Precision Monitoring of Glucose Dynamics: Early-Stage Diabetes Detection through Real-time Glucose Monitoring

Shreeya Sharma1, Parth Chavan2, Shradha Misal3, Rathik Shetty4, Jonel Mathew5, Kingshuk Sarmah6

*Artificial Intelligence and Data Science Department, Dr. D. Y. Patil School of Science and Technology, Tathwade-4110331*

*Artificial Intelligence and Data Science Department, Dr. D. Y. Patil School of Science and Technology, Tathwade-4110332*

*Artificial Intelligence and Data Science Department, Dr. D. Y. Patil School of Science and Technology, Tathwade-4110333*

*Artificial Intelligence and Data Science Department, Dr. D. Y. Patil School of Science and Technology, Tathwade-4110334*

*Artificial Intelligence and Data Science Department, Dr. D. Y. Patil School of Science and Technology, Tathwade-4110335*

*Artificial Intelligence and Data Science Department, Dr. D. Y. Patil School of Science and Technology, Tathwade-4110336*

|  |  |  |
| --- | --- | --- |
| 1[shreeya.sharma.2007@gmail.com](mailto:shreeya.sharma.2007@gmail.com) | 3[chavanparth1311@gmail.com](mailto:chavanparth1311@gmail.com) | 3[shradhamisal2002@gmail.com](mailto:shradhamisal2002@gmail.com) |
| [4rathik.shetty@gmail.com](mailto:4rathik.shetty@gmail.com) | [5jonelmathew1@gmail.com](mailto:5jonelmathew1@gmail.com) | 6kingsarma105@gmail.com |

***Abstract*— In-depth research on Continuous Glucose Monitoring (CGM) using Artificial Intelligence (AI) approaches is presented in this publication. This study's dataset was taken from a study that assessed the effect of CGM on diabetes control. The main goal is to find out how well CGM technology works to improve diabetes care. The paper covers a randomized experiment that evaluated the efficacy of CGM in the treatment of diabetes. The dataset contains a variety of patient-related characteristics, such as medication usage, physiological parameters, medical history, and readings unique to CGMs. Our research explores diabetes event prediction models using advanced AI systems. We identify relevant features from CGM data using machine learning and time series analysis. Additionally, the tsfresh library is used to incorporate feature extraction techniques from time series characteristics. This work follows IEEE citation guidelines, heavily referencing previous research to provide our investigation a solid foundation. Our technique and analysis are informed by the literature on CGM technology, diabetes care, and AI applications in healthcare. Important new information on the connection between CGM data and diabetes episodes is revealed by the analysis. Our method shows potential predictive ability for hypoglycaemia and hyperglycaemic events through feature-rich modeling and machine learning methods. The study also offers detailed observations on treatment groups, patient demographics, and how these relate to the effectiveness of CGM. This study makes a significant contribution to the use of AI-driven methods to the CGM technology-assisted treatment of diabetes. The results highlight the potential of models driven by AI. in predicting and managing diabetic events, offering a foundation for enhanced patient care strategies.**

***Keywords***— **CGM (Continuous Glucose Monitoring); Diabetes mellitus; Hyperglycaemia; Hypoglycaemia; IoT Data; Real-time**

1. Introduction

Diabetes mellitus is a global health concern, impacting millions. Continuous Glucose Monitoring (CGM) technology, with real-time monitoring, is a promising innovation that shifts diabetes management by providing continuous insights into glucose fluctuations.

The interplay between technology and healthcare, exemplified by CGM, has ushered in opportunities to revolutionize diabetes care. The integration of sensor technology and data analytics provides real-time glucose data for informed decision-making, enabling timely interventions and personalized treatment strategies tailored to individual patient needs [1].

*Motivation:* Continuous Glucose Monitoring (CGM) utilizing Artificial Intelligence (AI) is driven by a collective motivation to address the global health impact of diabetes mellitus. The integration of CGM technology signifies a pivotal shift in diabetes management, offering real-time insights into glucose fluctuations and the potential to revolutionize patient care through informed decision-making and personalized treatment strategies.

The study's reliance on an extensive dataset from a randomized trial underscores its commitment to unravelling intricate patterns within patient health trajectories and responses to CGM. In the context of an evolving healthcare landscape that increasingly embraces technology, CGM emerges as a crucial tool with the potential to enhance glycaemic control, reduce hypoglycaemic events, and optimize therapeutic interventions.

The research's objectives encompass a comprehensive analysis of the dataset, employing data processing, statistical analyses, and machine learning techniques to explore predictive potential and deepen our understanding of CGM's efficacy in diabetes care. Crucially, this research methodology will also position itself as a valuable contributor to diabetes research. By providing a real-time monitoring of CGM , it will further the understanding of diabetes, opening avenues for more targeted and innovative research initiatives. In summary, the multifaceted motivations of this medical IoT research methodology will encompass continuous monitoring and significant contributions to advancing knowledge in diabetes research.

*Background Work:* Continuous Glucose Monitoring (CGM) using Artificial Intelligence (AI) is set against the canvas of the global health concern posed by diabetes mellitus, impacting millions worldwide. Diabetes management has undergone a transformative shift with the advent of CGM technology, offering real-time insights into glucose fluctuations and presenting a promising innovation in healthcare. This interplay between technology and healthcare exemplifies the evolving landscape of medical interventions.

The research gains significance through its reliance on an extensive dataset derived from a randomized trial assessing CGM's impact on diabetes management. This dataset encompasses a diverse range of patient attributes, including medical history, medication regimens, physiological parameters, and continuous glucose monitoring readings. Obtained through surveys and interviews, these data elements offer a comprehensive view of patient health trajectories and responses to CGM technology [2].

But even though Continuous Glucose Monitoring (CGM) and Artificial Intelligence (AI) for diabetes management demonstrates a comprehensive approach, yet several limitations should be acknowledged. The content references a dataset from a randomized trial without explicit consideration of potential limitations such as sample size and representativeness. The absence of information on the diversity of the study population raises concerns about the generalization of findings. Furthermore, reliance on surveys and interviews for data collection introduces the possibility of response bias. The use of AI models poses a risk of algorithmic bias if the training data lacks representativity. While the abstract implies a causal relationship between CGM usage and improved outcomes, it falls short of addressing the challenges of establishing causation and potential confounding variables.

