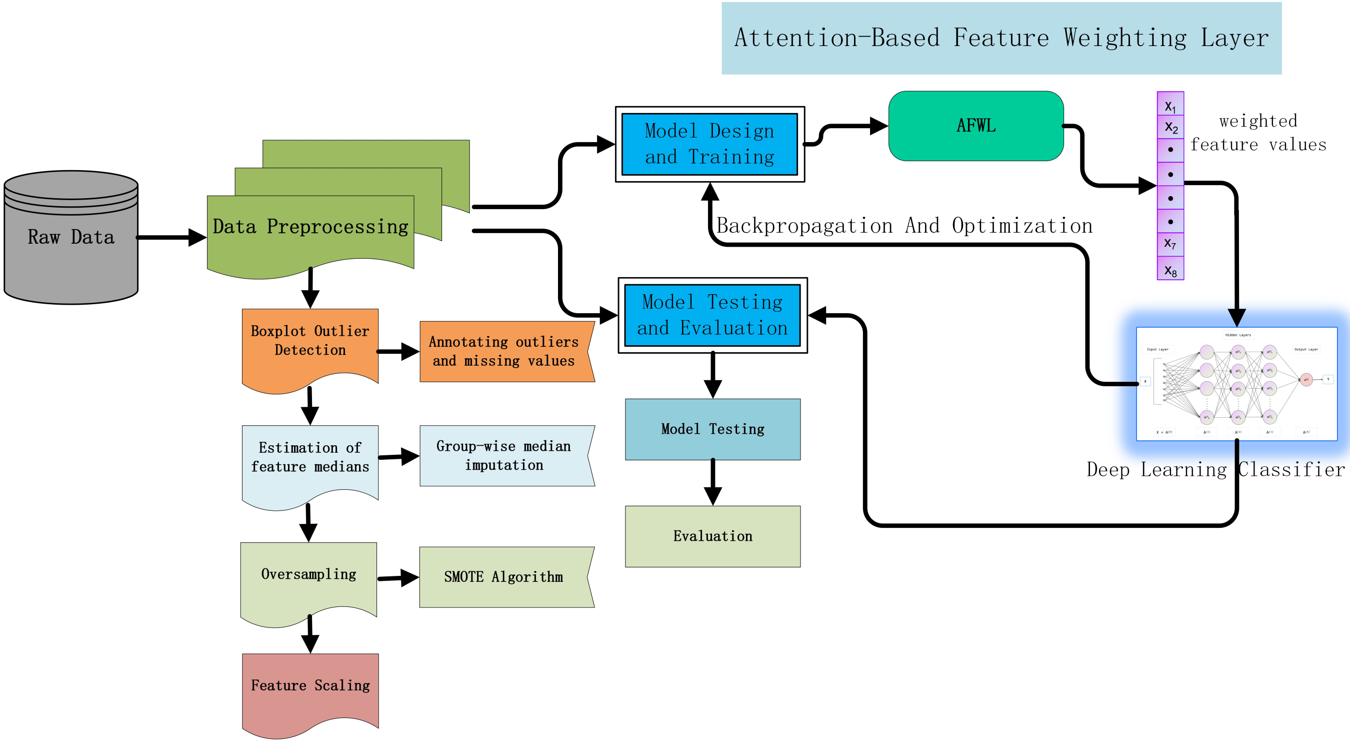

Attention-Based Feature Weighting Layer
Raw Data
Data Preprocessing
Boxplot Outlier Detection
Annotating outliers and missing values
Estimation of feature medians
Group-wise median imputation
Oversampling
SMOTE Algorithm
Feature Scaling
Model Design and Training
AFWL
x_1
x_2
x_7
x_8
weighted feature values
Backpropagation And Optimization
Model Testing and Evaluation
Model Testing
Evaluation
Deep Learning Classifier

Figure 2. The workflow and structure of the diabetes prediction system.

There are ones with multiple measurements, such as the following:

- “BPXDI1: Diastolic blood pressure (first reading) mm Hg”;
- “BPXDI11: Diastolic blood pressure (first reading) replicate 1 mm Hg”;
- “BPXDI12: Diastolic blood pressure (first reading) replicate 2 mm Hg”.

For these, the values were smoothed using a non-empty value average imputation method to mitigate the error across multiple readings. If all three measurements were available, the average of the three was used as the final value. If one measurement was missing, the average of the two available measurements was used instead. This dynamic averaging based on the number of valid data points significantly enhances data authenticity.

The NHANES dataset also contains a substantial amount of annotation codes, such as the following:

- BMDSADCM: Sagittal Abdominal Diameter Comment;
- BMIARMC: Arm Circumference Comment;
- BMISUB: Subscapular Skinfold Comment;
- BMITRI: Triceps Skinfold Comment;
- BMIWT: Weight Comment.

These codes were used to record methods or comments that might affect the measurement results during data collection. However, for the purposes of this study, such codes are irrelevant to diabetes prediction, so they were removed.

A strict sample exclusion strategy was designed. Samples with over 20% missing values for features directly related to diabetes were excluded, as were those with less than 50% completeness for features indirectly related to diabetes. After applying these criteria, a final dataset of 52,390 complete samples was obtained, resulting in a clean, organized, and user-friendly Diabetes-NHANES dataset.

The Diabetes-NHANES dataset contains 41 features and 52,390 samples, of which 11,168 are diabetic patients and 41,222 are non-diabetic individuals, with non-diabetic cases being nearly four times more prevalent than diabetic cases, indicating a significant class imbalance. In contrast, the Pima Indians Diabetes dataset consists of 768 samples with 8 features, including 268 diabetic patients and 500 non-diabetic patients, with the non-diabetic population being approximately twice the size of the diabetic population. Although the data have been meticulously collected and processed, there are still numerous missing values, outliers in features, and imbalanced data.

3.2. Data Preprocessing

The preprocessing phase focused on addressing missing values and outliers, where group median imputation was applied to handle absent and anomalous data. Oversampling techniques were also used to balance the diabetic and non-diabetic sample sizes. Lastly, data normalization was performed to ensure that all features were scaled consistently within the same range.

In this study, the data preprocessing steps included handling missing values, detecting and addressing outliers, oversampling, and data normalization. To ensure data quality and prevent biases during model training, the specific preprocessing steps are as follows:

1. Missing Value Handling

Both the Pima Indians Diabetes dataset and the Diabetes-NHANES dataset contain missing values in some features. Directly removing samples with missing values could result in significant information loss and a reduction in data volume. Therefore, this study employed a grouped median imputation method. This approach groups the data based on class labels (diabetic and non-diabetic), and the missing values in each group are imputed using the median value of the corresponding group. This method prevents the loss of valuable information and preserves the overall data distribution.

2. Outlier Detection and Treatment

To identify and handle outliers, this study employed the boxplot method. A boxplot determines outliers based on the interquartile range (IQR), defined by the first quartile (Q1) and third quartile (Q3). Outliers are defined as data points falling outside the "lower bound" (Q1 − 1.5 IQR) or "upper bound" (Q3 + 1.5 IQR). Detected outliers were treated using the grouped median imputation method, ensuring that the treatment of outliers does not distort the overall data distribution.

3. Class Imbalance Treatment

Due to the significant class imbalance between diabetic and non-diabetic patients, the model could be biased toward the majority class during training. To address this issue, the SMOTE (Synthetic Minority Oversampling Technique) method was applied. This technique generates synthetic minority class samples by interpolating between existing minority class samples and their nearest neighbors, thus balancing the class distribution in the dataset. By doing so, the imbalance between diabetic and non-diabetic samples was mitigated, reducing bias in the model training.

4. Data Normalization

Finally, all the features were normalized using the min–max normalization technique, which scales each feature to the [0, 1] range. Specifically, the normalization formula is given as

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

Here, X represents the original feature value, and X_{min} and X_{max} are the minimum and maximum feature values, respectively. Normalization helps eliminate the impact of feature scale differences on model training, improving both convergence speed and prediction accuracy.

Through these four preprocessing steps, this study ensured data integrity and balance, providing a more reliable foundation for model training and prediction.

3.2.1. Missing Values

The Pima Indians Diabetes dataset As shown in Table 1 and Diabetes-NHANES dataset contain a significant number of missing values. In the Diabetes-NHANES dataset, each feature exhibits an average of approximately 2300 missing values. These missing values may be due to human errors during data collection or equipment failures, leading to incomplete information on certain features. This incompleteness can introduce biases into the model and degrade the data quality, potentially causing inaccurate predictions when the model learns from incomplete data. Missing values are typically represented as 0 or NAN. Common approaches to handle missing values include deletion or imputation. Directly deleting missing values risks losing critical information and reducing the sample size, potentially distorting the model's learning outcomes. Therefore, this study adopts imputation methods to address missing values.

Table 1. Missing values and data ranges of features in the raw dataset.

Features	Number of Missing Values	Data Range
pregnant	0	0–17
Plasma_glucose_concentration	5	0–199
blood_pressure	35	1–122
Triceps_skin_fold_thickness	227	0–99
serum_insulin	374	0–846
BMI	11	0–67.1
Diabetes_pedigree_function	0	0.078–2.42
Age	0	21–81

3.2.2. Outliers

During the data collection process of the Pima Indians Diabetes dataset, human errors may inevitably introduce outliers. The Diabetes-NHANES dataset also contains a significant number of outliers. Therefore, it is necessary to conduct outlier detection on this dataset. In this study, the outlier detection method is based on the boxplot algorithm. A boxplot is a visual tool that intuitively displays the central tendency and dispersion of the data while also identifying potential outliers.

In a boxplot, the box represents the interquartile range (IQR), with the upper boundary corresponding to the third quartile (Q3) and the lower boundary to the first quartile (Q1). A line inside the box indicates the median (Q2). The IQR, which is the difference between Q3 and Q1, reflects the spread of the middle 50% of the data. To detect potential outliers, upper and lower limits are calculated using specific formulas:

- Upper bound: $Q3 + 1.5 \times IQR$;
- Lower bound: $Q1 - 1.5 \times IQR$.

Data points falling outside these bounds are considered outliers.

Figures 3 and 4 illustrate the distribution of outliers for each extracted feature. A total of 50 outliers were detected in the sample. The range of outliers for the number of pregnancies is 14 to 17. The outliers for plasma glucose concentration are 0. For blood pressure, the range of outliers is 24 to 122. The outliers for triceps skinfold thickness are 99. The range of outliers for serum insulin is 321 to 846. For BMI, the range of outliers is 52.3 to 67.1. The range of outliers for the function of the diabetes pedigree is 1.213 to 2.42. These outliers may be attributed to data collection errors.

Scatter plots allow for the intuitive identification of data anomalies, as shown in Figure 4, enabling targeted attention to these outliers during data imputation. For instance, as observed from the scatter plots, some data points for plasma glucose concentration are zero, which can be considered as omissions due to human error during data collection. Additionally, triceps skinfold thickness values of 99 are evident, which far exceed the normal range and are likely recording errors.

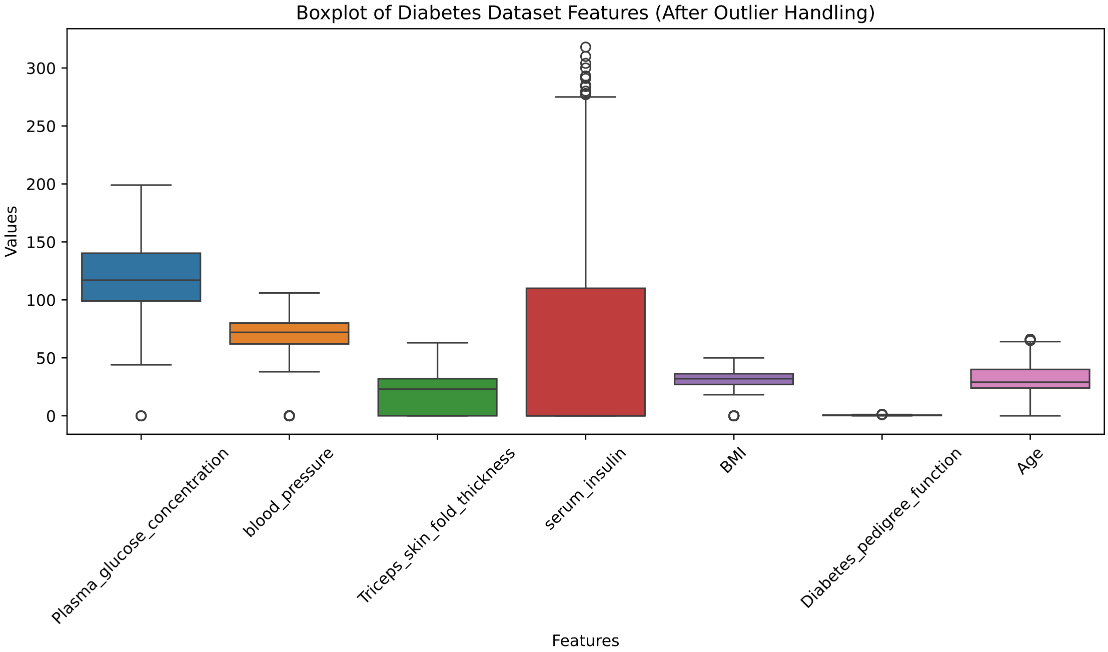


Figure 3. The Pima Indians Diabetes dataset employs boxplot annotation for the identification of outliers.

