Development of Deep Learning-Based Mobile Application for Predicting Diabetes Mellitus

Christopher G. Estonilo
Graduate Programs
Technological Institute of the Philippines
Manila, Philippines
mcgestonilo@gmail.com

Enrique D. Festijo
Technological Institute of the Philippines
Manila, Philippines
enrique.festijo@tip.edu.ph

2021 4th International Conference of Computer and Informatics Engineering (IC2IE) | 978-1-6654-4288-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/IC2IE53219.2021.9649235

***Abstract*—With the growing demand for intelligent services on mobile devices, deep learning-based mobile applications are expected to progress even further in the coming year. In the advent of this technology, a deep learning model embedded in a mobile application can play a vital role in predicting a certain kind of disease like diabetes mellitus. Many studies have been performed in the past few years to predict diabetes mellitus using various algorithms of machine learning and deep learning. However, these researches are mostly focused on the development of the predicting model. This study aimed for developing a mobile application that is deep learning-based for predicting diabetes mellitus. Using the TensorFlow platform, the Sequential function was used in building the diabetes prediction model. The model was then transformed into a 'tflite' format which was deployed in the development of mobile application using the Android Studio integrated development environment (IDE) to predict if a person has diabetes mellitus. The deep learning model demonstrated considerable accuracy of 93%. Additionally, the application also provides some important instructions for the end-users and facts about diabetes mellitus. The developed deep learning-based mobile application is an important new technology for diabetes mellitus early detection. If the prediction is positive, the lifestyle could change, and a serious complication will be avoided.**

Keywords — diabetes mellitus, deep learning-based mobile application, tensorflow, keras, sequential, tflite

I. INTRODUCTION

Diabetes mellitus, or commonly known as diabetes, is marked by a high level of sugar in the blood or blood glucose. Too much sugar in the bloodstream can lead to severe, life-threatening health issues. This disease is known to be one of the world's fatal diseases. In 2014, the World Health Organization (WHO) stated that 422 million people all over the world had diabetes mellitus. This disease kills 1.6 million people per year, with the majority of those affected living in low- and middle-income countries. Over the past few decades, both the number of cases and the incidence of diabetes mellitus have gradually risen [1].

However, with artificial intelligence (AI), a certain kind of disease like diabetes can be predicted to change the lifestyle and avoid unnecessary complications to the human body. Both machine learning and deep learning are under the category of artificial intelligence which are necessary for the prediction of the diseases. With a growing number of smartphone users, mobile application technology can be applied to enable patients to control their diabetes mellitus better [2]. Furthermore, the application of deep learning to mobile devices has become an interesting topic in academe and industry. With the growing demand for intelligent services on mobile devices, deep learning-based mobile application are expected to progress even further in the coming years [3].

In previous years, many researches have been carried out in the field of diseases prediction. To build a model for diabetes mellitus prediction, various algorithms of machine learning and deep learning9 were used by those researchers. Many of these studies, however, focused only on the development of the model, but not to the extent that these models are practically applied to be used by the potential end-users through the use of mobile application. Therefore, this research focused on predicting the diabetes mellitus using the performance of the deep neural network algorithm embedded in a mobile application. Specifically, it aimed to: a) perform data visualization and pre-processing of the Pima Indian Diabetes Dataset (PIDD) using Python programming language; b) build the diabetes mellitus prediction model using the TensorFlow platform and Keras library; c) evaluate the prediction model's success using a 5-fold cross validation, a classification report to assess the precision, recall, F1 score, and accuracy, a confusion matrix, and the area under receiver operating characteristic (ROC) curve ; and d) transform the developed model in a 'tflite' format for the mobile application development using the Android Studio IDE to predict if a person has diabetes mellitus.

