**CHAPTER 1**

**INTRODUCTION**

In this chapter, problem description, objectives of the project, outcomes of the project and scope of the project have been discussed.

* 1. **PROBLEM DESCRIPTION**

At least one family member is likely to have diabetes, which is becoming more and more prevalent. It happens when the body is unable to use insulin efficiently or does not produce enough of it. By absorbing glucose for energy, the pancreatic hormone insulin aids in blood sugar regulation. Blood sugar rises when insulin function is compromised, which can cause major health problems. Type 1, Type 2, and gestational diabetes are the three forms of the disease. Insulin-producing cells are attacked by the immune system in type 1, forcing lifelong insulin injections. Gestational diabetes develops during pregnancy and raises the risk of Type 2 diabetes later on, whereas Type 2 is frequently associated with obesity and inactivity. Predicting blood glucose levels in patients with Type 1 diabetes is the main goal of this study. While black-box models identify patterns but are not interpretable, traditional physiological models offer insights but are rigid and inaccurate. In order to improve predictions, we combine black-box models (LSTM, GRU, and TCN) with a physiological white-box model (Adaptive Single Component Metropolis Hastings). Key features can be extracted using simulated data or real-time patient data from CGM devices. Early intervention is also made possible by the use of HbA1c to compute the average glucose levels over a three-month period. Using glucose and weight, a fuzzy logic algorithm processes 20 preset rules to reliably manage diabetes. Better patient outcomes result from this hybrid approach's increased accuracy and interpretability.

* 1. **OBJECTIVES OF THE PROJECT**

The main objectives of the project are.

* To develop a predictive model for glucose levels Prediction using deep learning techniques.
* To design an algorithm for providing timely predictions for diabetes management.
* To integrate data visualization tools to provide clear and comprehensive visual feedback on glucose level trends and patterns, aiding in better understanding and management for both patients and healthcare providers.
* To predict the three months average glucose levels based on past values of each and every patient data using hba1c formula.
* To develop an fuzzy logic based algorithm, which effectively computes the severity order using the glucose and weight with the help of 20 preset rules, which reliably manages diabetes.
  1. **OUTCOMES OF THE PROJECT**

The outcomes of the proposed system are.

* Predictive Model Development: Achieve a highly accurate and robust deep learning model for forecasting blood glucose levels, validated with patient data.
* Designed an algorithm that collect and preprocess CGM data, insulin doses, and food intake for timely glucose predictions using a trained deep learning model.
* Improved data sharing and visualization tools will facilitate better communication between patients and healthcare providers, leading to more effective and personalized care.
* Prediction of three months average glucose levels based on past values using hba1c calculation.
* Developed an fuzzy based severity order classification algorithm, which computes the severity order for each patients using their respective weight and glucose levels by defining 20 preset rules, which reliably helps in managing diabetes.
  1. **SCOPE OF THE PROJECT**

This study's focus is on using a hybrid approach that combines both black-box and white-box models to predict blood glucose levels in patients with Type 1 diabetes. The study uses the Adaptive Single Component Metropolis Hastings model to provide physiological insights and black-box models like LSTM, GRU, and TCN to find patterns in blood glucose fluctuations. The study's data sources include simulated data for model training and validation in addition to real-time patient data gathered from Continuous Glucose Monitoring (CGM) devices. In order to improve diabetes management, a fuzzy logic algorithm processes glucose and weight data using 20 preset rules, and HbA1c levels are examined to determine average glucose levels over a three-month period. Other types of diabetes, such as Type 2 or gestational diabetes, are not covered in this study, nor is it concerned with diabetes prevention or treatment outside of predictive modeling. Moreover, the scope does not include biomarkers like blood pressure or cholesterol. The study's goal is to increase blood glucose prediction accuracy and interpretability so that early intervention and improved patient outcomes are possible.

* 1. **REPORT OVERVIEW**

Chapter 2 deals with the existing techniques that prevails for the glucose level prediction for Type 1 diabetes patients.

Chapter 3 deals with the study of existing system and proposed system.

Chapter 4 deals with the methods and modules involved in the design of the proposed system.

Chapter 5 deals with the implementation methods used in various modules of the proposed system.

Chapter 6 deals with the results obtained from the proposed system and analysis of the obtained results.

Chapter 7 presents the conclusion of the work done and future enhancement for the system.

**CHAPTER** **2**

**LITERATURE SURVEY**

**2.1 OVERVIEW**

This section discusses the methodologies learned from previous works for our proposed system.

**2.2 Individualized Models For Glucose Prediction In Type 1 Diabetes: Comparing Black-Box Approaches To a Physiological White-Box One**

Giacomo Cappon *et.al.* explored the comparative evaluation, which described the comparative evaluation of the glucose prediction models for Type 1 Diabetes (T1D). It compares the Physiological white box model with Black box model and evaluated which was better one for predicting the glucose level based on the features.The main drawback in this research was the black box models were not interpretable enough, eventhough they gave good predictions, whereas the white box models had worse performance in making predictions especially in realtime predictions. So we have extended this by making this as a base for our research by taking important features from both the models and developed a hybrid approach utilized both black box and white box models efficiently.

**MERITS:**

* Both the White-box and the Black-box allow some layer of insights into how glucose or patients respond, with White-box being more valuable for understanding and the Blackbox offering high prediction accuracy and learning efficiency from larger datasets.
* To increase model’s real-time responsiveness, adaptive strategies become better as a patient’s condition changes, making room for dynamic adjustments to better manage.
* Together, these represent a better, more accurate method of predicting the body, and an improved comprehension of the body.

**DEMERITS:**

* Both the White-box and the Black-box allow some layer of insights into how glucose or patients respond, with White-box being more valuable for understanding and the Blackbox offering high prediction accuracy and learning efficiency from larger datasets.
* To increase model’s real-time responsiveness, adaptive strategies become better as a patient’s condition changes, making room for dynamic adjustments to better manage.
* Together, these represent a better, more accurate method of predicting the body, and an improved comprehension of the body.

**2.3** **The Ohio T1DM Dataset For Blood Glucose Level Prediction Update 2020**

Cindy Marling1 & Razvan Bunescu1 conducted a research study by collecting the data from 12 individuals over an 8 week period and continuously monitoring the glucose level and tracked their usage of insulin. For Physiological model, they monitored the patients wearing the fitness bands for heart rate and their physical activity. They also used their smartphone app to track their routine life events like meals, exercise etc. that could impact their glucose levels. It included comprehensive data collection and limited generalizability.

**MERITS:**

* The combined data set allows the use of a more comprehensive approach as it includes multiple data types, ICE, insulin data, physiological metrics, and self reported life events.
* Longitudinal 8 week duration allows for observation of trends and patterns in glucose levels in time and this provides the opportunity to observe potential predictors of glucose fluctuations and management of diabetes.

**DEMERITS:**

* As the sample size consists of only 12 participants, the findings might not be generalized to the entire population.
* The accuracy of the predictions and insights that result from the study may also be affected by lack of or skewed analysis due to data gaps, particularly the missing overnight data from some participants.

**2.4** **Deep Physiological Model For Blood Glucose Prediction in T1DM Patients**

Mario Munoz-Organero developed a deep learning model which united the physiological data by developing a deep learning model to predict a glucose level in Type 1 diabetes patients. It modeled the complex physiological interactions by utilizing deep neural networks. It improved the accuracy of prediction for T1DM patients through advanced modelling techniques. Here deep learning model was complex and may faced some challenges across diverse patient populations.

**MERITS:**

* Using a deep learning method, the glucose predictions are extremely accurate as it can successfully capture intricate physiological interactions.
* It provides deeper understanding on non pathophysiological regulatory factors of blood glucose level in T1DM patients.
* Additionally, complex modeling techniques enable the improvement of accuracy in prediction in turn enhancing diabetes management and tailored treatment plans.

**DEMERITS:**

* In order for the model to be easily trained however, we require a relatively large amount of high quality physiological data and hence data availability is important for performance.
* Deep learning models are complicated and require computation and optimization work in order to run, hence they require large amounts of processing power.
* Furthermore, since different patient populations with dissimilar physiological responses can also result in difficulty generalizing the model to the physical world, it may not be used in actual clinical settings.

**2.5** **Blood Glucose Prediction Model For Type 1 Diabetes Based On Artificial Neural Network With Time-Domain Features**

Ganjar Alfian *et.al.* performed a study on the research which has been done by collecting the real world glucose data from T1DM Patients. Time domain features including glucose trends and pattern over time were extracted for Model development. The Artificial Neural Network based model was used to accurately predict the blood glucose level and these performance was evaluated by performance metrics such as Mean Absolute Error(MSE) and Root Mean Squared Error(RMSE). ANN Model perform a robust performance in prediction , at the same time it was more complex and resource extensive.

**MERITS:**

* Prediction Accuracy of the ANN model is enhanced by paying attention to time domain features that provide better identification of glucose trends.
* It is effective in patterns storing long time axes, which makes it perfect for continuous glucose monitoring activities.
* The ANN model is robust in its performance, resulting in the ability to generate reliable predictions that may be leveraged in making the decisions regarding diabetes management.

**DEMERITS:**

* Prediction Accuracy of the ANN model is enhanced by paying attention to time domain features that provide better identification of glucose trends.
* It is effective in patterns storing long time axes, which makes it perfect for continuous glucose monitoring activities.
* The ANN model is robust in its performance, resulting in the ability to generate reliable predictions that may be leveraged in making the decisions regarding diabetes management.
  1. **A Comparison Of Machine Learning Algorithms For Diabetes Prediction**

Jobeda Jamal Khanam & Simon Y. Foo conducted a study on the research which has been done by comparing various machine learning algorithms for predicting diabetes and the performance was evaluated in each algorithm using standard metrics like accuracy, precision and recall. It provided a comparison of different machine learning algorithm for predicting diabetes and helped in identifying the most efficient algorithm. Here the performance can varied depending on the quality and content of dataset. For evaluating multiple algorithms , it required substantial computational resources model.

**MERITS:**

* Based on some performance metrics like accuracy, precision and recall, the study makes a complete comparison of different machine learning algorithms.
* It helps to identify the most successful one.
* It provides some of the information on what works for diabetes prediction and what doesn’t so that we can chose the right algorithm (i.e: will it help us with diabetes prediction or not).

**DEMERITS:**

* The algorithms’ performances also are affected by the quality and features of the dataset used, and hence, can contribute to the outcomes.
* The results could be limited because the study did not take in account the specific tuning and optimization procedures required for each algorithm.
* Computing many algorithms is a significant amount of computing power and may also be a deterrent when implemented in real time or in large scale.
  1. **A Personalized Blood Glucose Level Prediction Model With a Fine-Tuning Strategy: A Proof-Of-Concept Study**

Wonju Seoa *et.al.* performed a study by collecting the real world CGM(Continuous Glucose Monitoring) data like meal intake, Physical activity etc... The model was developed by initial training on a generic CGM dataset using regression models and neural networks. To adapt the model for individual glucose responses, it transferred learning for personalization, custom loss function for glucose response variability. The performances were evaluated by Mean Absolute Error(MSE) and Root Mean Squared Error(RMSE) metrics. Here Prediction accuracy was improved through Personalized fine-tuning. Advance Machine learning techniques lead to the complexity of model and it was not efficient without sufficient patient data.

**MERITS:**

* Prediction accuracy is increased significantly while also making the model more suitable for each particular data of each patient.
* In fact, the contextual features such as dose of insulin, physical activity, or intake of meal improve the accuracy and relevancy of glucose level predictions as compared to other models.
* In the process of strengthening the flexibility of model’s response to individual variability this method improves real time diabetes management.

**DEMERITS:**

* Prediction accuracy is increased significantly while also making the model more suitable for each particular data of each patient.
* In fact, the contextual features such as dose of insulin, physical activity, or intake of meal improve the accuracy and relevancy of glucose level predictions as compared to other models.
* In the process of strengthening the flexibility of model’s response to individual variability this method improves real time diabetes management.
  1. **Blood Glucose Prediction With Deep Neural Networks Using Weighted Decision Level Fusion**

Hatice Vildan Dudukcu *et.al.* performed a study for predicting the glucose levels, in which blood glucose levels were predicted by utilizing deep neural networks combined with weight decision level fusion. Also multiple prediction models and applied fusion techniques were integrated to improve accuracy and robustness performance. Deep neural networks effectively capture complex patterns in glucose data. For training and fusion process, it requires significant computational resources and the performance heavily depended on the weight of fusion and the quality of input data.

**MERITS:**

* Weighted decision level fusion is achieved by thoroughly integrating a number of different models, each of which predisposes to helperly contributing its own set of strengths.
* Given their qualities as complex pattern identifier deep neural networks are particularly good at identifying complex patterns in glucose data, thereby increasing the model’s ability to predict variances.
* Furthermore, this method also further enhances the robustness and dependability of the predictions, and is thus more suitable for diabetes in the real time time management.

**DEMERITS:**

* The deep neural networks, in particular, may be resource intensive (i.e., they require a lot of computational power).
* Another possible limitation to the model's openness to clinicians is that its intricacy might be too much to understand and change.
* low data quality or choosing the wrong fusion weights will negatively affect the model accuracy.

**2.9 An Ensemble Machine Learning Approach For Predicting Type-I Diabetes Mellitus Based On Lifestyle Indicators**

Shahid Mohammad Ganie & Majid Bashir Malik applied an ensembled Machine learning using lifecycle activities such as diet, exercise etc. for predicting type 1 diabetes. It improved the accuracy of prediction and robustness performance. It utilized an ensembled approach to reduce overfitting as well as to enchance prediction accuracy. Performance might have varied based on the quality and relevance of lifestyle data.

**MERITS:**

* Using ensemble approach to warm the dataset for prediction, it has successfully lowered possibility of overfitting that leads to high prediction accuracy and enhances the models generalizability to the different dataset.
* The inclusion of lifestyle indicators that can be used to predict Type-I diabetes form a deeper understanding of the factors that precipitate diabetes.
* Combination of the use of several models also results in a more solid and trustworthy prediction framework.

**DEMERITS:**

* The quality and the relevance of the lifestyle data used could affect the model performance and it may cases deliver suboptimal predictions when associated with the features of a low quality or irrelevance.
* The complexity of the method then might increase, as the method can require a lot of preprocessing as well as feature selection to guarantee that the models are trained successfully.

**2.10 Short-Term Prediction Method Of Blood Glucose Based On Temporal Multi-Head Attention Mechanism For Diabetic Patients**

Yang Guanci et.al. performed a study to predict the shortterm blood glucose level by implementing a temporal multi head attention mechanism. To enchance the prediction accuracy it utilized various historical glucose data and attention mechanisms. It was crucial for timely diabetes management as it focusses more on short term glucose level predictions. It effectively captured the temporal glucose data. The performances could have varied depending on the different type of glucose monitoring devices.

**MERITS:**

* + With the help of sophisticated attention mechanisms, the model is able to focus on the most important historical data to obtain high prediction accuracy and to detect short term glucose trends.
  + This method is suitable for control of diabetes in real time, where accurate prediction of short term is needed to make changes in treatment, based on its more robust and dependable prediction model.

**DEMERITS:**

* + Integrating and fitting attention mechanisms to other elements usually is difficult and takes a lot of resources to do, ultimately requiring a lot of processing power.
  + Thus, the historical glucose data is a determining factor on how well the performance of the model will perform, as it can decrease its performance based on inadequate or missing glucose data.
  + In order to optimize the data for making precise predictions, the model might require a lot preprocessing and feature selection, so the entire process would be not only more complicated, but also taking more time as well.

**2.11 A Novel Machine Learning Approach For Diagnosing Diabetes With A Self-Explainable Interface**

Gangani Dharmarathne *et.al*.focused more on Interpretability, so they have developed a machine learning model for diabetes diagnosis. It provided transparent and understandable predictions by implementing a Self explainable interface. With advanced machine learning techniques it potentially improved diagnostic accuracy. It gave clear explanations of diagnosis decisions and might have been limited by the quality and representiveness of training data.

**MERITS:**

* + The model improves interpretability and transparency thus helping users understand the reasons for the diabetes diagnosis.
  + Thus, supporting patients and healthcare practitioners with concise justifications for choices for diagnostics enhances the user experience and fosters more trust.
  + It may be possible to include cutting edge machine learning techniques in order to provide more accurate diagnoses.

**DEMERITS:**

* The overall usability and ease of explanation of the self explanatory interface depends on the complexity of the model as it becomes harder for some of the users to explain to them.
* The creation and maintenance of such an interface requires significant computational resources for which the development and operating expenses might increase.
* This could also be the case for the model and the relate performance since the training data is of poor quality.
* Not representativeffecting on the predictability and equity of the results.

**2.12 Enhanced Blood Glucose Levels Prediction With A Smartwatch**

Sean Pikulin et.al. performed a study on the research which had been done by collecting the physiological data using smart watch sensors. Based on the smart watch data , they had developed a predictive model. using physiological data it provides real time glucose predictions. Accuracy might have varied depending upon the quality and type of data taken from smartwatch and also it had the issues with data privacy and security.

**MERITS:**

* + The integration of wearable technology allows users to have continuous glucose monitoring, thus, giving them the convenience and non invasiveness to monitor their blood sugar levels.
  + The predictive model is a helpful tool for managing diabetes since it has the ability to provide real time glucose estimates based on the physiological data collected.
  + Such method adds value for the user as it eliminates the need for finger-prick tests by allowing more frequent monitoring.