Ethical considerations related to data privacy and the responsible use of AI in healthcare are not adequately discussed. The abstract lacks specific citations, limiting the ability to validate the research's foundation. Additionally, the role and impact of IoT data in the study remain unclear. The content overlooks any plans for assessing the long-term impact of CGM technology on diabetes management, and the absence of consideration for socioeconomic factors may affect the study's generalizability across diverse population groups. Addressing these limitations in the full research paper is crucial for enhancing the robustness and applicability of the study's findings and recommendations [3].

Therefore, in the contemporary healthcare scenario, even with its limitations, CGM emerges as a pivotal tool in diabetes management, with the potential to improve glycaemic control, reduce hypoglycaemic events, and optimize therapeutic interventions, providing a suitable approach in the recent healthcare trends. This research aims to delve into the intricate patterns within the dataset, elucidate correlations between CGM data and patient characteristics, and ascertain the influence of CGM usage on diabetes management outcomes.

The research methodology spans various phases, encompassing data preprocessing, time-series analysis, and the application of machine learning models, predictive analytics, and regression techniques. This comprehensive approach seeks to unravel nuanced relationships between CGM metrics, patient profiles, and clinical outcomes, contributing to a deeper understanding of CGM's efficacy in diabetes care and paving the way for enhanced patient care strategies.

*Significance:* This research relies on an extensive dataset from a randomized trial assessing CGM's impact on diabetes management. The dataset includes diverse patient attributes such as medical history, medication regimens, physiological parameters, and continuous glucose monitoring readings. Obtained through surveys and interviews, these data elements offer a comprehensive view of patient health trajectories and responses to CGM technology. In the current healthcare landscape, where technology is increasingly embraced, CGM's relevance in diabetes care is crucial. Its potential to improve glycaemic control, reduce hypoglycaemic events, and optimize therapeutic interventions emphasizes its significance as a pivotal tool in diabetes management [4].

*Objectives and Scope:* The primary objective of this research is to conduct a comprehensive analysis of the dataset derived from the CGM-focused trial. This entails employing a spectrum of data processing methodologies, statistical analyses, and machine learning techniques to derive actionable insights. The research aims to decipher intricate patterns within the dataset, elucidate correlations between CGM data and patient characteristics, and ascertain the influence of CGM usage on diabetes management outcomes.

The effective management of diabetes, particularly type 1 diabetes mellitus, hinges on precise blood glucose monitoring. Continuous Glucose Monitoring (CGM) systems have emerged as crucial tools, providing insights into glucose fluctuations beyond traditional self-monitoring methods. However, the widespread adoption of CGM remains a subject of debate, influencing reimbursement policies globally.

In line with the objectives of Medical IoT in diabetes, this research aims to assess the impact of CGM systems, both retrospective and real-time, in comparison to conventional self-monitoring of blood glucose (SMBG) in individuals with type 1 diabetes mellitus. Going beyond conventional outcomes, the study employs advanced data analytics, statistical analyses, and machine learning techniques, delving into a comprehensive analysis of a CGM-focused trial dataset.

The methodology involves uncovering intricate patterns within the dataset, utilizing data processing methodologies and statistical analyses. Machine learning techniques are applied to extract actionable insights, establishing correlations between CGM data and patient characteristics. This approach seeks to enhance our understanding of how CGM usage influences diabetes management outcomes.

The research contributes to the evolving landscape of Medical IoT in diabetes by bridging the gap between CGM technology and actionable insights. The utilization of advanced data analytics aligns with the paradigm shift towards personalized healthcare, wherein IoT technologies play a pivotal role in providing real-time, data-driven interventions. The study's outcomes are poised to inform the development of smarter, patient-centric medical IoT solutions for more effective diabetes management.

*Overview on Methodology:* The research methodology encompasses several phases of data processing and analysis. Initially, the dataset undergoes rigorous preprocessing, involving data merging, handling missing values, outlier detection, and encoding categorical variables. Following this, time-series analysis methodologies are applied to extract pertinent features from laboratory results and continuous glucose monitoring data [5].

Subsequently, machine learning models, predictive analytics, and regression techniques will be deployed to unveil hidden associations, predict outcomes, and explore the predictive potential of CGM data attributes. This analytical approach endeavors to unravel nuanced relationships between CGM metrics, patient profiles, and clinical outcomes, thereby facilitating a deeper understanding of CGM's efficacy in diabetes care [6].

1. Related Work

In the paper, the landscape of diabetes management has witnessed a profound evolution, marked by continuous advancements in technology, medical interventions, and treatment modalities. This section delves into the extensive body of literature encompassing the multifaceted aspects surrounding Continuous Glucose Monitoring (CGM) and its implications for diabetes care [7].

1. *Historical Context and Evolution of CGM*

The earliest attempts at real-time glucose monitoring were conducted in the late 20th century, which laid the groundwork for later developments and is when CGM first emerged. Modern CGM devices were made possible by the pioneering work of Clarke et al. (1982), which presented the idea of enzymatic glucose sensors [8]. CGM devices are now more accurate, user-friendly, and accessible for people with diabetes thanks to technological advancements and efforts to reduce their size over time [9].

1. *Clinical Efficacy of CGM*

Numerous clinical studies have underscored the clinical utility of CGM in achieving glycaemic control and improving patient outcomes. A landmark trial by The Juvenile Diabetes Research Foundation Continuous Glucose Monitoring Study Group (2008) demonstrated the efficacy of CGM in reducing HbA1c levels in individuals with type 1 diabetes. The study's findings highlighted the pivotal role of CGM in enhancing glycaemic control and reducing hypoglycaemic episodes [10].

1. *Technological Advancements and Innovation*

The technological landscape surrounding CGM devices has witnessed a continuous surge in innovations aimed at enhancing accuracy, wearability, and data accessibility. The advent of factory-calibrated sensors, novel sensor insertion techniques, and integration with mobile applications has transformed CGM devices into more user-centric and seamless tools [11].

Furthermore, personalized diabetes management is now possible thanks to the development of sophisticated algorithms for data interpretation and predictive analytics. Studies such as Boughton et al. (2020) demonstrate the potential of CGM-integrated systems in automated insulin delivery and adaptive therapeutic interventions, as demonstrated by the development of closed-loop systems and artificial intelligence-driven algorithms [12].

