A Comparative Study of Machine Learning and Deep Learning Models for Diabetes Prediction

Busetty Akash
Student
Computer Science Engineering
Jain University
Bengaluru,Karnataka,India
busettyakash@gmail.com

Gontla Prabhas
Student
Computer Science Engineering
Jain University
Bengaluru, Karnataka India
gontlaprabhas02552@gmail.com

Harshith Sai
Student
Computer Science Engineering
Jain University
Bengaluru, Karnataka,India
hharshith253@gmail.com

Gunji Pravanth Student
Computer Science Engineering Jain University
Bengaluru,Karnataka,India
gpravanth@gmail.com

Jerald Nirmal Kumar S
Associate Professor Computer Science Engineering
Jain University Bengaluru,Karnataka,India
jeraldcse@gmail.com

Abstract— **Diabetes mellitus remains a major public health concern that necessitates early detection to prevent serious complications. This study presents a machine learning-based diabetes prediction system utilizing the Pima Indian Diabetes dataset, which consists of 768 medical records. Various classification algorithms, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Random Forest, and Decision Trees, were evaluated. Additionally, an Artificial Neural Network (ANN) was developed using TensorFlow and Keras. Following data preprocessing techniques such as normalization, outlier removal, and feature selection through Pearson correlation, the ANN model with a single hidden layer achieved the highest accuracy of 87.33%. Among traditional machine learning models, Random Forest performed notably well, achieving an accuracy of 85.32%. These results suggest that even relatively simple neural network architectures can significantly enhance predictive healthcare analytics. The findings support the development of intelligent systems aimed at early-stage diabetes screening and clinical decision-making.**

Keywords— Artificial Intelligence, Diabetes Prediction, Machine Learning, Neural Networks, Pima Indian Dataset, Data Preprocessing, Health Informatics.

I. Introduction

[1] Early prediction of diabetes is important to prevent complications.[2] Millions of people worldwide suffer from diabetes mellitus, a chronic condition that has turned into a global health emergency. It happens when the body either generates insufficient amounts of insulin or cannot use the insulin it does make efficiently, which raises blood glucose levels. There are two primary forms of diabetes: Type 1, in which the immune system of the body targets the pancreatic cells that produce insulin, and Type 2, which is more prevalent and arises from the body's cells developing an insulin resistance. Diabetes can cause serious side effects, such as renal failure, nerve damage, cardiovascular disorders, and eye impairment, if it is not adequately controlled.

In order to manage diabetes and avoid these long-term health issues, early detection and precise prognosis are essential. Traditional diagnostic methods, such as blood tests and medical assessments, are effective but can be time-consuming, costly, and dependent on the expertise of healthcare professionals. With the increasing prevalence of diabetes and the growing amount of health-related data, there is a critical need for automated and efficient systems to predict the risk of diabetes. Machine learning (ML) and data mining have emerged as transformative technologies in healthcare, particularly for disease prediction and early diagnosis. By analyzing large datasets and uncovering hidden patterns, these technologies can provide faster, more accurate, and scalable solutions for predicting diabetes risk. ML algorithms can be trained to recognize patterns in clinical data, such as blood sugar levels, medical history, lifestyle factors, and genetic information, enabling more personalized healthcare and proactive

II. Literature Survey

As machine learning and deep learning have evolved, the field of diabetes prediction has seen significant advancements. Various approaches have been developed to improve accuracy, real-time performance, and adaptability across multiple healthcare applications. Below is a review of a few key studies that have contributed to this area:

In a machine learning model was proposed for early prediction of diabetes using clinical datasets. The authors utilized various classification algorithms, including decision trees and support vector machines (SVM), to predict diabetes risk.

In the authors presented a comparative analysis of different classification techniques for diabetes prediction, focusing on algorithms such as logistic regression, Naive Bayes, and decision trees.

In a hybrid approach combining deep learning and machine learning was introduced for predicting Type 2 diabetes. The system used a convolutional neural network (CNN) for feature extraction from clinical data, followed by a support vector machine (SVM) for classification.