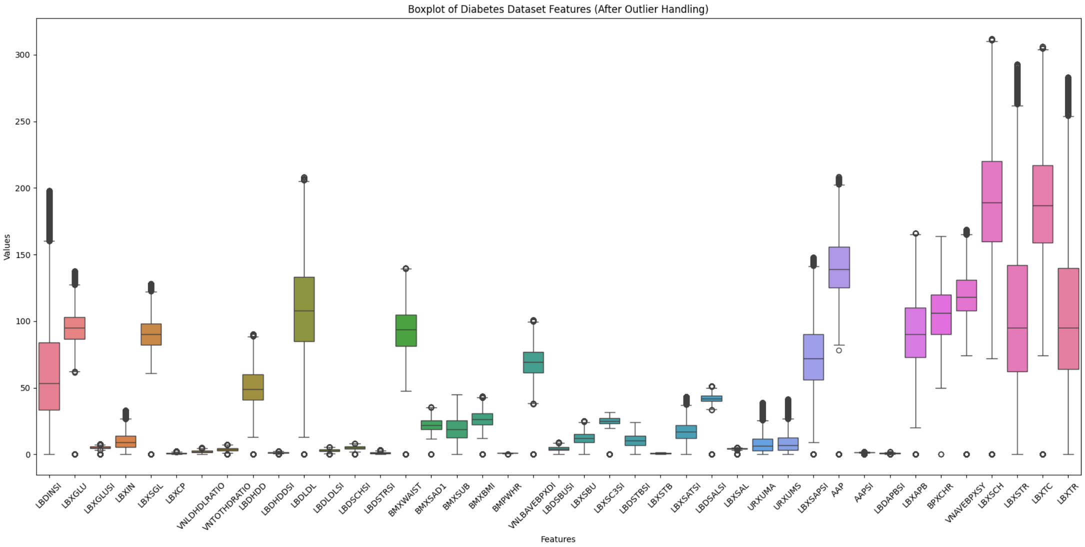


Figure 4. The Diabetes-NHANES dataset employs boxplot annotation for the identification of outliers.

3.2.3. Group-Wise Median Imputation

Common methods for handling missing and outlier data include mean imputation, mode imputation, and median imputation. Mean imputation can fail to reflect the distribution characteristics of the data, potentially leading to imputed values that deviate from the actual situation. Moreover, it may ignore correlations between different features, resulting in the inaccurate imputation of data. Mode imputation, which uses the most frequently occurring value in the dataset, is suitable for categorical data but not for continuous data. For example, features such as plasma glucose concentration, blood pressure, and BMI can take on any value within a certain range rather than fixed values. Furthermore, the mode does not accurately describe the central tendency of the data. Therefore, neither mean nor mode imputation effectively handles missing values in this dataset.

Given the significant differences between the diabetic and non-diabetic groups, where various health indicators change substantially after the onset of diabetes, relying solely on mean or mode imputation is inadequate. Imputing missing values based on the median of each group is a more scientific and data-consistent approach. The method of group median imputation involves dividing the data for the eight features into diabetic and non-diabetic groups and filling in missing values with the median of the respective group. This method accurately reflects the impact of disease status, ensuring data integrity and fully considering the influence of disease on the data.

The following validation demonstrates this point well. Histograms of the original data and the group median-imputed data are plotted to fully show the characteristics of the data distribution, as shown in Figures 5 and 6.

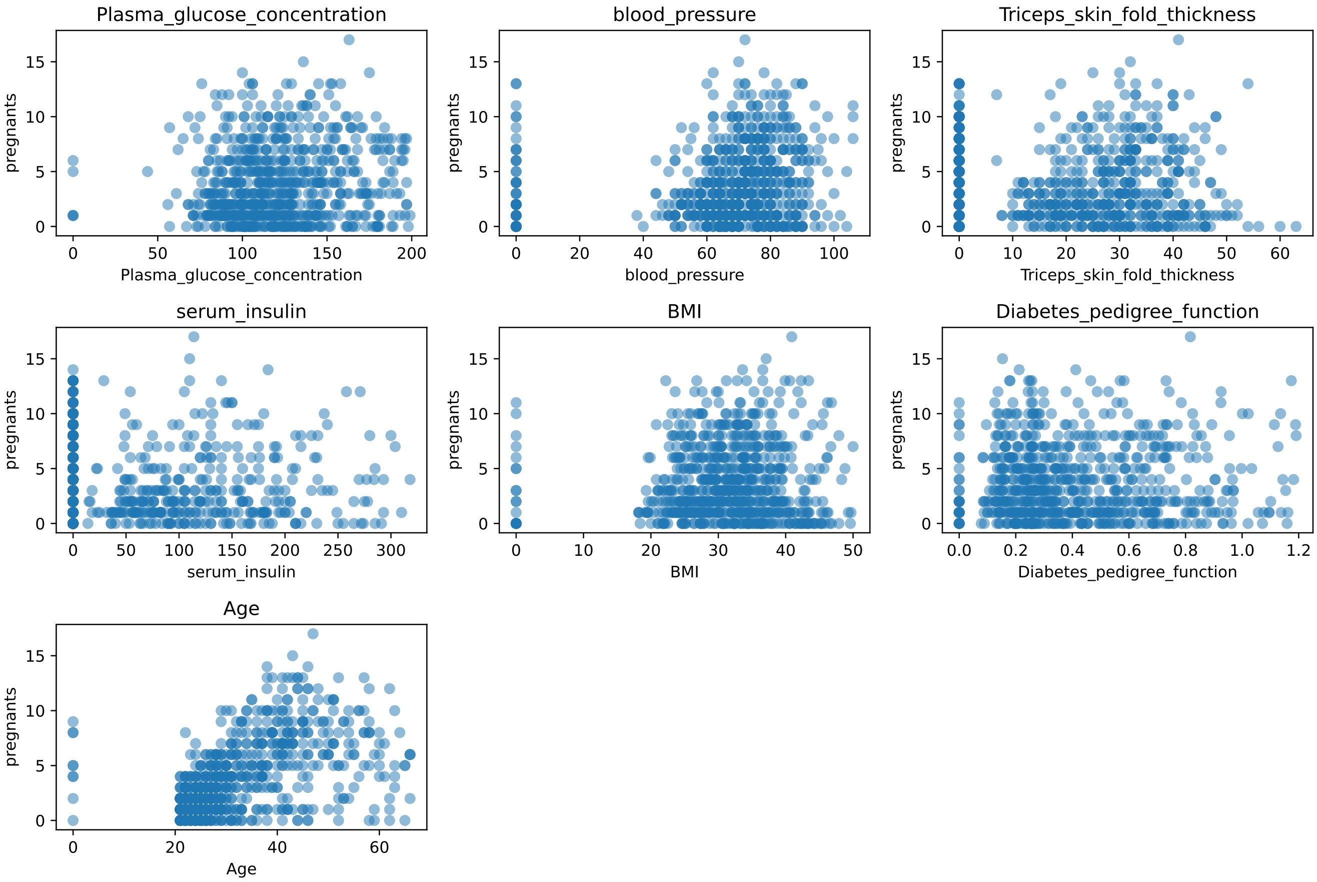


Figure 5. Scatter plots can visually display missing values and outliers.

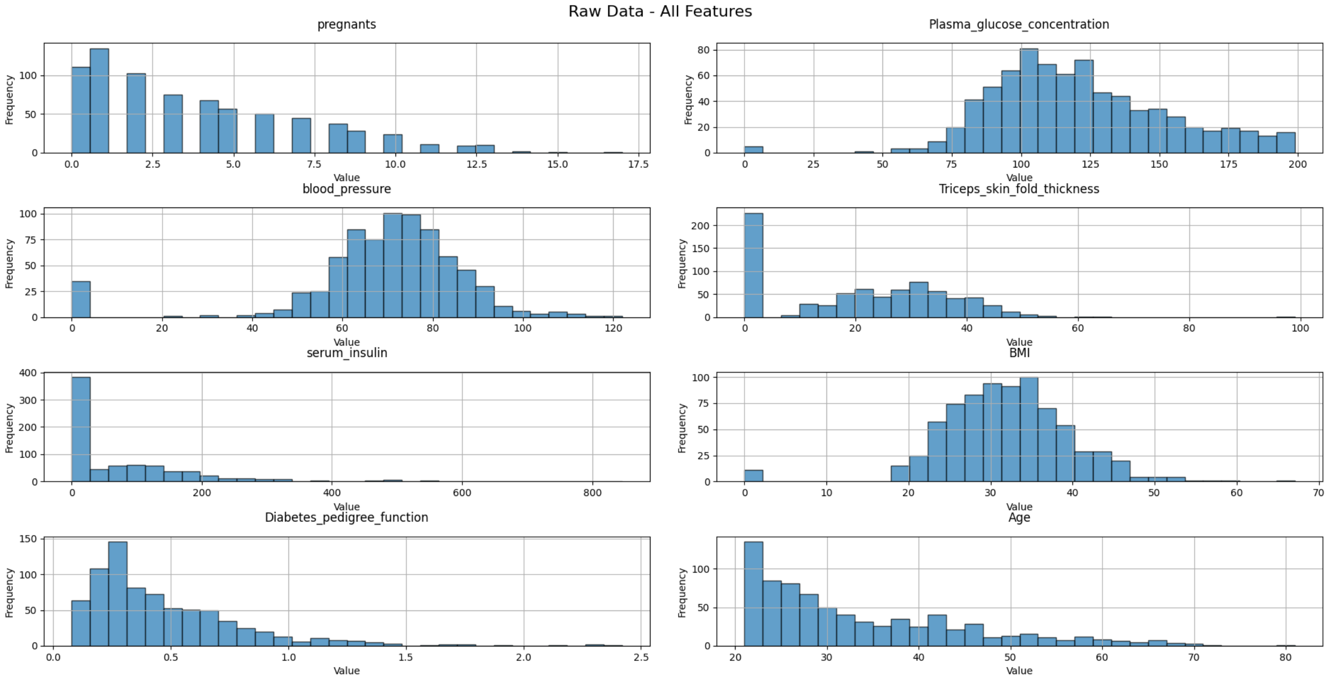


Figure 6. The histograms of the raw data, respectively, show the data distribution of the 8 features.

The following Figures 7 and 8 intuitively demonstrates the different predictive performances of the dataset under the same algorithm after applying three different data imputation methods. As can be seen from the figure, mode imputation yields the poorest results, with a test accuracy of only 75.5%. Mean imputation achieves a test accuracy of 80.5%, while median imputation outperforms the others with a test accuracy of 98.35%.