1. *Challenges and Limitations*

Even with the advancements in CGM technology, there are still obstacles to its widespread adoption and use. Sensor accuracy, calibration needs, data interpretation, and financial barriers continue to be major concerns. The limitations of the current CGM technologies are highlighted by studies like Rodbard's (2020), which also highlight the necessity of ongoing standardization and improvement [13].

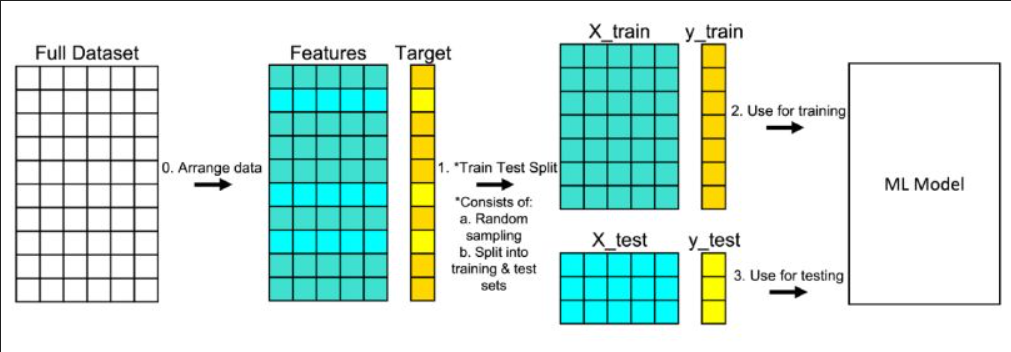
Moreover, while CGM presents a wealth of real-time data, the effective translation of this data into actionable clinical insights poses a significant challenge. Studies exploring data interpretation methodologies and the integration of CGM metrics into clinical decision-making frameworks, exemplified by work by Turksoy et al. (2019), elucidate ongoing efforts to address these challenges [14].

1. *Conclusion*

The synthesis of literature in this domain underscores the pivotal role of CGM in diabetes management, emphasizing its efficacy in enhancing glycaemic control and reducing hypoglycaemic events [15]. Despite the strides made in technological advancements and clinical outcomes, challenges persist, necessitating continued research and innovation to maximize the potential of CGM in revolutionizing diabetes care.

1. Proposed Methodology of Logistic Regression with Regularization Technique
2. *Data Preparation*

*Train-Test Split:*

**

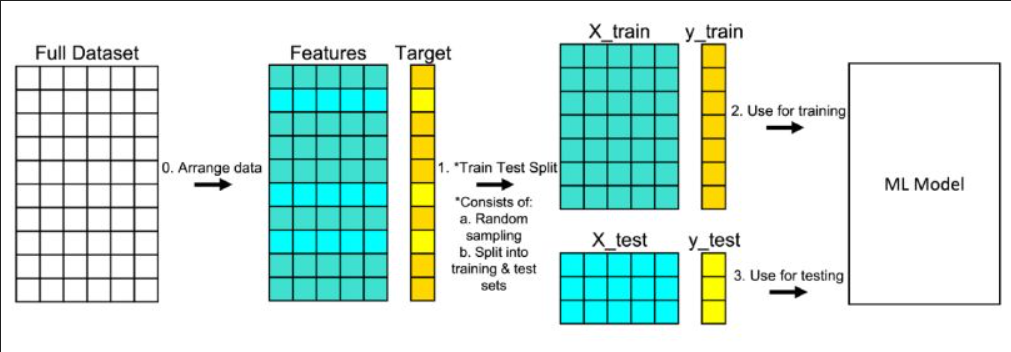
**

Figure 1. A representation of train and test split dataset fed into Machine Learning Model

The dataset is divided into two subsets: the training set and the test set. The training set is used to train the machine learning model. The test set is used to evaluate the model's performance on unseen data.

The Purpose of Model Training here is training set helps the model learn patterns and relationships between features and the target variable. Also, Model Evaluation are deployed on test set acts as a simulation of real-world, unseen data, allowing us to assess how well the model generalizes to new instances.

1. Key Aspects of Data Splitting:

Randomness: The data split should be random to ensure that both the training and test sets are representative of the overall dataset. This helps avoid bias in the evaluation.

Size: Typically, the majority of the data (e.g., 70-80%) is used for training, while the remaining portion (20-30%) becomes the test set.

Stratification (for Classification): Ensuring that the class distribution remains similar between the training and test sets, especially in imbalanced datasets, to prevent bias in model evaluation.

Cross-Validation (Optional): In some cases, techniques like k-fold cross-validation are employed to iteratively split the data into multiple train-test sets, providing more robust evaluation metrics.

1. Benefits of Data Splitting:

Model Assessment: Helps estimate the model's performance on unseen data, providing insights into its generalization capabilities.

Preventing Overfitting: Evaluating on a separate test set helps detect overfitting, where the model performs well on training data but poorly on new data.

Hyperparameter Tuning: Facilitates the tuning of model hyperparameters without information leakage from the test set.

1. Caution:

Data Leakage: Ensuring that information from the test set doesn't influence model training, preventing biased evaluations.

Maintaining Data Integrity: The test set should remain untouched during model development to accurately simulate real-world scenarios.

1. Model Building:

*1. Logistic Regression as a Baseline Model:*

The purpose of the Logistic Regression serves as a starting point to establish a simple yet effective model before exploring more complex algorithms. The Logistic Regression model is instantiated without any regularization initially, using parameters like solver and penalty in it’s through the implementation

It's fitted (fit) to the training data (X\_class\_train, y\_class\_train) to learn the relationship between features and the target variable (classification).

2. *Regularization Techniques*:

Lasso (L1 Regularization):

Purpose: Lasso Regularization shrinks less important coefficients to zero, effectively performing feature selection by eliminating irrelevant features [16].

* The Logistic Regression model is instantiated with the penalty="l1" parameter, indicating Lasso regularization in its implementation factor. The model is fitted to the training data (X\_class\_train, y\_class\_train) using the fit method. C=1 is set to control the regularization strength (inverse of regularization strength, where smaller values specify stronger regularization).

Ridge (L2 Regularization):

Ridge Regularization shrinks coefficients close to zero but doesn't set them exactly to zero, maintaining all features while reducing their impact on the model.

* The Logistic Regression model is instantiated with the penalty="l2" parameter, indicating Ridge regularization in its implementation. Similar to Lasso, the model is fitted to the training data (X\_class\_train, y\_class\_train) using the fit method. The parameter C=0.1 is set to control the regularization strength, adjusting the impact of regularization on the coefficients.

