Journal of Computing and Applied Informatics (JoCAI) Vol. 5, No. 2, 2021 | 49 – 59

DATA SCIENCE
Journal of Computing and Applied Informatics

A Web-Based Diabetes Prediction Application Using XGBoost Algorithm

Herlambang Dwi Prasetyo[1], Pandu Ananto Hogantara[2], and Ika Nurlaili Isnainiyah[3]

[1,2,3]Department of Computer Science, Faculty of Computer Science, University of Pembangunan Nasional Veteran Jakarta, Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, 12450, Jakarta, Indonesia

Abstract. One of the diseases that is generally characterized by symptoms of an increase in glucose levels in the blood and is one of the body diseases classified as chronic is diabetes. Diabetes suffered by a person from time to time can cause serious damage to other organs such as blood vessels, kidneys, heart and nerves. Machine learning provides various data mining algorithms that can be used to assist medical experts. The accuracy of machine learning algorithms is a measure of the effectiveness of decision support systems. Prediction of diabetes can be seen from the patient's medical record data, therefore the author wants to create a diabetes prediction system independently through a website-based application system. This application system will be combined with data observation, namely the science of data mining using the XGBoost algorithm. The dataset is divided into training data by 80% and testing data by 20%. Before creating the model, various parameter setting scenarios were carried out to evaluate several adjusted parameters, namely colsample_bytree, gamma, learning_rate, max_depth, n_estimators, reg_alpha, reg_lambda, and subsample. After sharing data and tuning parameters, the model results from the XGBoost algorithm have an accuracy of 74.67%, a precision value of 57.40%, a recall value of 65.94% and a specificity value of 78.50%.

Keyword: Data Mining, Diabetes, Knowledge Discovery in Database (KDD), Website, XGBoost Algorithm

Received 28 May 2021 | Revised 25 July 2021 | Accepted 26 July 2021

1 Introduction

One of the diseases that is generally characterized by symptoms of an increase in glucose levels in the blood and is one of the body diseases classified as chronic is diabetes. Diabetes suffered by a person from time to time can cause serious damage to other organs such as blood vessels, kidneys, heart and nerves. Generally, type 2 diabetes is found in adults, type 2 diabetes occurs

*Corresponding author at:Department of Computer Science, Faculty Of Computer Science, University Of Pembangunan Nasional Veteran Jakarta, Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, Jakarta, Indonesia

E-mail address: herlambangdwi.prasetyo@gmail.com (Herlambang Dwi Prasetyo), panduanantoh@gmail.com (Pandu Ananto Hogantara), nurlailika@upnvj.ac.id (Ika Nurlaili Isnainiyah)

ISSN: 2580-6769 | e-ISSN: 2580-829X | DOI: https://doi.org/10.32734/jocai.v5.i2-6290
Journal Homepage: https://jocai.usu.ac.id

when the body doesn't produce enough insulin. In the last 3 decades, it was noted that all countries in the world have shown that type 2 diabetes has increased dramatically every year. Type 1 diabetes is a chronic condition in which the pancreas cannot produce insulin or can only produce small amounts of insulin. Insulin for diabetics is very important for their survival. A global program agreed by many countries to stop the increase in diabetes and obesity by 2025 [1].

Machine learning provides various data mining algorithms that can be used to assist medical experts. The accuracy of machine learning algorithms is a measure of the effectiveness of decision support systems. So that the maximum level of accuracy is the main goal of building a decision support system that is used to predict and diagnose certain diseases. In this system we use the Pima Indian Diabetes dataset which is open source to evaluate the model [2]. Prediction of diabetes can be seen from the patient's medical record data, therefore the author wants to create a diabetes prediction system independently through a website-based application system [3]. This application system will be combined with data observation, namely the science of data mining using the XGBoost algorithm. To classify whether a user has a history of diabetes or not, researchers used a machine learning algorithm, namely Extreme Gradient Boosting (XGBoost). XGBoost is a method that uses the development of the basic gradient tree boosting model to become extreme gradient boosting. Based on a paper that is written by [4], the model used classification and regression tree (CART). XGBoost algorithm has several advantages including good scalability, so that it runs more rapidly than available famous machine learning algorithms, besides it consumes less memory [5]. Previous studies show good results of XGBoost implementation. Research conducted by Abdurrahman et al. that applies XGBoost Algorithm to classify parkinson's diseases has successfully obtained an accuracy of 85.60% without feature selection and 84.40% with feature selection [5]. Then, research conducted by Handayani et al. that applies XGBoost Algorithm to classify breast cancer has successfully obtained an accuracy of 97.83% [6]. Then, research conducted by Ogunleye et al. that applies XGBoost Algorithm to classify Chronic Kidney Disease has successfully obtained an accuracy of 100% [7]. Based on previous studies on various case studies, namely disease classification using the XGBoost algorithm, the results of these studies are that the XGBoost algorithm produces excellent accuracy. However, in these referenced studies the XGboost algorithm has not been implemented into media that is easily accessible by many users, so in this study, we apply the XGBoost algorithm to classify diabetes which is then implemented into a website-based application.