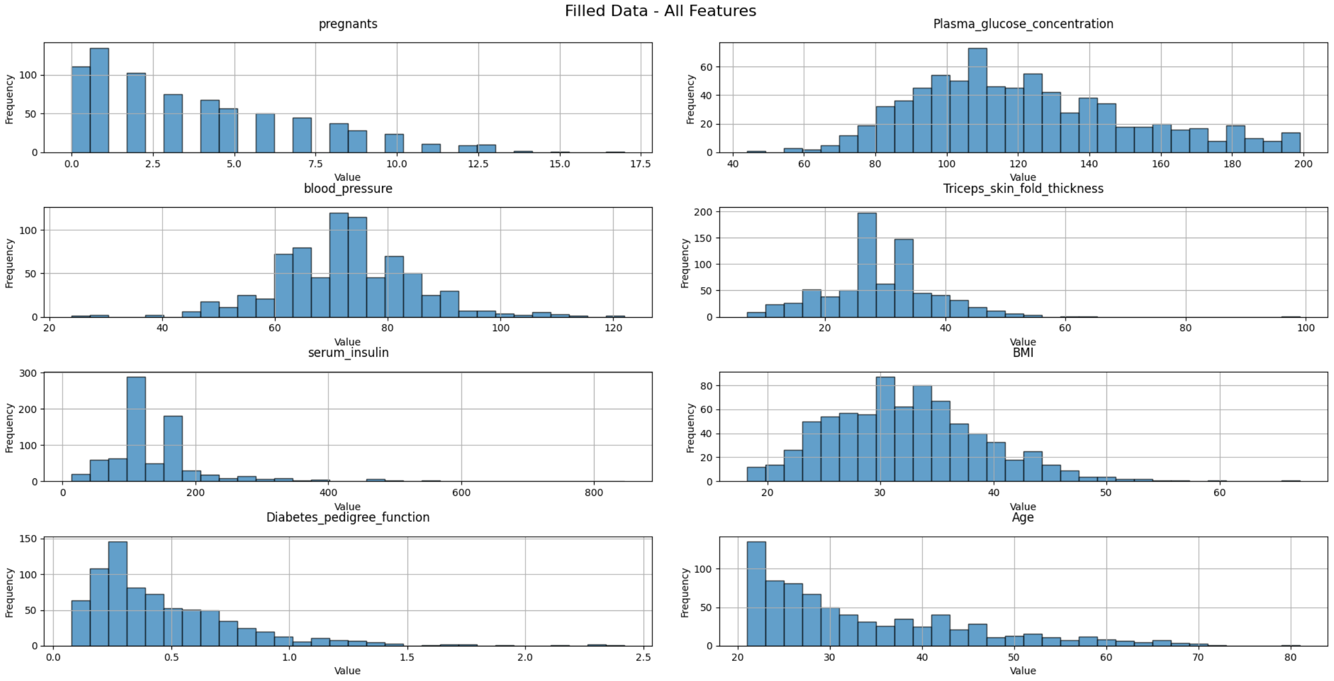


Figure 7. The histograms of the imputed data, obtained after group median imputation, can be compared with the histograms of the raw data.

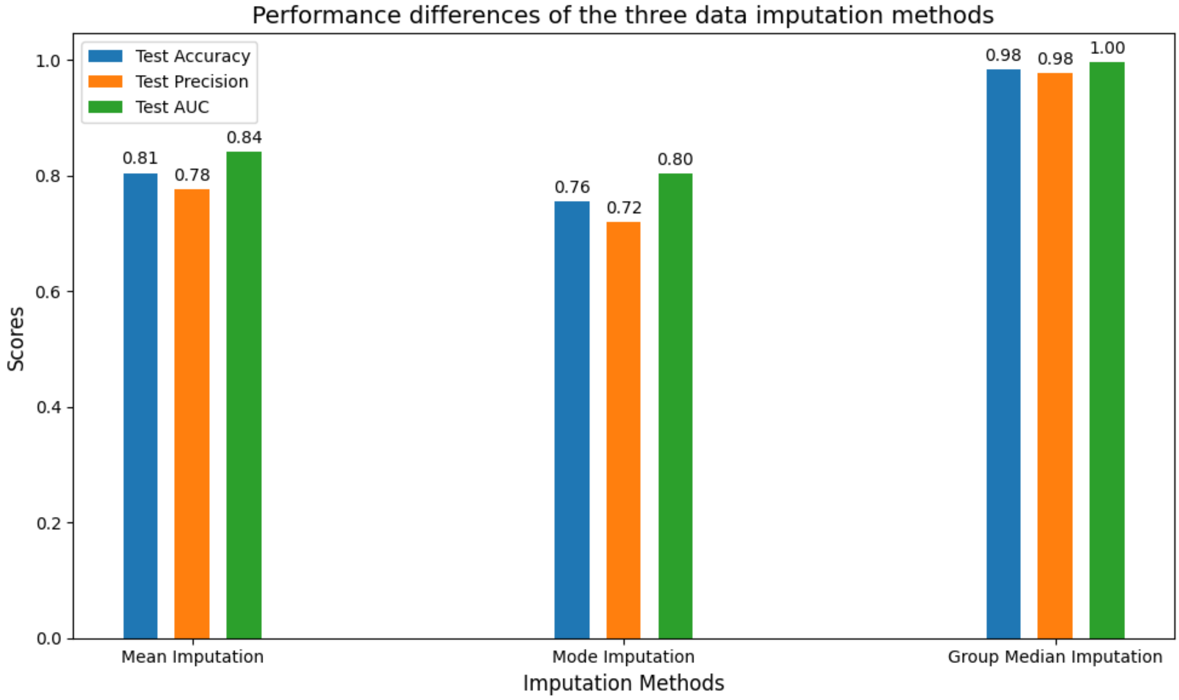


Figure 8. Performance differences in the same algorithm on the same dataset under different data imputation methods.

3.2.4. Oversampling of Data

The Pima Indians Diabetes dataset (PIDD) exhibits a significant class imbalance issue, with only 268 samples for diabetic patients and 500 samples for healthy individuals. This imbalance leads to machine learning models favoring the majority class (healthy individuals) and underperforming in identifying the minority class (diabetic patients), affecting the overall classification performance.

SMOTE is a method for addressing class imbalance by generating synthetic minority class samples [20]. It interpolates between minority class samples in the feature space to create new synthetic samples and balance the class distribution. The SMOTE algorithm finds k-nearest neighbors among minority class samples in the feature space and generates new minority class samples that lie on the lines connecting these neighbors. This approach increases data diversity and avoids overfitting issues caused by simple sample replication. In this study, we employed the SMOTE method from the imbalanced-learn library to resample the dataset. SMOTE is a straightforward technique for oversampling data. We first imported the SMOTE module in the code and then used the $fit\_resample()$ method to oversample the original data. The resampled dataset expands to 1000 samples, with 500 samples for diabetic patients and 500 samples for non-diabetic patients.

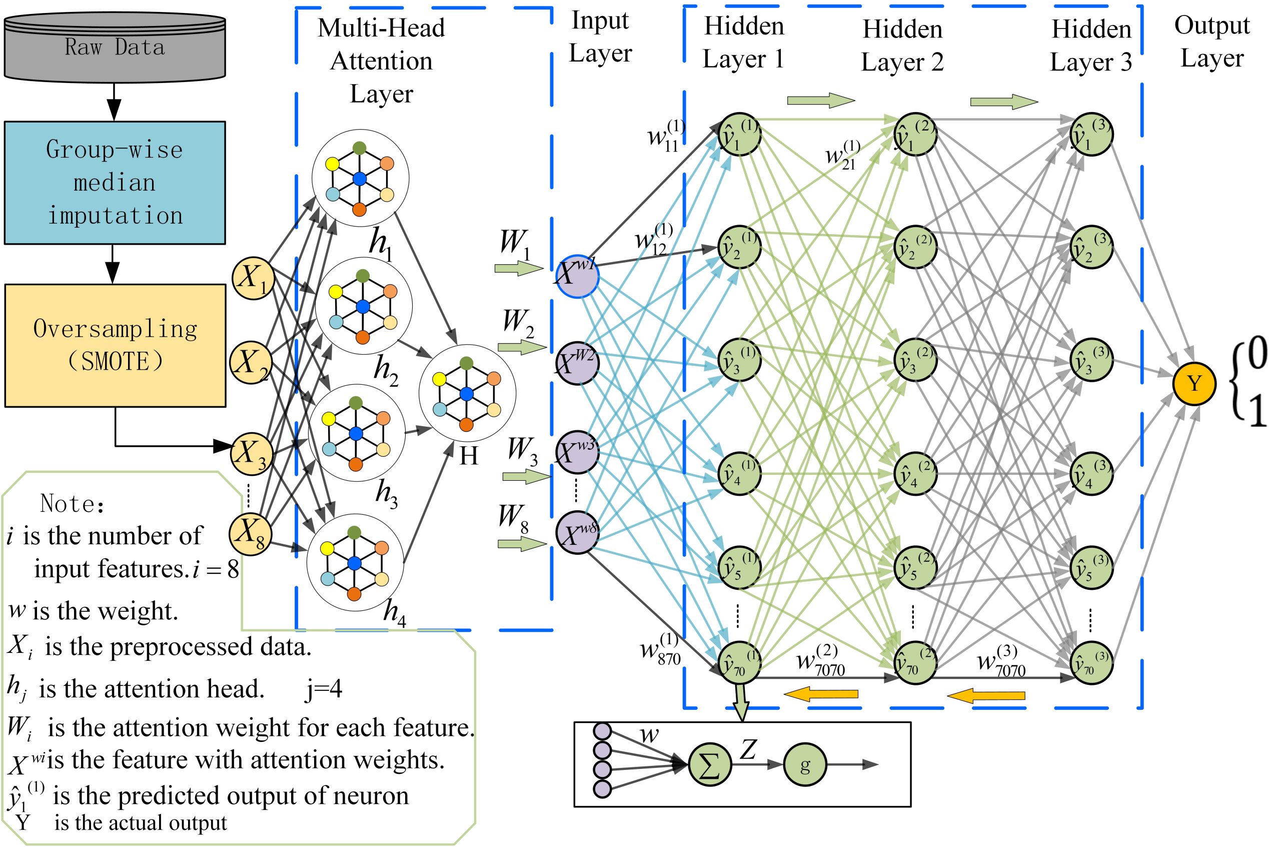
3.2.5. Data Normalization Processing

It is widely recognized that deep neural networks perform optimally when handling data with consistent scales. The Pima Indians Diabetes dataset consists of eight features, each varying in scale. As a result, feature scaling becomes essential. Two common scaling methods include normalization and standardization. Normalization adjusts feature values to a specified range, typically [0, 1] or [−1, 1], using min–max scaling. This method assigns the minimum feature value as 0, the maximum as 1, and linearly scales all other values accordingly. In this study, we utilized min–max normalization, as shown in Equation (1).

4. Model Design

In this section, we primarily elucidate the model design, which integrates the Attention-based Feature Weighting Layer with deep neural networks as shown in Figure 9.

Initially, the Attention-based Feature Weighting Layer is employed to compute the weights of each feature relative to the target outcome (diabetes diagnosis). The weighted data are then fed into the input layer of a fully connected neural network, which passes through three hidden layers and an output layer, ultimately producing a prediction on the presence of diabetes.



Note:
i is the number of input features. $i = 8$
w is the weight.
X_i is the preprocessed data.
h_j is the attention head. j=4
W_i is the attention weight for each feature.
X^{wi} is the feature with attention weights.
$\hat{y}_1^{(1)}$ is the predicted output of neuron
Y is the actual output

Figure 9. Attention-Enhanced Deep Neural Network.

4.1. Attention-Based Feature WeightingLayer

The innovation of this study lies in the training of deep neural networks guided by multi-head attention, utilizing the Pima Indians Diabetes (PIDD) dataset and the Diabetes-NHANES dataset for independent model training and validation, respectively. This approach enables the model to dynamically assign varying attention weights to different features, emphasizing those that significantly impact the prediction of diabetes outcomes.

The Attention-based Feature Weighting Layer enhances the model's representational capabilities by applying attention mechanisms to various subspaces of the input representation. It accomplishes this by using multiple attention heads to compute distinct attention representations in parallel, which are then combined to improve the model's expressiveness and generalization ability.

Attention mechanisms allow the model to dynamically allocate different attention weights to various parts of the input sequence, thereby enhancing performance in processing sequential data. By incorporating the number of attention heads as a hyperparameter in the computations, it was found that four attention heads are sufficient for processing both the large-scale Diabetes-NHANES dataset and the medium-sized Pima Indians Diabetes dataset (PIDD). Each head operates in parallel, focusing on different feature subspaces, thereby increasing the model's expressiveness and better capturing information from the input sequence. This approach enhances the model's robustness and generalization capability, as shown in Figure 10.