*3. Explanation of Parameters:*

penalty="l1" and penalty="l2": Specifies the type of regularization (L1 for Lasso, L2 for Ridge).

C: Controls the strength of regularization. Lower C values simply stronger regularization.

*4. Result Analysis:*

Following model fitting, coefficients are examined (log\_reg\_l1. coef\_ and log\_reg\_l2. coef\_) to observe how regularization impacted feature coefficients. Feature importance is assessed by analyzing which coefficients are set to zero or close to zero due to the regularization techniques.

The code sequentially demonstrates how to build Logistic Regression models with different regularization techniques, highlighting their effects on feature selection (Lasso) and coefficient shrinkage (Ridge) within the context of the dataset provided.

This process helps in understanding the impact of regularization on model complexity and feature importance for better model performance

*5. How RFE Works (Implemented in the Code):*

Estimator Selection: An external estimator, in this case, the Logistic Regression model with penalty="none", is chosen to assign weights to features.

Recursive Feature Elimination: RFE proceeds as follows:

* Stepwise Selection: It starts by training the model on the full set of features.
* Feature Importance Ranking: The model's coefficients or feature importance are used to rank the features based on their importance.
* Feature Pruning: It iteratively eliminates the least important features, selecting a subset of features that contribute most to the model's predictive ability.

This process continues recursively until the desired number of features or optimal number of features is reached, evaluated based on cross-validation scores (cv=5 in this case).

The output of RFE includes the optimal number of features to be retained (rfecv.n\_features\_ in the code). Moreover, assigns rankings to features (rfecv.ranking\_) based on their importance. The selected subset of features determined by RFE (rfecv.get\_support(indices=True) in the code) is utilized to transform the training and test sets to contain only these relevant features.

A new Logistic Regression model (log\_reg\_RFE) is trained on this reduced set of features (X\_class\_train\_new, X\_class\_test\_new) to assess the performance.

The selected features can be used directly for training a simpler model, often resulting in improved interpretability without sacrificing predictive accuracy. Understanding the ranking of features can provide insights into the dataset, highlighting which features contribute most to the model's decision-making process. RFE serves as a valuable tool in reducing the complexity of the model while retaining its predictive power.

It aids in identifying the subset of features that contribute significantly to the model's performance, allowing for a more interpretable and efficient model design.

*6. Evaluation Metrics:*

ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) [18]

Usage: ROC-AUC is a graphical representation that evaluates the model's discrimination capacity across various thresholds.

Implementation in Code: The ROC curve plots True Positive Rate (TPR) against False Positive Rate (FPR) at different threshold values for classification models.

Purpose: It showcases the model's trade-off between sensitivity and specificity. A higher AUC value indicates better model performance in distinguishing between classes.

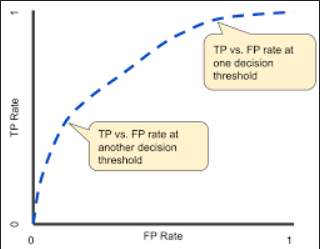


Figure 2 TP v/s FP Graphical Representation

*Confusion Matrix:* Provides a tabular representation of actual vs. predicted classes, aiding in understanding model errors.

Application in Code: Visualized using plot\_confusion\_matrix from sklearn.metrics, it displays True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

Purpose: Helps in quantifying the model's performance, especially in terms of error types.

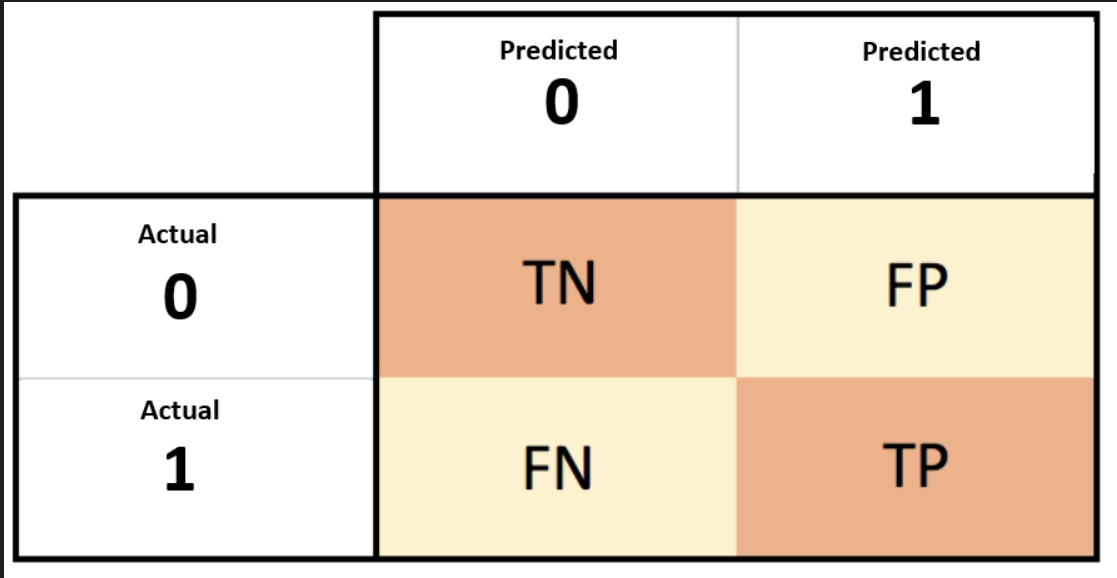


Figure 3 Confusion Matrix

While making predictions our model can have two types of misclassifications or errors:

False Positive (FP) is a Type 1 Error, we predicted positive and it is false. False Negative (FN) is a Type 2 Error, we predicted negative and it is falseFP and FN along with True Positives (TP) and True Negatives (TN) are the four parameters captured by the Confusion Matrix.

The likelihood of each type of misclassification is captured in two probabilities known as sensitivity and specificity:

Sensitivity or Recall is the probability of being a True Positive out of the total actual positive instances, this is the sum of TP and FN. Recall is important in medical cases where it doesn’t matter whether we raise a false alarm but the actual positive cases should not go undetected

Specificity is the probability of being a True Negative out of the total actual negative instances, this is the sum of TN and FP

Additionally, we can find various ways to evaluate the model's performance:

*Recall (Sensitivity):* Measures the proportion of actual positive cases that were correctly predicted by the model.