2 Research Methods

In this section the authors describe the XGBoost algorithm and the proposed model used in this study.

2.1 XGBoost Algorithm

XGBoost is more efficient and Scalable. XGBoost can perform multiple functions Such as regression, classification and ranking. XGBoost is a Tree ensemble algorithm composed of multiple sets classification and regression trees (CART). The most important thing behind the success of XGBoost is scalability in various situations. This scalability is due to optimize from the previous algorithm. Given innovation including a novel tree learning algorithm to handle data is sparse. XGBoost proves this success is one of the widely used methods applicable to various situations in machine learning [6]. The objective function usually has the form of two parts (training loss and regularization) [8]:

$$Obj = L(\theta) + \Omega(\theta) \tag{1}$$

where L is the training loss function, and Ω is the regularization term. The training loss measures the performance of the model on the training data. The regular term controls the complexity of the model, and usually controls overfitting. The complexity of each tree is defined as follows:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \tag{2}$$

There is, of course, more than one way to define the complexity, and this particular one works well in practice. The objective function in XGBoost is defined as:

$$obj = \sum_{j=1}^{T} \left[G_j \omega_j + \frac{1}{2}\left(H_j +\right)\omega_j^2 \right] + \gamma T \tag{3}$$

2.2 Proposed Model

Supervised learning algorithms learn the pattern from pre-existing data and try to predict new results based on the previous learning [9]. The method used in this research uses several approaches that include research and development model which incorporates the Knowledge Discovery in model Database (KDD) and Prototype Model. Knowledge Discovery in Database (KDD) used to parse a pattern in data with the help of algorithmic calculations specific. As for