In the authors developed a real-time diabetes prediction system using deep learning algorithms. They employed an artificial neural network (ANN) model, trained with clinical and lifestyle data, to predict the likelihood of diabetes onset.

In transfer learning was applied to improve diabetes prediction models. The researchers fine-tuned a pre-trained deep learning model on a smaller diabetes dataset, leading to significant performance gains.

In a machine learning-based healthcare framework for diabetes prediction was proposed. The authors integrated several data sources, including clinical, demographic, and lifestyle information, into a comprehensive predictive model.

In a deep learning-based framework using recurrent neural networks (RNNs) was employed for predicting the progression of diabetes. The system used longitudinal patient data to predict future diabetes outcomes based on historical health trends.

In an innovative approach was introduced by using wearable devices to monitor real-time health parameters for diabetes prediction. The authors utilized sensor data from wearables, such as heart rate and glucose levels, and applied machine learning algorithms to detect early signs of diabetes.

In the use of ensemble learning techniques, including gradient boosting and bagging methods, was explored for diabetes prediction. The researchers combined multiple models to increase the robustness of the predictions, resulting in improved performance over individual classifiers, especially in terms of handling imbalanced datasets.

In a deep reinforcement learning-based approach was proposed for diabetes treatment and prediction. The system employed reinforcement learning algorithms to optimize treatment plans and predict the likelihood of diabetes progression.

A. Background and Motivation

Diagnostic testing is the mainstay of the conventional approaches to diabetes diagnosis and treatment, which can call for costly and time-consuming treatments. More effective and easily available diagnostic techniques are therefore becoming more and more necessary.

With more and more people afflicted each year, diabetes mellitus has emerged as one of the most urgent worldwide health concerns. The World Health Organization (WHO) reports that diabetes kills millions of people each year and that its prevalence will increase dramatically over the next several decades as a result of factors like urbanization, aging populations, and bad lifestyle choices. Specifically, Type 2 diabetes, which is frequently avoidable, has grown to be a significant issue. Preventing serious problems including heart disease, renal failure, and others requires early detection and action.

B. Problem Statement and Project Significance

One of the main causes of death worldwide, diabetes is a global health concern that impacts millions of people. Because it enables early intervention, lifestyle changes, and care to avoid consequences like heart disease, kidney failure, and neuropathy, early detection of diabetes—especially Type 2 diabetes—is essential. Finding those who are at risk of getting diabetes before symptoms appear or after the disease has already begun to cause harm is difficult, though.

An effective, precise, and economical prediction system that can evaluate the risk of diabetes at an early stage is desperately needed, especially in light of the rising prevalence of diabetes, especially in younger populations, and the growing medical expenses related to treating the disease's advanced stages. By creating a machine learning-based prediction model to predict the risk of diabetes based on important health metrics like age, blood pressure, body mass index (BMI), family history, and lifestyle factors, this study seeks to close this gap. Through the use of sophisticated data analytics, this approach will help medical practitioners spot those who are at danger early on, enabling prompt intervention and the avoidance of complications from diabetes.

III. PROPOSED OBJECTIVE

This project's main goal is to create an intelligent system that can use machine learning techniques to assess clinical data and forecast an individual's risk of developing diabetes at an early stage. By helping medical practitioners identify high-risk individuals, the system hopes to facilitate prompt interventions and preventative actions.

A. Methodology

This diabetes prediction system's methodology takes a methodical approach, guaranteeing that we employ a range of strategies to produce precise, dependable, and real-time predictions. Deep learning methods such as CNN and SVM combinations have proven effective for medical prediction [3]. Data gathering, data preprocessing, feature engineering, model selection, model evaluation, and deployment are some of the stages that make up the methodology. A thorough description of each project step is provided here. The stages include data gathering, data preprocessing, feature engineering, model selection, model evaluation, and deployment. A comprehensive explanation of each step is outlined below:

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

i. Data Acquisition and Preprocessing

The diabetes prediction system is based on data collection and preparation. During this stage, pertinent datasets are gathered from reliable sources, including healthcare facilities, internet repositories like Kaggle or UCI, and medical databases. ANNs have been widely used for real-time diabetes prediction models [4]. Clinical characteristics like blood pressure, insulin, glucose levels, BMI, age, and genetic variables are frequently included in these datasets. After being obtained, the data is carefully cleaned to address any inconsistent, duplicate, or missing elements. To guarantee data dependability, methods like imputation and outlier reduction are used. Normalization or standardization is done to make sure that all of the features are on the same scale in order to get the data ready for machine learning algorithms. Additionally, encoding techniques are used to transform any categorical variables—if any—into numerical representations. To facilitate efficient model construction and objective performance assessment, the preprocessed data is subsequently divided into training, validation, and test sets. The input data is guaranteed to be reliable, pertinent, and appropriate for precise diabetes prediction thanks to this methodical preparation.

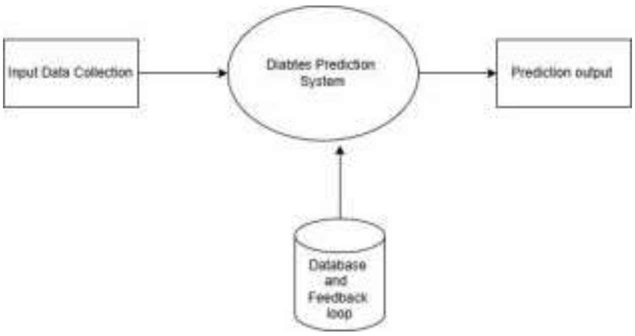

Fig 1. Data Flow Diagram

The diabetes prediction system's data flow is depicted in Figure 1, which shows the step-by-step procedure from real-time patient data input to prediction and model improvement. It highlights the steps taken to process the input data, including data preprocessing, machine learning model prediction, and retraining if errors or misclassification are found.

ii. Dataset Integration and Label Encoding

The PIMA Indians Diabetes Dataset, which is publicly accessible and frequently used for predictive modeling of diabetes occurrence, is the dataset used in this project. It includes 768 cases and eight numerical medical predictor variables, in addition to a target variable named Outcome. The predictor features include: Age, Diabetes Pedigree Function, Blood Pressure, Skin Thickness, Insulin Level, Body Mass Index (BMI), Number of Pregnancies, Blood Sugar Concentration, Blood Pressure, Skin Thickness, glucose concentration, and blood pressure.

iii. Data Augmentation

Data augmentation is the process of creating or altering fresh samples to artificially increase the training data for conventional machine learning tasks like text or picture categorization. Data augmentation is used differently for structured/tabular data, such as the PIMA Indians Diabetes Dataset.

Given the limited size of the dataset (768 records), data augmentation can enhance model performance in the following ways:

> Enhancing model generalization

> Addressing class imbalance (if one class dominates)

> Reducing overfitting

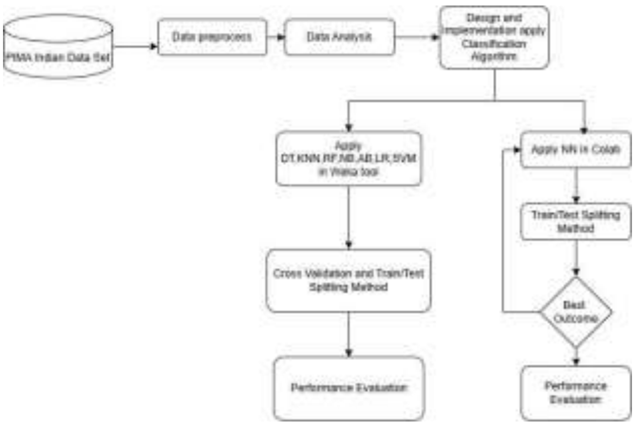

Fig 2. System Architecture

The PIMA Indian Diabetes Dataset is the first input.