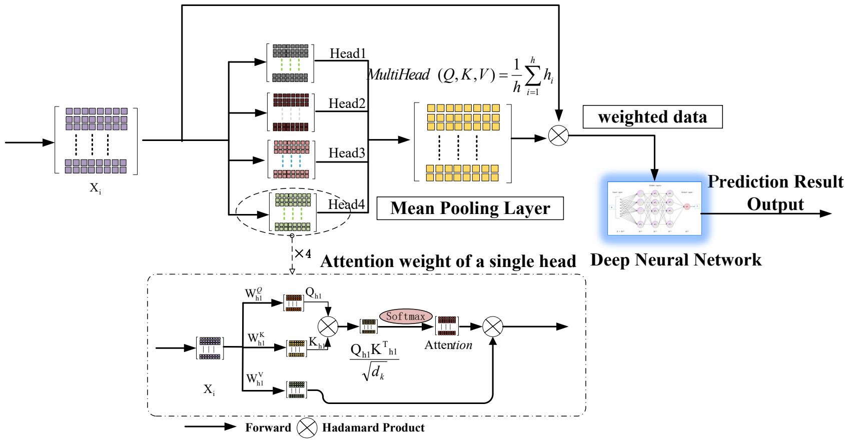


Figure 10. Attention-based Feature Weighting Layer.

1. **Single-Head Attention Mechanism**
In the single-head attention mechanism, assume the input sequence is X, which comprises n feature vectors x_i, each with a dimension of d. Given a query vector q and a set of key vectors K, we can compute the attention weights α_i and then use these weights to perform a weighted sum of the value vectors V, obtaining the final attention representation.
$$\alpha_i = \frac{\exp(q \cdot k_i)}{\sum_{j=1}^{n} \exp(q \cdot k_j)} \tag{2}$$
where k_i is the i-th key vector, and $q \cdot k_i$ denotes the dot product of the query vector q and the key vector k_i.
The final attention representation A is calculated as follows:
$$A = \sum_{i=1}^{n} \alpha_i \cdot v_i \tag{3}$$

2. **Single Attention Head Calculation**
$$\text{Head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{4}$$
where W_i^Q, W_i^K, and W_i^V are the query, key, and value weight matrices for the i-th head, respectively.
3. **Attention-Based Feature Weighting Layer**
In the Attention-based Feature Weighting Layer, we introduce multiple independent attention heads, each with its own query, key, and value weight matrices. Each head computes a set of attention weights and produces an attention representation. These multiple attention representations are then concatenated and passed through a linear transformation to obtain the final multi-head attention representation.
Assume there are h attention heads, each with a dimension of d_h. Given the input sequence X, we obtain h attention representations $A^{(1)}, A^{(2)}, \ldots, A^{(h)}$. These representations are then concatenated to form the final multi-head attention representation.
4. **Multi-Head Attention Calculation**
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Head}_1, \text{Head}_2, \ldots, \text{Head}_h)W^O \tag{5}$$

where W^O is the output weight matrix, and Concat denotes the concatenation operation.

As shown in Algorithm 1, After processing with four attention heads, we obtain an average attention weight for each feature through an averaging layer, forming a comprehensive attention weight matrix. The input data are then element-wise multiplied with this attention weight matrix, emphasizing key features, such as blood glucose levels and BMI. For less important features, multiplying by attention weights close to zero reduces their impact on the model, thus minimizing noise and enhancing model robustness. This quantification of feature importance increases the model's interpretability, allowing us to better understand how the model evaluates each feature and derive the basis for the model's classification decisions.

Algorithm 1 Multi-Head Attention Algorithm Implementation

```
Require: Input vector $X$, number of attention heads $H$, sets of query weight matrices $\{W_Q^h\}_{h=1}^H$, key weight
    matrices $\{W_K^h\}_{h=1}^H$, value weight matrices $\{W_V^h\}_{h=1}^H$, scaling factor $\sqrt{d_k}$
Ensure: Multi-head attention output
for $h = 1$ to $H$ do
    Step 1: Split the input for each attention head
        Split $X$ to obtain subvector $X_h$
    Step 2: Compute attention weights
            Compute query vector $Q_h = X_h \cdot W_Q^h$
        Compute key vector $K_h = X_h \cdot W_K^h$
    Compute value vector $V_h = X_h \cdot W_V^h$
    Step 3: Calculate attention scores
    Compute Attention Score$_h$ = softmax$\left(\frac{Q_h \cdot K_h^T}{\sqrt{d_k}}\right)$
      Step 4: Perform weighted sum
      Compute Attention$_h$ = Attention Score$_h \cdot V_h$
end for
Step 5: Concatenate multiple heads
 Concatenate Multi-head Attention = $[\text{Attention}_1, \text{Attention}_2, \ldots, \text{Attention}_H]$
return Multi-head Attention
```

After obtaining the multi-head attention weights through the aforementioned steps, the weights are processed through a Mean Pooling Layer to derive the weight for each feature. This new set of weights is then element-wise multiplied with the original data, resulting in a new dataset imbued with these attention weights. This weighted dataset is subsequently fed into a deep neural network for training, yielding the prediction results.

4.2. Deep Neural Network Layer

After processing through the Attention-based Feature Weighting Layer, a new dataset is generated. For the Pima Indians Diabetes (PIDD) dataset, this consists of 8 features and 10,000 samples, whereas for the Diabetes-NHANES dataset, it comprises 41 features and 52,390 samples. The input layer receives the data, which propagate through the hidden layers to the output layer. The hidden layers employ the ReLU activation function, while the output layer utilizes the Sigmoid function. The deep neural network begins at the input layer, performing linear transformations and activation function calculations layer by layer, propagating outputs forward. The model's loss function is calculated using cross-entropy loss, and the backpropagation algorithm is employed to determine the gradients of the loss with respect to the model's parameters. To optimize learning, the Adaptive Moment Estimation (Adam) optimizer adjusts the learning rates for each parameter individually, incorporating momentum to accelerate convergence toward the global minimum. During each epoch, the process involves forward propagation, loss computation, backpropagation, and parameter updates, iterating until the model attains optimal predictive accuracy. After the training phase, the model is tested on the evaluation set to assess its predictive performance. Throughout the entire process, backpropagation optimizes the model by minimizing the loss function, ensuring that the model fits the training data well and generalizes effectively to unseen data.

The specific implementation details are provided in Algorithm 2.

Algorithm 2 DNN Algorithm

```
1. Initialize weights $W$ and biases $b$ randomly
2. Iterate over the dataset for $n$ iterations:
    (a) Compute the linear combination of input features and weights plus bias:
            $Z = W^T \cdot X + b$    (6)
    (b) Apply the activation function $g$ to $Z$:
            $\hat{y} = g(Z)$    (7)
        (Note: $g$ can be ReLU or Sigmoid)
    (c) Apply Dropout layer after each hidden layer:
            $\hat{y}_{\text{dropout}} = \text{Dropout}(\hat{y}, p)$    (8)
        where $p$ is the probability of keeping a node
    (d) Compute the loss function $L$:
            $L(\hat{y}, Y) = -[Y \cdot \log(\hat{y}) + (1 - Y) \cdot \log(1 - \hat{y})]$    (9)
    (e) Compute the cost function $J$, including an L2 regularization term:
            $J(W, b) = -\frac{1}{m} \sum_{i=1}^{m} [Y \cdot \log(\hat{y}) + (1 - Y) \cdot \log(1 - \hat{y})] + \frac{\lambda}{2m} \sum_{j=1}^{n} W_j^2$    (10)
    (f) The learning rate adjusts the step size of gradient descent:
            $W := W - \alpha \cdot \frac{\partial J(W, b)}{\partial W}$    (11)
            $b := b - \alpha \cdot \frac{\partial J(W, b)}{\partial b}$    (12)
    (g) Compute gradients:
            $dZ = \hat{y} - Y$    (13)
            $dW = \frac{1}{m}(X \cdot dZ) + \lambda \cdot W$    (14)
            $db = \frac{1}{m} \sum_{i=1}^{m} dZ_i$    (15)
    (h) Update weights and biases using the gradients:
            $W := W - \alpha \cdot dW$    (16)
            $b := b - \alpha \cdot db$    (17)
    (i) Determine diagnosis based on prediction:
            If $\hat{y} \geq 0.5$, diagnose as "Diabetes"
            Otherwise, diagnose as "Non-Diabetic"
3. End of the algorithm
```

Note:

Input:

1. X: Input features.
2. Y: Target labels.
3. α: Learning rate.
4. θ: Weights and biases.
5. n: Number of iterations.
6. m: Number of samples.

7. $\hat{y}$: Predicted output.

   Output:

   Updated weights W and biases b.

- **Attention-Based Feature Weighting Layer**
  The multi-head attention model possesses a robust capability to learn the relationships between features. The importance of each feature is computed in parallel by multiple attention heads, each of which can focus on different aspects of the data, thus learning the contribution of each feature to the prediction outcome. Each feature is assigned four attention weights, which are then averaged through an aggregation layer. This process ensures that the weighted features more effectively highlight their impact on the prediction target while suppressing the influence of irrelevant features. The output of the Attention-based Feature Weighting Layer is represented in Equation (18):
  For the Pima Indians Diabetes (PIDD) dataset, $n = 8$, whereas for the Diabetes-NHANES dataset, $n = 41$.

$$[X'_1, X'_2, \ldots, X'_8] = [A(X_1), A(X_2), \ldots, A(X_8)] \tag{18}$$

- **Input Layer**
  The input layer receives eight features with weights $[X'_1, X'_2, \ldots, X'_8]$.
- **Hidden Laye**
  Considering the number of features and the complexity of the Pima Indians Diabetes dataset, the number of hidden layers was systematically increased using a trial-and-error method. A grid search was conducted to identify the optimal architecture, which revealed that three hidden layers offered the best performance while avoiding overfitting. The neuron count per hidden layer was also optimized through a grid search, with each layer containing 70 neurons, yielding the most favorable results.
- **Linear Connection Function (Z)**
  The linear connection function is established as shown in Equation (19), where $n = 8$ and ω represents the weight of each feature:

$$Z = \sum_{i=1}^{n} W^T \cdot X_i^{\omega} + b \quad (2) \tag{19}$$

- **Initialization**
  The weights W and biases b are initialized using the Kaiming uniform distribution, as specified in Equation (20):

$$W \sim U\left(-\sqrt{\frac{1}{\text{fan-in}}}, \sqrt{\frac{1}{\text{fan-in}}}\right) \tag{20}$$

  Given that the dataset has eight features, the initialization range for the weights is as follows:

$$W \sim U\left(-\sqrt{\frac{1}{8}}, \sqrt{\frac{1}{8}}\right) = U(-0.3536, 0.3536) \tag{21}$$

$$Z = W_0^T \cdot X_1^{\omega} + W_0^T \cdot X_2^{\omega} + W_0^T \cdot X_3^{\omega} + \cdots + W_0^T \cdot X_n^{\omega} + b \tag{22}$$

- **Activation Function (g)**
  The activation function g is applied, using ReLU for the hidden layers and Sigmoid for the output layer, to obtain the predicted output $\hat{y}$:

$$\hat{y} = g(Z) \tag{23}$$

- **Dropout Layer**
  A dropout layer is added:

$$\hat{y}_{\text{dropout}} = \text{Dropout}(\hat{y}, p) \tag{24}$$

- **Loss Function (L)**
  The loss function L is computed as follows:

$$L(\hat{y}, Y) = -[Y \cdot \log(\hat{y}) + (1 - Y) \cdot \log(1 - \hat{y})] \tag{25}$$

- **Cost Function (J) with L2 Regularization**
  The cost function $J(W, b)$, including an L2 regularization term, is computed as follows, where m is the total number of samples, Y represents the actual target vector, and λ denotes the regularization strength or penalty parameter:

$$J(W, b) = -\frac{1}{m}\sum_{i=1}^{m}[Y \cdot \log(\hat{y}) + (1 - Y) \cdot \log(1 - \hat{y})] + \frac{\lambda}{2m}\sum_{j=1}^{n} W_j^2 \tag{26}$$

- **Gradient Calculation**
  The gradients are computed as follows:

$$dZ = \hat{y} - Y \tag{27}$$

$$dW = \frac{1}{m}(X \cdot dZ) + \lambda \cdot W \tag{28}$$

$$db = \frac{1}{m}\sum_{i=1}^{m} dZ_i \tag{29}$$

- **Weight and Bias Update**
  The weights and biases are updated using gradient descent with the learning rate adjusting the step size:

$$W := W - \alpha \cdot \frac{\partial J(W, b)}{\partial W} \tag{30}$$

$$b := b - \alpha \cdot \frac{\partial J(W, b)}{\partial b} \tag{31}$$

5. Model Training and Optimization Phase

5.1. Model Training and Validation

In this study, the dataset was divided into 80% training and 20% testing sets, with a 5-fold cross-validation performed using StratifiedKFold. For each fold, the model was trained and validated, and performance metrics including the accuracy, precision, and AUC were recorded.