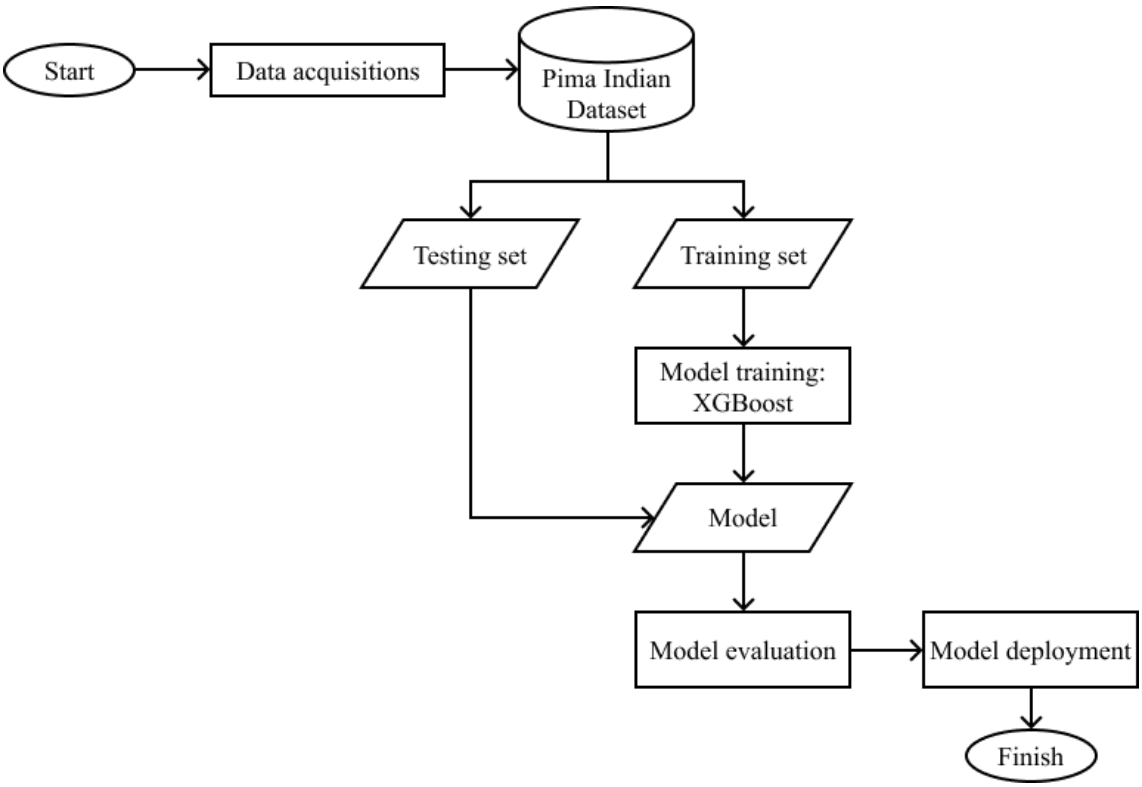

Figure 1 Research Methodology

the prototype model used as a connecting road between the design of work and software systems use of the system work as a whole [3]. Figure 1 indicates the ample architecture of our proposed model. According to our model, patients are required to provide their medical data for successful diagnosis of their diabetes test.

3 Results and Discussion

This section will discuss the steps that need to be done in order to obtain all the results following research methodology as described in Section 2, starting from data acquisitions, data pre-processing, model training, model evaluation, and model deployment.

3.1 Data Acquisitions

Dataset used in this research are secondary dataset obtained from Kaggle [2]. The dataset is called Pima Indian Diabetes Dataset. The total number of data samples in the dataset used is 768 which were divided into two classes, namely diabetes class with a total of 268 data and non-diabetes

class with a total of 500 data. The dataset also consists of eight medical predictors or features and one target variable which is described in Table 1.

Table 1 Diabetes Dataset Attributes

No	Attribute	Attribute Description
1	Pregnancies	Number of times pregnant
2	Glucose	Plasma glucose concentration: a two hours in an oral glucose tolerance test
3	BloodPreasure	Diastolic blood pressure (mm Hg)
4	SkinThickness	Triceps skin fold thickness (mm)
5	Insulin	2-Hour serum insulin (mu U/ml)
6	BMI	Body Mass Index (weight in kg/height in m square)
7	DiabetesPedigreeFunction	Diabetes pedigree function
8	Age	Age (years)
9	Outcome	Class variable (0 or 1)

3.2 Data Pre-Processing

As we can see from Table 1, the features of the dataset are not in the same measurement, i.e. BloodPreasure were measured in mmHg, meanwhile SkinThickness were measured in mm. Therefore, it is important to do feature normalization to ensure all features are in the same range. The feature normalization method used in this research is the min-max normalization, so that all features are ensured to be within the range of {0, 1}. The next pre-processing step is to divide dataset into training set, validation set, and testing set. The dataset is firstly divided into training and testing set with 80:20 ratio. Then, part of the training data that has been generated will be divided again as validation data using stratified k-fold cross validation with k = 3. Because we use k-fold cross validation, the feature normalization step have to be applied separately within each fold to ensure there is no data or information leakage to the model. To simplify this process, we use a method called “pipeline” from scikit-learn library.

3.3 Model Training

Before we implement the XGBoost model, the model needs to be trained first toward data training in order for the model to find recognizable patterns among the data. Every model has its own parameters, so does the XGBoost model. These parameters need to be optimized because each study case or problem requires different parameters to obtain the best accuracy to predict the output of the problem. Therefore, in this step we also perform hyperparameter tuning to find the best parameter for the XGBoost model. The hyperparameters being tested are *colsample_bytree*, *gamma, learning_rate, max_depth, n_estimators, reg_alpha, reg_lambda,* and *subsample*. Table 2 shows the values that are being tested for each hyperparameter.

Table 2 Hyperparameter Values

Hyperparamater	Values
colsample_bytree	[0.4, 0.6, 0.8]
gamma	[1, 5, 10]
learning_rate	[0.01, 0.1, 1]
max_depth	[3, 6, 10]
n_estimators	[100, 150, 200]
reg_alpha	[0.01, 0.1, 10]
reg_lambda	[0.01, 0.1, 10]
subsample	[0.4, 0.6, 0.8]

This hyperparameter tuning process is done toward data training and data validation using a method called grid search, meaning the model will be trained with all possible combinations of hyperparameters, resulting in a model with the best parameter and accuracy to predict the outcome. In total, there is 6561 possible parameter combination and because we use cross validation with $k = 3$, the total number of training done is multiplied by 3, totalling 19683 training performed. The best parameters and score from the model training process is shown in Table 3.

