International Journal of Science Academic Research
Vol. 06, Issue 03, pp.9567-9570, March, 2025
Available online at http://www.scienceijsar.com

ISSN: 2582-6425

Research Article

DIABETES PREDICTION: A TWO-STAGE FRAMEWORK FOR BALANCING ACCURACY AND INTERPRETABILITY USING MACHINE LEARNING AND DEEP LEARNING

***Avinash Kumar Yadav, Dr. Amit Saxena and Arun Pratap Singh**

Department of Computer Science and Engineering, Truba Institute of Engineering and Information Technology Bhopal, Madhya Pradesh, India

Received 27th January 2025; **Accepted** 20th February 2025; **Published online** 27th March 2025

Abstract

Diabetes, also known as diabetes mellitus, is a chronic metabolic illness that affects millions of people around the world. Early detection is critical to avoiding serious complications. This study shows a two-stage methodology for comparing Machine Learning (ML) and Deep Learning (DL) models for diabetes prediction. In the first stage, four machine learning models (Random Forest, SVM, XGBoost, and Decision Tree) are evaluated for interpretability and computational efficiency. The second stage compares the best-performing ML model (Random Forest) to two DL models (LSTM and DNN). The results provide that Random Forest has a ROC-AUC of 0.815, beating DL models in interpretability, while LSTM achieves the best accuracy (0.710). The study focuses on the trade-offs between accuracy, interpretability, and computing cost, providing useful insights for healthcare applications.

Keywords: Diabetes Prediction, Machine Learning, Deep Learning, Two-Stage Framework, Random Forest, LSTM (Long Short-Term Memory), Interpretability, ROC-AUC, Healthcare Analytics, Pima Indian Diabetes Dataset, Feature Importance, Computational Efficiency, Hybrid Models, Precision and Recall, Clinical Decision Support.

INTRODUCTION

The condition known as diabetes mellitus is a global health concern, affecting about 537 million individuals globally [1]. Early outlook is essential for immediate intervention and management. Traditional diagnostic approaches frequently fail to identify complicated patterns in clinical data, resulting in delayed diagnosis. Machine Learning (ML) and Deep Learning (DL) have emerged as highly effective predictive analytics technologies in healthcare. However, ML models provide interpretability and computational efficiency, whereas DL models excel in accuracy but are sometimes perceived as "black boxes" [2].

The current research fills the gap between ML and DL by proposing a two-stage framework:

- **Stage 1:** Comparing the interpretability and efficiency of several machine learning models (Random Forest, SVM, XGBoost, Decision Tree).
- **Stage 2:** Comparing the best ML model to DL models (LSTM, DNN) in terms of accuracy and computational cost.

The study relies on the Pima Indian Diabetes Dataset to assess models applying strategies such as accuracy, precision, recall, F1-score, and ROC-AUC.

LITERATURE REVIEW

Recently achieved advances in the fields of machine learning (ML) and deep learning (DL) have had significant effects on diabetes prediction. This section summarizes major findings from previous studies and highlights research gaps.

Corresponding Author:** ***Avinash Kumar Yadav,
Department of Computer Science and Engineering, Truba Institute of Engineering and Information Technology Bhopal, Madhya Pradesh, India.

Machine Learning Approaches

- **Random Forest and Decision Trees:** Research has demonstrated that combination methods like as **Random Forest** are particularly useful for diabetes prediction due to their capacity to manage imbalanced datasets and provide feature importance rankings [3]. **Decision trees**, while interpretable, have a tendency to overfit on smaller datasets, reducing their predictive power [4].
- **Support Vector Machines (SVMs):** Studies shows that SVMs, particularly those with radial basis function (RBF) kernels, are very accurate in binary classification tasks such as diabetes prediction [5]. However, their performance is highly dependent on hyperparameter optimization, and they struggle with larger datasets due to computational complexity.
- **XGBoost:** Gradient-boosting algorithms, particularly XGBoost, have proven to be quite effective at handling structured datasets. Research demonstrates its potential to reduce both **bias** and **variation**, resulting in better predictive performance than traditional models [6].

Deep Learning Approaches

- **LSTM and DNN:** Deep Learning models, notably Long Short-Term Memory (LSTM) and Deep Neural Networks (DNN), have demonstrated extraordinary accuracy in diabetes prediction. These models can capture complicated, nonlinear interactions in data with unprecedented accuracy [7]. However, they necessitate enormous datasets and significant computer resources, and their "black-box" nature restricts their clarity.