II. DIFFERENT DEEP LEARNING APPROACHES AND SYSTEM DEVELOPMENT STRATEGIES

In deep learning algorithms, there's only few studies attempting to use the performance of deep learning to predict diabetes compared to machine learning algorithms. For instance, the study of [4] performed a thorough review of the most widely used methods, including Deep Neural Networks (DNN) and other algorithms to identify diabetes. The best technique that was found has highest accuracy was by using DNN. In the study of [5], the used of PIDD shows that using a deep learning approach, an effective system for diabetes prediction was developed. In addition, [6] used the PIDD to for data analysis and evaluation to develop a model. The dataset is normalized before being used in the prediction model, which is based on the Deep Belief Neural Network (DBNN). The results showed that the DBNN approach outperforms well-known classifiers such as the Decision Tree

(DT), Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), and Naive Bayes (NB) in terms of overall efficiency.

Another method is to use functional classifiers such as Artificial Neural Network (ANN), NB, DT, and deep learning algorithms to achieve various levels of accuracy [7]. According to this study, deep learning outperforms the other three in terms of diabetes prediction when combined with PIDD. In addition, [8] used four well-known models, namely Convolutional Neural Network (CNN), Conv-LSTM, Traditional LSTM, and CNN-LSTM. The Conv-LSTM-based model identified the diabetes patients with the highest precision in the initial trial, which grouped the dataset into different testing and training sets. Meanwhile, according to [9], ANN is useful in medical science because they reduce classification error and improve accuracy. The results of the study determined that ANN system achieved higher accuracy.

All these researches conducted in the past years used deep learning algorithm with Pima Indian Diabetes Dataset to predict diabetes mellitus. However, these studies end-up with the development of the model but has no mobile application where model are embedded.

There are also researchers whereas different machine learning and deep learning algorithms were investigated, and the best model was implemented in a web or mobile application. The study [10] also used the PIDD and applied the ANN algorithm for prediction, which received the highest accuracy among other algorithms such as K-Nearest Neighbor (KNN), NB, and SVM. They used JavaScript and PHP programming language for the web development and TensorFlow was used to developed the prediction model. During training sessions, the proposed architecture extracted dataset values from a SQL database and trained the ANN model. The users must provide some information as feedback during the prediction process, so that the established web application can predict whether the test result is positive or not.

Also, the study [11] examined several datasets for the prediction such as the type 2 diabetes dataset and other datasets for chronic kidney disease, prehypertension, and hypertension. The research used ensemble learning techniques, SMOTE-Tomek, and iForest method. The MongoDB V4.0.2 was used for the database. The Web API was created using Python V3.6.5, Hug V2.4.1, and PyMongo V3.7.1, which were run on a Python Web Server Gateway Interface (WSGI) HTTP Server called Gunicorn V19.9.0. Also, to build the mobile application, the AngularJS V1.5.3 and Ionic V1.3.3 were used.

Furthermore, the study [12] developed a model for predicting diabetes complications such as Diabetic Eye Disease, Diabetic Kidney Disease, Diabetic Heart Disease, and Diabetic Hyperlipidemia Disease. The prediction model was built using the Decision Tree and evaluated using a ten-fold cross validation. Using a mobile application, the prediction model was tested with general patients.

In addition, the study [13] examined five algorithms which includes DT, ANN, LR and NB, but only RF classifier was applied in the web application since it received the highest accuracy performance. PHP was the programming used for the interface and MySQL for the backend. In the study of [14], researchers applied the C4.5 algorithm using WEKA and the results were applied in the development of the mobile application. The study of [15], a comparison of four machine learning algorithms was carried out, where DT classifier outperforms amongst other algorithms and it was applied for the development of mobile application for diabetes prediction. Lastly, the study of [16], Java using the IntelliJ IDE in Android Studio IDE was used to develop the interface of the system and NB classifier algorithm was utilized for the prediction.

To improve the graphical user interface (GUI) of the mobile application, this research also used the Android Studio IDE and the Kotlin programming language to deployed the developed prediction model using Deep Neural Network.

III. Methods of Data Analysis and Mobile Application Development

The research performed three major methods: the dataset pre-processing, prediction model building and evaluation, and the mobile application development. The conceptual framework of the deep learning-based mobile application is presented in Fig. 1.