As illustrated in Figure 11, 80% of the PIDD data were divided into five approximately equal-sized subsets. In each iteration, one fold was selected as the test set, while the remaining data were used for training. The performance metrics were recorded for each iteration, and after five iterations, the average metrics were calculated to summarize the overall performance.

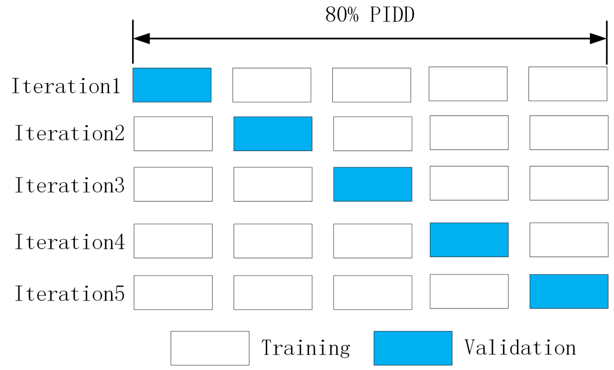


Figure 11. The 5-fold cross-validation process.

After concluding the cross-validation, the final model was trained on the complete training set and subsequently assessed on an independent test set. The model was optimized using the Adam algorithm with a learning rate of 0.0001 over 10,000 epochs. Predictions were then generated for the test set, and the key performance indicators, such as the accuracy, precision, and AUC, were calculated and documented for evaluation.

GridSearchCV was employed to identify the optimal hyperparameters, such as the number of hidden layers, neurons per layer, activation functions, optimizers, learning rates, batch sizes, and epochs. A range of values were explored to find the most effective combination. For instance, the number of hidden layers was tested from 3 to 100, and the number of neurons per layer was varied between 10 and 100. The best configuration selected consisted of three hidden layers, each containing 70 neurons.

The optimizers tested included seven types: “SGD”, “$RMSprop$”, “$Adagrad$”, “$Adadelta$”, “$Adam$”, “$Adamax$”, and “$Nadam$”. The activation functions included the following: [“$tanh$”, “$softmax$”, “$softplus$”, “$softsign$”, “$relu$”, “$sigmoid$”, “$hard_sigmoid$”, “$linear$”]. The L2 regularization parameters were set as “alpha”: [0.001, 0.01, 0.02, 0.03, 0.04, 0.05], and the learning rates (lr) were tested at [0.0001, 0.001, 0.01, 0.1].

The deep learning model was implemented using the PyTorch and TensorFlow frameworks, utilizing the Adam optimizer to reduce error during forward propagation. The network architecture consisted of three hidden layers, each with 70 neurons. ReLU was applied as the activation function for the hidden layers, while Sigmoid was utilized for the output layer. A learning rate of 0.0001 was selected, with binary cross-entropy serving as the loss function. The model was trained over 10,000 epochs. Data splitting was performed using the train_test_split method from scikit-learn, and the feature data were standardized using StandardScaler.

5.2. Model Optimization

In our model optimization process, we initially performed weight and bias initialization. To further refine the model, we employed various techniques to fine-tune the backpropagation process and select the most significant features. The specific optimization steps are as follows:

- **Hyperparameter Tuning**
  To systematically explore and optimize the model’s hyperparameters, we utilized the Keras Tuner tool and applied the RandomSearch algorithm to randomly sample within a predefined hyperparameter space. This approach allowed us to evaluate multiple potential hyperparameter combinations and determine the optimal set based on performance on the validation set.

  1. Units (number of units in the attention mechanism): This hyperparameter directly affects the effectiveness of feature weighting. Smaller values may lead to information loss, while larger values could result in overfitting. We explored a range from 1 to 16, with a step size of 1, to fine-tune this parameter. The optimal number of units was ultimately determined to be 6.

2. Num-heads (number of attention heads): Multi-head attention allows the model to capture different patterns in the data. We set the range from 1 to 8, with a step size of 1, to identify the most appropriate number of attention heads for the task at hand. The upper limit of 8 was chosen as a balance between computational complexity and model efficiency. The optimal number of attention heads was determined to be 4.
3. Hidden-units (number of neurons in the hidden layers): The number of neurons in the hidden layers directly affects the model's learning capacity. To balance model complexity and the risk of overfitting, we set the range between 32 and 256, with a step size of 32. This range allowed us to test models of varying sizes and find the most appropriate hidden layer size for learning patterns in the data.
4. Dropout-rate (dropout rate): Dropout is a common regularization technique used to prevent overfitting. We selected a range from 0.0 to 0.5, with a step size of 0.1, ensuring that we could identify the optimal dropout rate that prevents overfitting without losing too much information.
5. Learning-rate: The learning rate directly impacts the convergence speed and stability of the model. We selected common values such as 1×10^{-2}, 1×10^{-3}, and 1×10^{-4}, representing fast, moderate, and slow learning rates, respectively. This range helped us find the optimal update step for the model, avoiding issues of either too rapid or too slow convergence.
   Additionally, we designed the number of hidden layers as a tunable hyperparameter, further expanding the search space to identify the best combination of network depth configurations. This comprehensive approach ensured a balance between model complexity and performance, leading to the identification of the optimal hyperparameter set for the task.

- **Incorporation of L2 Regularization**
  To prevent overfitting, we added an L2 regularization term to the loss function. L2 regularization evaluates model complexity by adding the squared values of the weight coefficients and ensures that the weights remain constrained within a small range. The specific formula is as follows:

$$J(W,b) = -\frac{1}{m}\sum_{i=1}^{m}[Y \cdot \log(\hat{y}) + (1-Y) \cdot \log(1-\hat{y})] + \frac{\lambda}{2m}\sum_{j=1}^{n} W_j^2 \tag{32}$$

  where λ is the regularization strength hyperparameter. Through this approach, we were able to prevent overfitting while preserving the model's learning capacity. Several values for "alpha" were tested [0.001, 0.01, 0.02, 0.03, 0.04, 0.05], and the optimal parameter was determined to be 0.04.
- **Inclusion of Dropout Layers**
  Dropout layers were added after each hidden layer. Dropout is a regularization technique that randomly drops a fraction of neurons during each training iteration, preventing over-reliance on specific neurons and enhancing the model's generalization capability. This further mitigated the risk of overfitting.
- **Optimization Algorithm**
  We employed the Adam optimization algorithm for model training. Adam is an adaptive learning rate method that computes individual adaptive learning rates for each parameter, accelerating convergence and improving model performance. By dynamically adjusting the learning rate during training, Adam enables the model to reach an optimal state more quickly and stably.

Through these optimization strategies, we effectively adjusted the model's hyperparameters, prevented overfitting, and improved the model's generalization ability on the test set. These efforts resulted in a model that performs better on complex tasks, with higher accuracy and robustness. The final model achieved an impressive prediction accuracy of 98.2% on the Pima Indians Diabetes dataset (PIDD). However, given that the PIDD is a moderately small-scale dataset, it may not fully demonstrate the robust performance of the AEDNN model. To further evaluate its capabilities, we conducted validation on a large-scale dataset, the Diabetes-NHANES dataset. The results revealed that the model's performance was significantly enhanced when applied to this large-scale dataset, achieving an outstanding accuracy of 99.82%, representing a 1.65% improvement compared to the performance on the PIDD. This finding indicates that the increase in data volume has a substantial impact on enhancing the learning capabilities of the model, making it particularly well suited for large-scale diabetes prediction tasks.

6. Comparative Experiments

To validate the effectiveness of our proposed methods, we performed two comparative experiments.

6.1. Experiment 1: Impact of Attention-Based Feature Weighting Layer

We compared the model's performance with and without the Attention-based Feature Weighting Layer. We used the same dataset and preprocessing methods for both models to ensure a fair comparison. The baseline model was without the Attention-based Feature Weighting Layer and used traditional feature selection methods. The proposed model incorporated the Attention-based Feature Weighting Layer as described in Section 4.

The results of the comparative experiment are summarized in Table 2. The baseline model achieved an accuracy of 87%, while the model with the Attention-based Feature Weighting Layer achieved an accuracy of 98%. The ROC curve of the baseline model (without attention) is shown in Figure 12.

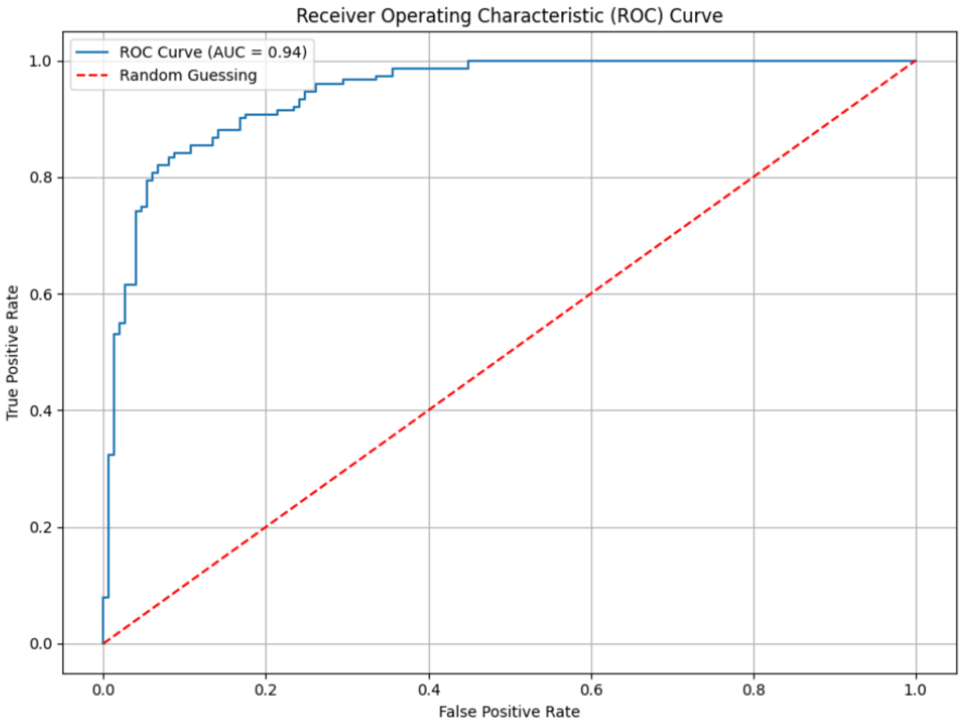


Figure 12. The ROC curve of the baseline model (without attention).

Table 2. Comparison of the model's performance with and without the Attention-based Feature Weighting Layer.

Model	Accuracy	Precision	AUC
Baseline model (without attention)	0.87	0.90	0.94
Proposed model (with attention)	0.98	0.97	0.99

The inclusion of the Attention-based Feature Weighting Layer led to a 12.6% improvement in prediction accuracy, demonstrating its effectiveness in capturing complex relationships within physical examination data. This significant increase provides strong evidence that the Attention-based Feature Weighting Layer is crucial for enhancing the accuracy of diabetes prediction. The dynamic weighting and adaptive dimension adjustment capabilities of the attention mechanism enable the model to better utilize the available data, resulting in more accurate predictions. These findings highlight the substantial performance gains achieved by incorporating the Attention-based Feature Weighting Layer, underscoring its value in predictive modeling tasks involving complex and multi-dimensional data.

6.2. Experiment 2: Attention Mechanism-Based Feature Extraction Versus Lasso and Ridge for Feature Selection

When employing attention mechanisms for feature selection, a multi-dimensional comparative analysis was conducted against the classical feature selection methods, Lasso and Ridge. The comparative experiments were carried out on the same dataset, the Diabetes-NHANES dataset, utilizing a variety of evaluation metrics, including the accuracy, precision, recall, F1 score, mean squared error (MSE), and area under the curve (AUC), as detailed in Table 3.