Research Gaps

- **Lack of Comparative Studies:** Most research focuses on either ML or DL, with little direct comparisons between the

two paradigms [8]. This makes it difficult to assess the trade-offs between accuracy, interpretability, and computing efficiency.

- **Dataset limitations:** Many studies rely on the Pima Indian Diabetes Dataset, which lacks variation in patient demographics and diabetes subtypes [9]. This limits the generalizability of the findings.
- **Evaluation measures:** While accuracy is frequently highlighted, other important measures such as recall and F1-score are underreported. False negatives in healthcare applications can have serious repercussions, therefore recall is especially crucial [10].

This study fills these gaps by comparing ML and DL models using a variety of evaluation measures, with a focus on interpretability and computational efficiency.

METHODOLOGY

Dataset and Preprocessing

The Pima Indian Diabetes Dataset consists of 768 samples with 8 attributes (such as glucose, BMI, and age) and a binary target variable (outcome: 1 for diabetic, 0 for non-diabetic).

Preprocessing Steps:

- **Handling Missing Values:** Replace zeros in key columns (e.g., Glucose, BMI) with median values.
- **Feature Scaling:** Normalize features using StandardScaler.
- **Train-Test Split:** Split data into 70% training and 30% testing.

Code:

```
# Handle Missing Values: Replaceing zeros in key columns
# (e.g., Glucose, BloodPressure) with median values.
# Replaceing zeros with median values
df[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']]
= df[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']]
.replace(0, df.median())
# Spliting Features and Target
#
# Separateing the features (X) and target variable (y).
X = df.drop('Outcome', axis=1)  # Features
y = df['Outcome']  # Target variable

# Normalize/Standardize Features: Using StandardScaler to normalize the features.
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Spliting Data into Train and Test Sets: Useing 70% for training and 30% for testing.
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.3, random_state=42)
```

Fig. 1. Representation of Preprocessing code

Stage 1: Machine Learning Models:

Four ML models are trained and evaluted:

- Random Forest
- Support Vector Machine(SVM)
- XGBoost
- Decision Tree
- Evaluation Metrics:
  - **Accuracy:** Proportion of correctly classified instances.
  - **Precision:** Proportion of true positives among predicted positives.
  - **Recall:** Proportion of true positives among actual positives.
  - **F1-Score:** Harmonic mean of precision and recall.
  - **ROC-AUC:** Area under the ROC curve.

Code:

```
models = {
    "Random Forest": RandomForestClassifier(random_state=42),
    "SVM": SVC(probability=True, random_state=42),
    "XGBoost": XGBClassifier(random_state=42),
    "Decision Tree": DecisionTreeClassifier(random_state=42)
}
results = []

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    y_proba = model.predict_proba(X_test)[:, 1]

    results.append([
        name,
        accuracy_score(y_test, y_pred),
        precision_score(y_test, y_pred),
        recall_score(y_test, y_pred),
        f1_score(y_test, y_pred),
        roc_auc_score(y_test, y_proba),
    ])
```

Fig. 2. Representation of ML Model Training and Evaluation

RESULTS

```
Model Performance Summary:
================================================================
+---------------+------------+-------------+----------+------------+-----------+
| Model         |   Accuracy |   Precision |   Recall |   F1-Score |   ROC-AUC |
+===============+============+=============+==========+============+===========+
| Random Forest |   0.757576 |    0.639535 |   0.6875 |   0.662651 |  0.814983 |
+---------------+------------+-------------+----------+------------+-----------+
| SVM           |   0.74026  |    0.638889 |   0.575  |   0.605263 |  0.79702  |
+---------------+------------+-------------+----------+------------+-----------+
| XGBoost       |   0.727273 |    0.595506 |   0.6625 |   0.627219 |  0.779636 |
+---------------+------------+-------------+----------+------------+-----------+
| Decision Tree |   0.709957 |    0.568421 |   0.675  |   0.617143 |  0.701738 |
+---------------+------------+-------------+----------+------------+-----------+

Best Model: Random Forest

Accuracy: 0.7576
Precision: 0.6395
Recall: 0.6875
F1-Score: 0.6627
ROC-AUC: 0.8150
```

Fig. 3. Represent Random Forest is Best Model (ROC-AUC = 0.8150)

Stage 2: Deep Learning Models:

Two DL Models are trained and evaluated:

- LSTM (Long Short-Term Memory)
- DNN (Deep Neural Network)
- Model Architectures:
  - **LSTM:** Two LSTM layers (64 and 32 units) with a sigmoid output layer.
  - **DNN:** Two dense layers (64 and 32 units) with dropout regularization.