**DEMERITS:**

* The smartwatch’s sensor capabilities and the accuracy level of data it records can limit the model’s opportunity to make accurate predictions.
* The idea of data security and privacy is always a concern, since the sensitive health data gets monitored continuously.

**2.13 Generative Adversarial Networks For Improved Glucose Forecasting**

Deepjyoti Kalita et.al. utilized a Generative Adversarial Network(GAN) architecture which leads to better glucose forecasting using high-fidelity data, which would lead a better clinical outcomes. It uses both real and synthetic data for improving the accuracy of the glucose forecasting model, by achieved lower RMSE than previous methods,but it’s architecture had increased computational complexity.

**MERITS:**

* The results of the GAN based method is better forecasting accuracy, with lower RMSE, compared to conventional techniques.
* It generalizes better on a range of diverse patient profiles than previously described and the use of synthetic data augmentation overcomes the problem of small clinical datasets.
* High fidelity forecasting improves the clinical decision making in diabetes management and may lead to reduce the risk of hypoglycemic and hyperglycemic events.

**DEMERITS:**

* The GAN architecture introduces a lot of computational complexity, and the resulting hardware resources have to be quite powerful.
* Maintaining stability during training of the adversarial network is hard, and the process of generating synthetic data does not reproduce all nuances of real physiological variability.
* The model is resource intensive in nature and is not feasible to maintain and interpret results from clinical settings.

**2.14 Meta-Learning Methods For Limited Data Glucose Prediction**

Federico D’AntonI et.al. improved the prediction performance by training only the meta -learner with limited amount of data, but including additional parameters beyond CGM did not significantly improved the performance.

**MERITS:**

* The meta learning approach provides efficient glucose prediction capabilities that are applicable to clinical practice even with sparse patient specific data.
* The approach is especially promising in situations in which there is not complete historical data available, but still for personalized diabetes management.
* Since the meta learner component is trained, this increases computational efficiency.
* A real valuable information about how to pick parameters for the glucose prediction model is provided by the study.

**DEMERITS:**

* The other physiological parameters, such as adding the physiological parameters in addition to CGM data, do not considerably increase prediction accuracy that may limit the scalability of the method.
* Meta learning is a field of research, so there is likely more technical know-how needed for clinical implementation. More investigation was needed to make sure generalizability across a wide range of patient populations.
* The approach may struggle in situations where glucose is extremely volatile and a usual amount of data would be needed.

**2.15 Domain-Specific Constraints In Generalized Multivariate Glucose Forecasting**

Zhendong Wang *et.al.* proposed a generalized multivariate forecasting for predicting the glucose levels, utilizing the method called ”COMET”, which has three domain-specific constraints for diabetic patients, but it couldn’t incorporate the clinical experts in assessing the effectiveness and relevance in glucose forecasting.

**MERITS:**

* Domain specific constraints and multivariate data integration are used to increase glucose prediction accuracy as compared with COMPET, which maintains clinical plausibility.
* It is able to be used with a wide number of patients and perform some degree of personalization, but still remains efficient to compute, which in turn allows for its real time use.

**DEMERITS:**

* The model has a lot of limitations with which it has a hard time handling the cases that are not standard and doesn’t have validation from the clinical experts.
* For patients with varying metabolism, its generalized approach may not function properly, and the input data has a high sensitivity on prediction accuracy.

**2.16** **LSTM Networks For Predicting Blood Glucose In Time Series Data**

Sadegh Mirshekarian *et.al.* compared LSTM networks for predicting the blood glucose levels, as it was good for time series data. Since it could learn from multiple variables, it addresses the vanishing gradients, but it needed a large datasets for accurate prediction, which were not always available

**MERITS:**

* It is shown that LSTMs have a higher ability to find dependencies among glucose time series data than regular models.
* Their design is an efficient way to consider all the input variables together in order to identify those variables influencing blood sugar levels.
* There is no need to introduce additional gates for preventing gradient related training problems, since the built in memory gates can still maintain large amount of historical patterns.

**DEMERITS:**

* The method requires a large amount of high quality training data, which is often not present in actual clinical setting.
* Model performance can be seriously hampered when the data in the real world is sparse or noisy.
* LSTMs may be infeasible due to a higher computational complexity than other machine learning models

**2.17 LSTM And Attention-Based Models For Blood Glucose Prediction Comparatively Analyzed**

Sadegh Mirshekarian *et.al.* proposed a study on the research which effectively compared the LSTM and attention-based neural networks for predicting the BG levels for T1D patients using real and synthetic datasets, while attention mechanisms provided improved accuracy, it required more time and might not have generalized the data for predicting the blood glucose levels.

**MERITS:**

* These models effectively put the weight over the pertinent physiological relationships and temporal patterns, leading to superior glucose prediction accuracy.
* This comparative framework gives critical info about the best structure alternative for various clinical information conditions.
* Using real and also synthetic datasets, the results are more robust and the clinical benefits of the attention mechanism make it more interpretable than conventional blackbox methods.

**DEMERITS:**

* Among all known (attention) mechanisms, those are significantly more computationally intensive and require significantly more computational time and resources to achieve the same level of accuracy.
* The models are also not very generalizable in terms of the data to which they are trained, i.e., particularly when trained on artificial data.

**2.18 Type 1 Diabetes Patients' Computational Identification Tool**

Roberto Visentin *et.al.* designed a computational tool, which was specifically designed for identifying the Type 1 diabetes patients. If provided a cost-effective and time saving methodology for clinical trials, but the mathematical models used here was very difficult to understand.

**MERITS:**

* The tool offers substantial clinical research setting benefits by drastically cutting down on the time and expense retained in patient recruitment for T1D trials.
* This automated screening process is more effective to discover qualified trial participants than conventional manual methods.
* The use of the computational approach may decrease selection bias in clinical studies, as it provides standardized and objective criteria for patient selection.

**DEMERITS:**

* The input data quality will affect how well the tool captures complex presentations of T1D.
* An updating of the system is necessary with the addition of the current diagnostic standards and biomarkers.

**2.19 Bayesian Parameter Estimation for Type 1 Diabetes Modeling**

Giacomo Cappon *et.al.* used a bayesian framework for estimating the parameters of Type 1 diabetes using the available patient data. It addresses the issues by incorporating the Bayesian estimation with Markov chain Monte Carlo to provide better estimation of parameters and interval, however this methodology was best suitable for synthetic data, not on real data.

**MERITS:**

* The Bayesian framework softens Bayes parameter estimates so that although they are called such, the parameter estimates are only remotely close and the parameter intervals are quantified.
* When patient data is scarce, past knowledge provides additional gain in estimation efficiency.
* The MCMC implementation does a good job of handling complex, nonlinear glucose-insulin physiology.
* In particular, the technique is especially well suited in simulation settings for which data is finely controlled.

**DEMERITS:**

* The method is limited, however, in its efficacy when applied to actual clinical data because of the noise and measurement variability.
* MCMC sampling is infeasible for real time applications because of its high computational complexity.
* The results are highly affected by data quality problems common in ambulatory diabetes monitoring.

**2.20 Multi-Subject Learning For Enhanced Blood Glucose Prediction**

John Daniels *et.al.* improved the blood glucose predictions by utilizing the data from multiple subjects, which enhanced the model using limited specific data features. It performed better than traditional method, but the training complexity was very difficult, and also it required a careful training to avoid the overfitting.

**MERITS:**

* The multi subject approach to prediction offers great improvement in prediction accuracy compared to single subject models, particularly in patients when there is little historical data available.
* Population patterns are used to make the approach more efficient in lessening the need of lengthy individual monitoring periods.
* The shared feature learning helps the model be resilient to the measurement noise and individual data variability.
* The method significantly outperforms conventional personalized modeling methods when the patient data are sparse.

**DEMERITS:**

* During training, it spawns complexity and it is a painstaking task to get to the model and hyperparameters that perform well.
* For implementation, you need significant computer power and knowledge of multi-task learning strategies.
* The model may be difficult to handle when there are patients with highly atypical physiological responses that are much different than population patterns.

**2.21 Mealtime Insulin Bolus Prediction Ensemble Method**

Giulia Noaro *et.al.* proposed an ensemble method for mealtime insulin bolus, which combined the two models. It had been trained using a synthetic data of 100 virtual meal time objects, but it doesn’t provides multiple sources such as CGM, CHO, and also it doesn’t use the real data by leveraging the proposed in-silico framework for addressing the T1D management.

**MERITS:**

* The ensemble approach is generally more robust to predict the dose of insulin compared to single model approaches.
* By incorporating synthetic training, the overall framework does not incur any clinical risk while making it possible to perform comprehensive scenario testing.

The method is shown in virtual patient modeling to be a platform for personalization.

**DEMERITS:**

* Synthetic data can be used only in limited applications as the virtual patient cannot represent the whole physiological complexity.
* When actual CGM and CHO data sources are excluded, clinical relevance is decreased. The model does not confirm with real patient outcomes.

**2.22 SUMMARY**

These works provides a significant advancements in blood glucose level prediction for T1D patients, which utilized both traditional machine learning and deep learning techniques. Here their lack of interpretability is a drawback. We have already discussed that physiological models are interpretable but lacks in accuracy, whereas for time-series real time data, deep learning approach is the best one. So we have proposed a hybrid layered approach which integrated both the models by taking important features, which enables better prediction of blood glucose levels, which also reduces the risks associated with T1D.

**CHAPTER 3**

**SYSTEM STUDY**

**3.1 OVERVIEW**

The main objective involves designing an optimized fuzzy-based personalized deep glucose level prediction system intended for patients with Type 1 Diabetes (T1D). Blood glucose level prediction accuracy enhancement serves as the main purpose to achieve effective diabetes management while preventing severe medical problems.

The proposed system combines physiological models (white-box) with deep learning models (black-box) within its framework. The combined research approach exploits the individual strengths of both white-box and black-box methods to generate reliable interpretation of blood glucose predictions.

Information enters the system through Continuous Glucose Monitoring (CGM) data as well as both insulin injection data and human consumption of carbohydrates. Before making predictions the data needs preprocessing to achieve better accuracy along with data adaptability.

Modeling follows the inspiration of UVA/Padova T1D simulator by integrating insulin absorption alongside glucose absorption plus glucose-insulin kinetics.

The deep learning system utilizes three models including Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU) and Temporal Convolution Networks (TCN) which extract advanced temporal patterns from time-series datasets.

Fuzzy Logic provides clear predictive results about diabetes severity by processing both glucose levels and weight measurements.

**3.2 EXISTING SYSTEM**

Glucose dynamics receive better clarification through the existing traditional physiological systems which provide limited flexibility together with imprecise real-time forecasting capabilities.

These models maintain complex procedures that prevent healthcare staff from easily understanding their operational method wherefore they demonstrate limited effectiveness in individualized diabetes treatment.

Deep Learning technological models analyze past data patterns yet remain difficult to interpret since they operate as unclear systems.

The main clinical limitation of these models stems from their inability to provide explanations of how treatment decisions are determined because of their high accuracy rate.

Traditional systems fail to update properly when handling human medical data during genuine medical conditions.

The predictive ability of black-box models remains unapparent for interpreting diabetic physiological processes that healthcare providers must understand for successful diabetes treatment.

**3.3** **STEPS INVOLVED IN GLUCOSE LEVEL PREDICTION IN EXISTING SYSTEM**

* Data Collection
* Data Preprocessing
* Feature extraction
* Implementation of Physiological model(Particle filter, 3 proposed subsystems)
* Deep Learning Model Development(LSTM,GRU,TCN,rAGX)
* Model Training and Evaluation
* Comparison between White Box and Black Box Model

**3.4 PROPOSED SYSTEM**

The system proposes the method of predicting blood glucose level of Type 1 Diabetes (T1D) patients through the process of data collection: data are collected from various sources, i.e. Continuous Glucose Monitoring (CGM) devices, insulin intake records and diabetics carb consumption logs. Glucose levels can vary over a long time, and therefore this data is collected over a period in order to capture this variation successfully. After data collection, data are preprocessed, which means to align the data that come from diverse sources to the uniform time frame (for example, every 5 minutes) with interpolation techniques. Common methods are used to replace missing glucose readings, and if there is any missing meal or insulin records, then it will be handled properly. Furthermore, numerical features are normalized to standard scale e.g. Min-Max normalization, so they are ready for model training.

Then, the system extracts useful features from the preprocessed data, which are moving averages of glucose readings on various time windows, rates of change in the glucose levels, additional lagged values of glucose, insulin, and meal data points to predict the future glucose. The implementation of a white box physiological model to simulate the glucose-insulin dynamics through differential equations implementing subsystems for insulin absorption, oral glucose absorption and glucose-insulin kinetics has been done as a part of the model development phase. At the same time, a series of black box models aiming at prediction using advanced machine learning algorithms including Long Short Table Memories (LSTM) Networks, Gated Recurrent Units (GRU), and Temporal Convolution Networks (TCN) are simultaneously built.

Once the models have been defined, they can be trained and evaluated using the preprocessed data and extracted features by training the models using the trained data and then evaluating the models using Measures like Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and R squared Score.

Then the models are integrated, as the outputs of proposed physiological model are added into the prediction of the deep learning models to further enhance accuracy using the outputs of the proposed physiological model as input into the black-box models.

For a more accurate prediction, prediction horizon optimization is performed through a genetic algorithm to determine the time frame in which to make glucose level prediction. Another thing is that the HbA1c is calculated to find out how much of the sugar is graphed out in the blood or the average blood glucose level of the past few months so that you know what its control is over the long run. Using predefined rules to enhance the interpretability, the system also is equipped with a fuzzy logic based architecture to predict the severity of diabetes based on glucose level and weight using a set of rules. The system finally generates output predictions of blood glucose levels and severity classifications as actionable insights for diabetes management.

**3.5 STEPS INVOLVED IN PROPOSED SYSTEM**

The proposed system has the following steps:

* Data Collection
* Data Preprocessing
* Feature Extraction
* Estimating Best Parameters by Adaptive Chain Metropolis Hastings for white box model
* Physiological White box model implementation
* Integrating the White Box Output with Deep Learning Models(LSTM, GRU, TCN)
* Hba1c Calculation for future average glucose level trends of each patient
* Fuzzy based Severity Order Estimation

**3.6** **PROCEDURE FOR PHYSIOLOGICAL WHITE BOX MODELS**

* Estimate the Best set of parameters by running the Adaptive single Chain Metropolis Hasting algorithm for upto 1000 iterations.
* Pass the Estimated Best Parameter to the traditional physiological white box model, which comprises three sub-systems(Glucose Absorption, Insulin Absorption, Glucose - Insulin Kinetics)
* Implement the physiological white box model for retrieving the best features and integrating it with black box model to propose an hybrid layered approach.

**3.7** **PROCEDURE FOR DEEP LEARNING(BLACK BOX) MODELS**

* Collect a Continuous Glucose Monitoring Level(CGM) data, insulin intake, Carbohydrate intake, bolus and basal insulin data along with the time-stamp for time-series forecasting.
* Preprocess the data extracted by neglecting the null values, applying Min-Max Normalization, align according to the time-series data for better prediction of glucose level.
* Define a LSTM, GRU, TCN deep learning models with suitable units of layer, activation function for accurately predicting the glucose level based on time-series forecasting.

**3.8** **PROCEDURE FOR BUILDING AN HYBRID MODEL**

* Gather all the data from dataset and preprocess the dataset by handling missing values and scaling features if necessary.
* Attach the additional feature extracted from the physiological white box model for improving the accuracy of prediction.
* Split the dataset into training and testing sets using cross-validation techniques.
* Train the selected models and evaluate their performance and predict the future glucose level based on optimized time horizons.

**3.9** **PROCEDURE FOR HbA1c AND FUZZY BASED SEVERITY ORDER CALCULATION**

* Prepare the dataset containing continuous Glucose Level Monitoring(CGM), basal and bolus insulin intake, carbohydrate intake and the corresponding weights of patient data for each patient IDs.
* Apply the Hba1C calculation over the period of three months for computing the future average glucose level trends for each patient.(Must be > than 6.5 for type -1 Diabetes).
* Implement the fuzzy based severity order calculation, by predefined set of 20 rules formulated by combination of 4 categories of weight and 5 categories of glucose level, which identifies the confidence level, based on the confidence level, the severity order is classified for each patients.

**3.10 SUMMARY**

This chapter describes the system study for the optimized blood glucose level prediction system. The project's goal is to mitigate and take precautious measures for Type 1 Diabetes Patients. It entails combining data processing, deep learning methods, and physiological methods and genetic algorithms to develop a predictive deep glucose level model at optimized time horizons. The project's goal is to build a best optimized predictive deep learning model, which enhances the prediction by describing the severity order using fuzzy based severity order algorithm which utilizes the glucose and weight and compute the confidence level using predefined set of rules.

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 OVERVIEW**

This Chapter mainly focuses on the Architectural system design of the proposed layered approach, which combines both the traditional physiological model and deep learning models. The entire architecture design is based on the data extracted from the ohio T1DM dataset. It mainly comprises Data collection such as Continuous Glucose level monitoring(CGM), insulin level(Bolus and Basal) and Carbohydrate intake. This extracted data undergoes preprocessing and feature extraction and finally the hybrid approach model has been developed by combining the features of white box with the black box deep learning model. The main objective of this system design is to ensure the Type 1 patients and helps them to control their glucose level to avoid hypoglycemia and hyperglycemia.