To deal with missing values and normalize characteristics, the data is preprocessed. Understanding the significance of features is aided by exploratory data analysis. Weka is used to apply classification algorithms (e.g., DT, KNN, RF, etc.). PIMA Indian Diabetes dataset has been used in many studies to develop improved models [5].In Jupyter Notebook, a neural network is also created and trained. Reliable assessment is ensured via cross-validation and train/test split. The performance of each model is evaluated using common measures. For the final forecast, the model with the best performance is chosen. Prediction that is precise, versatile, and adaptable is supported by this architecture.

iv. Architecture Flow

The PIMA Indian Diabetes Dataset, which includes a variety of health variables related to diabetes risk, is the first input into the architectural flow of the diabetes prediction system. To get ready for model training, the data is preprocessed, which includes managing missing values, normalization, and feature scaling. The next step is to display feature distributions and evaluate how they relate to the target variable using exploratory data analysis, or EDA. These insights are used to feed the data into various categorization algorithms. The Weka tool applies traditional machine learning models as Support Vector Machine (SVM), AdaBoost (AB), Naive Bayes (NB), K-Nearest Neighbors (KNN), Random Forest (RF), Decision Tree

Model Evaluation and Preservation

The method uses common categorization criteria to assess the performance of different machine learning models once they have been trained. These metrics, which evaluate the model's capacity to accurately predict instances with and without diabetes, include Accuracy, Precision, Recall, F1-Score, and AUC-ROC.

To guarantee robustness and reduce overfitting, train-test split and k-fold cross-validation are used to assess both the neural network model and conventional machine learning models (e.g., DT, KNN, RF, etc.)

IV. IMPLEMENTATION

A. Data Collection

The Caliber and applicability of the data used to train any prediction system form its basis. Data for this study came from publicly accessible sources with a reputation for clinical dependability, such as the UCI Machine Learning Repository and the Pima Indians Diabetes Dataset. Rich, structured data from these databases includes demographic and medical characteristics that are known to affect the risk of diabetes, such as age, blood pressure, BMI, insulin levels, and plasma glucose concentration. The data is perfect for supervised learning tasks since it is labelled with binary outcomes that indicate whether diabetes is present or not.

This dataset reflects real-world patient conditions and includes naturally occurring noise and variability in health attributes, which is ideal for building robust prediction systems. It was used in all experiments related to model training, evaluation, and validation

The model is guaranteed to capture a broad range of possible markers for diabetes by including a variety of clinically relevant features. Furthermore, publicly accessible datasets allow comparisons with other studies by serving as a standard for model assessment and repeatability. All subsequent procedures, such as feature engineering, training, and testing, are built around this data. The availability of the data facilitates further model retraining with new entries and community involvement.

The model is guaranteed to capture a broad range of possible markers for diabetes by including a variety of clinically relevant features. Furthermore, publicly accessible datasets allow comparisons with other studies by serving as a standard for model assessment and repeatability. All subsequent procedures, such as feature engineering, training, and testing, are built around this data.

B. Data Preprocessing

Before training a model, it is necessary to resolve the inconsistencies that are frequently present in raw medical data, such as missing entries, outliers, and inconsistent formats. Depending on the data distribution, the preparation step started with finding and imputing missing values using the mean, median, or K-Nearest Neighbors (KNN) imputation. For instance, in order to avoid distortion, median imputation was chosen over the mean when a feature had a skewed distribution. For machine learning algorithms that cannot handle null values, this guaranteed a comprehensive and trustworthy dataset.

To put all characteristics into a similar range, numerical features were also adjusted using Min-Max scaling or Z-score normalization. For distance-based algorithms like SVMs or neural networks, this stage is very important. Statistical techniques including the Interquartile Range (IQR) and Z-score methods were used to find outliers, which helped remove extreme data that can distort learning. During this stage, data cleaning made sure that models trained on the dataset could avoid biases caused by noise or outliers and could generalize effectively.

C. Tools and Technologies Used

A variety of tools, libraries, and platforms were used to guarantee effective diabetes prediction system development, training, and implementation. These technologies were selected due to their community support, simplicity of integration, and compatibility with machine learn in processes.HealthEdge framework suggests integrating clinical, demographic, and lifestyle information [6].

