Interpretable Deep Learning Models for Improved Diabetes Diagnosis

Vadada Yamuna,Durvasula Venkata Sai Bhavana,Amiti Sudhish,Animireddy Tharunima

Introduction:

Diabetes, a chronic condition marked by persistent high blood sugar, poses major global health challenges due to complications like cardiovascular disease and neuropathy. Traditional diagnostic methods, though common, are invasive, time-consuming, and prone to interpretation errors. To overcome these issues, this project proposes a novel machine learning framework that integrates structured data (e.g., demographics, test results) and unstructured data (e.g., retinal images, clinical notes) using deep learning models like CNNs, RNNs, and transformers. Explainable AI techniques, such as SHAP and attention mechanisms, are incorporated to make predictions transparent and trustworthy. An interactive diagnostic tool allows clinicians to explore model insights, enhancing adoption in real-world settings. With continuous learning capabilities, this framework aims to improve diagnostic accuracy, personalize treatment, and reduce healthcare burdens.

Abstract:

The existing system leverages multiple machine learning models, including Decision Tree, K-Nearest Neighbor, Support Vector Classification, and Extreme Gradient Boosting.The proposed system leverages a deep neural network (DNN) to capture complex patterns in medical data, further enhancing diagnostic precision.The developed system integrates an interactive interface, designed to assist healthcare professionals and patients in understanding predictions with clarity. This enables more informed decision-making by highlighting key contributing factors in a user-friendly manner.

Existing System:

The existing system for diabetes diagnosis primarily relies on traditional machine learning models such as Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Classification (SVC), and Extreme Gradient Boosting (XGB). These models were trained using structured clinical data from publicly available datasets. Among these, the XGB model demonstrated the best performance, showing higher accuracy and fewer false negatives compared to others. To enhance interpretability, SHAP (Shapley Additive Explanations) was used to explain the model predictions, and an initial self-explanatory interface was developed.

Challenges:

1. Limited Interpretability
2. Lack of Local Explanations
3. Dependence on Structured Data Only
4. Limited Use of Explainable AI (XAI)
5. No Adaptive Learning Mechanism
6. Basic User Interface

Proposed System:

The proposed system presents a Deep Neural Network (DNN)-based framework for the diagnosis of diabetes, designed to address the limitations associated with traditional machine learning methodologies. In contrast to earlier models that primarily depend on structured data and offer limited interpretability, the proposed approach leverages the DNN's capacity to learn complex, nonlinear patterns from high-dimensional datasets, thereby facilitating more accurate and robust predictions. The system incorporates an intuitive graphical user interface (GUI) that enables healthcare professionals to input patient data and receive predictive diagnostics in an accessible manner. Furthermore, to support real-time decision-making, the framework integrates Google's Gemini API, a state-of-the-art large language model (LLM), which facilitates real-time prediction and contextual interpretation of patient data. This integration enhances the system's responsiveness and adaptability in clinical environments. Overall, the framework combines predictive accuracy, interpretability, usability, and real-time inference, offering a practical and intelligent tool for early diabetes detection and personalized medical decision support.

Pre Processing Steps:

- **Dataset Loading:** Loaded the PIMA Indian Diabetes Dataset using pandas.
- **Label Encoding**: Convert categorical variables (gender, smoking_history) to numeric using LabelEncoder.
- **Missing Value Handling:** Drop or fill missing NaNs using mean/mode.
- **Feature Scaling:** Applied StandardScaler to normalize numerical features.
- **Data Splitting:** Divided the dataset into training and testing sets using train_test_split.

Literature Survey:

1. "A novel machine learning approach for diagnosing diabetes with a self-explainable interface"(2024) by Gangani Dharmarathne,Upaka Rathnayake - This research presents a machine learning model for diabetes diagnosis with a self-explainable interface, enhancing interpretability for healthcare professionals while ensuring high prediction accuracy.
2. The research article "Deep Learning-Based Diabetes Prediction Using Medical Data" (2020) by Dr. P. Kumar, S. Mehta, and R. Verma - The study uses CNNs for diabetes prediction from medical records, highlighting explainable AI for better decision-making.
3. The study "A Hybrid Deep Learning Framework for Diabetes Detection" (2019) by R. Singh, M. Patel, and N. Kaur - The study combines CNNs and Random Forest for diabetes diagnosis, using medical imaging and health indicators to enhance interpretability in AI diagnostics.
4. "Deep Learning for Medical Diagnosis: A Focus on Diabetes" (2019) by H. Kim, T. Park, and S. Choi - The study explores ResNet and DenseNet for diabetes detection and uses Grad-CAM to enhance AI transparency in medical imaging.

Objectives:

1. Enhanced Accuracy – Develop a Deep Neural Network (DNN) to capture intricate patterns for improved diabetes diagnosis.
2. User-Friendly Interface – Design an intuitive interface for easy interpretation by healthcare professionals and patients.
3. Actionable Insights – Provide early diagnosis support and personalized treatment recommendations.
4. Clinical Data Integration – Train on diverse datasets for better generalization across demographics.
5. Scalability – Build a framework extendable to other medical conditions.

Functional Requirements:

1. Data Preprocessing
2. Feature Extraction
3. Defining the DNN Architecture
4. Splitting Data (Train-Test Split)
5. Prediction & Decision Support
6. User Interface & Reporting

HARDWARE REQUIREMENTS

1. Processor(i5 or i7)
2. Minimum 8 GB RAM
3. Minimum 256 GB SSD
4. Graphics Card
5. Monitor

SOFTWARE REQUIREMENTS

1. Windows 10/11,macOS
2. Operating System
3. Programming Language:Python
4. Data Analysis Libraries (Pandas,NumPy)