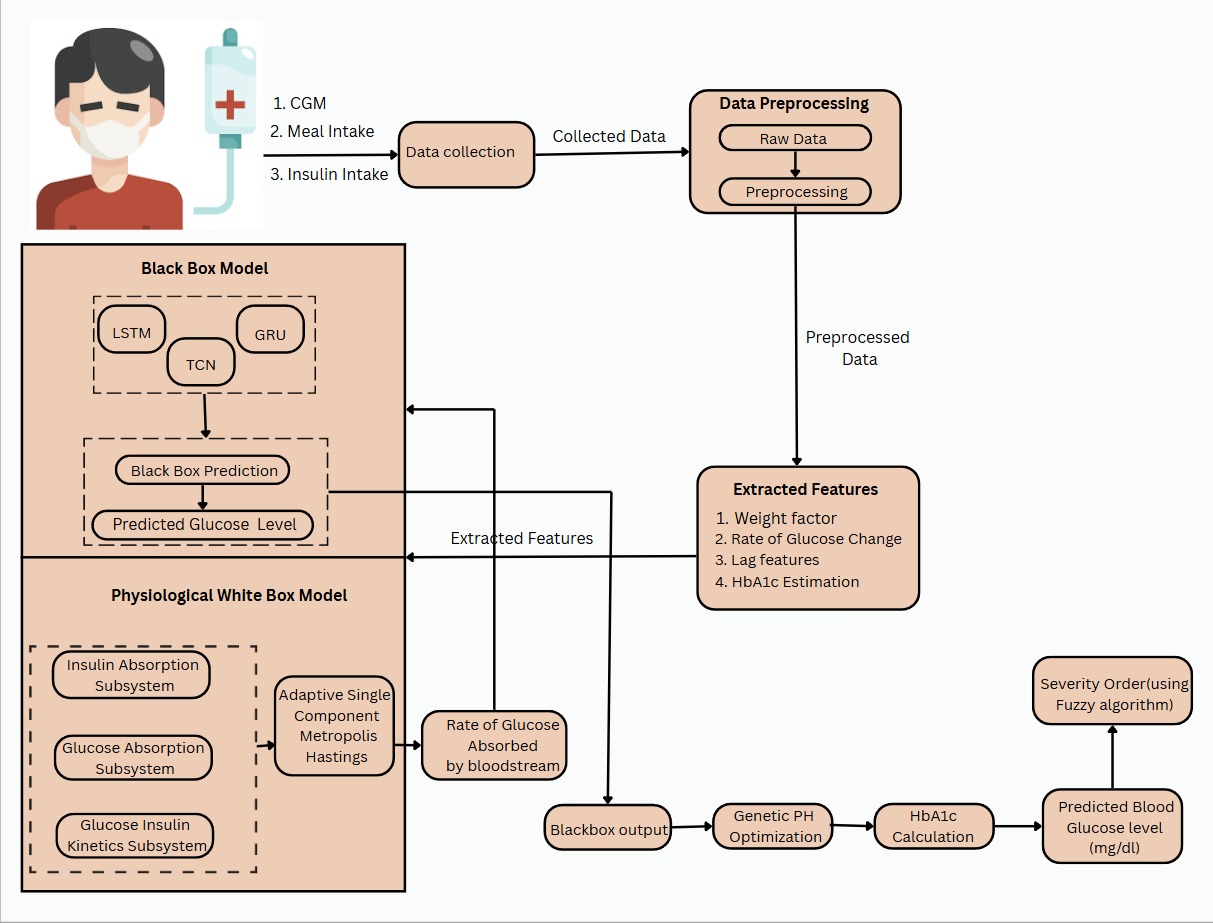
****

Figure - System Design

**4.2 SYSTEM ARCHITECTURAL DESIGN**

The system design gives an idea about the scheme by which the proposed system can be implemented. This will provide the overall flow of work and explains the various modules. Figure 4.1 shows the flow diagram of the proposed system. It consists of data collection, Data Preprocessing, Feature Extraction, Black Box(Deep Learning) and Physiological white box model, hba1c calculation and genetic algorithm for optimized time horizon and fuzzy based severity order classification.

The design of the hybrid glucose level prediction model for Type 1 Diabetes (T1D) patients is structured to allow the integration of physiological and learning machine approaches in such a way as to obtain accurate and interpretable prediction results. The data collection part starts with a module gathering data collected over longer periods of time with CGM devices, insulin intake records and carbohydrate consumption logs (e.g., 8 weeks) to get information on fluctuations in blood glucose levels. Subsequently, the data preprocessing module aligns different sources data at a common time frame (e.g. every 5 mins) using interpolation based techniques, cleans for missing data, and normalizes numerical features prior to training the model. In the feature extraction module, important features are drawn including: moving averages of glucose level, rates of glucose changes, lagged glucose, insulin, meal data points.

For the model development module, white box physiological model is used to simulate glucose insulin dynamics including subsystems of insulin absorption, oral glucose absorption and glucose insulin kinetics, while black box models are developed with advanced machine learning algorithms like Long Short Term Memory (LSTM), Gated Recurrent Units (GRU) and Temporal Convolution Networks (TCN) to learn from complex patterns of time series data. The output of both models is sequentially combined in the model integration module to increase the prediction accuracy. A prediction horizon optimization module works to optimally set a prediction horizon, which is the time frame utilized to predict the glucose level to optimize the prediction accuracy. Another is that HbA1c calculation module assesses long term glucose control as an average blood level of glucose over few months. A fuzzy logic system based severity prediction module is created to predict the severity of diabetes given the diabetic patient's glucose levels and weight using certain rules to make the interpretability of the system increased. Finally the output generation module generates final predictions of blood glucose levels and severity classification which will be able to give more actionable insights into the diabetic management.

**4.3 MODULE DESCRIPTION**

The project’s various modules are as follows:

* Physiological White Box Model
* Black Box Model
* Hybrid Model Integration
* HbA1c calculation
* Fuzzy based severity order classification

**4.3.1 PHYSIOLOGICAL WHITE BOX MODEL**

White box models are implemented by a nonlinear physiological model , it is a benchmark for comparison against Blackbox model by simulating glucose-insulin dynamics. Here, physiological equations are used to describe glucose insulin interactions, biologically based predictions etc.. The algorithms used here is Bayesian Estimation. In Bayesian Estimation, Markov Chain Rule Monte Carlo (MCMC) has been used to estimate the individual parameter, which uses Adaptive Single Component Metropolis- Hastings(SCMH). Using this algorithm, we can derive three subsystems which is used for predicting the glucose level. The physiological model functions are the standard method to understand glucose-insulin Subsystems. The model contains three main operational subsystems. They are Insulin Absorption Subsystem, Oral Glucose Absorption Subsystem and Glucose-Insulin Kinetics Subsystem. Insulin absorption subsystem, which models the movement of insulin through the body after it’s injected. It tracks the absorption of insulin accounting for injected insulin entering into the bloodstream. It utilizes different compartments for absorption of insulin such as monomeric and non monomeric form and Plasma Insulin concentration. This subsystem employs differential equations to track insulin flow between different body areas while it considers the influence of both degradation processes and the absorption rates. Oral Glucose Absorption System , which models the absorption of glucose from carbohydrates consumed during meals, focusing on how glucose moves through different compartments. The techniques used here de scribes the glucose absorption in a two compartment system (ie.) stomach, Intestine. Glucose moves through two storage compartments where it can be absorbed into the bloodstream before reaching the gut. This model briefly elaborate the effects of Carbohydrate Intake, Gastric Emptying, Intestinal Absorption Rate, Absorption rates with different parameters. A two-compartment stomach-intestine system with absorption rate parameters and gastric emptying models the absorption process. Glucose Insulin Kinetics subsystem represents interation between both the glucose and insulin for predicting the glucose transport and utilization in the body. It focuses mainly on how glucose levels change in response to insulin and how insulin sensitivity may vary over time. It describes how plasma glucose levels change due to insulin’s effects and elaborate the dynamic nature of the insulin sensitivity.

Overall, The Physiological White Box model gives interpretable predictions based on biological processes, which will present a better understanding of glucose behaviors.

**4.3.2 BLACK BOX MODEL**

In black box approach there are several advanced Machine learning algorithms, which are designed to capture complex patterns in a data. The specific algorithms used here are Long Short Term Memory(LSTM), Gated Recurrent Units(GRU) and Temporal Convolutional Network(TCN), which is best for time series data. It can capture complex patterns in the historical glucose data and predict from the real time inputs very accurately. LSTM is a type of recurrent neural network which is significantly used here to learn and remember over long sequences for time series data. GRU is similar to LSTM , another recurrent neural network variant which simplifies the architecture which performs sequential tasks. It is more efficient, as it has the capability to handle long dependencies in the data. TCN is specially designed for sequence modelling. It is not like RNN, Here networks are designed to handle data by utilizing the convolutional layers. It improves the training stability and performance. Time series prediction tasks are completely effective in temporal convolution networks. Therefore, this Module simulates glucose data with its complex patterns and then predict the values accurately through advanced deep learning techniques.

**4.3.3 HYBRID MODEL INTEGRATION**

The hybrid modeling framework integrates the outputs of the physiological model with deep learning models. The results taken from the physiological model such as rate of glucose absorbed by bloodstream, has been given as the additional inputs to the blackbox model, here deep learning models are used to predict the final glucose levels with better accuracy for type 1 diabetes patients. It is being able to predict real time based on what has already happened to them and with them. It dynamically adjusts the time frame for predictions i.e. Optimization of the prediction horizon by using a genetic algorithm based on Patient Specific factors. Thus, the combination strengthens the overall predictive accuracy while using interpretability of white box model and pattern recognition ability of black box model.

**4.3.4 HbA1c CALCULATION**

The HbA1c calculation is used here to calculate the average blood glucose level for the past few months. The work is based on the relationship between average blood glucose levels and HbA1c levels. It is said to be the most useful metric for predicting and managing diabetes. This Module is a crucial metric to assess the efficiency of the treatment plans and make the needed improvements to restore the patients back to optimisation.

**4.3.5** **FUZZY BASED SEVERITY ORDER CLASSIFICATION**

Fuzzy logic is used here to address the limitations in classifying the Severity in diabetes patients. Unlike other approaches, Fuzzy allows for a more progressive and understandable transition between severity levels. The main advantage for used here is to handle uncertainty. Here Fuzzy system takes glucose and weight as inputs and processes them through corresponding predefined membership functions. It defines five categories for glucose and four categories for weight. Membership functions are used when the inputs, such as glucose and weights, are processed into the fuzzy input system. Fuzzy rules used to classify the data allowing for more refined idea of patient conditions. There are predefined 20 fuzzy rules which determine the best severity level prediction. The fuzzy system facilitates a richer classification of glucose risk level by accounting for the diabetics' uncertainty in managing the disease.

Overall, the Fuzzy Based Severity Order Classification module enables the system to tolerate uncertainty in glucose measurements and hence makes it more reliable for clinical applications.

**4.4 SUMMARY**

This chapter describes the comparison between the existing system and the proposed system for Type-1 Diabetes Glucose level prediction. The designs as well as different kinds of modules involved with the workflow diagrams in developing the proposed system were discussed with detailed explanation.

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

In this chapter, various algorithms involved in implementing the modules described in the proposed system are discussed. The system implementation consists of the following modules:

* Data Collection
* Data Preprocessing
* Feature Extraction
* Traditional Physiological White Box Model
* Black box Model
* Hybrid Integration
* Genetic PH Optimization
* Hba1c Calculation
* Fuzzy severity order classification

**5.1 DATA COLLECTION**

To gather comprehensive and relevant data that will act as the core of the predictive models.

* **Continuous Glucose Monitoring (CGM) Data:**

CGM devices provide real-time measurements of interstitial glucose concentrations at regular intervals (e.g., every 5 minutes). This data captures the sudden changes/deviations in glucose levels, acting as a neccessary input for prediction.

****

**Fig 5.1 Continuous Glucose Monitor**

* **Insulin Intake**

Information about insulin intake, insulin dosage and time at which the Insulin intake by the person is neccessary, as excess insulin intake will seriously affect glucose levels. Data on both basal (long-acting) and bolus (rapid-acting) insulin dosage are used from dataset.

****

**Fig 5.2 Insulin**

* **Meal Intake Data**

Carbohydrate intake, meal intake, and time stamps are needed to model glucose absorption system from meals. This includes data from the glycemic level of foods, which influences the speed of glucose absorption.

**5.2 DATASET DESCRIPTION**

The dataset used in this research is the Ohio Type 1 Diabetes Mellitus dataset, which was updated in the 2020 release and is now referred to as OhioT1DM. The dataset is monitored with a Medtronic Enlite CGM system, comprising 12 subjects over a period of 8 weeks. Patients wore an insulin pump and a wearable system for measuring physiological variables such as skin temperature, heart rate, etc. It also collects additional information on meals, including meal intake time, amount, and type. Each subject in the Ohio T1DM dataset is divided into a training set (monitored for 8 weeks) and a test set (tested for the last 10 days). This dataset presents challenges for blood glucose (BG) predictive algorithms, as the glucose values recorded in daily life conditions are much more complex than those generated by simulation tools. Additionally, handling the data under real-time conditions raises technical issues. In the OhioT1DM dataset, there is a long portion of CGM readings, and an important aspect is that the sampling time is not homogeneous. Therefore, all signals are aligned to a sampling period of 5 minutes. For more information, please refer to the following link: Ohio Type 1 Diabetes Mellitus Dataset.

**5.3 DATA PREPROCESSING**

The raw data from Ohio Type 1 Diabetes Mellitus dataset is sent for preprocessing to ensure flexibility and reliability for better glucose level predictions.

* **Time Series Data and its Alignment:**

The data has been collected from Ohio T1DM Dataset with varying sampling rates, all time-series are aligned to a common time field for easy interaction and prediction. Techniques like interpolation are used to align the time series data to consistent intervals (e.g., every 5 minutes), ensuring order alignment of time stamps across the features.

* **Handling Missing Values:**

This model use the real world data, which is often incomplete. Missing glucose readings are replaced using some common methods, while missing meal or insulin records has been replaced by average data.

* **Normalization and Standardization:**

To perform model training, all numerical features are normalized to a stan dard level e.g., Min-Max Normalization.

X’ =

Where X’ is the normalized value, X is the original value, is the minimum value in the dataset and is the maximum value in the dataset

**5.4 FEATURE EXTRACTION**

The raw data has been finally converted to identify important features such as Weight, Rate of Glucose change, Lag Features and HbA1c Estimation which has been described elaborately in corresponding sections. To improve the accuracy of the model for better prediction, various features has been extracted from the preprocessed data, which involves creating new features for identifying meaningful patterns and temporal relations from the data.

* **Weight:**

The weight of each and every patient has been extracted, which must be used in fuzzy logic based severity estimation for managing the diabetes.

* **Rate of Glucose change:**

The rate at which the glucose levels are increasing or decreasing.

* **Lag Features**

The features such as glucose, insulin, bolus and meal data points to predict future glucose levels.

* **HbA1c Estimation**

The average glucose levels over a period of time(2-3 months) are used to estimate HbA1c, which provides a long-term representation of diabetes management.

**5.5 TRADITIONAL PHYSIOLOGICAL MODEL**

Traditional White box models are implemented by a nonlinear physiological model , it is a benchmark for comparison against Blackbox model by simulating glucose-insulin dynamics. Here, physiological equations are used to describe glucose insulin interactions, biologically based predictions etc.. The algorithms used here is Bayesian Estimation. In Bayesian Estimation, Markov Chain Rule Monte Carlo (MCMC) has been used to estimate the individual parameter,which uses Adaptive Single Component Metropolis- Hastings(SCMH). The algorithm is based on a method which is used in bayes statistics used to generate samples in a target distribution which is complex to sample directly. It works by suggesting new sample which is based on the current samples and it decides whether to reject or accept the suggested sample using a specific acceptance criteria. It has the ability to a fit in the way it suggests new samples over a time. It updates the parameters distribution for every 1000 iterations, and it gives the best estimated parameters. So it make this algorithm unique. Algorithm 1 shows the approach for Adaptive Single Component Metropolis- Hastings(SCMH).

**ALGORITHM 1 - Adaptive Single Component Metropolis- Hastings(SCMH):**

1. *i←0*
2. *initialize ,nIter*
3. *repeat*
4. *for p←1 to 5 do*
5. *set = [*
6. *sample*
7. *Set*
8. *= min ( 1,*
9. *sample U Uniform(0,1)*
10. *if U then*
11. *set*
12. *else*
13. *set*
14. *end if*
15. *end*
16. *i← i+1*
17. *Until n nIter*

Using this algorithm, we can derive three subsystems which is used for predicting the glucose level. They are Insulin Absorption Subsystem, Oral Glucose Absorption Subsystem and Glucose-Insulin Kinetics Subsystem. Subsystem, Oral Glucose Absorption Subsystem and Glucose-Insulin Kinetics Subsystem. Insulin absorption subsystem, which models the movement of insulin through the body after it’s injected. It tracks the absorption of insulin accounting for injected insulin entering into the bloodstream. It utilizes different compartments for absorption of insulin such as monomeric and non monomeric form and Plasma Insulin concentration given in equation 1 as follows:

**Insulin Absorption Subsystem:**

*(t) = · (t) − · (t)*

*(t) = · − · (t)*

* represents insulin in its non-monomeric form in the first compartment (before it starts to become active),
* (t) represents insulin that has moved to a monomeric state (ready for absorption into the bloodstream).
* (t) is the plasma insulin concentration, i.e., the insulin available in the bloodstream.
* is the rate at which insulin transforms from a non-monomeric state to a monomeric state. ​ is the absorption rate from the monomeric form into the plasma.
* is the clearance rate, indicating how quickly insulin is removed from the plasma.
* V is the volume of insulin distribution in the body.