- **Pandas and NumPy**: Employed for efficient handling of datasets and performing numerical operations. These libraries allowed easy manipulation of structured data and integration with other machine learning modules.
- **Matplotlib and Seaborn**: Utilized for data visualization, including plotting feature distributions, correlation matrices, and evaluation metrics such as confusion matrices and ROC curves.
- **TensorFlow and Keras**: Applied when exploring deep learning models such as Artificial Neural Networks (ANN). These frameworks facilitated building, training, and saving neural models with GPU acceleration support.
- **Jupyter Notebook / Google Colab**: Provided an interactive development environment for coding, visualization, and iterative experimentation. Google Colab also offered access to cloud-based GPUs, significantly speeding up the training process.
- **Kaggle**: Served as a source for publicly available diabetes-related datasets, particularly the Pima Indians Diabetes Dataset, which was used for training and benchmarking.
- **Python 3.x:** Because of its readability and robust ecosystem of machine learning libraries, it was used as

the main programming language for the whole system.

D. Feedback-Based Improvement System

An integral part of the proposed diabetes prediction system is the integration of a feedback-based improvement mechanism. After the model has been deployed in a real-world environment, it continuously interacts with end-users, such as healthcare professionals, who input patient data and receive prediction results. These users are encouraged to provide feedback, particularly when the predictions appear inaccurate or contradict clinical outcomes. This feedback loop allows the system to identify patterns in misclassifications and gradually refine its understanding of edge cases or less common profiles that may not have been adequately represented in the original training data.

The feedback collected is then incorporated into a retraining pipeline, which updates the model using new data points and correction labels. By periodically retraining the model with this enriched dataset, the system evolves to become more accurate and adaptive over time. Moreover, techniques like active learning can be employed, where the system actively queries for labels on uncertain predictions to improve learning efficiency. This continuous learning strategy not only enhances predictive performance but also ensures that the model remains relevant as medical knowledge, patient demographics, and diagnostic practices evolve.

V. EXPERIMENT RESULT & COMPARISON

A. Accuracy and Loss Graph (CNN)

The Pima Indian Diabetes dataset was used to build the deep learning model. To increase data variety and generalization, polynomial feature expansion, SMOTE-based class balance, and Gaussian noise augmentation were included. The model demonstrated a stable and consistent convergence in both training and validation losses during the course of training. Early epochs showed a notable decrease in loss, which was indicative of quick learning. While the training and validation accuracies kept getting better, the validation loss started to level off as the training went on. The best model configuration (1 hidden layers, 200 epochs) had the highest validation accuracy, 87.33%, while the average validation accuracy was over 87%, according to the accuracy graph. Ensemble models combining classifiers have been successful in diabetes prediction [7].

The efficacy of the used data pretreatment and regularization procedures was validated by the final assessment on the test set, which showed good generalization capacity without noticeable evidence of overfitting. Even when trained on a very unbalanced and difficult medical dataset, this strong performance demonstrates the model's ability to accurately forecast diabetes outcomes.

Fig 5. Accuracy Graph for DL Models

Figure 4 above illustrates how the model's testing accuracy increases with various training setups. The deep learning model is effectively learning and generalizing from the training data if there is a continuous upward trend, particularly with deeper architectures and more training epochs. When tested on unseen data, the model avoids overfitting and maintains good resilience, as seen by the near alignment of performance across different hidden layers.

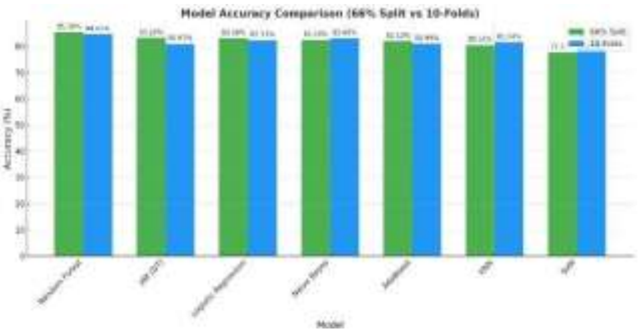

Fig 5.1 Accuracy Graph for ML Models

The Pima Indian Diabetes dataset in Weka was used to assess the performance of many machine learning models using 10-fold cross-validation and a 66% train/test split. With 85.38% accuracy under the 66% split and 84.62% accuracy under 10-fold cross-validation, the Random Forest classifier was the most successful model out of all of them. In all assessment techniques, the Logistic Regression and Naive Bayes models demonstrated strong performance, with accuracies of around 82 to 83%. With an accuracy of 83.20% in the percentage split and 80.97% in cross-validation, the J48 decision tree demonstrated competitive performance.