Table 3 Result of Model Training

Hyperparamater	Values	Score
colsample_bytree	0.4	78,67%
gamma	1	
learning_rate	1	
max_depth	3	
n_estimators	200	
reg_alpha	10	
reg_lambda	0.01	
subsample	0.6	

3.4 Model Evaluation

After the model has been trained and the best parameters for the model have been found, we can then proceed to the model evaluation step which uses the testing set as the input. Here, the model tries to predict the outcome from the testing set using the best parameter that has been obtained before. To evaluate the result we use a confusion matrix which is shown in Table 4.

Table 4 Confusion Matrix

Confusion Matrix		Predicted Class	
		Diabetes	**Non-Diabetes**
Actual Class	**Diabetes**	31	23
	Non-Diabetes	16	84

Based on Table 4 above, the prediction results from the model against the testing set show fair results. The explanation of the confusion matrix is as follows:

1. The prediction of non-diabetes class data that are correctly classified as non-diabetes class is 84 data and there are 16 data that are misclassified as diabetes class data.
2. The prediction of diabetes class data that are correctly classified as diabetes class is 31 data and there are 23 data that are misclassified as non-diabetes class data.

Based on the confusion matrix shown in Table 4, we calculate several metrics to evaluate the model. The first metrics is accuracy which is the ratio of correctly predicted data from both classes to the total of data [3] which can be calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{31 + 84}{31 + 84 + 23 + 16} \times 100\% = 74{,}67\%$$

Next, we calculate precision which is the ratio of the total number of positive data that are correctly predicted as positive to the total number of data predicted as positive [10]. Precision can be calculated as follows:

$$Precision = \frac{TP}{TP + FP} = \frac{31}{31 + 23} \times 100\% = 57{,}40\%$$

The third metric that we calculate is recall (sensitivity) which is the ratio of total number of positive data to the total of positive data either correctly predicted as positive class or misclassified as negative class [11]. Recall can be calculated as follows:

$$Recall = \frac{TP}{TP + FN} = \frac{31}{31 + 16} \times 100\% = 65{,}94\%$$

Another metric that we use to evaluate the model is specificity which is the ratio of total number of negative data to the total number of data predicted as negative [7]. Specificity can be calculated as follows:

$$Specificity = \frac{TN}{TN + FP} = \frac{84}{84 + 23} \times 100\% = 78{,}50\%$$

The model performance was further studied with ROC curves which is the plot of sensitivity against "1-specificity". The area under the ROC curve indicated the performance of a particular classifier [12]. Evaluation with the ROC curve can be seen in Figure 2 showing that the classification using the XGBoost algorithm on the Pima Indian Diabetes data shows a diagnostic value of 0.82 so it is included in the good classification.