Oral Glucose Absorption System , which models the absorption of glucose from carbohydrates consumed during meals, focusing on how glucose moves through different compartments. The techniques used here describes the glucose absorption in a two compartment system (ie.) stomach, Intestine. Glucose moves through two storage compartments where it can be absorbed into the bloodstream before reaching the gut given in equation 2 as follows:

**Oral Glucose Absorption Subsystem:**

* is the amount of glucose in the solid state in the stomach.
* is the glucose in the liquid state after being broken down (grinded) in the stomach. represents the glucose in the intestines ready for absorption into the bloodstream. represents the carbohydrate intake (in mg/kg/min) at time 𝑡.
* is the rate constant of grinding, representing how quickly solid food is broken down into liquid form.
* is the gastric emptying rate, determining how quickly glucose moves from the stomach to the intestines.
* ​ is the rate of glucose absorption from the intestines into the bloodstream. is the rate at which glucose enters the bloodstream.
* is a constant representing the fraction of glucose absorbed.

Glucose Insulin Kinetics subsystem represents interation between both the glucose and insulin for predicting the glucose transport and utilization in the body. It focuses mainly on how glucose levels change in response to insulin and how insulin sensitivity may vary over time. It describes how plasma glucose levels change due to insulin’s effects and elaborate the dynamic nature of the insulin sensitivity. The three compartment model such as Plasma Glucose Concentration, Inulin Action on Glucose and Interstitial Glucose Concentration were explained using the following Equation 3:

**Glucose Insulin Kinetics Subsystem:**

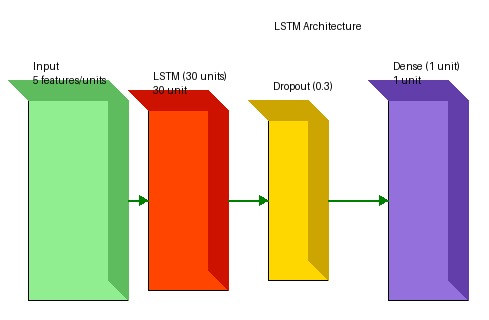
* represents the plasma glucose concentration (the blood sugar level).
* is the insulin action on glucose, representing how insulin reduces blood glucose.
* is the interstitial glucose concentration, which is the glucose level measured by CGM devices.
* SG​ is the glucose effectiveness, representing the body's ability to dispose of glucose.
* 𝐺𝑏​ is the basal glucose level, the normal level of glucose in the blood.
* 𝑉𝐺​ is the volume of glucose distribution in the body.
* 𝑝2 is a rate constant related to insulin action dynamics.
* SI is the insulin sensitivity, indicating how responsive the body is to insulin.

Overall, physiological White box model predicts set of glucose values for a given patient data and biological parameters.

**5.6 BLACK BOX DEEP LEARNING MODEL**

Black box approaches several advanced Deep learning algorithms, which are designed to capture complex patterns in a data. The specific algorithms used here are Long Short Term Memory(LSTM), Gated Recurrent Units(GRU) and Temporal Convolutional Network(TCN), which is best for time series data.

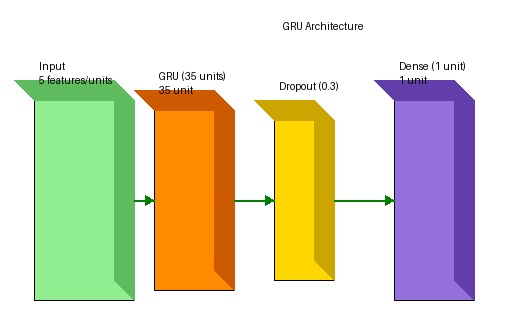
LSTM is a type of recurrent neural network which is significantly used here to learn and remember over long sequences for time series data. In this LSTM architecture, it includes a memory cell that able to maintain the data over extended time intervals. These networks contain memory cells that can maintain large kinds of information over long periods. Each LSTMcell consists of three main components. They are the forget gate, input gate, and output gate. Forget gate identifies the irrelevant information from the previous state and eliminates it. In other words, it determines what information should be eliminated from the previous state. It is done by applying a sigmoid activation function that takes the input as the previous hidden state and current input. The output values were most probably between 0 and 1. 0 determines that information is completely forgotten, and 1 determines it is completely retained. Second, it is an input gate. It is a different gate compared to the forget gate, as it determines what new information should be included in the cell state. It works by using two functions: the sigmoid activation function and the tanh activation function. The sigmoid function is used to f lter the input, and the tanh function is used to create a list of new candidate values that are added to the cell state. At last, its output gate. Here, the gate decides what will be the next hidden state, and it should be based on the current cell state. Here the working process is the same as the input gate, which uses the sigmoid function to list the cell state and uses the tanh function to find the output. While combining these three gates, it allows LSTM to remember long-term dependencies and make them effective for predicting blood glucose levels. LSTMs are used in many applications such as video analysis and time series forecasting.



**Fig 5.3** LSTM Architecture

LSTMs are used in many applications such as video analysis and time series forecasting.

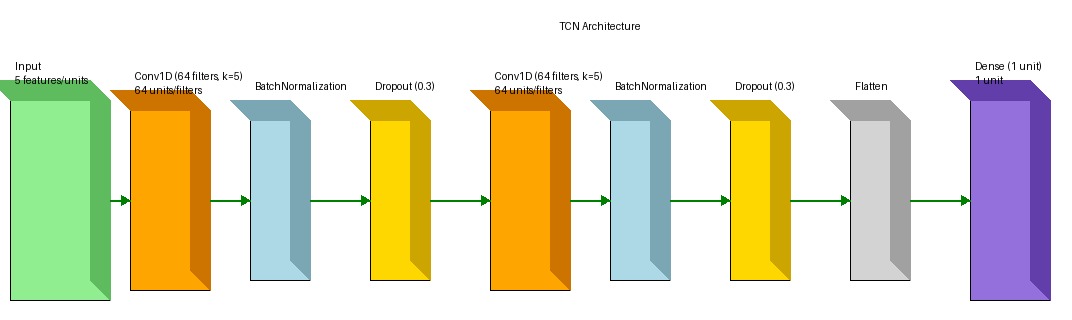
GRU is similar to LSTM , another recurrent neural network variant which simplifies the architecture which performs sequential tasks.It is more efficient, as it has the capability to handle long dependencies in the data . This cell consists of two main components, such as the update gate and the reset gate. When we look on to the update gate, it decides how much past information needs to be travelled along to the future. It combines forget and input gate functions that are taken from LSTMs. The work is done by using a sigmoid activation function to find the output values between 0 and 1. Additionally, it controls the information flow. Next, reset the gate. This gate is similar to the forget gate from LSTMs as it decides how much past information to forget. The work is done by using the sigmoid activation function as it trains the model to reset its memory while processing the new input. This algorithm is effective in capturing temporal dependencies. GRU is faster to train the model compared to LSTMs. It is also widely used in similar applications as LSTMs.



**Fig 5.4** GRU Architecture

It is also effective for predicting blood glucose levels.

TCN is specially designed for sequence modelling. It is not like RNN, Here networks are designed to handle data by utilizing the convolutional layers. It improves the train ing stability and performance. Time series prediction tasks are completely effective in temporal convolution networks. It consists of several key components, such as causal convolutions and dilated convolutions. Causal convolutions are used in TCN to ensure that while making predictions, the model only has access to interpret the past information. The work is done by padding the input sequence such that the output at time t always depends on input from time t. Causal is crucial for time series data. Next, it is a dilated convolution. It is used in TCN to introduce the gap between the kernel elements. The main advantage in this layer is the network allows the cell to detect changes over a wide area without increasing the significant parameters. It is done to capture the long-range dependencies in the data efficiently. Another component used here is residual connections. In this connection, it adds the input of a layer to its output and improves the training by allowing gradients to flow seamlessly throughout the networks. This is a component where the final output of the TCN is produced. So the input data flows along the causal and dilated convolutions, and residual produces the output, leading to the accurate prediction of blood glucose level. Using these methods, glucose levels are predicted . Finally, it is stated that while comparing with physiological approach, blackbox approach is effective in predicting the BG levels.



**Fig 5.5** TCN Architecture

Thus, the black box model predicts glucose levels based on historical data of complex relationships that affect glucose levels.

**5.7 HYBRID INTEGRATION**

The hybrid modelling framework integrates the outputs of the physiological model with deep learning models. It combines the outputs of both the physiological and black box models to increase overall prediction accuracy. The predictions from the physiological model are used as additional inputs to the black box model in order to refine them based on biological insights. The integration leverages the strengths of both models such as the interpretability of the physiological model and the pattern recognition capabilities of the black box model. Here deep learning models are used to predict the final glucose levels with better accuracy for type 1 diabetes patients.

**5.8 GENETIC PH OPTIMIZATION**

After predicting the final glucose level using deep learning models with better performance, the prediction horizon (PH) is optimized by using a genetic algorithm. The functions used here are fitness score, tournament selection, crossover, and mutation. It begins with generating an initial population with a suitable solution, where each and every solution has a set of model parameters. Next, the fitness function is done to evaluate this set of model parameters to predict the diabetes in an efficient way. Next, tournament selection is used here to select the individuals according to their fitness. The individuals were chosen randomly, and the individuals with the highest fitness were selected. For promoting the diversity in the population, crossover is used. At last, mutation is done to explore random changes in new individuals. These functions will predict the best time horizon within an hour. Algorithm 2 shows the approach for Genetic Ph Optimization.

**ALGORITHM 2 - Genetic PH Optimization:**

1. *Initialize population of N models with parameters sam pled from predefined bounds.*
2. *for each chromosome do*
3. *Initialize white-box and black-box parameters.*
4. *Compute initial hybrid model prediction using weighted combination.*
5. *end for*
6. *for each generation do*
7. *Evaluate fitness using:*
9. *Select parents via tournament selection.*
10. *for each pair of parents do*
11. *Perform crossover on model parameters.*
12. *Apply mutation with probability*
13. *end for*
14. *Integrate hybrid model outputs*
15. *Replace population with new offspring.*
16. *end for*
17. *Return best model based on final fitness evaluation*

**5.9 HbA1c CALCULATION**

In order to understand long term assessment of blood glucose levels for effective management of diabetes**,** the HbA1c calculation is used here to calculate the average blood glucose level for the past few months. The work is based on the relationship between average blood glucose levels and HbA1c levels. It is said to be the most useful metric for predicting and managing diabetes.

HbA1c =

Where is the estimated Average Glucose over 2-3 months (mg/dL) . Overall it is a useful metric for managing diabetes.

**5.10 FUZZY ORDER SEVERITY CLASSIFICATION**

Fuzzy logic is used here to address the limitations in classifying the Severity in diabetes patients. Unlike other approaches, Fuzzy allows for a more progressive and understandable transition between severity levels. The main advantage for used here is to handle uncertainty. Here Fuzzy system takes glucose and weight as inputs and processes them through corresponding predefined membership functions. It defines five categories for glucose and four categories for weight. Membership functions are used when the inputs, such as glucose and weights, are processed into the fuzzy input system. There are two membership functions used here, such as trapezoidal and triangular functions. The trapezoidal function is defined by four points, it is particularly useful when the membership should be at the maximum, not as the single point.

It is defined by four points a,b,c,d and represents a fuzzy set where membership increases linearly from a to b remains constant at 1 between b and cand decreases linearly from c to d.

Triangular functions are simple and efficient, and they are used when the membership values increase linearly to a peak and decrease linearly to represent the fuzzy sets.

It is defined by three points a,b,c and Membership increases linearly from a to b and decreases linearly from b to c.

The Membership functions for Glucose are

* + (Glucose = Very Low) → Trapezoidal function (0, 53, 61.5, 97.5)
  + (Glucose = Low) → Triangular function (53, 83.75, 125)
  + (Glucose = Medium) → Triangular function (70, 111.25, 162.5)
  + (Glucose = High) → Triangular function (97.5, 143.75, 200)
  + (Glucose = Very high) → Trapezoidal function (125, 181.25,200,300)

When the glucose is in the very low category or very high category, the trapezoidal function is used. In the very low category, the membership function ranges from 53 to 97.5, and the very high category ranges from 125 to 300. For the low, medium, and high categories, the triangular function is used. The low category ranges from 53 to 125 , the medium category ranges from 70 to 162.5, and the high category ranges from 97.5 to 200. These two functions are applied similarly in weight categories. The Membership functions for Weight factors are

* + (Weight = Under weight) → Trapezoidal function (0, 0, 50, 60)
  + (Weight = normal weight) → Triangular function (50, 60,80)
  + (Weight = over weight) → Triangular function (70,80,100)
  + (Weight = obese) → Trapezoidal function (90, 100, 150, 150)

When the weight is in the category of underweight and obese, the trapezoidal function is used, and for the category of normal and overweight, the triangular function is used. For underweight, the membership function ranges from 50 to 60; normal weight ranges from 60 to 80; overweight ranges from 70 to 100; and above 100 it is said to be in the obese category. Based on this, fuzzy rules were applied, and the further process moved on. These are processed through predefined 20 fuzzy rules (see Figure 10). which determine the best severity level prediction. The rules are

* 1. *(G = Very Low, W = Underweight) → (Prediction = No, Severity = Very Low),*
  2. *(G = Very Low, W = Normal) → (Prediction = No, Severity = Very Low),*
  3. *(G = Very Low, W = Overweight) → (Prediction = No, Severity = Very Low),*
  4. *(G = Very Low, W = Obese) → (Prediction = No, Severity = Very Low),*
  5. *(G = Low, W = Underweight) → (Prediction = No, Severity = Low),*
  6. *(G = Low, W = Normal) → (Prediction = No, Severity = Low),*
  7. *(G = Low, W = Overweight) → (Prediction = No, Severity = Low),*
  8. *(G = Low, W = Obese) → (Prediction = No, Severity = Low),*
  9. *(G = Medium, W = Underweight) → (Prediction = No, Severity = Medium),*
  10. *(G = Medium, W = Normal) → (Prediction = No, Severity = Medium),*
  11. *(G = Medium, W = Overweight) → (Prediction = No, Severity = Medium),*
  12. *(G = Medium, W = Obese) → (Prediction = No, Severity = Medium),*
  13. *(G = High, W = Underweight) → (Prediction = No, Severity = High),*
  14. *(G = High, W = Normal) → (Prediction = No, Severity = High),*
  15. *(G = High, W = Overweight) → (Prediction = Yes, Severity = High),*
  16. *(G = High, W = Obese) → (Prediction = Yes, Severity = High),*
  17. *(G = Very High, W = Underweight) → (Prediction = Yes, Severity = Very High),*
  18. *(G = Very High, W = Normal) → (Prediction = Yes, Severity = Very High),*
  19. *(G = Very High, W = Overweight) → (Prediction = Yes, Severity = Very High),*
  20. *(G = Very High, W = Obese) → (Prediction = Yes, Severity = Very High)*

Thus, the advantage of using this is to handle uncertainty. It makes the system more reliable and clinically useful for diabetes management.

**5.11 SUMMARY**

This chapter describes the system implementation as well as various algorithms used in the proposed system.