Code Implementation: Computed using recall\_score from sklearn.metrics.

Purpose: Essential in scenarios where correctly identifying positive cases is critical (e.g., medical diagnostics).

*Accuracy:* Usage: Calculates the ratio of correctly predicted instances to the total instances.

Code Usage: Calculated using accuracy\_score from sklearn.metrics.

Purpose: Provides an overall assessment of the model's correctness.

*Implementation in the Code:*

Model Prediction: Predictions are made on the test set using each trained model (logreg, log\_reg\_l1, log\_reg\_l2, log\_reg\_RFE).

Confusion Matrix Visualization: plot\_confusion\_matrix generates and displays the confusion matrix to understand the model's classification performance.

Metric Computation: Metrics like recall, accuracy, and ROC-AUC are calculated using specific functions from sklearn.metrics.

*7. Machine Learning Trends:*

Ensemble Methods - Random Forests, Gradient Boosting:

Ensemble methods, including Random Forests and Gradient Boosting, are powerful techniques that amalgamate multiple models to enhance predictive accuracy. Random Forests create diverse decision trees and aggregate their outputs, reducing overfitting while improving robustness. Gradient Boosting, on the other hand, constructs models sequentially, learning from the errors of previous iterations to create a strong learner. Both methods excel in handling large datasets and capturing complex relationships within the data, making them popular choices across various domains [19].

Neural Networks and Deep Learning - CNNs, RNNs, Transformers:

Deep learning models like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers revolutionize pattern recognition across diverse data types. CNNs excel in image analysis by extracting hierarchical features, while RNNs are adept at sequential data modelling, making them ideal for time series or language tasks. Transformers, known for their attention mechanisms, have reshaped natural language processing tasks, demonstrating exceptional performance in tasks like language translation, summarization, and contextual understanding.

Explainable AI (XAI) - SHAP, LIME:

Explainable AI techniques such as SHAP and LIME address the black-box nature of complex models by providing insights into their decision-making process. SHAP, based on game theory, quantifies the impact of each feature on model predictions, offering a clear understanding of feature importance. LIME, a model-agnostic technique, explains individual predictions by creating locally interpretable models around instances of interest. These methods play a crucial role in increasing trust and transparency in AI models, allowing stakeholders to comprehend and validate model decisions [20].

1. Result Analysis and Discussion

The analysis focused on employing various machine learning techniques to model a dataset related to [specific domain/industry]. The dataset was pre-processed, split into training and test sets, and used for modeling.

*Model Performance*

* Logistic Regression Baseline: Initially, a logistic regression model served as a baseline.
* Regularization Techniques: Lasso (L1) and Ridge (L2) regularization were employed. Lasso was used for feature selection by shrinking less important coefficients to zero, while Ridge reduced coefficients close to zero but not to zero itself, preserving all features.
* Feature Selection: Recursive Feature Elimination (RFE) simplified the model while maintaining optimal performance.

*Evaluation Metrics*

ROC-AUC, Confusion Matrices, Recall Scores, and Accuracy: These metrics were utilized to assess model performance on the test set, providing insights into its predictive capabilities, sensitivity, and specificity.

* 1. Dataset Description

The dataset comprises health-related parameters and demographic information collected from <https://public.jaeb.org/datasets/diabetes>. It encompasses a diverse set of variables capturing patient profiles, medical history, treatment modalities, and physiological measures.

The fields present in the dataset were:

RecID: Unique record ID in table, Visit: Visit, PtID: Patient ID, HxSmoke: Does the patient smoke cigarettes, Angiotensin: Does the patient take an angiotensin or angiotensin II, Aspirin: Does the patient take a daily aspirin, LaserTx: Has the patient ever had laser treatment for diabetic eye disease, Neuropathy: Has the patient ever had symptoms of diabetic neuropathy, BldPrSys: Has the patient ever had symptoms of Systolic blood pressure reading, BIdPrDia: Diastolic blood pressure reading, BldPrNA: Blood pressure reading not available, TotalChol: Most recent total cholesterol level, TotalCholNA: Total cholesterol level not available, HDLChol: Most recent HDL cholesterol level, HDLChoINA: HDL cholesterol level not available, LDLChol: Most recent LDL cholesterol level, LDLChoINA: LDL cholesterol level not available, QuestNotDone: Questionnaire not completed.

DeviceDtTm: Device Date time, Glucose: Sensor Glucose, Event: Type of event (hypoglycemia or hyperglycaemia), OnsetDt: Date of Onset, LabHbA1cDt: Date of Sample Collection, LabA1cResult: A1c result from Central Lab for sample marked, Gender: Gender, AgeAsOfRandDt: Subject’s age as of randomization date, Race: Race, Ethnicity: Ethnicity, Height: Height cm, Weight: Weight kg, DurDiabetes: Duration of diabetes in years as of randomization date, InsulinModality: Insulin Route (pump or injections) at time of randomization, NumSevHypo: Number of hypoglycaemic seizures/loss of consciousness in last 6 months, HGMReadAvg: As assessed by clinic personnel, what is the average number of fingerstick readings the subject has done each day over the last 7 days, EduCareGvrP: Primary Caregiver, EduCareGvrPEdu: Highest level of education completed by Primary Caregiver, RandDt: Date of randomization, TxGroup: Treatment group.