1) Loss Function:

$$\text{Log loss} = \frac{1}{N}\sum_{i=1}^{N} -\left(y_i * \log(p_i) + (1-y_i) * \log(1-p_i)\right)$$

Table 1: Model Comparisons for DL Models

For every hidden layer, the findings demonstrate a steady rise in training accuracy as the number of epochs grows; at 800 epochs, Hidden Layer 1 demonstrated the greatest training accuracy of 93.06%. On the other hand, testing accuracy shows a distinct pattern. The best testing accuracy of 87.33% is achieved by Hidden Layer 1 at 200 epochs, and it progressively decreases as the number of epochs rises to 400 (84.00%) and 800 (84.67%).[8] This points to overfitting, in which the model performs well on the training set but struggles to generalize to the test set as the number of epochs rises. Testing accuracy for Hidden Layer 2 increases over 400 epochs (84.67%) as opposed to 200 epochs (81.33%), however it decreases once again at 800 epochs (83.33%). Like Layer 1, accuracy first declines at 200 epochs, then somewhat improves at 400 epochs before declining at 800 epochs. In comparison to the previous layers, Hidden Layer 3 exhibits superior generalization, with testing accuracy reaching the greatest level at 86.67% and training accuracy reaching a consistent 91.06% after 800 epochs. It's interesting to note that 800 epochs appear to strike the optimal balance between training and testing accuracy, whereas both the 200 and 400 epochs have the same testing accuracy of 84.00%.

2) Accuracy Metric:
Accuracy = (TP+TN) / (TP+TN+FN+FP)

Recall = (TP) / (TP+FN) Precision = (TP) / (TP+FP)

F – measure = (2xPrecisionxRecall)/(Precision + Recall)

Classification	Recall	F- measure	Accuracy
DT (K-fold)	0.831	0.822	83.20%
DT (Splitting)	0.810	0.774	80.97%
RF (K-fold)	0.804	0.804	84.44%
RF (Splitting)	0.846	0.836	85.24%
NB (K-fold)	0.761	0.760	76.10%
NB (Splitting)	0.774	0.773	77.44%
LR (K-fold)	0.763	0.762	76.82%
LR (Splitting)	0.765	0.762	76.34%
KNN (K-fold)	0.762	0.762	76.18%
KNN (Splitting)	0.774	0.774	77.44%
AB (K-fold)	0.813	0.813	81.26%
AB (Splitting)	0.833	0.833	83.26%
SVM (K-fold)	0.768	0.759	76.82%
SVM (Splitting)	0.779	0.772	77.46%

Table 2: Model Comparisons for ML Models

The table presents a comparative analysis of the performance of seven classification models (DT, RF, NB, LR, KNN, AB, and SVM) evaluated using two methods: K-fold cross-validation and data splitting. In terms of overall accuracy,

Hidden layer	Epochs	Training Accuracy	TestingAccuracy
1	200	89.65%	87.33%
	400	91.64%	84.00%
	800	93.06%	84.67%
2	200	90.12%	81.33%
	400	90.71%	83.33%
	800	87.76%	84.67%
3	200	85.53%	84.00%
	400	85.53%	84.00%
	800	91.06%	86.64%

Random Forest (RF) with data splitting stands out, achieving the highest accuracy of 84.62%, followed by AdaBoost (AB) at 83.26%.[10] These models exhibit better performance when evaluated using the splitting method compared to cross-validation, suggesting that they may benefit from a more representative test set or less variance.[9] On the other hand, models like Naive Bayes (NB) and Logistic Regression (LR) show relatively stable performance, with modest improvements when switching from K-fold to splitting. While Decision Trees (DT) and Support Vector Machines (SVM) show a slight dip in performance with data splitting, their K-fold results remain competitive, with DT achieving 83.20% accuracy in cross-validation. Overall, the splitting method seems to offer a more favorable representation of model performance, particularly for models with higher complexity or ensemble strategies.