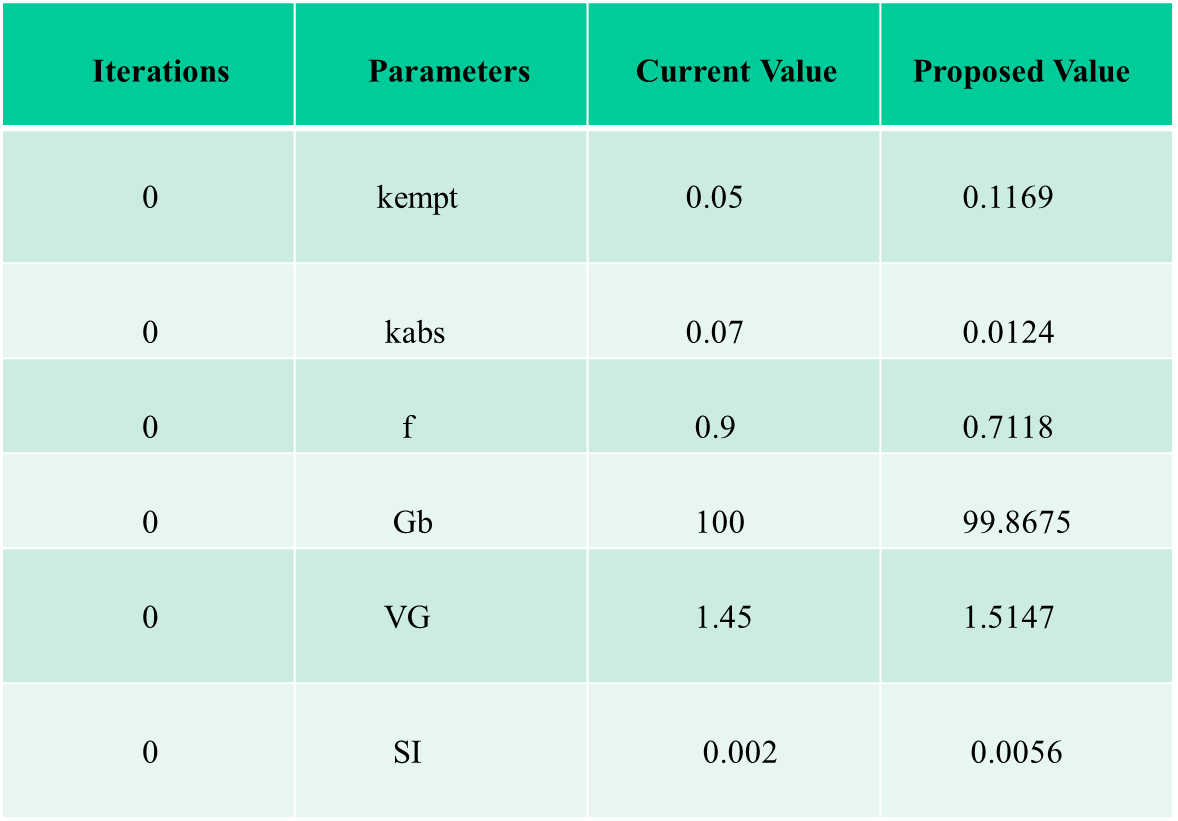
**CHAPTER 6**

**RESULTS AND DISCUSSION**

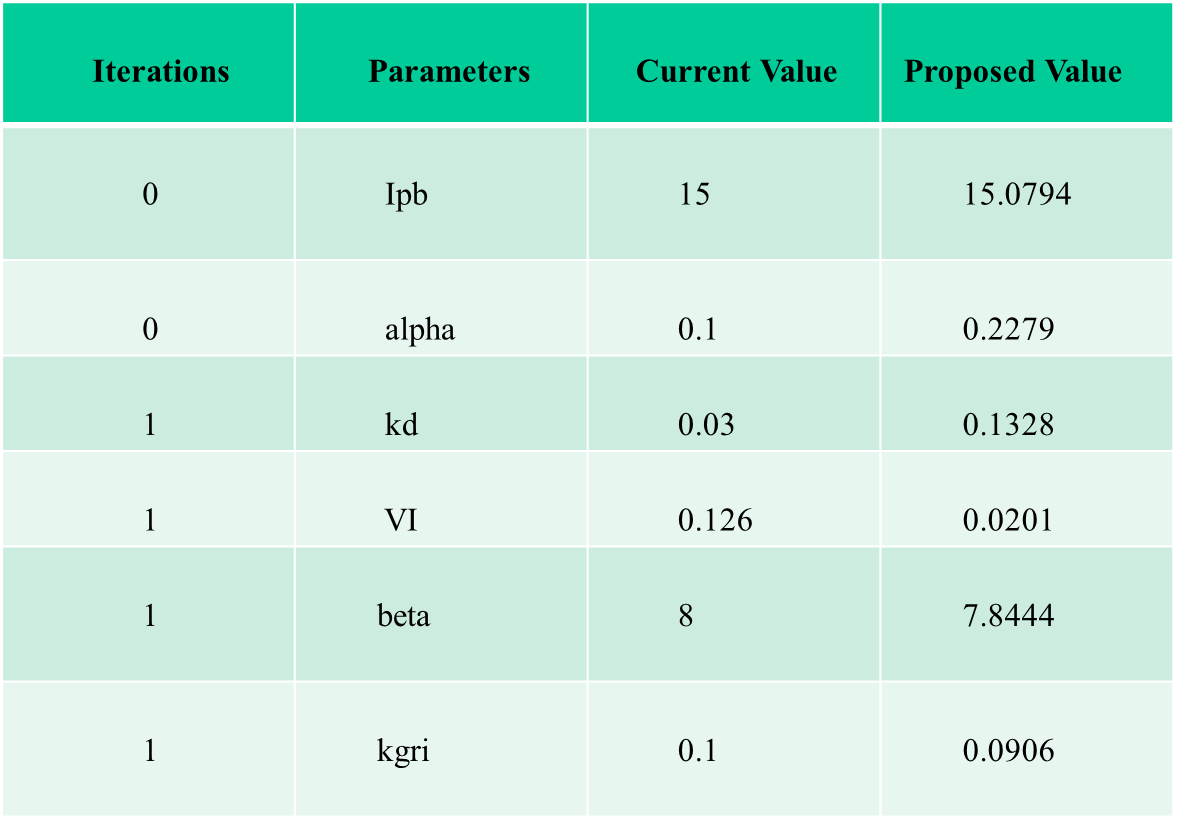
In this chapter, the results obtained by glucose level prediction model and severity order classification are discussed.

**6.1 ESTIMATING THE BEST PARAMETER FROM ADAPTIVE SINGLE CHAIN METROPOLIS HASTINGS:**

The best parameter for implementing the traditional physiological white box model estimated from adaptive single chain metropolis hastings are given in Table 6.1, 6.2



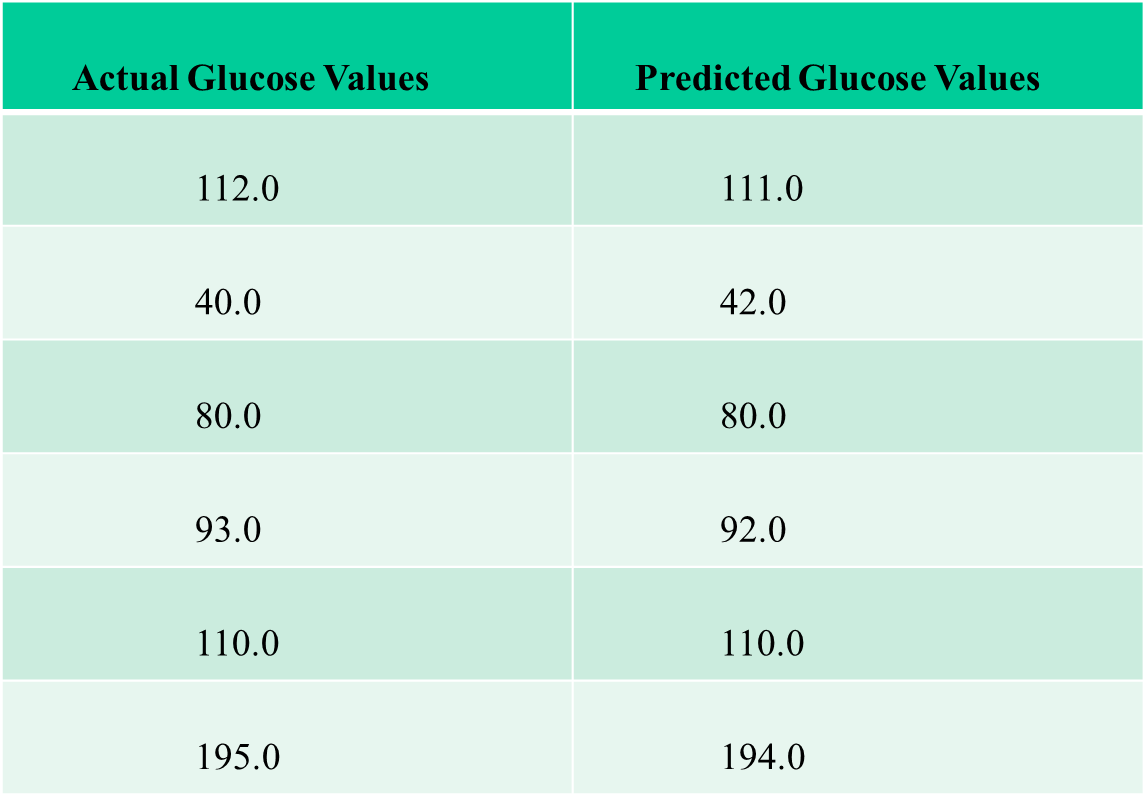
**Table 6.1** Sample Best Parameters from Adaptive SCMH-1



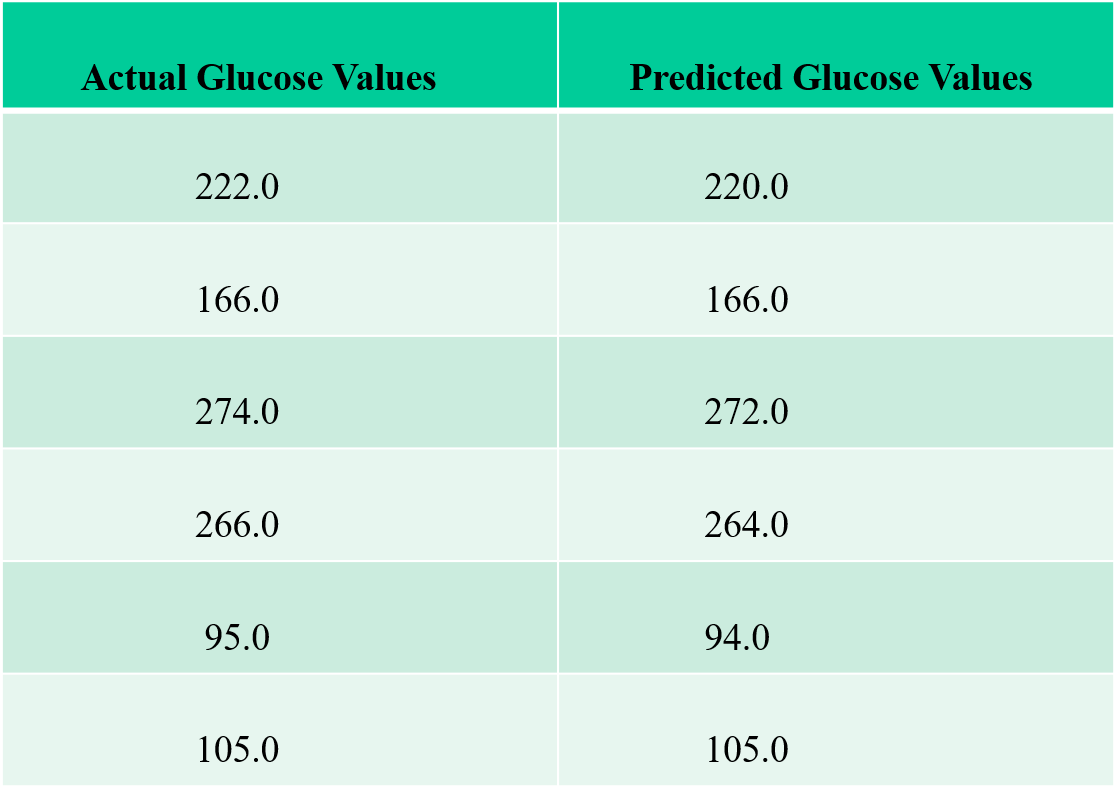
**Table 6.2** Sample Best Parameters from Adaptive SCMH-2

**6.2 HYBRID MODEL INTEGRATION PREDICTIONS**

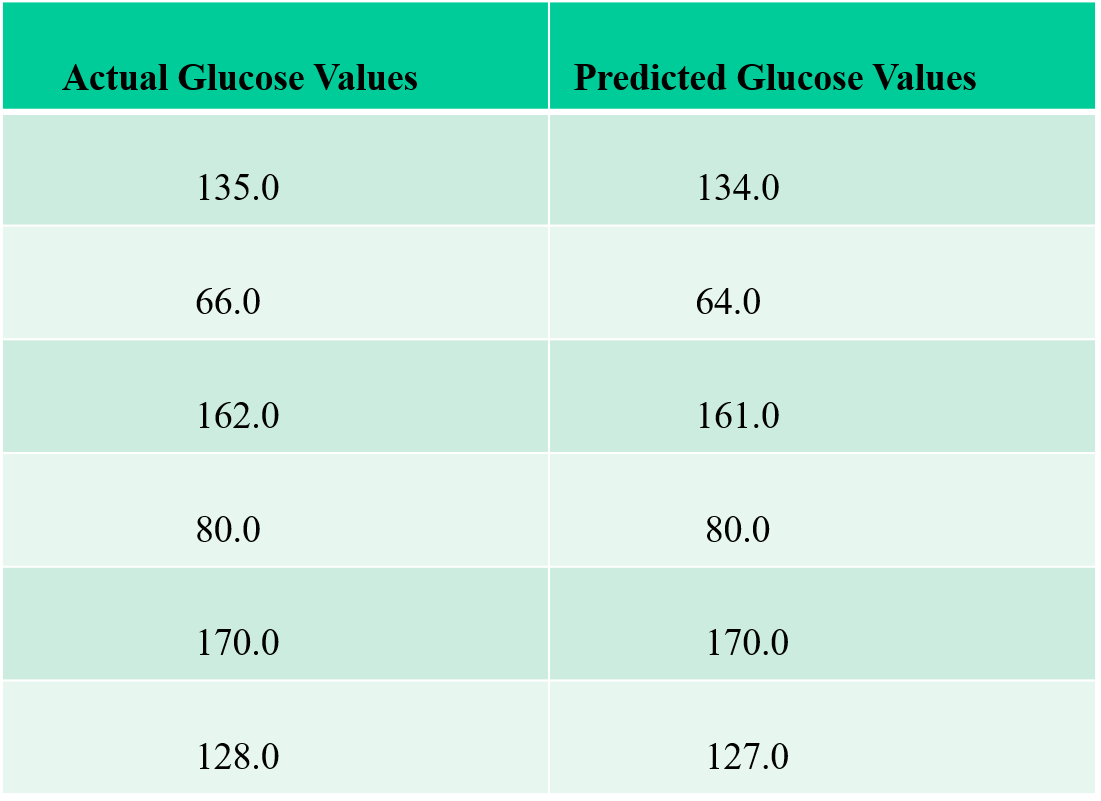
The Sample Glucose Level Prediction from different prediction models such as LSTM, GRU and TCN Models are given in Table 6.3, 6.4, 6.5



**Table 6.3** Sample LSTM Prediction



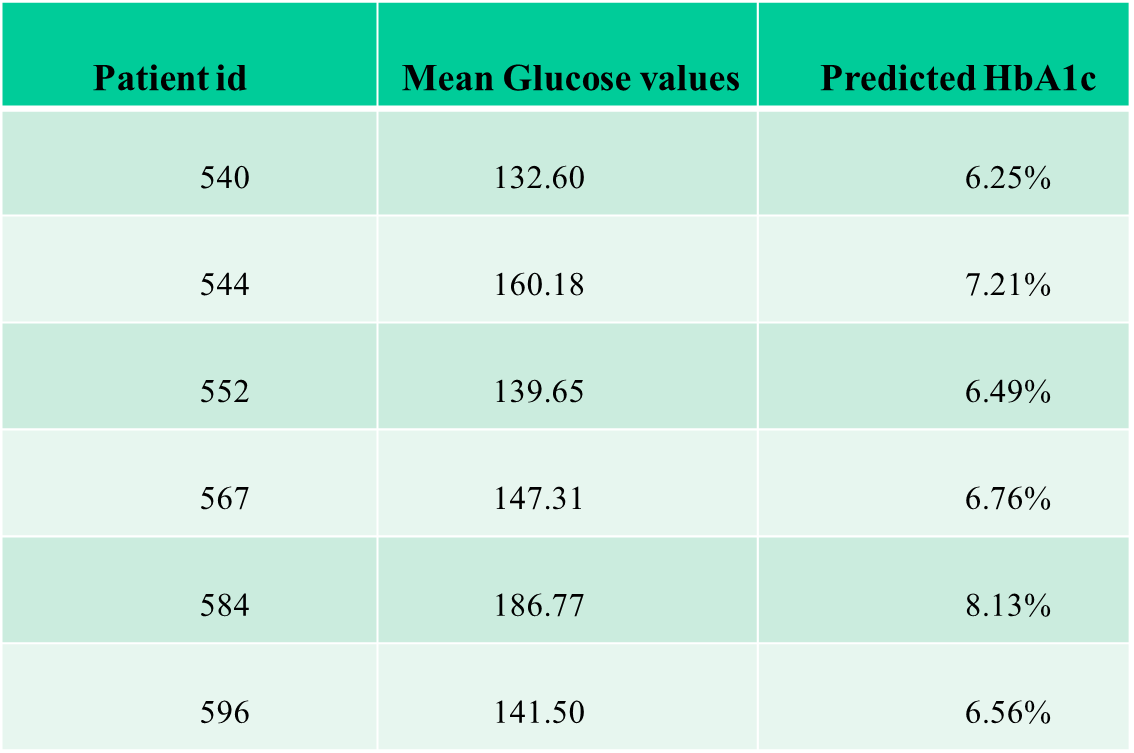
**Table 6.4** Sample GRU Prediction



**Table 6.5** Sample TCN Prediction

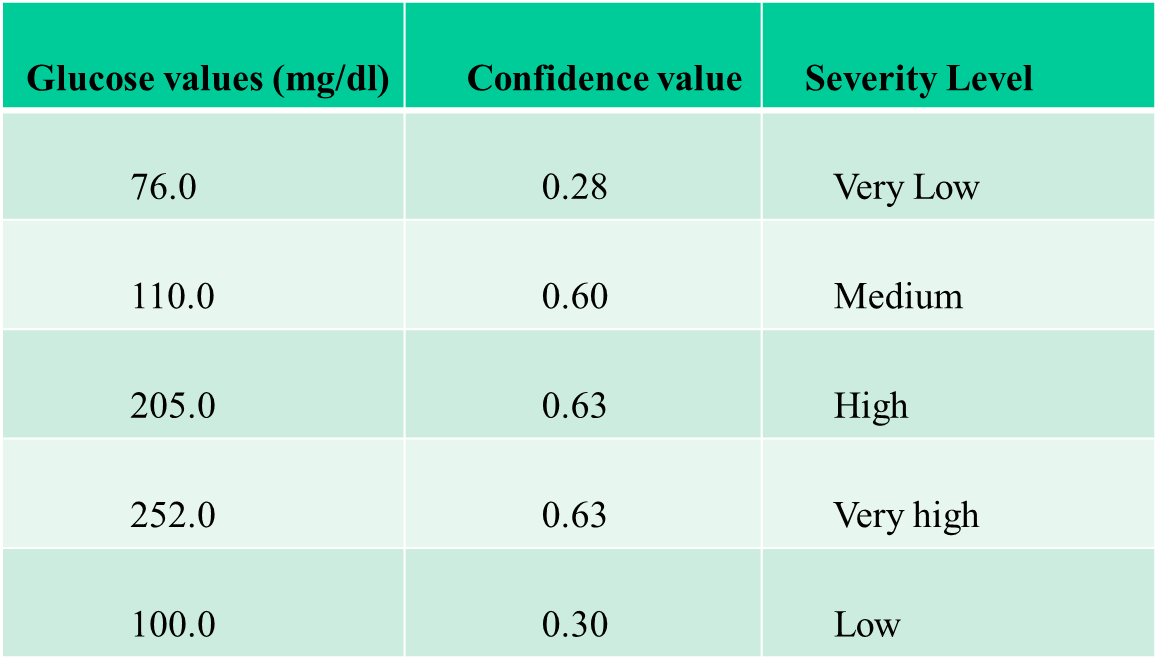
**6.3 HBA1C PREDICTION**

The Hba1c Calculation for the period of 3 months average of each patient is computed and is given in Table 6.6