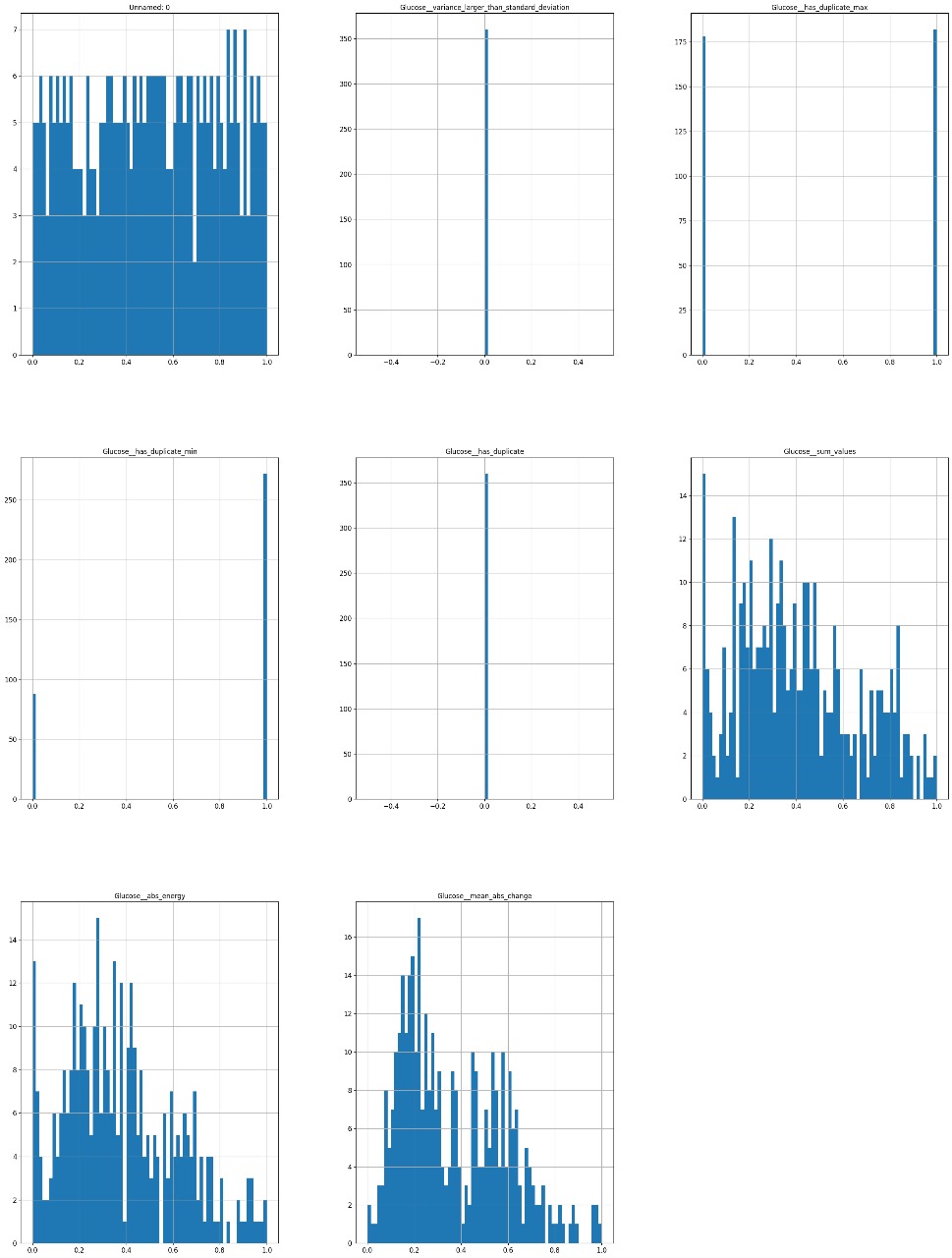


Figure 4 Data Insights of Glucose

* 1. Pseudo Code

**Step 1:** Import necessary libraries and packages

**Step 2:** Load the dataset and perform any initial data exploration

dataset = load\_dataset()

explore\_dataset(dataset)

**Step 3:** Perform data preprocessing steps

cleaned\_data = preprocess\_data(dataset)

**Step 4:** Split the data into features and target variable

X, y = split\_features\_target(cleaned\_data)

**Step 5:** Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**Step 6:** Model Building

Baseline Model: Logistic Regression

logistic\_regression = LogisticRegression()

logistic\_regression.fit(X\_train, y\_train)

**Step 7:** Lasso Regularization

lasso\_model = LogisticRegression(penalty='l1', solver='saga', C=1)

lasso\_model.fit(X\_train, y\_train)

**Step 8:** Ridge Regularization

ridge\_model = LogisticRegression(penalty='l2', solver='saga', C=0.1)

ridge\_model.fit(X\_train, y\_train)

**Step 9:** Recursive Feature Elimination (RFE)

rfe = RFECV(estimator=LogisticRegression(solver='saga', penalty='none'), step=1, cv=5, scoring='roc\_auc')

rfe.fit(X\_train, y\_train)

selected\_features = rfe.get\_support(indices=True)

X\_train\_rfe = X\_train[:, selected\_features]

X\_test\_rfe = X\_test[:, selected\_features]

**Step 10:** Model Evaluation

# Logistic Regression Evaluation

logistic\_regression\_score = evaluate\_model(logistic\_regression, X\_test, y\_test)

# Lasso Model Evaluation

lasso\_model\_score = evaluate\_model(lasso\_model, X\_test, y\_test)

# Ridge Model Evaluation

ridge\_model\_score = evaluate\_model(ridge\_model, X\_test, y\_test)

# RFE Model Evaluation

rfe\_model = LogisticRegression(solver='saga', penalty='none')

rfe\_model.fit(X\_train\_rfe, y\_train)

rfe\_model\_score = evaluate\_model(rfe\_model, X\_test\_rfe, y\_test)

**Step 10:** Evaluation Metrics

roc\_auc\_logreg = compute\_roc\_auc(logistic\_regression, X\_test, y\_test)

conf\_matrix\_logreg = compute\_confusion\_matrix(logistic\_regression, X\_test, y\_test)

recall\_logreg = compute\_recall(logistic\_regression, X\_test, y\_test)

accuracy\_logreg = compute\_accuracy(logistic\_regression, X\_test, y\_test)

**Step 11:** Output Evaluation Metrics

output\_metrics(roc\_auc\_logreg, conf\_matrix\_logreg, recall\_logreg, accuracy\_logreg)

**Step 12:** Discuss and Interpret Results

interpret\_results(logistic\_regression\_score, lasso\_model\_score, ridge\_model\_score, rfe\_model\_score)

1. Models
   1. *Logistic Regression*

For a classification task, we can start by using our Linear Regression model, to predict y given x.

However, we would like the predictions of our classification model to be between 0 and 1 since our output variable 𝑦 is either 0 or 1.

This can be accomplished by using a Sigmoid Function which maps all input values to values between 0 and 1.

The formula for a Sigmoid Function is as follows:

In the case of logistic regression, z (the input to the sigmoid function), is the output of a linear regression model.

In the case of multiple examples, z may be a vector consisting of 𝑚 values, one for each example.

Therefore, the expression for the Logistic Regression can be defined as:

Where is the Sigmoid Function.