Table 3. Performance Comparison of Feature Extraction Using Attention Weights, Lasso, and Ridge.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	MSE	AUC
AEDNN with Attention	99.82	99.80	99.86	99.81	0.0018	0.9995
Lasso Feature Selection	94.05	94.39	93.67	94.03	0.0595	0.9571
Ridge Feature Selection	94.34	94.40	94.27	94.34	0.0566	0.9742

The AEDNN model achieves a test accuracy of 99.82%, significantly higher than the results obtained using Lasso (94.05%) and Ridge (94.34%). This indicates that the attention-based deep learning model can capture more relevant and nuanced information from the input features, providing superior prediction capabilities. The AEDNN's precision and recall scores (both 99.8%) vastly exceed those of the Lasso (precision 94.39% and recall 93.67%) and Ridge (precision 94.40% and recall 94.27%) models. This difference shows the model's ability to better balance between true positives and false positives. Lasso and Ridge, which rely on linear regularization techniques, may fail to capture the complexity of nonlinear relationships between features and the target variable (diabetes). The AEDNN has a near-perfect F1 score of 99.81%, whereas Lasso and Ridge achieve around 94%. The F1 score indicates the balance between precision and recall, and the AEDNN's superior performance suggests it excels at distinguishing between diabetic and non-diabetic cases. The AEDNN model significantly reduces the MSE to 0.0018 compared to Lasso (0.0595) and Ridge (0.0566), indicating its superior prediction accuracy. The R2 score of the AEDNN (0.9907) also far exceeds that of Lasso (0.7621) and Ridge (0.7737), further proving the AEDNN's strong capability to explain the variance in the data. The AEDNN achieves a near-perfect AUC of 0.9995, demonstrating its excellent capability in distinguishing between classes. In contrast, Lasso and Ridge achieve 0.9571 and 0.9742, respectively, which, while respectable, fall short of the AEDNN's near-perfect performance.

The attention mechanism plays a critical role in identifying and prioritizing the most relevant features for prediction, dynamically adjusting its focus on important parts of the input data. This contrasts with Lasso and Ridge, which are based on predefined assumptions about feature importance through linear regularization. Lasso and Ridge are linear methods that assign importance to features by applying regularization penalties, which may eliminate or shrink less important features. However, these methods can struggle with complex, nonlinear relationships between features, which are common in medical data, like diabetes prediction. In contrast, attention mechanisms allow the model to focus dynamically on different features depending on the input context. This leads to better handling of data complexity, which is reflected in the AEDNN's improved performance metrics.

Diabetes prediction often involves intricate interactions between features (e.g., age, BMI, and glucose levels). Lasso and Ridge may oversimplify these relationships due to their linear nature, while the attention mechanism can capture higher-order interactions between features. This capability allows the AEDNN to perform better on both small (Pima Indians Diabetes) and large (NHANES) datasets. The attention mechanism has the advantage of flexibly adapting to imbalanced data scenarios, such as in medical diagnosis where diabetic cases may be fewer than non-diabetic ones. This is especially visible in the recall scores of the AEDNN, which captures positive cases with a much higher accuracy. Lasso and Ridge, being more sensitive to imbalanced data, often struggle to prioritize minority class samples effectively, leading to lower recall and F1 scores.

The attention mechanism allows for better generalization by focusing on features that are most significant across a wide variety of samples rather than applying uniform regularization, like Lasso or Ridge. This leads to the superior performance of the AEDNN model, especially in complex datasets, such as the NHANES. In conclusion, the attention mechanism not only provides significant improvement in feature selection compared to Lasso and Ridge but also demonstrates clear advantages in terms of capturing complex relationships, handling imbalanced data, and achieving higher performance metrics in diabetes prediction tasks.

6.3. Experiment 3: Impact of Data Preprocessing

To demonstrate the importance of data preprocessing, we conducted a comparative experiment using the same predictive model with and without data preprocessing. The data preprocessing steps included anomaly detection, median imputation for missing values, and SMOTE for class balancing.

For this experiment, we constructed two versions of the dataset:

1. Raw Data: The original dataset without any preprocessing.
2. Preprocessed Data: The dataset after applying the aforementioned preprocessing steps.

Both datasets were then fed into the same predictive model incorporating the Attention-based Feature Weighting Layer and deep learning architecture. The performance of the model on both datasets was evaluated and compared. The results are summarized in Table 4. The ROC curve of the same model using raw data is shown in Figure 13.

The comparative experiment highlights the crucial role of data preprocessing in enhancing model performance. As indicated in Table 4, the model trained on raw data yielded an accuracy of 72%, precision of 59%, and an AUC of 74%. Conversely, the model trained on preprocessed data achieved substantially better results, with an accuracy of 98%, precision of 97%, and an AUC of 99%. These improvements—26% in accuracy, 38% in precision, and 25% in AUC—demonstrate the substantial performance gains attributed to preprocessing. Addressing data anomalies, handling missing values, and correcting class imbalances during preprocessing ensures data quality and consistency, which ultimately enables more effective model learning and enhances predictive accuracy. This underscores the necessity of data preprocessing as a fundamental step in optimizing the performance of predictive models.

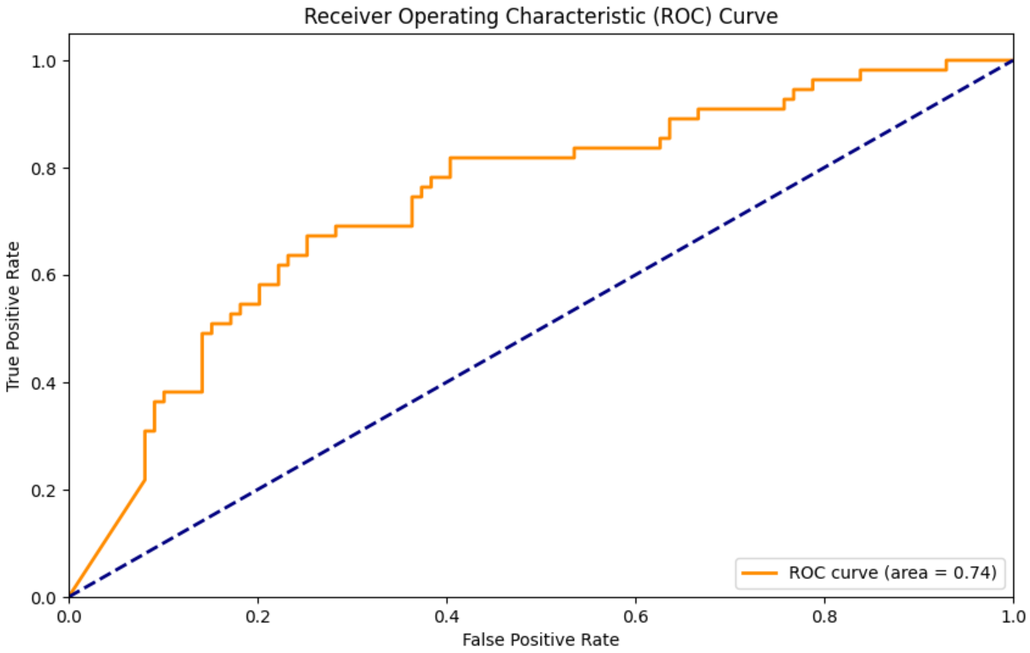


Figure 13. The ROC curve of the same model using raw data.

Table 4. The performance comparison of the same model on raw data and preprocessed data.

Dataset	Accuracy	Precision	AUC
Raw Data	0.72	0.59	0.74
Preprocessed Data	0.98	0.97	0.99

6.4. Experiment 4: Comparative Analysis of AEDNN with Several Models

We selected models including L1-regularized logistic regression, support vector machine (SVM), random forest, K-nearest neighbors (KNNs), AdaBoost, XGBoost, and the latest semi-supervised XGBoost approach [21]. These methods were independently tested on both the Pima Indians Diabetes dataset and the Diabetes-NHANES dataset. For data preprocessing, a consistent strategy was applied across all models. Specifically, for each feature, the median value of both the diabetic and non-diabetic groups was computed, which was then used to impute outliers and missing values. Additionally, oversampling and data normalization techniques were employed uniformly, resulting in two fully harmonized datasets. This preprocessing ensured that any differences in model performance could not be attributed to variations in data handling. The models were subsequently evaluated in terms of the accuracy, precision, recall, F1 score, and training time, with the results presented in Tables 5 and 6.

Table 5. Performance comparison of different models on the Diabetes-NHANES dataset.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Training Time (s)
Logistic Regression with F1 Regularization	94.48	94.52	94.54	94.53	28.08
SVM	95.22	94.50	96.12	95.30	46.10
Random Forest	92.77	91.22	73.07	81.14	17.98
KNN	93.07	89.89	97.19	93.40	0.01
AdaBoost	94.00	92.00	96.00	94.98	0.34
XGBoost	93.98	92.00	94.95	93.98	0.34
Semi-Supervised XGBoost	94.94	93.00	95.88	94.94	1.14
AEDNN	99.82	99.80	99.86	99.81	900.00

Table 6. Performance comparison of different models on the Pima Indians Diabetes dataset.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Training Time (s)
Logistic Regression with F1 Regularization	80.00	77.98	84.16	80.95	0.00
SVM	84.50	82.41	88.12	85.17	0.01
Random Forest	89.00	88.35	90.10	89.22	0.12
KNN	83.00	77.69	93.07	84.68	0.00
AdaBoost	87.50	86.54	89.11	87.80	0.07
XGBoost	90.50	91.84	89.11	90.45	0.07
Semi-Supervised XGBoost	86.50	87.76	85.15	86.43	0.16
AEDNN	98.30	97.80	98.90	98.00	270.00

Our AEDNN model achieved a remarkable accuracy of 99.82% on the Diabetes-NHANES dataset, significantly outperforming all the other models. The support vector machine (SVM) achieved an accuracy of 95.22%, the closest performance among the traditional machine learning models. Logistic regression and XGBoost demonstrated relatively stable performance with accuracies of 94.48% and 93.98%, respectively, though still lower than that of the AEDNN model. Random forest and KNN had accuracies of 92.77% and 93.07%, respectively, showing moderate performance. On the Pima Indians Diabetes dataset, the AEDNN model achieved an accuracy of 98.30%, markedly higher than the other models. XGBoost had the best performance among the traditional models, with an accuracy of 90.50%. Random forest and SVM achieved accuracies of 89% and 84.50%, respectively, with SVM underperforming compared to the AEDNN. Logistic regression had the lowest accuracy at only 80.00%, significantly below that of the AEDNN.

In terms of precision, on the Diabetes-NHANES dataset, the AEDNN model achieved the best result with a precision of 99.80%, followed by SVM with 94.50%. Other models, such as logistic regression and XGBoost, had precision scores in the 92–94% range, showing decent performance. Random forest achieved a precision of 91.22%, indicating more moderate results. On the Pima Indians Diabetes dataset, the AEDNN model's precision was 97.80%, significantly outperforming other models. XGBoost had a precision of 91.84%, showing good performance, while random forest and SVM achieved 88.35% and 82.41%, respectively. Logistic regression had the lowest precision at 77.98%, substantially below that of the AEDNN.

Regarding recall, the AEDNN model achieved an outstanding recall of 99.86% on the Diabetes-NHANES dataset, while SVM achieved 96.12%, the best performance among the traditional models. KNN had a recall of 97.19%, performing well in capturing positive samples, although its F1 score was lower than the AEDNN. Logistic regression and XGBoost both achieved recall rates around 94-95%, while random forest had a much lower recall of 73.07%, far below the other models. On the Pima Indians Diabetes dataset, the AEDNN model had a recall of 98.90%, significantly higher than the other models. XGBoost and SVM achieved recall rates of 89.11% and 88.12%, respectively, showing the best performance among the traditional models. Random forest and logistic regression had recall rates of 90.10% and 84.16%, respectively, with logistic regression underperforming in its ability to capture positive samples.