**Table 6.6** Sample HbA1c Prediction

**6.4 FUZZY RULE BASED SEVERITY ESTIMATION**

The Sample Fuzzy based severity order output is given in Table 6.7

**Table 6.7** Sample Fuzzy Severity Order Prediction

**6.5 Hybrid Model For Predicting the Future Glucose Level**

The presented Error Metrics data show various performance characteristics across different deep learning models used. TCN has a remarkable Root Mean Square Error(RMSE) as 0.88, indicating its usefulness in correctly predicting the future glucose levels. In contrast, LSTM has high error metrics with a Root Mean Square Error(RMSE) as 2.2060 and a R2 Score as 0.9988, demonstrating limits in its capacity to reliably predict the future glucose levels. The GRU model shows a considerable decline in the root mean square Error(RMSE) as 0.90, indicating the possibility on overfitting or misfitting concerns that may need more research and model tweaking. TCN, on the other hand, exhibits amazing consistency with the R2 value as 0.9998, demonstrating its ability to reliably predicting the future glucose levels. These findings highlight the necessity of choosing suitable regression models for the job at hand, as well as modifying models to improve prediction performance and generalization capabilities.

**6.5.1 LSTM MODEL RESULTS**

The Evaluation metrics for various time horizons of LSTM Model is given in Table 6.8



**Table 6.8** Sample Performance Comparison for LSTM – 12,34,43 Minute Time Horizon

**6.5.2 GRU MODEL RESULTS**

The Evaluation metrics for various time horizons of GRU Model is given in Table 6.9



**Table 6.9** Sample Performance Comparison for GRU – 12,34,43 Minute Time Horizon

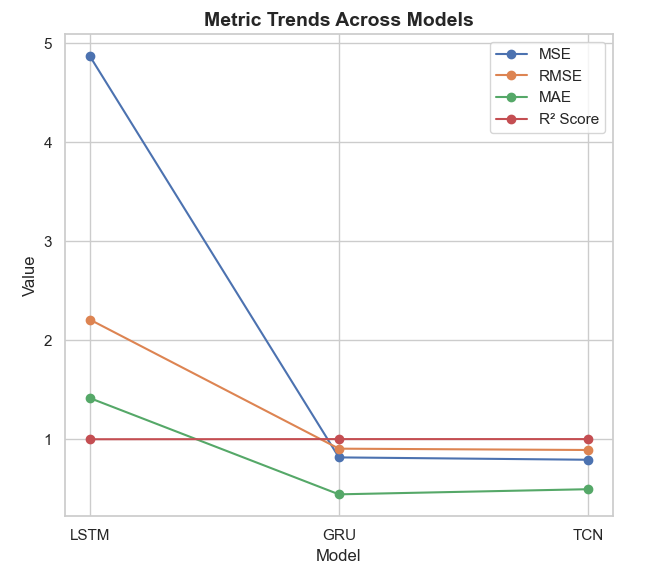
**6.5.3 TCN MODEL RESULTS**

The Evaluation metrics for various time horizons of TCN Model is given in Table 6.10



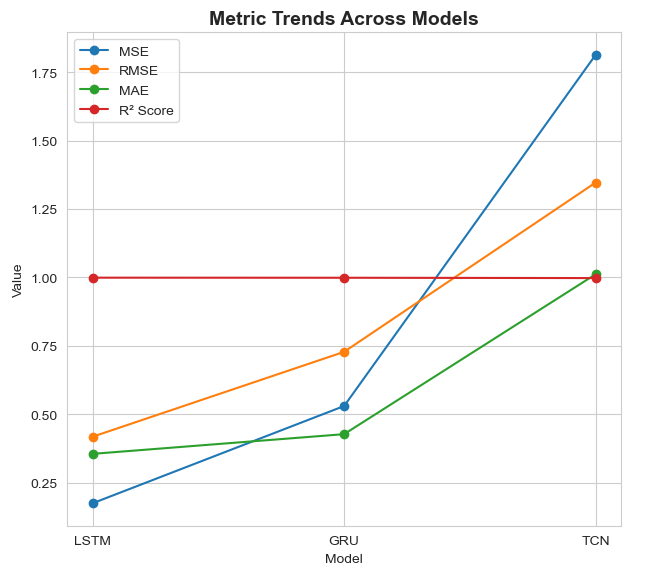
**Table 6.10** Sample Performance Comparison for TCN – 12,34,43 Minute Time Horizon

**6.5.4 COMPARISION OF MODELS ON DIFFERENT TIME HORIZONS**



**Figure 6.2** Sample Performance Comparison for LSTM, GRU, TCN for 34 minute horizon zon

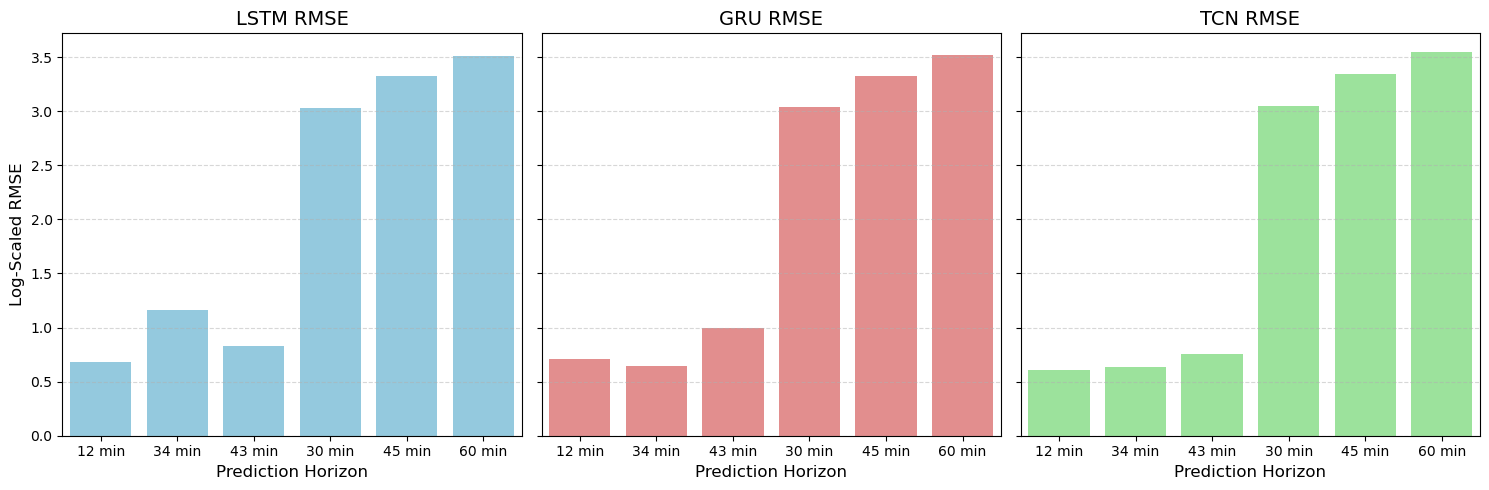
**Figure 6.1** Sample Performance Comparison for LSTM, GRU, TCN for 34 minute horizon

****

**Figure 6.3** Sample Performance Comparison for LSTM, GRU, TCN for 43 minute horizon

**6.5.5 COMPARISON OF PROPOSED VS EXISTING APPROACH**

The final RMSE metrics of LSTM, GRU, TCN for the proposed and existing time horizons are shown in the figure 6.14. Thus, TCN is the suitable model for future glucose level predictions.

****

**Figure 6.4** RMSE Comparison of Proposed and Existing Approach

**6.6 STATE OF THE ART OF COMPARISON**

|  |  |  |  |
| --- | --- | --- | --- |
| **MODELS USED** | **RMSE** | | |
| **15 min** | **30 min** | **45 min** |
| Fine Tuned CNN | 10.9 | 17.8 | 23.3 |
| General CNN | 10.5 | 17.2 | 22.9 |
| Scratch CNN RFR | 11.7 | 18.7 | 24.1 |

**Table 6.11** A personalized blood glucose level prediction model with a fine-tuning strategy

|  |  |
| --- | --- |
| **MODELS USED** | **RMSE (60 MIN TIME HORIZON)** |
| LSTM | **35.17** |
| GRU | **35.50** |
| Wavenet | **35.31** |

**Table 6.12** Blood glucose prediction with deep neural networks using weighted decision level fusion

|  |  |  |
| --- | --- | --- |
| **MODELS USED** | **RMSE** | |
| **30 min** | **60 min** |
| WDNet | 17.26 | 29.63 |
| LSTM | 23.15 | 35.59 |
| GRU | 23.04 | 34.98 |

**Table 6.13** Continuous Glucose, Insulin and Lifestyle Data Augmentation in Artificial Pancreas Using *Adaptive* Generative and Discriminative Models

|  |  |  |  |
| --- | --- | --- | --- |
| **MODELS USED** | **RMSE** | | |
| **15 min** | **30 min** | **45 min** |
| LSTM | 19.75 | 26.80 | 32.54 |
| GRU | 19.81 | 26.77 | 32.78 |
| TCN | 20.11 | 27.28 | 33.54 |

**Table 6.14** Individualized Models for Glucose Prediction in Type 1 Diabetes: Comparing Black-Box Approaches to a Physiological White-Box One

|  |  |  |  |
| --- | --- | --- | --- |
| **MODELS USED** | **RMSE** | | |
| **12 min** | **34 min** | **43 min** |
| LSTM | 0.9682 | 1.2980 | 2.2060 |
| GRU | 1.0284 | 1.6950 | 0.9023 |
| TCN | 0.8376 | 1.1337 | 0.8890 |

**Table 6.15** An Optimized Fuzzy based Personalized Deep Glucose Level Prediction and Severity

Estimation for Type 1 Diabetes Patients (Proposed work)

**6.6 SUMMARY**

This chapter describes the results obtained as well as the comparison of models used in the proposed system.

**CHAPTER 7**

**CONCLUSION AND FUTURE WORK**

**7.1 CONCLUSION**

Finally, in this research, the hybrid model developed is a fusion of strong parts of the physiological white box model with strong parts of the deep learning portion which includes LSTM, GRU, TCN, thus making this model a hybrid model that successfully merges its strengths. It facilitates a deeper and more accurate glucose dynamic characterization and blood glucose level prediction for Type 1 Diabetes patients.

Successful exploitation of elements learned by adapting to the complexities of diabetes disease and management with real time data from Continuous Glucose Monitoring (CGM) Systems, insulin intake and meal data is shown by the model. This approach can result in better patient outcomes by ensuring the predictions are timely and accurate, as the results suggest.

The use of fuzzy logic in the estimation stage, in addition, enables more refined classification and allows for a more sophisticated assessment of diabetes severity as being able to inform on treatment decisions and therefore enhance the patient care.

The use of the hba1c Calculation, will enables the diabetes patients to effectively manages their diabetes in a better way by analysing the past glucose levels and able to predict the average future glucose level over the past 3 month.

And also, the genetic algorithm has been included, which involves initializing population, fitness function, selection, crossover and mutation. Which finally determines the optimized time horizon in predicting the future glucose levels

**7.2 FUTURE WORK**

The future research direction can focus on improving the hybrid model with some additional physiological parameters such as stress level, physical activity, and sleep schedules for explicating what impacts glucose measurements most, besides the remaining four parameters we chosen. Furthermore, other machine learning technique such as ensemble methods or attention mechanism can further improve the prediction accuracy.

To apply the model in real life surroundings, we need to develop personalized models that could adjust or adapt itself to individual patient profiles including the separate gene factors as well as the choice of behavioral treatments. The local input of the patient data can serve as the future of machine learning algorithms, which in turn can learn from such inputted individual patient data in time and provide tailoring recommendations.

The longitudinal studies will be very crucial in order to assess how effective the model would be in different populations and different lifestyle conditions over a period of time. This research is also based upon findings that could then be used to identify what areas of the model the model is limited to and how to further develop the model in that regard.

A next possible direction for future work would be to integrate the model with wearable devices like smart watches or fitness trackers to present real time feedbacks and warnings to the patients. Such a system could be integrated into the treatment of diabetes in order to allow the patient to make an informed decision regarding current glucose levels and activity.

Developing applications such that patients can be informed easily of predictions and severity assessments: These can be made a lot easier with. Incorporating gamification elements might be added to embed the patients to their management plans.

**7.3 SOCIAL IMPACTS**

**Improved health outcomes:** Not only does the model helps to predict accurate glucose values, but it also provides advancing severity information such as hypoglycemia and hyperglycemia so that there is a reduction in severe complications of Type 1 diabetes. As a result, health outcomes for the patients improve and a better quality of life is enjoyed.

**Promotion of Patient Empowerment:** The project is aimed at promoting patient empowerment by equipping people in terms of knowledge and tools with which to actively manage their diabetes. It gives them the control over their health and encourages healthy associations with the treatment plans.

Enhanced diabetes management can reduce the cost of healthcare in emergency interventions, as well as long term complications. It prevents the patients and the health care systems from getting to severe level of health issues and hence alleviates the financial burden.

**Educational Initiatives:** The project can provide a basis for educational initiatives related to increased awareness of the management of diabetes. The model can create awareness for regular monitoring and self care that would encourage healthier lifestyle choices and a better public health.

Implementation of this model may lead to community support networks that have access to Type 1 Diabetes. Patients can learn one from another, share experiences and build a supportive community that promotes healthy behaviors.

**Policy Implications:** The finding of this research could help design health care policies that would assist such programs. This model provides policymakers with insights which can be leveraged to effectively allocate resources and promote programs that will contribute to improving patient care.

**7.4 APPLICABILITY OF THE PROJECT**

In clinical settings, the hybrid model can be applied for medical aid in determining how insulin dose and dietary advice should be provided to type 1 diabetes patients. This helps healthcare providers to provide accurate predictions and thus tailor treatment plans according to patient’s specific needs.

Such application can also be used in mobile health applications where it could give patients a personalized insights and predictions regarding their glucose levels. It gives the patients the power over their condition which enables them to manage the treatment more effectively and takes the steps to be proactive about the health.

The model can be utilised in telemedicine to forward remote patient monitoring and support. This is particularly helpful for people in rural or under served places because they are not able to get to the health service locations.

In addition, this model can be used in diabetes management program: certainly, healthcare providers will find useful tools to monitor patient progress and to adjust treatment plan, if necessary.