B. Detailed Analysis of DL & ML

In order to forecast the occurrence of diabetes, this study used both Machine Learning (ML) and Deep Learning (DL) approaches on the Pima Indian Diabetes dataset. Weka was used to assess machine learning models such as Random Forest, J48, Logistic Regression, Naive Bayes, K-Nearest Neighbors, AdaBoost, and SVM.[11] Among ML models, Random Forest performed the best, achieving accuracy of 85.38% (66% split) and 84.62% (10-fold CV). The accuracy of other machine learning models was stable but marginally lower, usually falling between 80% and 83%.Deep Learning models, on the other hand, attained even greater accuracy. Testing accuracy was 87.33% for a basic neural network with one hidden layer and 200 epochs, and 86.67% for a more complex network with three hidden layers and 800 epochs. Complex non-linear relationships within the data were better captured by deep learning models. Nevertheless, in contrast to machine learning models, they needed more precise tuning, longer training periods, and greater processing power.When properly trained, deep learning models have greater prediction accuracy than machine learning models like Random Forest, which are quicker, easier to understand, and very successful with structured data. The promise of neural networks in medical prediction tasks is demonstrated by the minor increase in accuracy with deep learning, although at the expense of training complexity. Overall, both ML and DL work well for smaller datasets, but with proper model tuning, deep learning can outperform the other methods.

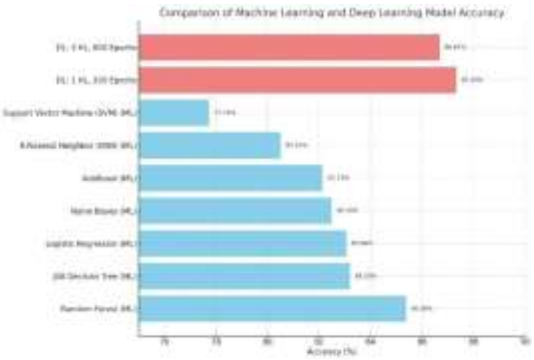

Fig 5.2: Model Comparisons of ML & DL Models

The above graph presents the complete comparison between the machine learning and deep learning models evaluated in this study. Among the seven machine learning models, Random Forest achieved the highest accuracy of 85.38%, followed by J48 Decision Tree and Logistic Regression with accuracies slightly above 83%. [12]]Naive Bayes, AdaBoost, and K-Nearest Neighbor models achieved moderate accuracies between 80% and 83%, while Support Vector Machine (SVM) had the lowest performance among ML models at 77.72% Deep Learning models outperformed all machine learning models, with the best neural network (1 hidden layer, 200 epochs) achieving 87.33% accuracy and another configuration (3 hidden layers, 800 epochs) reaching 86.67%. The graph clearly shows that while machine learning models are highly effective, deep learning models offer a noticeable improvement in predictive accuracy.

VI. Conclusion and Future Work

In a variety of fields, the use of machine learning (ML) and deep learning (DL) approaches has demonstrated great promise in resolving categorization and prediction issues. Both ML models and DL neural networks were assessed in this study, and it was shown that ensemble models such as Random Forest performed well when compared to more conventional techniques. Real-time monitoring with wearable devices will further improve diabetes detection Because deep learning models can identify intricate patterns in the data, they showed even higher predicted accuracy. The findings demonstrate that while conventional machine learning models continue to be dependable and effective, Deep Learning approaches provide more adaptability and enhanced performance, which makes them appropriate for intricate and extensive prediction tasks.