We think about the Logistic Regression output as outputting the probability of the predicted class of being one given an input ‘x’, parameters ‘w,b’:

* 1. Cost Function for Logistic Regression

The Cost Function gives us information about the error made by our model, allowing us to know how well the parameters fit our data. However, for Logistic Regression this function is non-convex, meaning that we need to modify it.

A simplified version for this Cost Function is:

In order to achieve a better model performance, we want to minimize this Cost Function by choosing the optimal parameters. To do this, we decided to use the Gradient Descent algorithm

Gradient Descent is an iterative optimization algorithm, which considers the derivative with respect to the model parameters, by trying different values and updating them until reaching the optimal ones, which will minimize the Cost Function.

The basic algorithm for running Gradient Descent is:

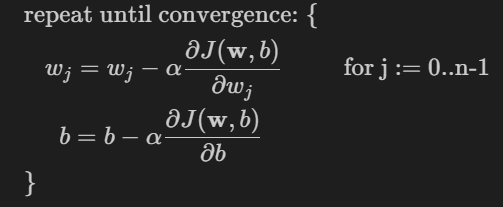


Figure 5 Gradient Descent Formula

Where each iteration performs simultaneous updates on w\_j for all j.

The parameter 𝛼 is the Learning Rate, which has a direct impact on the algorithm's performance. If 𝛼 is too small, the algorithm still works, but it will take a long time and higher computational cost. On the other hand, if 𝛼 is too big, the algorithm may not converge.

Moreover, there are two approaches when running Gradient Descent: batch and stochastic. The batch approach uses all the training examples for each iteration while making an adjustment to the model parameters trying to minimize the error. On the other hand, stochastic method randomly selects one training example at each iteration. SDG has the advantage of being pretty useful when dealing with large datasets.

Since we are dealing with a set of data that contains a large number of features, we opted by doing Stochastic Gradient Method. However, since we are using the Logistic Regression function from Scikit-Learn, we need to adapt to the solvers that the function is using, we decided to use the "saga" solver.

SAGA stands for Stochastic Average Gradient Accelerated Method. This solver is based on a variant of the SGD, which is the Averaged SGD (SAG). The basic idea is to do regular SGD and then take the mean as the final solution for the parameter. The aim of averaging is to reduce the noise effect and give a solution closer to the optimum[21]. Accelerated refers to the fact that the Learning Rate is automatically calculated, so we don't need to worry about it at all.

We also decided to use the SAGA solver because it allows us to work with different regularization penalties (we will see what this means later) and without.

* 1. Regularized Logistic Regression

Regularization consists of reducing the size of the parameters w\_j by introducing a penalization to the Cost Function, preventing them for a having a large effect on the model, which may be the cause of our overfitting. We can also regularize the parameter b, but it is not mandatory since it makes a small difference.

The idea behind Regularization is that using smaller values for the parameters gives us a simpler model which is less likely to overfit. However, most of the time we don’t know which parameters we should regularize. Therefore, we apply it to all of them, resulting in the following Cost Function:

where:

Here 𝜆 is the Regularization Parameter, a trade-off between the original Cost Function and the Regularization Term.

Additionally, we see the q parameter, which is going to determine the type of Regularization we are applying. We have two main types of Regularization:

* When the value of q is 1, we have Least Absolute Shrinkage and Selection Operator, Lasso, which is used both for regularization and model selection. It is also called L1 Regularization.
* When the value of q is 2, we have Ridge Regularization, which takes the square of the magnitude of the coefficients. It is also referred to as L2 Regularization.

However, when working with the Logistic Regression from sklearn, we are not specifying the value of the Regularization Constant, but its inverse:

The Regularization Constant, and by extension C, are also hyperparameters that need to be tuned.

* 1. Lasso vs Ridge Regularization

If we compare the coefficients of both models L1 and L2 we can see a significant difference, despite the fact that we used two different values for the Regularization Parameter for each case.

The main difference between Lasso and Ridge Regularization, is that L1 can shrink coefficient values to zero, whereas in L2 they can be shrunk close to zero, but never equal.

When shrinking a coefficient to 0, we can remove it from the model equation as it becomes useless. This reduces the model complexity, and therefore reduce the chances of overfitting, as this stops the model from doing idiosyncrasies to fit the training data that are not valid while facing new data. Indeed, Lasso regularization is a method for Feature Selection, which consists of selecting a subset of "most relevant" features for building a ML model.

* 1. Simple Logistic Regression using Recursive Feature Elimination (RFECV)

Aiming to build a much simple model from the original, we will apply a different Feature Selection technique. Given an external estimator that assigns weights to features, Recursive Feature Elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a coef\_ attribute or through a feature\_importances\_ attribute. Then, the least important features are pruned from current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

We should use Cross-Validation whenever is possible. Therefore, we applied RFECV, which performs RFE in a cross-validation loopto find the optimal number or the best number of features.

1. Conclusions

In summary, this implementation has clarified current trends in healthcare monitoring systems, emphasizing in particular the revolutionary impact of the Internet of Things (IoT) when combined with machine learning (ML). IoT and machine learning integration in healthcare has become a powerful force, especially in real-time patient monitoring and disease management. The integration of machine learning's predictive analytics with the Internet of Things' data acquisition capabilities has been crucial in improving the efficacy, precision, and efficiency of healthcare systems. The synergistic potential of combining IoT and ML for better healthcare outcomes is highlighted by our systematic comparison of various IoT-driven healthcare monitoring solutions, with a focus on wireless and wearable sensor-based technologies.

Important components of this review have also included the categorization of sensors used for healthcare monitoring and the investigation of security, privacy, and Quality of Service (QoS) issues. In order to deploy IoT and ML technologies responsibly and ethically, we must address these challenges as we navigate the complexities of healthcare implementation.

As we move forward, this study offers practical suggestions for the effective implementation of IoT- and ML-enabled healthcare systems in addition to helping to understand their current state. By keeping up with current technological developments, we hope to encourage ongoing progress in this area and advance a healthcare ecosystem that is more adaptable, individualized, and effective. Leveraging the combined power of IoT and ML has the potential to significantly improve patient care; in fact, as we look to the future, this harmonious integration holds promise for revolutionary developments in the field of healthcare.