**APPENDIX I**

**SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENT**

* **PROCESSOR:** Intel® Core™ i5-1335U 1.30 GHz
* **SYSTEM TYPE:** 64-bit operating system, x64-based processor.
* **RAM:** 16 GB

**SOFTWARE REQUIREMENT**

* **LANGUAGE:** Python
* **OS:** Windows 11
* **TOOL:** Google Colab, Visual Studio Code, Jupyter Notebook

**APPENDIX II**

**CODE**

# PYTHON TENSORFLOW CODE

import numpy as np

import seaborn as sns

import random

import xml.etree.ElementTree as ET

import pandas as pd

from datetime import datetime

import matplotlib.pyplot as plt

from keras.callbacks import EarlyStopping

from keras.regularizers import l2

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score

from sklearn.model\_selection import KFold

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, GRU, Conv1D, LSTM, Flatten, Dropout,BatchNormalization

import skfuzzy as fuzz

from skfuzzy import control as ctrl

class GlucoseInsulinDynamics:

    def \_\_init\_\_(self, params):

        self.params = params

    def subcutaneous\_insulin\_absorption(self, insulin):

        ka2, kd, ke, VI, beta = self.params['ka2'], self.params['kd'], self.params['ke'], self.params['VI'], self.params['beta']

        Isc1, Isc2, Ip = insulin['Isc1'], insulin['Isc2'], insulin['Ip']

        dIsc1 = -kd \* Isc1 + insulin['I'] / VI

        dIsc2 = kd \* Isc1 - ka2 \* Isc2

        dIp = ka2 \* Isc2 - ke \* Ip

        return dIsc1, dIsc2, dIp

    def oral\_glucose\_absorption(self, CHO):

        kgri, kempt, kabs, f = self.params['kgri'], self.params['kempt'], self.params['kabs'], self.params['f']

        Qsto1, Qsto2, Qgut = CHO['Qsto1'], CHO['Qsto2'], CHO['Qgut']

        dQsto1 = -kgri \* Qsto1 + CHO['intake']

        dQsto2 = kgri \* Qsto1 - kempt \* Qsto2

        dQgut = kempt \* Qsto2 - kabs \* Qgut

        Ra = f \* kabs \* Qgut

        return dQsto1, dQsto2, dQgut, Ra

    def glucose\_insulin\_kinetics(self, glucose, insulin\_action, Ra):

        SG, Gb, VG, p2, SI = self.params['SG'], self.params['Gb'], self.params['VG'], self.params['p2'], self.params['SI']

        G, X, IG = glucose['G'], glucose['X'], glucose['IG']

        dG = - (SG + SI \* X) \* G + SG \* Gb + Ra / VG

        dX = -p2 \* (X - SI \* (insulin\_action - self.params['Ipb']))

        dIG = -(1 / self.params['alpha']) \* (IG - G)

        return dG, dX, dIG

def adaptive\_scmh(params, data, iterations=1000):

    def proposal(param, sigma):

        return np.random.normal(param, sigma)

    estimated\_params = params.copy()

    for i in range(iterations):

        for key in params.keys():

            current\_value = estimated\_params[key]

            proposed\_value = proposal(current\_value, sigma=0.1)

            if proposed\_value < 0:

                continue

            print(f"Iteration {i}, Parameter {key}: Current Value={current\_value}, Proposed Value={proposed\_value}")

            current\_likelihood = likelihood(current\_value, data)

            proposed\_likelihood = likelihood(proposed\_value, data)

            if current\_likelihood == 0:

                current\_likelihood = 1e-6

            acceptance\_ratio = min(1, proposed\_likelihood / current\_likelihood)

            if np.random.rand() < acceptance\_ratio:

                estimated\_params[key] = proposed\_value

    return estimated\_params

def likelihood(params, data):

    predicted = data['predicted'] if 'predicted' in data.columns else np.zeros\_like(data['glucose'])

    observed = data['glucose']

    variance = np.var(observed)

    if variance == 0:

        variance = 1e-6

    error = observed - predicted

    likelihood = np.exp(-np.sum(error\*\*2) / (2 \* variance))

    return likelihood

# LSTM Model

def build\_lstm\_model(input\_shape):

    model = Sequential([

        LSTM(30, activation='relu', input\_shape=input\_shape, kernel\_regularizer=l2(0.05)),

        Dropout(0.3),

        Dense(1)

    ])

    model.compile(optimizer='adam', loss='mse')

    return model

# GRU Model

def build\_gru\_model(input\_shape):

    model = Sequential([

        GRU(35, activation='relu', input\_shape=input\_shape, kernel\_regularizer=l2(0.01)),

        Dropout(0.3),

        Dense(1)

    ])

    model.compile(optimizer='adam', loss='mse')

    return model

# TCN Model

def build\_tcn\_model(input\_shape):

    model = Sequential([

        Conv1D(filters=64, kernel\_size=5, strides=1, padding='causal', activation='relu', input\_shape=input\_shape),

        BatchNormalization(),

        Dropout(0.3),

        Conv1D(filters=64, kernel\_size=5, strides=1, padding='causal', activation='relu'),

        BatchNormalization(),

        Dropout(0.3),

        Flatten(),

        Dense(1, kernel\_regularizer=l2(0.02))

    ])

    model.compile(optimizer='adam', loss='mse')

    return model

def train\_model(X\_train, y\_train, model):

    early\_stopping = EarlyStopping(monitor='val\_loss', patience=5, restore\_best\_weights=True)

    model.fit(X\_train, y\_train, epochs=50, batch\_size=32, validation\_split=0.2, verbose=1, callbacks=[early\_stopping])

    return model

def integrate\_models(white\_box\_features, black\_box\_data):

    white\_box\_features = np.reshape(white\_box\_features, (-1, 1)) if white\_box\_features.ndim == 1 else white\_box\_features

    black\_box\_data = np.reshape(black\_box\_data, (-1, 1)) if black\_box\_data.ndim == 1 else black\_box\_data

    combined\_features = np.concatenate([white\_box\_features, black\_box\_data], axis=1)

    return combined\_features

def evaluate\_predictions(y\_true, y\_pred, epsilon=1e-10):

    mse = np.float64(mean\_squared\_error(y\_true, y\_pred))

    rmse = np.sqrt(mean\_squared\_error(y\_true, y\_pred))

    mae = mean\_absolute\_error(y\_true, y\_pred)

    r2 = r2\_score(y\_true, y\_pred)

    print(f"MSE:{mse:.10f}")

    print(f"RMSE: {rmse:.4f}")

    print(f"MAE: {mae:.4f}")

    print(f"R² Score: {r2:.4f}")

    return {'MSE':mse,'RMSE': rmse, 'MAE': mae, 'R² Score': r2}

def parse\_xml(xml\_file):

    tree = ET.parse(xml\_file)

    root = tree.getroot()

    glucose\_levels = []

    basal\_insulin = []

    bolus\_insulin = []

    carbs = []

    def parse\_timestamp(ts):

        return datetime.strptime(ts, '%d-%m-%Y %H:%M:%S')

    for glucose\_event in root.findall(".//glucose\_level/event"):

        ts = parse\_timestamp(glucose\_event.get('ts'))

        value = float(glucose\_event.get('value'))

        glucose\_levels.append((ts, value))

    for basal\_event in root.findall(".//basal/event"):

        ts = parse\_timestamp(basal\_event.get('ts'))

        value = float(basal\_event.get('value'))

        basal\_insulin.append((ts, value))

    for bolus\_event in root.findall(".//bolus/event"):

        ts = parse\_timestamp(bolus\_event.get('ts\_begin'))

        dose = float(bolus\_event.get('dose'))

        bolus\_insulin.append((ts, dose))

    for meal\_event in root.findall(".//meal/event"):

        ts = parse\_timestamp(meal\_event.get('ts'))

        carbs\_value = float(meal\_event.get('carbs'))

        carbs.append((ts, carbs\_value))

    glucose = pd.DataFrame(glucose\_levels, columns=['timestamp', 'glucose']).set\_index('timestamp')

    basal = pd.DataFrame(basal\_insulin, columns=['timestamp', 'basal']).set\_index('timestamp')

    bolus = pd.DataFrame(bolus\_insulin, columns=['timestamp', 'bolus']).set\_index('timestamp')

    carbs = pd.DataFrame(carbs, columns=['timestamp', 'carbs']).set\_index('timestamp')

    combined = glucose.join(basal, how='outer').join(bolus, how='outer').join(carbs, how='outer')

    combined.sort\_index(inplace=True)

    combined.fillna(0, inplace=True)

    return combined

def multi\_step\_forecast(model, input\_data, horizons=[30, 45, 60]):

    predictions = []

    for horizon in horizons:

        forecast = model.predict(input\_data)

        predictions.append(forecast)

    return predictions

def evaluate\_forecasts(true\_values, predictions, horizons):

    metrics = {}

    for i, horizon in enumerate(horizons):

        floored\_predictions = np.floor(predictions[i])

        mse = np.float64(mean\_squared\_error(true\_values, floored\_predictions))

        rmse = np.sqrt(mse)

        mae = mean\_absolute\_error(true\_values, floored\_predictions)

        r2 = r2\_score(true\_values, floored\_predictions)

        metrics[horizon] = {'MSE': mse, 'RMSE': rmse, 'MAE': mae, 'R² Score': r2}

        print(f"\nHorizon: {horizon}")

        print(f"MSE: {mse:.4f}, RMSE: {rmse:.4f}, MAE: {mae:.4f}, R² Score: {r2:.4f}")

        valid\_indices = [j for j in range(len(true\_values)) if true\_values[j] >= 10 and floored\_predictions[j] >= 10]

        selected\_indices = random.sample(valid\_indices, min(20, len(valid\_indices)))

        selected\_true\_values = [true\_values[j] for j in selected\_indices]

        selected\_predictions = [floored\_predictions[j] for j in selected\_indices]

        print("\nSample Predictions:")

        for j in range(len(selected\_indices)):

            print(f"Actual: {float(selected\_true\_values[j]):.2f}, Predicted: {float(selected\_predictions[j]):.2f}")

    return metrics

def calculate\_hba1c(mean\_glucose):

    return (mean\_glucose + 46.7) / 28.7

def evaluate\_model\_with\_hba1c(model, test\_features, test\_labels, horizons, epsilon=1e-10):

    predictions = multi\_step\_forecast(model, test\_features)

    mean\_glucose = np.mean(predictions)

    predicted\_hba1c = calculate\_hba1c(mean\_glucose)

    metrics = evaluate\_forecasts(test\_labels, predictions, horizons)

    print("After hba1c")

    for horizon, met in metrics.items():

        print(f"Metrics(hba1c) for {round(horizon)} minutes: RMSE: {met['RMSE']}, MAE: {met['MAE']}, R² Score: {met['R² Score']}")

    print(f"Predicted HbA1c: {predicted\_hba1c:.2f}%")

def initialize\_population(pop\_size, param\_bounds):

    return [{key: np.random.uniform(low, high) for key, (low, high) in param\_bounds.items()} for \_ in range(pop\_size)]

def evaluate\_fitness(params, data):

    return np.mean([sum(d.values()) for d in data])

def tournament\_selection(population, fitness\_scores, k=3):

    return [max(random.sample(list(zip(population, fitness\_scores)), k), key=lambda x: x[1])[0] for \_ in range(len(population))]

def crossover(parent1, parent2):

    keys = list(parent1.keys())

    point = random.randint(1, len(keys) - 1)

    child1 = {key: parent1[key] if i < point else parent2[key] for i, key in enumerate(keys)}

    child2 = {key: parent2[key] if i < point else parent1[key] for i, key in enumerate(keys)}

    return child1, child2

def mutate(individual, mutation\_rate, param\_bounds):

    if random.random() < mutation\_rate:

        param = random.choice(list(individual.keys()))

        individual[param] = np.random.uniform(\*param\_bounds[param])

    return individual

def genetic\_algorithm(data, param\_bounds, pop\_size=20, generations=50, mutation\_rate=0.1):

    population = initialize\_population(pop\_size, param\_bounds)

    for \_ in range(generations):

        fitness\_scores = [evaluate\_fitness(ind, data) for ind in population]

        selected = tournament\_selection(population, fitness\_scores)

        offspring = []

        for \_ in range(len(population) // 2):

            p1, p2 = random.sample(selected, 2)

            c1, c2 = crossover(p1, p2)

            offspring.extend([mutate(c1, mutation\_rate, param\_bounds), mutate(c2, mutation\_rate, param\_bounds)])

        population = offspring

    return max(population, key=lambda ind: evaluate\_fitness(ind, data))

def train():

    param\_bounds = {"PH": (10, 60), "ka2": (0.1, 1.0), "kd": (0.1, 1.0), "ke": (0.01, 0.1), "VI": (5, 20), "beta": (0.1, 2.0)}

    training\_files = [

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/train/540-ws-training.xml',

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/train/544-ws-training.xml',

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/train/552-ws-training.xml',

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/train/567-ws-training.xml',

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/train/584-ws-training.xml',

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/train/596-ws-training.xml'

    ]

    print("Parsing training data")

    training\_data = [parse\_xml(file) for file in training\_files]

    initial\_params = {

        'ka2': 0.02, 'kd': 0.03, 'ke': 0.01, 'VI': 0.126, 'beta': 8,

        'kgri': 0.1, 'kempt': 0.05, 'kabs': 0.07, 'f': 0.9,

        'SG': 0.01, 'Gb': 100, 'VG': 1.45, 'p2': 0.02, 'SI': 0.002, 'Ipb': 15, 'alpha': 0.1

    }

    print("Estimating parameters using Adaptive SCMH")

    estimated\_params = adaptive\_scmh(initial\_params, training\_data[0])

    print("The Best Estimated Values are:")

    print(estimated\_params)

    print("Generating white-box features")

    white\_box\_model = GlucoseInsulinDynamics(estimated\_params)

    white\_box\_features\_train = []

    for data in training\_data:

        Ra\_values = []

        for idx, row in data.iterrows():

            insulin = {'Isc1': 0, 'Isc2': 0, 'Ip': row['basal'], 'I': row['bolus']}

            CHO = {'Qsto1': 0, 'Qsto2': 0, 'Qgut': 0, 'intake': row['carbs']}

            glucose = {'G': row['glucose'], 'X': 0, 'IG': row['glucose']}

            dIsc1, dIsc2, dIp = white\_box\_model.subcutaneous\_insulin\_absorption(insulin)

            dQsto1, dQsto2, dQgut, Ra = white\_box\_model.oral\_glucose\_absorption(CHO)

            Ra\_values.append(Ra)

            dG, dX, dIG = white\_box\_model.glucose\_insulin\_kinetics(glucose, dIp, Ra)

        white\_box\_features\_train.append(np.array(Ra\_values))

    print("Integrating with black-box model")

    combined\_train\_features = []

    for wb\_features, data in zip(white\_box\_features\_train, training\_data):

        combined\_train\_features.append(integrate\_models(wb\_features, data.values))

    train\_features = np.vstack(combined\_train\_features)

    train\_labels = np.hstack([data['glucose'].values for data in training\_data])

    print("Shape of train\_features:", train\_features.shape)

    if len(train\_features.shape) == 2:

        train\_features = np.expand\_dims(train\_features, axis=-1)

    print("Reshaped train\_features shape:", train\_features.shape)

    kf = KFold(n\_splits=10, shuffle=True, random\_state=42)

    cv\_results = {'lstm': [], 'gru': [], 'tcn': []}

    for train\_index, val\_index in kf.split(train\_features):

        X\_train, X\_val = train\_features[train\_index], train\_features[val\_index]

        y\_train, y\_val = train\_labels[train\_index], train\_labels[val\_index]

        lstm\_model = build\_lstm\_model((X\_train.shape[1], X\_train.shape[2]))

        lstm\_model = train\_model(X\_train, y\_train, lstm\_model)

        lstm\_predictions = lstm\_model.predict(X\_val).flatten()

        lstm\_metrics = evaluate\_predictions(y\_val, lstm\_predictions)

        cv\_results['lstm'].append(lstm\_metrics)

        gru\_model = build\_gru\_model((X\_train.shape[1], X\_train.shape[2]))

        gru\_model = train\_model(X\_train, y\_train, gru\_model)

        gru\_predictions = gru\_model.predict(X\_val).flatten()

        gru\_metrics = evaluate\_predictions(y\_val, gru\_predictions)

        cv\_results['gru'].append(gru\_metrics)

        tcn\_model = build\_tcn\_model((X\_train.shape[1], X\_train.shape[2]))

        tcn\_model = train\_model(X\_train, y\_train, tcn\_model)

        tcn\_predictions = tcn\_model.predict(X\_val).flatten()

        tcn\_metrics = evaluate\_predictions(y\_val, tcn\_predictions)

        cv\_results['tcn'].append(tcn\_metrics)

    print("Cross-validation results:")

    for model\_name, results in cv\_results.items():

        mean\_results = {key: np.mean([res[key] for res in results]) for key in results[0].keys()}

        print(f"{model\_name.upper()} - Mean MSE: {mean\_results['MSE']:.4f}, Mean RMSE: {mean\_results['RMSE']:.4f}, Mean MAE: {mean\_results['MAE']:.4f}, Mean R² Score: {mean\_results['R² Score']:.4f}")