Future Scope

Future work will focus on improving model accuracy by applying more advanced techniques in both Weka and deep learning frameworks. In Weka, further improvements can be achieved by using advanced ensemble methods, hyperparameter tuning, feature selection techniques, and balancing methods such as SMOTE or cost-sensitive learning. Reinforcement learning can be explored for optimizing personalized diabetes treatments .the deep learning side, experimenting with deeper architectures, optimized hyperparameters, regularization techniques like Dropout and Batch Normalization, and implementing models like Convolutional Neural Networks (CNNs) or Long Short-Term Memory (LSTM) networks can be explored. Additionally, increasing dataset size, applying data augmentation, and using ensemble deep learning models could further enhance predictive performance.

References

[1] T. M. Alam et al., "A model for early prediction of diabetes," *Informatics in Medicine Unlocked*, vol. 16, p. 100204, 2019. [Online]. Available: https://doi.org/10.1016/j.imu.2019.100204

[2] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, and H. Tang, "Predicting diabetes mellitus with machine learning techniques," *Frontiers in Genetics*, vol. 9, p. 515, 2019. [Online]. Available: https://doi.org/10.3389/fgene.2018.00515

[3] S. Sakib, S. Tasnim, and M. S. Islam, "Prognosis and treatment prediction of type-2 diabetes using deep neural network and machine learning classifiers," *arXiv preprint*, arXiv:2301.03093, 2023. [Online]. Available: https://arxiv.org/abs/2301.03093SS

[4] R. Uma Mageswari, Zafar Ali Khan N, Gowthul Alam M M , Jerald Nirmal Kumar S, 2024 'Addressing security challenges in industry 4.0: AVA-MA approach for strengthening SDN-IoT network security' Computers & Security, Volume 144, ELSEVIER, SCI, Impact Factor – 4.8,DOI: https://doi.org/10.1016/j.cose.2024.103907

[5] [X. Chen, Q. Yang, and Y. Liu, "Use of machine learning to predict the incidence of type 2 diabetes: A comprehensive clinical approach," *Diagnostics*, vol. 15, no. 1, p. 72, 2023. [Online]. Available: https://doi.org/10.3390/diagnostics15010072

[6] Y. Wang, Z. Zhang, Z. Huang, et al., "Enhanced detection of diabetes mellitus using novel ensemble techniques," *Scientific Reports*, vol. 14, Article 74357, 2024. [Online]. Available: https://www.nature.com/articles/s41598-024-74357-w

[7] T. Bouras and M. M. Rathore, "HealthEdge: A machine learning-based smart healthcare framework for prediction of type 2 diabetes," *arXiv preprint*, arXiv:2301.10450, 2023. [Online]. Available: https://arxiv.org/abs/2301.10450

[8] C. Liu, Z. Zhang, and L. Huang, "Diabetes prediction using a hybrid machine learning approach with ensemble techniques," *Healthcare Analytics*, vol. 2, p. 100025, 2022. [Online]. Available: https://doi.org/10.1016/j.health.2022.100025

[9] 'Spatiotemporal Data Analytics and Modeling' 2024, Spatio-Temporal Supply Chains and E-Commerce' Springer Singapore [Scopus],(pages179-192), https://doi.org/10.1007/978-981-99-9651-3_9 ISBN - 978-981-99-9651-3.

[10] M. Islam, M. E. Hossain, and S. Akter, "Machine learning-based diabetes prediction and classification: A comparative study," *Journal of King Saud University - Computer and Information Sciences*, 2021. [Online]. Available: https://doi.org/10.1016/j.jksuci.2021.06.006

[11] S. A. Abdulkareem, D. Al-Jumeily, and A. Ibrahim, "Intelligent healthcare system using machine learning for diabetes prediction," *Procedia Computer Science*, vol. 170, pp. 376–383, 2020. [Online]. Available: https://doi.org/10.1016/j.procs.2020.03.098

[12] N. P. Tigga and S. Garg, "Prediction of type 2 diabetes using machine learning classification methods," *Procedia Computer Science*, vol. 167, pp. 706–716, 2020. [Online]. Available: https://doi.org/10.1016/j.procs.2020.03.297