In terms of the F1 score, the AEDNN model achieved an exceptional score of 99.81% on the Diabetes-NHANES dataset, approaching a near-perfect model. SVM and KNN had F1 scores of 95.30% and 93.40%, respectively, demonstrating good performance. XGBoost achieved an F1 score of 93.98%, also at a high level, while logistic regression had an F1 score of 94.53%, showing decent performance but still with a substantial gap compared to the AEDNN. Random forest, however, had a much lower F1 score of only 81.14%, significantly worse than the other models. On the Pima Indians Diabetes dataset, the AEDNN model achieved an F1 score of 98.00%, showing excellent performance. XGBoost's F1 score was 90.45%, the best among the traditional models, while random forest and SVM achieved F1 scores of 89.22% and 85.17%, respectively, also demonstrating good performance. Logistic regression had an F1 score of only 80.95%, notably lower, particularly when handling relatively imbalanced data, where it performed worse than the AEDNN model.

In terms of the training time, the AEDNN model required 15 min on the Diabetes-NHANES dataset, significantly longer than the traditional machine learning models, as the deep learning architecture requires processing complex network structures and large-scale data. On the Pima Indians Diabetes dataset, the training time was 4.5 min, which, although slightly longer than the other models, was relatively reasonable considering its performance advantages. XGBoost's training time was 0.07 s, much shorter than the AEDNN, though it did not match the AEDNN's performance. Random forest and SVM had shorter training times of 0.12 s and 0.01 s, respectively, but their accuracy and F1 scores were inferior to the AEDNN.

Overall, the AEDNN model significantly outperformed other models in terms of accuracy, precision, recall, and F1 score, particularly on large-scale and complex datasets such as Diabetes-NHANES, where it demonstrated exceptional generalization and classification capabilities. Its near-perfect F1 score indicates its ability to balance the classification of both positive and negative samples. The AEDNN model, by leveraging deep learning structures, can capture complex patterns and nonlinear relationships within the data, a capability particularly evident on relatively small and complex datasets, like the Pima Indians Diabetes dataset. In contrast, linear models like logistic regression performed poorly in capturing nonlinear features. The high recall and precision of the AEDNN model show its strong performance in handling classification tasks across different class distributions, especially in imbalanced datasets like the PIDD, where it effectively captures minority classes. This also highlights its potential for real-world applications in medical diagnosis tasks. The AEDNN model also performed excellently in terms of MSE and R2 scores, demonstrating its ability to fit the training data well while avoiding both overfitting and underfitting issues. Although the AEDNN's training time is longer than that of traditional models, it demonstrates significant advantages when dealing with complex data and features. For tasks requiring high accuracy and stability, the trade-off in training time is entirely justified.

7. Model Testing and Evaluation

Following the comparative experiments, we performed comprehensive testing and evaluation of the model using the preprocessed data and the attention-based feature weighting mechanism. The results are discussed in detail in this section.

7.1. Model Testing

The dataset was initially loaded from a CSV file, with the features and target variables extracted separately. To ensure consistency in feature scaling, StandardScaler was applied, followed by an 80%–20% split of the standardized data into training/validation and test sets. Oversampling techniques were utilized to balance the dataset, resulting in a total of 1000 samples. Of this, 20% (200 samples) was reserved as an independent test set to rigorously assess the generalization performance of the model.

7.2. Model Evaluation

During model evaluation, key performance metrics such as the accuracy, precision, recall, F1 score, mean squared error (MSE), and ROC-AUC were used to assess both the classification and regression aspects of the model's performance. Accuracy and precision reflect the model's correctness in identifying diabetic patients, while the F1 score balances precision and recall. The MSE quantifies the average squared deviation between predicted and actual values, providing insight into prediction accuracy. To ensure robustness, cross-validation and a learning curve analysis were conducted, further supporting the model's generalization capability across different datasets. The metrics accuracy, precision, recall, F1 score, and MSE were calculated using the following formulas:

Accuracy: Reflects the proportion of correctly classified samples out of the total number of samples, indicating the model's correctness in classifying patients as diabetic or non-diabetic.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{33}$$

Precision: Reflects the proportion of samples predicted by the model as diabetic patients that are indeed diabetic.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{34}$$

F1 Score: The weighted harmonic mean of the precision and recall, providing a balance between the two metrics.

$$\text{F1} = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \tag{35}$$

Mean Squared Error (MSE): The average squared difference between the predicted values and the actual values.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{36}$$

Here, y_i denotes the actual value of the i-th observation, $\hat{y}_i$ denotes the predicted value of the i-th observation, and n represents the number of samples.

8. Results

To evaluate the generalization performance of the model on both the Diabetes-NHANES dataset and the Pima Indians Diabetes (PIDD) dataset, a five-fold cross-validation was conducted. For the PIDD, the accuracy scores obtained were [0.9795, 0.9845, 0.981, 0.985, 0.982], resulting in an average accuracy of 98.2%. For the Diabetes-NHANES dataset, the accuracy scores were [0.9982,0.9982,0.9980,0.9980,0.9986], yielding an average accuracy of 99.82%, as shown in Figure 14.

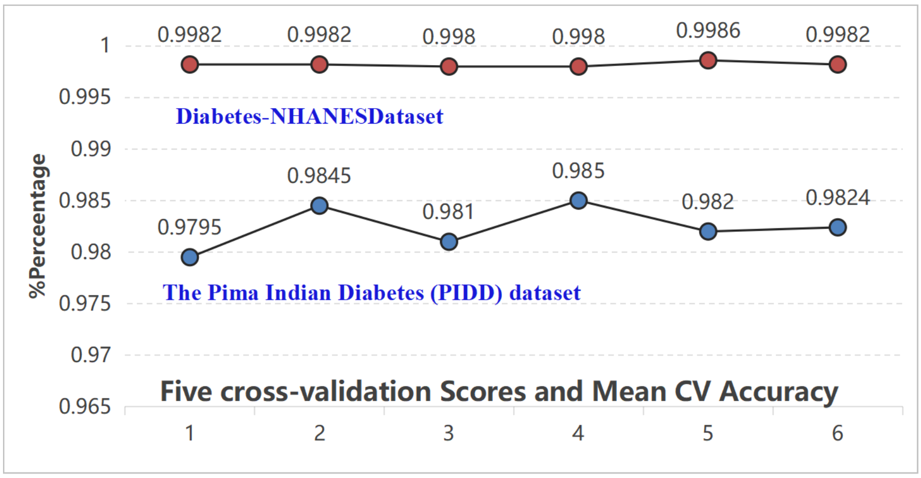


Figure 14. Five cross-validation scores and mean CV scores.

As shown in Figure 15, the proposed model achieved a test accuracy of 98.4% on the Pima Indians Diabetes dataset (PIDD), with a precision of 98%, a recall of 99%, and an F1 score of 98%. Additionally, the R-squared score reached 93.4%, and the mean squared error (MSE) was as low as 1.65%. On the larger-scale Diabetes-NHANES dataset, the model demonstrated an accuracy of 99.82%, highlighting the minimal deviation between the predicted and actual values. These results indicate that the model performs better on large-scale datasets, with an improvement of 1.65% in prediction accuracy compared to the PIDD. As further confirmed by Figure 16, the model's superior classification performance is evident, reinforcing its robustness and effectiveness in accurately predicting diabetes outcomes.

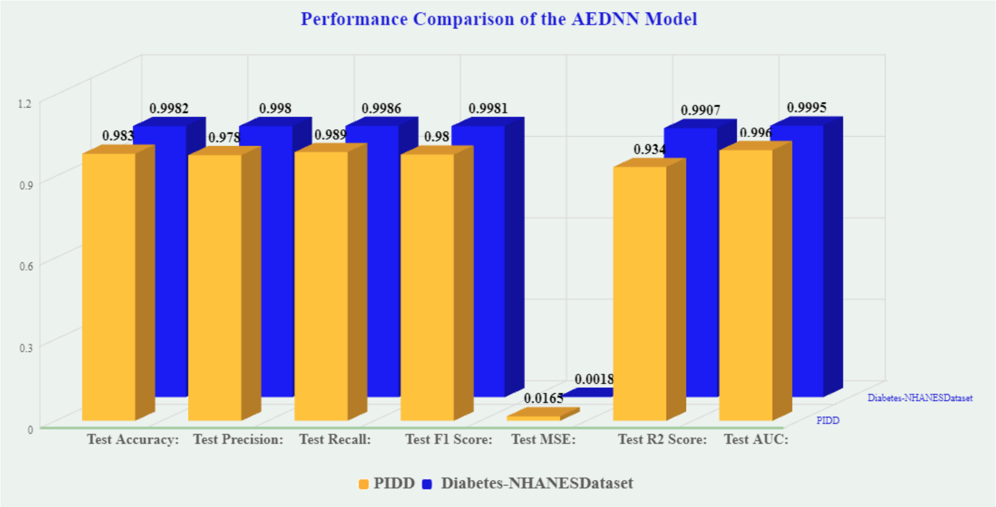


Figure 15. Performance comparison of the AEDNN model on the PIDD and the Diabetes-NHANES dataset.

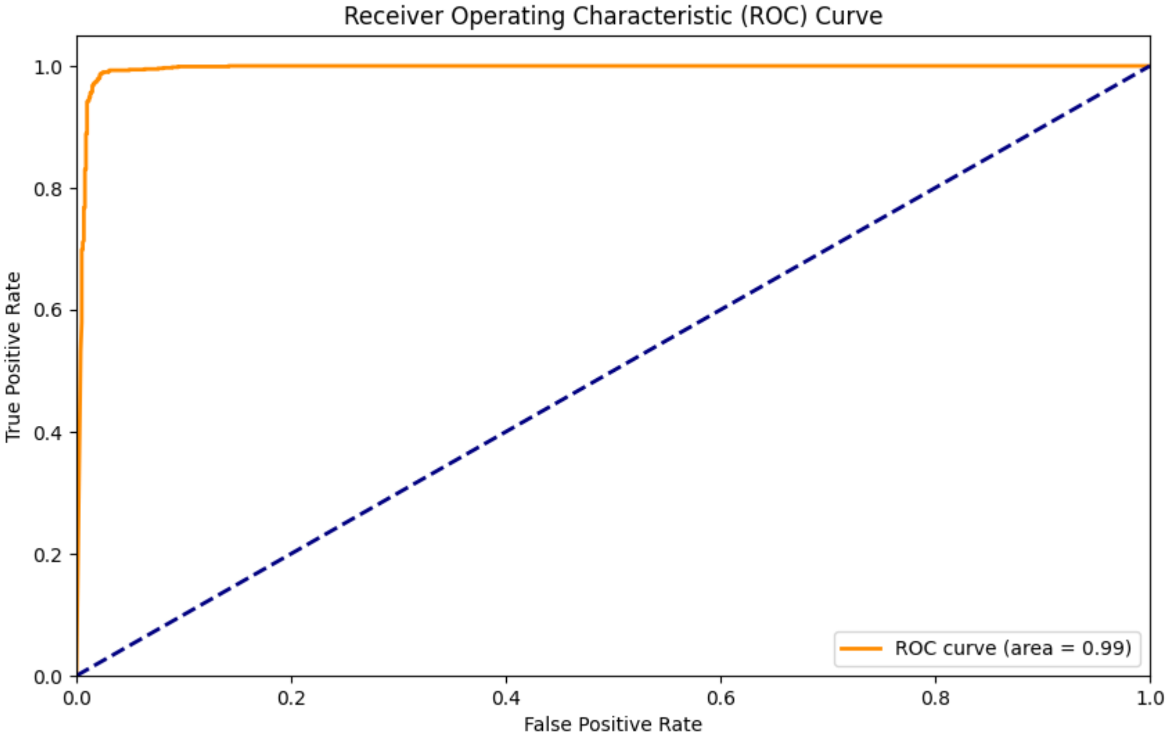


Figure 16. The ROC curve and AUC reached 0.99.

9. Discussion

Table 7 presents the datasets and data preprocessing methods employed by various diabetes prediction algorithms, along with the accuracy of each study. It also provides a comparative analysis between our proposed Attention-Enhanced Deep Neural Network algorithm and other algorithms. The results indicate that our algorithm achieves higher accuracy.

The Attention-Enhanced Deep Neural Network (AEDNN-DP) proposed in this study stands out by innovatively incorporating an Attention-based Feature Weighting Layer. This allows for the effective extraction of feature weights, leading to more accurate and efficient diabetes prediction. Especially for high-dimensional and large-scale samples, the introduction of the Attention-based Feature Weighting Layer effectively reduces dimensionality and removes redundant features. This is particularly evident in the testing of the Diabetes-NHANES dataset, where the large-scale data better demonstrated the performance of the AEDNN model, with a 1.65% improvement in performance compared to the PIDD. This suggests that the attention weights played a critical role during the model's operation, further supporting the advantages of incorporating attention mechanisms, as highlighted in Experiment 1 (Section 6.1).