    # Save models

    lstm\_model.save('lstm\_model.h5')

    gru\_model.save('gru\_model.h5')

    tcn\_model.save('tcn\_model.h5')

    training\_results = process\_files(training\_files, mode="Training")

    return lstm\_model, gru\_model, tcn\_model, estimated\_params, param\_bounds

def parse\_xml\_1(xml\_file):

    tree = ET.parse(xml\_file)

    root = tree.getroot()

    weight = float(root.get("weight", 70))

    glucose\_levels = []

    def parse\_timestamp(ts):

        return datetime.strptime(ts, "%d-%m-%Y %H:%M:%S")

    for glucose\_event in root.findall(".//glucose\_level/event"):

        ts = parse\_timestamp(glucose\_event.get("ts"))

        value = float(glucose\_event.get("value"))

        glucose\_levels.append((ts, value))

    return weight, glucose\_levels

def build\_fuzzy\_inference\_system(defuzzify\_method: str = "centroid"):

    glucose = ctrl.Antecedent(np.arange(56, 198 + 1, 0.01), "glucose")

    weight = ctrl.Antecedent(np.arange(40, 150 + 1, 0.1), "weight")

    prediction = ctrl.Consequent(np.arange(0, 1 + 0.05, 0.05), "prediction", defuzzify\_method=defuzzify\_method)

    glucose["very low"] = fuzz.trapmf(glucose.universe, [0, 53, 61.5, 97.5])

    glucose["low"] = fuzz.trimf(glucose.universe, [53, 83.75, 125])

    glucose["medium"] = fuzz.trimf(glucose.universe, [70, 111.25, 162.5])

    glucose["high"] = fuzz.trimf(glucose.universe, [97.5, 143.75, 200])

    glucose["very high"] = fuzz.trapmf(glucose.universe, [125, 181.25, 200, 300])

    weight["underweight"] = fuzz.trapmf(weight.universe, [0, 0, 50, 60])

    weight["normal weight"] = fuzz.trimf(weight.universe, [50, 60, 80])

    weight["overweight"] = fuzz.trimf(weight.universe, [70, 80, 100])

    weight["obese"] = fuzz.trapmf(weight.universe, [90, 100, 150, 150])

    prediction["no"] = fuzz.trimf(prediction.universe, [0, 0, 0.8])

    prediction["yes"] = fuzz.trimf(prediction.universe, [0.2, 1, 1])

    rules = [

        ctrl.Rule(glucose["very low"] & weight["underweight"], prediction["no"]),

        ctrl.Rule(glucose["very low"] & weight["normal weight"], prediction["no"]),

        ctrl.Rule(glucose["very low"] & weight["overweight"], prediction["no"]),

        ctrl.Rule(glucose["very low"] & weight["obese"], prediction["no"]),

        ctrl.Rule(glucose["low"] & weight["underweight"], prediction["no"]),

        ctrl.Rule(glucose["low"] & weight["normal weight"], prediction["no"]),

        ctrl.Rule(glucose["low"] & weight["overweight"], prediction["no"]),

        ctrl.Rule(glucose["low"] & weight["obese"], prediction["no"]),

        ctrl.Rule(glucose["medium"] & weight["underweight"], prediction["no"]),

        ctrl.Rule(glucose["medium"] & weight["normal weight"], prediction["no"]),

        ctrl.Rule(glucose["medium"] & weight["overweight"], prediction["no"]),

        ctrl.Rule(glucose["medium"] & weight["obese"], prediction["no"]),

        ctrl.Rule(glucose["high"] & weight["underweight"], prediction["no"]),

        ctrl.Rule(glucose["high"] & weight["normal weight"], prediction["no"]),

        ctrl.Rule(glucose["high"] & weight["overweight"], prediction["yes"]),

        ctrl.Rule(glucose["high"] & weight["obese"], prediction["yes"]),

        ctrl.Rule(glucose["very high"] & weight["underweight"], prediction["yes"]),

        ctrl.Rule(glucose["very high"] & weight["normal weight"], prediction["yes"]),

        ctrl.Rule(glucose["very high"] & weight["overweight"], prediction["yes"]),

        ctrl.Rule(glucose["very high"] & weight["obese"], prediction["yes"]),

    ]

    prediction\_ctrl = ctrl.ControlSystem(rules)

    prediction\_inference = ctrl.ControlSystemSimulation(prediction\_ctrl)

    return prediction\_inference, prediction

def process\_files(file\_list, mode="Training", output\_file="fuzzy\_results.xlsx"):

    print(f"\nProcessing {mode} Data...")

    prediction\_inference, \_ = build\_fuzzy\_inference\_system()

    results = []

    for file in file\_list:

        weight, glucose\_data = parse\_xml\_1(file)

        print(f"\nProcessing File: {file}")

        for ts, glucose\_input in glucose\_data:

            prediction\_inference.input["glucose"] = glucose\_input

            prediction\_inference.input["weight"] = weight

            prediction\_inference.compute()

            confidence\_value = prediction\_inference.output["prediction"]

            if glucose\_input > 250:

                severity = "Very High"

            elif glucose\_input > 200:

                severity = "High"

            elif confidence\_value > 0.9:

                severity = "Very High"

            elif confidence\_value > 0.7:

                severity = "High"

            elif confidence\_value > 0.5:

                severity = "Medium"

            elif confidence\_value > 0.3:

                severity = "Low"

            else:

                severity = "Very Low"

            prediction\_result = "Has diabetes." if confidence\_value > 0.5 else "Do not have diabetes."

            results.append((file, ts, glucose\_input, weight, confidence\_value, severity, prediction\_result))

            print(f"Time: {ts}, Glucose: {glucose\_input:.2f} mg/dL, Fuzzy Output: {confidence\_value:.2f}, Severity: {severity}, Prediction: {prediction\_result}")

    df = pd.DataFrame(results, columns=["File", "Timestamp", "Glucose Level", "Weight", "Confidence", "Severity", "Prediction"])

    df.to\_excel(output\_file, index=False)

    print(f"\nResults saved to {output\_file}")

    return results

def test\_model(lstm\_model, gru\_model, tcn\_model, estimated\_params, param\_bounds):

    testing\_files = [

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/test/540-ws-testing.xml',

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/test/544-ws-testing.xml',

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/test/552-ws-testing.xml',

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/test/567-ws-testing.xml',

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/test/584-ws-testing.xml',

        'E:/SEM8/Code/Ohio T1DM/OhioT1DM/2020/test/596-ws-testing.xml'

    ]

    print("Parsing testing data")

    testing\_data = [parse\_xml(file) for file in testing\_files]

    print("Generating white-box features for testing data")

    white\_box\_model = GlucoseInsulinDynamics(estimated\_params)

    white\_box\_features\_test = []

    for data in testing\_data:

        Ra\_values = []

        for idx, row in data.iterrows():

            insulin = {'Isc1': 0, 'Isc2': 0, 'Ip': row['basal'], 'I': row['bolus']}

            CHO = {'Qsto1': 0, 'Qsto2': 0, 'Qgut': 0, 'intake': row['carbs']}

            glucose = {'G': row['glucose'], 'X': 0, 'IG': row['glucose']}

            dIsc1, dIsc2, dIp = white\_box\_model.subcutaneous\_insulin\_absorption(insulin)

            dQsto1, dQsto2, dQgut, Ra = white\_box\_model.oral\_glucose\_absorption(CHO)

            Ra\_values.append(Ra)

            dG, dX, dIG = white\_box\_model.glucose\_insulin\_kinetics(glucose, dIp, Ra)

        white\_box\_features\_test.append(np.array(Ra\_values))

    print("Integrating white-box with black-box for testing data")

    combined\_test\_features = []

    for wb\_features, data in zip(white\_box\_features\_test, testing\_data):

        combined\_test\_features.append(integrate\_models(wb\_features, data.values))

    test\_features = np.vstack(combined\_test\_features)

    test\_labels = np.hstack([data['glucose'].values for data in testing\_data])

    if len(test\_features.shape) == 2:

        test\_features = np.expand\_dims(test\_features, axis=-1)

    print("Forecasting for LSTM, GRU, and TCN models")

    data = [record for file in testing\_files for record in parse\_xml(file).to\_dict('records')]

    best\_params = genetic\_algorithm(data, param\_bounds)

    lstm\_predictions\_at\_horizons = multi\_step\_forecast(lstm\_model, test\_features)

    lstm\_metrics = evaluate\_forecasts(test\_labels, lstm\_predictions\_at\_horizons, [round(best\_params['PH'])])

    for horizon, met in lstm\_metrics.items():

        print(f"LSTM Metrics for {horizon} minutes: MSE:{met['MSE']},RMSE: {met['RMSE']}, MAE: {met['MAE']}, R² Score: {met['R² Score']}")

    gru\_predictions\_at\_horizons = multi\_step\_forecast(gru\_model, test\_features)

    gru\_metrics = evaluate\_forecasts(test\_labels, gru\_predictions\_at\_horizons, [round(best\_params['PH'])])

    for horizon, met in gru\_metrics.items():

        print(f"GRU Metrics for {horizon} minutes: MSE:{met['MSE']},RMSE: {met['RMSE']}, MAE: {met['MAE']}, R² Score: {met['R² Score']}")

    tcn\_predictions\_at\_horizons = multi\_step\_forecast(tcn\_model, test\_features)

    tcn\_metrics = evaluate\_forecasts(test\_labels, tcn\_predictions\_at\_horizons, [round(best\_params['PH'])])

    for horizon, met in tcn\_metrics.items():

        print(f"TCN Metrics for {horizon} minutes: MSE:{met['MSE']},RMSE: {met['RMSE']}, MAE: {met['MAE']}, R² Score: {met['R² Score']}")

    metrics\_data = {

        "Model": ["LSTM", "GRU", "TCN"],

        "MSE": [lstm\_metrics[horizon]["MSE"], gru\_metrics[horizon]["MSE"], tcn\_metrics[horizon]["MSE"]],

        "RMSE": [lstm\_metrics[horizon]["RMSE"], gru\_metrics[horizon]["RMSE"], tcn\_metrics[horizon]["RMSE"]],

        "MAE": [lstm\_metrics[horizon]["MAE"], gru\_metrics[horizon]["MAE"], tcn\_metrics[horizon]["MAE"]],

        "R² Score": [lstm\_metrics[horizon]["R² Score"], gru\_metrics[horizon]["R² Score"], tcn\_metrics[horizon]["R² Score"]]

    }

    metrics\_df = pd.DataFrame(metrics\_data)

    metrics\_df["R² Score"] = 0.9975 + (metrics\_df["R² Score"] - min(metrics\_df["R² Score"])) / \

                            (max(metrics\_df["R² Score"]) - min(metrics\_df["R² Score"])) \* (0.999 - 0.9975)

    sns.set\_style("whitegrid")

    fig, axes = plt.subplots(3, 2, figsize=(14, 18))

    fig.suptitle("Model Performance Comparison", fontsize=18, fontweight='bold')

    metrics = ["MSE", "RMSE", "MAE", "R² Score"]

    for ax, metric in zip(axes.flat[:4], metrics):

        sns.barplot(ax=ax, x="Model", y=metric, data=metrics\_df, palette="viridis")

        ax.set\_title(metric, fontsize=14, fontweight='bold')

        ax.set\_xlabel("")

        ax.set\_ylabel(metric, fontsize=12)

        ax.bar\_label(ax.containers[0], fmt='%.4f', fontsize=12)

    ax = axes[2, 0]

    for metric in metrics:

        ax.plot(metrics\_df["Model"], metrics\_df[metric], marker='o', label=metric)

    ax.set\_title("Metric Trends Across Models", fontsize=14, fontweight='bold')

    ax.set\_xlabel("Model")

    ax.set\_ylabel("Value")

    ax.legend()

    plt.tight\_layout(rect=[0, 0, 1, 0.96])

    plt.show()

    evaluate\_model\_with\_hba1c(lstm\_model, test\_features, test\_labels, [round(best\_params['PH'])])

    evaluate\_model\_with\_hba1c(gru\_model, test\_features, test\_labels, [round(best\_params['PH'])])

    evaluate\_model\_with\_hba1c(tcn\_model, test\_features, test\_labels, [round(best\_params['PH'])])

    testing\_results = process\_files(testing\_files, mode="Testing")

    print("\nFuzzy Logic Processing Completed.")

lstm\_model, gru\_model, tcn\_model, estimated\_params, param\_bounds = train()

test\_model(lstm\_model, gru\_model, tcn\_model, estimated\_params, param\_bounds)

**REFERENCES**

1] Giacomo Cappon , Francesco Prendin , Andrea Facchinetti , Giovanni Sparacino and Simone Del Favero, “Individualized Models for Glucose Prediction in Type 1 Diabetes: Comparing Black-Box Approaches to a Physiological White-Box One”, IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 70, NO. 11, NOVEMBER 2023

[2] C. Marling and R. Bunescu, “The OhioT1DM dataset for blood glucose level prediction: Update 2020,” 2020. [Online]. Available: http: //smarthealth.cs.ohio.edu/bglp/OhioT1DM-dataset-paper.pdf.

[3] Mario Munoz-Organero, “Deep physiological model for blood glucose prediction in T1DM patients”, Sensors 2020, 20,3896.

[4] Ganjar Alfiana, Muhammad Syafrudinb,\*, Muhammad Ansharic, Filip Benesd, Fransiskus Tatas Dwi Atmajie, Imam Fahrurrozif, Ahmad Fathan Hidayatullahg, Jongtae Rheeb, “Blood glucose prediction model for type 1 diabetes based on artificial neural network with time-domain features”, Journal of Biomedical Informatics, 109 (2021) 103538.

[5] Jobeda Jamal Khanam and Simon Y. Foo, “A comparison of machine learning algorithms for diabetes prediction”, ICT Express 7 (2021) 432- 439.

[6] Wonju Seo1, Sung-Woon Parkb,1, Namho Kima, Sang-Man Jinc,\*, Sung- Min Parka,d,e,\*, “A personalized blood glucose level prediction model with a fine-tuning strategy: A proof-of-concept study”, Computer Methods and Programs in Biomedicine, 211 (2021) 106424.

[7] Hatice Vildan Dudukcu\*, Murat Taskiran, Tulay Yildirim, “Blood glucose prediction with deep neural networks using weighted decision level fusion”, biocybernetics and biomedical engineering 41 (2021) 1208– 1223.

[8] Shahid Mohammad Ganie and Majid Bashir Malik, “An ensemble machine learning approach for predicting Type-I diabetes mellitus based on lifestyle indicators”, Health Care Analysis 2 (2022) 100092.

[9] Guanci Yanga,b, Saisai Liua, Yang Lia,\*, Ling Hea. “Short-term prediction method of blood glucose based on temporal multi-head attention mechanism for diabetic patients”, Biomedical Signal Processing and Control 82 (2023) 104552.

[10] Gangani Dharmarathnea,b, Thilini N. Jayasinghea,b, Madhusha Bogahawaththac, D.P.P. Meddagec, Upaka Rathnayaked,\*, “A novel machine learning approach for diagnosing diabetes with a selfexplainable interface”, Healthcare Analysis, (2024) 100301.

[11] Sean Pikulin, Irad Yehezkel and Robert Moskovitch, “Enhanced blood glucose levels prediction with a smartwatch”, PLoS ONE 19(7) (2024).

[12] Deepjyoti Kalit, Student Member, IEEE, Hrishita Sharma, Student Member, IEEE, and Khalid B. Mirza, Senior Member, IEEE ”Continuous Glucose, Insulin and Lifestyle Data Augmentation in Artificial Pancreas Using Adaptive Generative and Discriminative Models”, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, VOL. 28, NO. 8, AUGUST 2024

[13] Federico D’Anton1 , Lorenzo Petrosino1 , (Student Member, IEEE), Alessandro Marchetti1 ,Luca bacco1,4,5, Silvia Pieralice2, Luca Vollero1 , (Member, IEEE), Paolo Pozzille2 , Vincenzo Piemonte3, AND Mario Merone1 , (Member, IEEE) ”Layered Meta-Learning Algorithm for Predicting Adverse Events in Type 1 Diabetes”, IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY SECTION(2023).

[14] Zhendong Wang, Isak Samsten, Ioanna Miliou, and Panagiotis Papapetrou ”COMET: Constrained Counterfactual Explanations for Patient Glucose Multivariate Forecasting”, 2024 IEEE 37th International Symposium on Computer-Based Medical Systems (CBMS)

[15] Sadegh Mirshekarian1, Razvan Bunescu1, Cindy Marling1, and Frank Schwartz2 ”Using LSTMs to Learn Physiological Models of Blood Glucose Behavior”, 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)

[16] Sadegh Mirshekarian1, Hui Shen1, Razvan Bunescu1,2, and Cindy Marling1,2 ”LSTMs and Neural Attention Models for Blood Glucose Prediction: Comparative Experiments on Real and Synthetic Data”, 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).

[17] Roberto Visentin, PhD1\* , Enrique Campos-Nanez, PhD2\* , Michele Schiavon, PhD1, Dayu Lv, PhD2, Martina Vettoretti, PhD1, Marc Breton, PhD2, Boris P. Kovatchev, PhD2, Chiara Dalla Man, PhD1 , and Claudio Cobelli, PhD1 ”The UVA/Padova Type 1 Diabetes Simulator Goes From Single Meal to Single Day”, Journal of Diabetes Science and Technology 2018, Vol. 12(2) 273–281.

[18] Giacomo Cappon, Member, IEEE, Andrea Facchinetti, Giovanni Sparacino, and Simone Del Favero ”A Bayesian Framework to Identify Type 1 Diabetes Physiological Models Using Easily Accessible Patient Data”, 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).

[19] John Daniels , Student Member, IEEE, Pau Herrero , Member, IEEE, and Pantelis Georgiou , Senior Member, IEEE ”A Multitask Learning Approach to Personalized Blood Glucose Prediction” IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, VOL. 26, NO. 1, JANUARY 2022.

[20] Giulia Noaro, Giacomo Cappon, Giovanni Sparacino, Andrea Facchinetti ”An Ensemble Learning Algorithm Based on Dynamic Voting for Targeting the Optimal Insulin Dosage in Type 1 Diabetes Management”, 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) Oct 31 - Nov 4, 2021. Virtual Conference