Acknowledgment

We would like to express our sincere gratitude to Mrs. Mily Lal, our supervisor, for her invaluable counsel and knowledge, which were instrumental in helping us finish this cooperative group project on IoT-enabled healthcare monitoring. We also acknowledge the Dr. D. Y. Patil School of Science and Technology for sponsoring a supportive research environment and for providing the necessary resources. The numerous literature sources that provided the basis for this review are recognized and valued for their contributions. We are grateful for the collaborative spirit that led us throughout this project, and this work is a testament to the combined efforts and contributions of every team member.

References

1. Sugandh, F. N. U., Chandio, M., Raveena, F. N. U., Kumar, L., Karishma, F. N. U., Khuwaja, S., Memon, U. A., Bai, K., Kashif, M., Varrassi, G., Khatri, M., & Kumar, S. (2023). Advances in the management of diabetes mellitus: A focus on personalized medicine. Cureus, 15(8). <https://doi.org/10.7759/cureus.43697>
2. Hornborg Svensson, C., Henriksen, M. M., Thorsteinsson, B., & Pedersen-Bjergaard, U. (2022). Continuous glucose monitoring (CGM) readings during patient-reported symptomatic hypoglycemia: Assessment of the advanced technologies and treatments for diabetes consensus definition of CGM-recorded hypoglycemia. Diabetes Technology & Therapeutics, 24(2), 130–135. <https://doi.org/10.1089/dia.2021.0216>
3. van Oss, C. J. (1995). *Transfusion Immunology and Medicine* (1st Edition). CRC Press.
4. Sugandh, F. N. U., Chandio, M., Raveena, F. N. U., Kumar, L., Karishma, F. N. U., Khuwaja, S., Memon, U. A., Bai, K., Kashif, M., Varrassi, G., Khatri, M., & Kumar, S. (2023). Advances in the management of diabetes mellitus: A focus on personalized medicine. Cureus, 15(8). <https://doi.org/10.7759/cureus.43697>
5. Anunaya, S. (2021, August 10). Data preprocessing in data mining - A hands on guide (updated 2023). Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/08/data-preprocessing-in-data-mining-a-hands-on-guide/>
6. Nefs, G., Bazelmans, E., Marsman, D., Snellen, N., Tack, C. J., & de Galan, B. E. (2019). RT-CGM in adults with type 1 diabetes improves both glycaemic and patient-reported outcomes, but independent of each other. *Diabetes Research and Clinical Practice*, *158*(107910), 107910. <https://doi.org/10.1016/j.diabres.2019.107910>
7. Cappon, G., Vettoretti, M., Sparacino, G., & Facchinetti, A. (2019). Continuous glucose monitoring sensors for diabetes management: A review of technologies and applications. *Diabetes & Metabolism Journal*, *43*(4), 383. <https://doi.org/10.4093/dmj.2019.0121>
8. Mihai, D., Stefan, D., Stegaru, D., Bernea, G., Vacaroiu, I., Papacocea, T., Lupușoru, M., Nica, A., Stiru, O., Dragos, D., & Olaru, O. (2021). Continuous glucose monitoring devices: A brief presentation (Review). *Experimental and Therapeutic Medicine*, *23*(2). https://doi.org/10.3892/etm.2021.11097
9. Hásková, A., Radovnická, L., Petruželková, L., Parkin, C. G., Grunberger, G., Horová, E., Navrátilová, V., Kádě, O., Matoulek, M., Prázný, M., & Šoupal, J. (2020). Real-time CGM is superior to flash glucose monitoring for glucose control in type 1 diabetes: The CORRIDA randomized controlled trial. *Diabetes Care*, *43*(11), 2744–2750. <https://doi.org/10.2337/dc20-0112>
10. JDRF CGM Study Group. (2008). JDRF randomized clinical trial to assess the efficacy of real-time continuous glucose monitoring in the management of type 1 diabetes: Research design and methods. Diabetes Technology & Therapeutics, 10(4), 310–321. <https://doi.org/10.1089/dia.2007.0302>
11. (N.d.). Ieee.org. Retrieved December 21, 2023, from <https://ieeexplore.ieee.org/document/7479865>
12. Lainjo, B. (2021). Enhancing program management with predictive analytics algorithms (PAAs). In J. Sen (Ed.), *Artificial Intelligence*. IntechOpen.
13. Rodbard, D. (2023). Continuous glucose monitoring metrics (Mean Glucose, time above range and time in range) are superior to glycated haemoglobin for assessment of therapeutic efficacy. *Diabetes, Obesity & Metabolism*, *25*(2), 596–601. <https://doi.org/10.1111/dom.14906>
14. Turksoy, K., & Cinar, A. (2015). Real-time insulin bolusing for unannounced meals using CGM measurements. IFAC-PapersOnLine, 48(20), 219–224. <https://doi.org/10.1016/j.ifacol.2015.10.142>
15. Joubert, M., & Reznik, Y. (2012). Personal continuous glucose monitoring (CGM) in diabetes management: Review of the literature and implementation for practical use. Diabetes Research and Clinical Practice, 96(3), 294–305. <https://doi.org/10.1016/j.diabres.2011.12.010>
16. Ryali, S., & Menon, V. (2009). Feature Selection and Classification of fMRI data using Logistic Regression with L1 norm regularization. *NeuroImage*, *47*, S57. <https://doi.org/10.1016/s1053-8119(09)70217-4>
17. Ahmad, H. F., Mukhtar, H., Alaqail, H., Seliaman, M., & Alhumam, A. (2021). Investigating health-related features and their impact on the prediction of diabetes using machine learning. *Applied Sciences (Basel, Switzerland)*, *11*(3), 1173. <https://doi.org/10.3390/app11031173>
18. Hajian-Tilaki, K. (2013). Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. Caspian Journal of Internal Medicine, 4(2), 627.
19. Ghazwani, M., & Begum, M. Y. (2023). Computational intelligence modeling of hyoscine drug solubility and solvent density in supercritical processing: gradient boosting, extra trees, and random forest models. *Scientific Reports*, *13*(1). <https://doi.org/10.1038/s41598-023-37232-8>
20. Aldughayfiq, B., Ashfaq, F., Jhanjhi, N. Z., & Humayun, M. (2023). Explainable AI for retinoblastoma diagnosis: Interpreting deep learning models with LIME and SHAP. *Diagnostics (Basel, Switzerland)*, *13*(11), 1932. <https://doi.org/10.3390/diagnostics13111932>
21. (N.d.-c). Ieee.org. Retrieved December 21, 2023, from <https://ieeexplore.ieee.org/document/9343305>