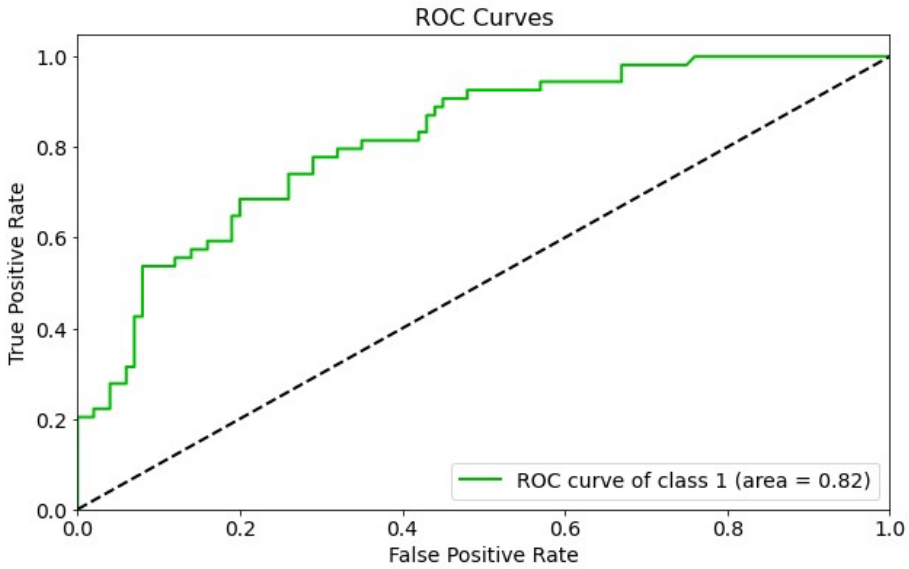

Figure 2 ROC Curve

3.5 Model Deployment

At the deployment model stage, the model deployed on the application system is a model that has an accuracy of 74.67%, a train value of 78.67%. The model with this evaluation value is deployed in a website-based application using Python Flask and Bootstrap, this application uses the Indonesian language as shown in Figure 3 and Figure 4.

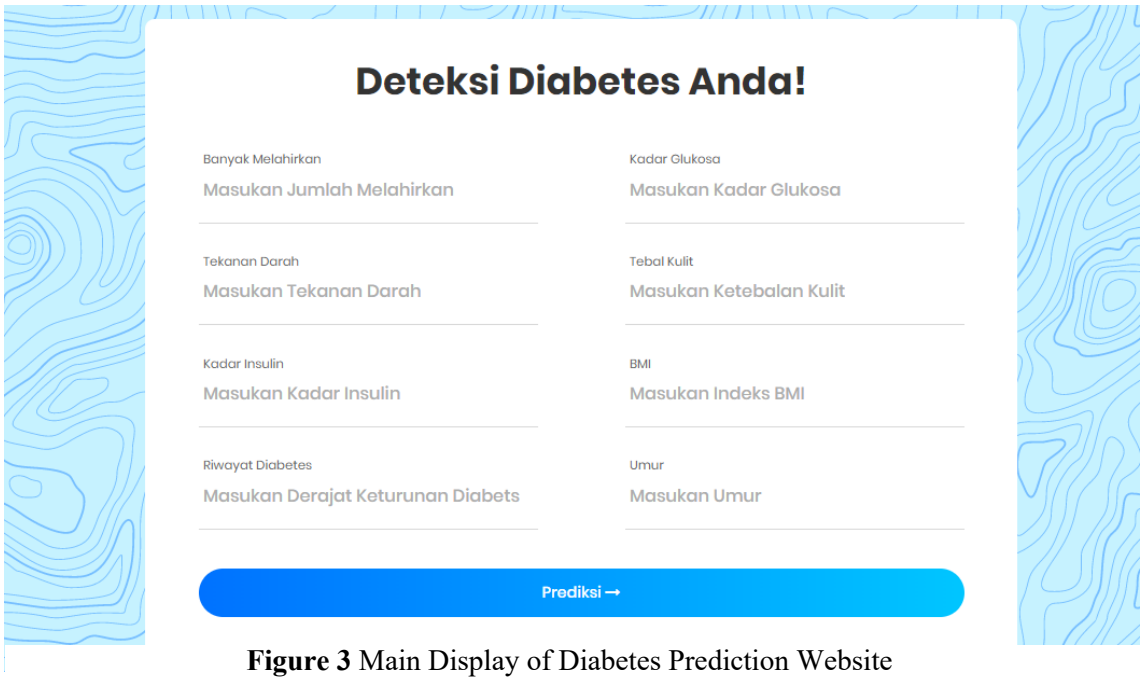

Figure 3 Main Display of Diabetes Prediction Website

On the main page of the application available in Figure 3, users are required to fill in the questions in the application, the questions in this application serve as an indicator whether the user has a history of diabetes or not. The user must fill in each question according to the user's body condition. After the user fill in each question, the user will be directed to the result page which will show the result based on model prediction as shown in Figure 4.