Our approach also demonstrates significant advantages in data preprocessing. In binary classification problems, issues such as missing data and data imbalance are common. Considering the substantial differences in various indicators between the negative and positive cases, it is insufficient to impute missing values using simple means or modes. Instead, we should consider the median of each feature separately for negative and positive groups. The grouped median imputation algorithm effectively handles missing and outlier values. In addition, oversampling techniques can be employed to balance the two classes, thus helping the model improve learning. The experimental results indicate that the AEDNN-DP significantly enhances performance in diabetes prediction tasks.

Table 7. Comparison of current diabetes prediction models with the proposed model.

Study	Year	Dataset	Algorithm	Data Preprocessing Technique	Accuracy
[21]	2024	private dataset	Semi-supervised approach combined with XGBoost	Oversampling Technique (SMOTE)	97.4%
[11]	2023	PIDD	iDP framework integrating various ML techniques	Mean replacement for missing values	95.26%
[22]	2023	PIDD	CNN, DNN, and MLP	Mean replacement for missing values	98.1%
[23]	2022	PIDD	SVM, LR, ANN, etc.	Mean replacement for missing values	81%
[24]	2021	PIDD	DT, NN, KNN, etc.	Mean and Pearson correlation analysis	88.6%
[25]	2021	PIDD	VAE, SAE, MLP, CNN	Normalization and augmentation	92.31%
The Proposed Model	2024	PIDD	Attention-Enhanced DNN	Median imputation, SMOTE, and normalization	98.4%
The Proposed Model	2024	Diabetes-NHANES database	Attention-Enhanced DNN	Median imputation, SMOTE, and normalization	99.8%

10. Conclusions

In this study, we conducted effective research on the early screening and prediction of diabetes. We proposed a novel technique, the Attention-Enhanced Deep Neural Network (AEDNN), which introduces the Attention-Based Feature Weighting Layer. This mechanism creates a weight matrix based on the influence of each feature on the outcome, multiplying each element by its respective weight to enhance or filter specific aspects of the data. The weighted data are then input into a deep neural network for learning.

For hyperparameter optimization, we employed the Keras Tuner tool in combination with the RandomSearch algorithm, systematically exploring key parameters within a predefined hyperparameter space. These parameters included the number of units in the attention mechanism, the number of attention heads, the number of neurons in the hidden layers, the dropout rate, and the learning rate. After a comprehensive search, the optimal configuration was identified to balance model complexity and performance. Additionally, we extended the search space by adjusting the number of hidden layers, ensuring that the best configuration was found across various network depths. To prevent overfitting, we introduced a dropout layer and incorporated an L2 regularization term in the loss function.

The model achieved an overall accuracy of 98.4% and a precision of 98% on the PIDD and an average accuracy of 99.82% and a precision of 99.8% in the Diabetes-NHANES dataset. As the size of the dataset increased, the model performance improved significantly, indicating robust generalizability. Particularly on large-scale datasets like Diabetes-NHANES, the model achieved near-optimal performance with an accuracy of 99.82% and an AUC of 0.9995. This demonstrates that the model continues to exhibit outstanding performance even when exposed to more diverse samples, further highlighting its robustness and ability to adapt to a broader range of data distributions. Early screening and prediction of diabetes are crucial for timely treatment, aiding doctors with diagnostic tools. This model can also handle structured biomedical data for other diseases, showing great potential in high-dimensional, multi-feature, large-scale data scenarios.

Author Contributions: Conceptualization, H.H. and W.D.; methodology, H.H. and J.Y.; software, H.H. and W.D.; validation, H.H., W.D. and J.Y.; formal analysis, H.H.; investigation, H.H. and W.D.; resources, J.Y.; data curation, H.H.; writing—original draft preparation, H.H.; writing—review and editing, W.D. and J.Y.; visualization, S.G. and X.Z.; supervision, J.Y. and X.Z.; project administration, J.Y.; funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the 2023 Annual Municipal Basic Research Program (Joint Special Fund) grant number HABL2023025. The APC (Article Processing Charge) was funded by the same grant.

Institutional Review Board Statement: Ethical review and approval were not required for this study because publicly available datasets were used. The data used in this study are from the Pima Indians Diabetes dataset, which is openly accessible and de-identified.

Informed Consent Statement: Not applicable. This study utilized the publicly available Pima Indians Diabetes dataset, which consists of de-identified data, and did not involve direct interaction with human participants.

Data Availability Statement: The data supporting the reported results can be found in the publicly available Pima Indians Diabetes dataset. This dataset is accessible via the UCI Machine Learning Repository at https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes (accessed on 17 October 2024). The Diabetes-NHANESDataset is publicly available at https://github.com/hongweihaha/Diabetes-NHANESDataset.git (accessed on 17 October 2024).

Acknowledgments: The authors would like to thank Wenbo Dong for his contributions to the conceptualization and methodology of this research. Jianming Yu is acknowledged for providing essential resources and for his role in validation, supervision, project administration, and funding acquisition. Shiyan Guan and Xiaofei Zhu are thanked for their efforts in visualization.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

PIDD	Pima Indian Diabetes Dataset
CNN	Convolutional Neural Network
SVM	Support Vector Machine
LR	Logistic Regression

ANN	Artificial Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
DNN	Deep Neural Network
DT	Decision Tree
KNN	K Nearest Neighbors
RF	Random Forest
NB	Naive Bayes
AB	AdaBoost
VAE	Variational Autoencoder
SAE	Sparse Autoencoder
MLP	Multilayer Perceptron
XGBoost	eXtreme Gradient Boosting
SMOTE	Synthetic Minority Oversampling Technique
iDP	Intelligent Diabetes Prediction

References

1. Gromova, L.V.; Fetissov, S.O.; Gruzdkov, A.A. Mechanisms of glucose absorption in the small intestine in health and metabolic diseases and their role in appetite regulation. *Nutrients* **2021**, *13*, 2474. [CrossRef] [PubMed]
2. Jiang, L.; Yang, Z.; Wang, D.; Gong, H.; Li, J.; Wang, J.; Wang, L. Diabetes prediction model for unbalanced community follow-up data set based on optimal feature selection and scorecard. *Digit. Health* **2024**, *10*, 20552076241236370. [CrossRef]
3. Statista. Chart: Where Diabetes Burdens Are Rising. Available online: https://www.statista.com/chart/23491/share-of-adults-with-diabetes-world-region/ (accessed on 24 May 2019).
4. World Health Organization (WHO). Global Action Plan for the Prevention and Control of Noncommunicable Diseases: 2013–2020. 2013. Available online: http://apps.who.int/iris/bitstream/10665/94384/1/9789241506236_eng.pdf (accessed on 26 July 2019).
5. International Diabetes Federation (IDF). *IDF Diabetes Atlas*, 10th ed.; IDF: Brussels, Belgium, 2021. Available online: https://diabetesatlas.org/atlas/tenth-edition/ (accessed on 14 October 2019).
6. Ahmed, B.M.; Ali, M.E.; Masud, M.M.; Naznin, M. Recent trends and techniques of blood glucose level prediction for diabetes control. *Smart Health* **2024**, *32*, 100457. [CrossRef]
7. Aslan, M.F.; Sabanci, K. A novel proposal for deep learning-based diabetes prediction: Converting clinical data to image data. *Diagnostics* **2023**, *13*, 796. [CrossRef] [PubMed]
8. Bell, K.; Shaw, J.E.; Maple-Brown, L.; Ferris, W.; Gray, S.; Murfet, G.; Flavel, R.; Maynard, B.; Ryrie, H.; Pritchard, B.; et al. A position statement on screening and management of prediabetes in adults in primary care in Australia. *Diabetes Res. Clin. Pract.* **2020**, *164*, 108188. [CrossRef]
9. Richards, B.A.; Lillicrap, T.P.; Beaudoin, P.; Bengio, Y.; Bogacz, R.; Christensen, A.; Clopath, C.; Costa, R.P.; de Berker, A.; Ganguli, S.; et al. A deep learning framework for neuroscience. *Nat. Neurosci.* **2019**, *22*, 1761–1770. [CrossRef]
10. Shojaee-Mend, H.; Velayati, F.; Tayefi, B.; Babaee, E. Prediction of Diabetes Using Data Mining and Machine Learning Algorithms: A Cross-Sectional Study. *Healthc. Inform. Res.* **2024**, *30*, 73. [CrossRef]
11. Kumar, A.; Bawa, S.; Kumar, N. iDP: ML-driven diabetes prediction framework using deep-ensemble modeling. *Neural Comput. Appl.* **2024**, *36*, 2525–2548. [CrossRef]
12. El-Bashbishy, A.E.-S.; El-Bakry, H.M. Pediatric diabetes prediction using deep learning. *Sci. Rep.* **2024**, *14*, 4206. [CrossRef]
13. Chen, T.-C.T.; Wu, H.-C.; Chiu, M.-C. A deep neural network with modified random forest incremental interpretation approach for diagnosing diabetes in smart healthcare. *Appl. Soft Comput.* **2024**, *152*, 111183. [CrossRef]
14. Zhang, Z.; Ahmed, K.A.; Hasan, M.R.; Gedeon, T.; Hossain, M.Z. A Deep Learning Approach to Diabetes Diagnosis. *arXiv* **2024**, arXiv:2403.07483.
15. An Y.; Huang, N.; Chen, X.; Wu, F.; Wang, J. High-risk prediction of cardiovascular diseases via attention-based deep neural networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2019**, *18*, 1093–1105. [CrossRef] [PubMed]
16. Djenouri, Y.; Belhadi, A.; Yazidi, A.; Srivastava, G.; Lin, J.C.-W. Artificial intelligence of medical things for disease detection using ensemble deep learning and attention mechanism. *Expert Syst.* **2024**, *41*, e13093. [CrossRef]
17. Zou, X.; Luo, Y.; Huang, Q.; Zhu, Z.; Li, Y.; Zhang, X.; Zhou, X.; Ji, L. Differential effect of interventions in patients with prediabetes stratified by a machine learning-based diabetes progression prediction model. *Diabetes, Obes. Metab.* **2024**, *26*, 97–107. [CrossRef] [PubMed]
18. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, 1–33. [CrossRef]
19. Pal, K.; Poonia, R.C.; Singh, V.; Bhardwaj, H.; Kumar, V. Risk Analysis of Diabetic Patient Using Map-Reduce and Machine Learning Algorithm. In *Handbook of Research on Disease Prediction Through Data Analytics and Machine Learning*; IGI Global: Hershey, PA, USA, 2021.
20. Fernández, A.; Garcia, S.; Galar, M.; Prati, R.C.; Krawczyk, B.; Herrera, F. Synthetic minority oversampling technique for multiclass imbalance problems. *Pattern Recognit.* **2018**, *91*, 313–330.

21. El-Sofany, H.F.; Al-Khassaweneh, M.; Taha, I. A Proposed Technique Using Machine Learning for the Early Prediction and Classification of Diabetes. *Int. J. Intell. Syst.* **2024**, *2024*, 6688934. [CrossRef]
22. Wee, B.F.; Sivakumar, S.; Lim, K.H.; Wong, W.K.; Juwono, F.H. Diabetes detection based on machine learning and deep learning approaches. *Multimed. Tools Appl.* **2023**, *83*, 24153–24185. [CrossRef]
23. Naseem, A.; Habib, R.; Naz, T.; Atif, M.; Arif, M.; Allaoua, C.; Chelloug, S. Novel Internet of Things based approach toward diabetes prediction using deep learning models. *Front. Public Health* **2022**, *10*, 914106. [CrossRef]
24. Khanam, J.J.; Foo, S.Y. A comparison of machine learning algorithms for diabetes prediction. *ICT Express* **2021**, *10*, 432–439. [CrossRef]
25. García-Ordás, M.T.; Benavides, C.; Benítez-Andrades, J.A.; Alaiz-Moretón, H.; García-Rodríguez, I. Diabetes detection using deep learning techniques with oversampling and feature augmentation. *Comput. Methods Programs Biomed.* **2021**, *202*, 105968. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.