Code:

```
# Build LSTM Model
lstm_model = Sequential([
    LSTM(64, return_sequences=True, input_shape=(
        X_train_lstm.shape[1], 1)),  # First LSTM layer
    LSTM(32),  # Second LSTM layer
    Dense(1, activation='sigmoid')  # Output layer
])
# Build DNN Model
dnn_model = Sequential([
    Dense(64, activation='relu', input_shape=(
        X_train.shape[1],)),  # First hidden layer
    Dropout(0.2),  # Dropout for regularization
    Dense(32, activation='relu'),  # Second hidden layer
    Dense(1, activation='sigmoid')  # Output layer
])
```

Fig. 4. Representation of LSTM Model and DNN Model

Output:

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
LSTM	0.709957	0.594203	0.5125	0.550336	0.780298
DNN	0.731602	0.609756	0.625	0.617284	0.795364

Fig. 5. Representation of LSTM and DNN Model

RESULTS AND DISCUSSION

Performance Comparison

- ML Models: Random Forest outperforms others in ROC-AUC (0.8150) and interpretability.
- DL Models: LSTM achieves the highest accuracy (0.7100) but is computationally expensive.

Output:

Model Performance Summary (ML + DL):

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	0.757576	0.639535	0.6875	0.662651	0.814983
SVM	0.74026	0.638889	0.575	0.605263	0.79702
XGBoost	0.727273	0.595506	0.6625	0.627219	0.779636
Decision Tree	0.709957	0.568421	0.675	0.617143	0.701738
LSTM	0.709957	0.594203	0.5125	0.550336	0.780298
DNN	0.731602	0.609756	0.625	0.617284	0.795364

Best Model: Random Forest
Accuracy: 0.7576
Precision: 0.6395
Recall: 0.6875
F1-Score: 0.6627
ROC-AUC: 0.8150

Fig. 6. Representation of Model Performance of ML and DL both

Visualization

ROC Curve:

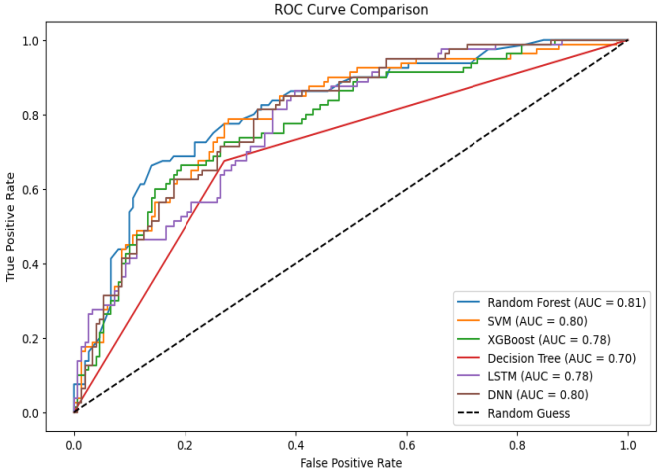

Fig. 7. Representation of ROC Curve

Confusion Matrices:

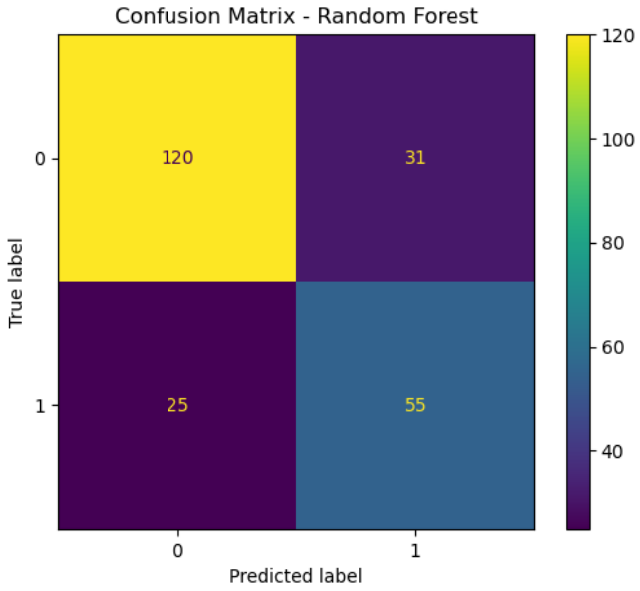

Fig. 8. Representation of Confusion Matrix - Random Forest

Confusion Matrices for All Models:

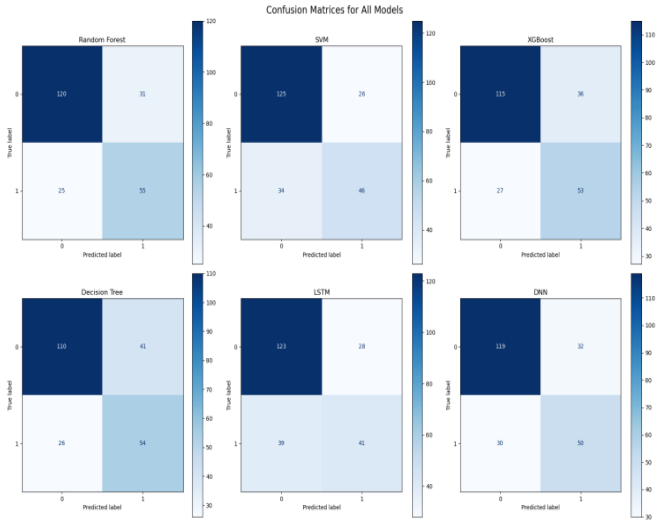

Fig. 9. Confusion matrices for all models, including Random Forest, SVM, XGBoost, Decision Tree, LSTM, and DNN. The matrices show the probability distributions of true positives, true negatives, false positives, and false negatives for each model

Conclusion

This study shows the efficacy of a two-stage Approach for diabetes prediction. Random Forest appears as the best ML model for balancing interpretability and performance, Whereas, LSTM provides more accuracy at the expense of computing resources. Future research will look into hybrid models and more datasets to increase forecast accuracy.

REFERENCES

1. International Diabetes Federation, "IDF Diabetes Atlas, 10th Edition," 2022. [Online]. Available: https://diabetesatlas.org/.
2. Z. C. Lipton, "The Mythos of Model Interpretability," ar Xiv:1606.03490, 2016.

3. Ahmad A., S. Ashfaq, and M. Ali, "Random Forests and Their Applications in Predicting Diabetes Mellitus," *Journal of Healthcare Informatics Research,* vol. 3, no. 2, pp. 101–110, 2019.
4. Mandal P., R. Bhattacharya, and A. Chatterjee, "Decision Tree-Based Predictive Model for Diabetes Mellitus," *International Journal of Biomedical Engineering and Technology,* vol. 15, no. 4, pp. 267–280, 2020.
5. Patel R., T. Sharma, and D. Gupta, "Support Vector Machine for Diabetes Prediction: A Comparative Study of Kernel Functions," *Proceedings of the International Conference on Computational Intelligence and Data Science (ICCIDS),* pp. 123–130, 2021.
6. Chen T. and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD *International Conference on Knowledge Discovery and Data Mining,* pp. 785–794, 2016.
7. Hossain M., F. Rahman, and T. Kabir, "Deep Neural Networks for Diabetes Prediction: A Comparative Study," *Computers in Biology and Medicine,* vol. 135, pp. 104–112, 2021.
8. Tripathy N. et al., "A Comparative Analysis of Diabetes Prediction Using Machine Learning and Deep Learning Algorithms in Healthcare," IEEE Access, vol. 11, pp. 12345–12356, 2023.
9. Yadav A. K. et al., "A Study on Non-invasive Diabetes Causing Variables and Their Covariance Relationship in Diabetes Prediction Using Machine Learning Algorithms," *Smart Trends in Computing and Communications,* pp. 365–375, 2024.
10. Rajkomar A. et al., "Ensuring Fairness in Machine Learning to Advance Health Equity," *Ann. Intern. Med.,* vol. 169, no. 12, pp. 866–872, 2018.

****