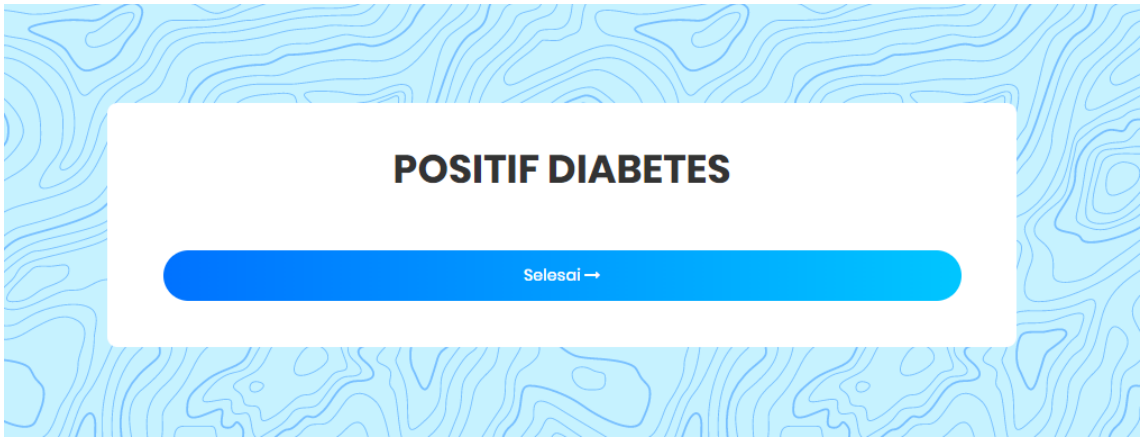

Figure 4 Display of Diabetes Prediction Results

4 Conclusion

In this paper, we built XGBoost to classify diabetes disease. The dataset is divided into training data by 80% and testing data by 20%. Before creating the model, various parameter setting scenarios were carried out to evaluate several adjusted parameters, namely colsample_bytree, gamma, learning_rate, max_depth, n_estimators, reg_alpha, reg_lambda, and subsample. After sharing the data and tuning parameters, the resulting model by applying the XGBoost algorithm has an accuracy of 74.67%, the resulting precision value is 57.40%, the resulting recall value is 65.94%, the resulting specificity value is 78, 50%.

REFERENCES

[1] World Health Organization (WHO), "Diabetes." https://www.who.int/health-topics/diabetes (accessed May 08, 2021).

[2] UCI Machine Learning, "Pima Indians Diabetes Database," 2016. https://www.kaggle.com/uciml/pima-indians-diabetes-database (accessed May 01, 2021).

[3] M. B. Hanif and Khoirudin, "SISTEM APLIKASI PREDIKSI PENYAKIT DIABETES MENGGUNAKAN FITURE," *Pengemb. Rekayasa dan Teknol.*, vol. 16, no. 2, pp. 199–205, 2020.

[4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[5] G. Abdurrahman and M. Sintawati, "Implementation of xgboost for classification of parkinson's disease," *J. Phys. Conf. Ser.*, vol. 1538, no. 1, 2020, doi: 10.1088/1742-6596/1538/1/012024.

[6] A. Handayani, A. Jamal, and A. A. Septiandri, "Evaluasi Tiga Jenis Algoritme Berbasis Pembelajaran Mesin untuk Klasifikasi Jenis Tumor Payudara," vol. 6, no. 4, pp. 394–403, 2017.

[7] A. Ogunleye and Q. G. Wang, "XGBoost Model for Chronic Kidney Disease Diagnosis," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 17, no. 6, pp. 2131–2140, 2020, doi: 10.1109/TCBB.2019.2911071.

[8] L. Zhang and C. Zhan, "Machine Learning in Rock Facies Classification: An Application of XGBoost." [Online]. Available: https://github.com/seg/2016-ml-contest.

[9] S. K. Dey, A. Hossain, and M. M. Rahman, "Implementation of a Web Application to Predict Diabetes Disease: An Approach Using Machine Learning Algorithm," *2018 21st Int. Conf. Comput. Inf. Technol. ICCIT 2018*, pp. 1–5, 2019, doi: 10.1109/ICCITECHN.2018.8631968.

[10] A. Mir and S. N. Dhage, "Diabetes Disease Prediction Using Machine Learning on Big Data of Healthcare," *Proc. - 2018 4th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2018*, pp. 1–6, 2018, doi: 10.1109/ICCUBEA.2018.8697439.

[11] S. S. Dhaliwal, A. Al Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Inf.*, vol. 9, no. 7, 2018, doi: 10.3390/info9070149.

[12] M. Maniruzzaman *et al.*, "Comparative approaches for classification of diabetes mellitus data: Machine learning paradigm," *Comput. Methods Programs Biomed.*, vol. 152, pp. 23–34, 2017, doi: 10.1016/j.cmpb.2017.09.004.