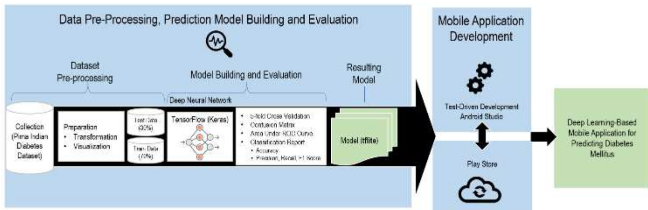

Fig. 1. Deep Learning-based Mobile Application Conceptual Framework

A. *Dataset pre-processing*

1) *Dataset collection*

The researcher used the Pima Indian Diabetes Dataset (PIDD) found in www.kaggle.com, which is one the largest data science online community that provides different kinds of dataset. Pima Indians are American Indian people of Southern Arizona and Northern Mexico. The National Institute of Diabetes, Digestive, and Kidney Disease is the origin of PIDD. In this study, PIDD was selected as a benchmark dataset because it is well-known and widely used for comparing data analysis results in predicting diabetes mellitus.

2) *Dataset preparation*

During the first phase, all attributes of the dataset was already selected and cleaned. The researcher only executed the transformation of the dataset in CSV (comma-separated values) file format and uploaded using Python programming language and TensorFlow platform. After uploading the dataset, the researcher ensured that all data are floating point values and encoded strings to integer. Also, the PIDD shows non-missing values in every attributes. So, cleaning and replacing missing value was not necessary. The dataset was already in a numerical category to perform the necessary equations. The dataset was split into different ratio. Then the typical 30:70 was chosen, which mean 30 percent of dataset take in testing and 70% of dataset take in training.

The PIDD shows eight (8) attributes which is limited to the number of pregnancies, level of sugar or glucose in the blood, blood pressure, thickness of the skin, number of insulins, body mass index (BMI), diabetes pedigree function (DPF), age, and one classification output of whether a person has diabetes mellitus or non-diabetes mellitus. The histogram of the dataset is shown in Fig. 2.

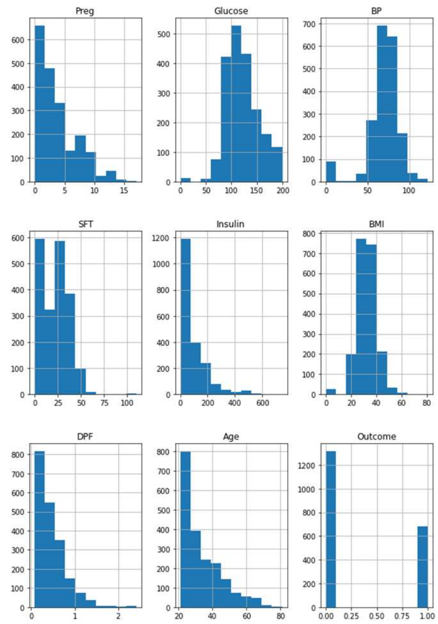

Fig 2. Histogram of the Dataset

The input range of each attribute of the dataset is presented in Table 1.

TABLE I. INPUT RANGE OF THE EIGHT ATTRIBUTES

Attribute	Range
Pregnancy	0 – 17
Glucose	0 – 199
Blood Pressure	0 – 122
Skinfold Thickness	0 – 99
Insulin	0 – 846
Body Mass Index	0 – 67.1
Diabetes Pedigree Function	0.078 – 2.42
Age	21 – 81
Outcome	0/1

B. *Prediction Model Building and Evaluation*

The diabetes prediction model was built using a Sequential function. It is the simplest way to construct a model in Keras, since it helps you to build one layer at a time. Each layer has weights that match the weights of the layer above it. Three layers and an output layer were applied to the model. Dense of 64 neurons was defined as standard layer type. Both nodes in the previous layer bind to the nodes in the current layer in a dense layer. The final layer is the output layer which has two (2) nodes for prediction. Fig. 3 shows the architecture plot of the deep neural network model.

After compiling the prediction model, a 5-fold cross validation was executed to evaluate its performance. The evaluation of the model was also performed through a classification report whereas the precision, recall, F1 score, and accuracy were measured with (1) to (4) In addition, the evaluation performed the confusion matrix and the area under receiver operating characteristic (ROC) curve.

Lastly, the prediction model was saved into TensorFlow lite (**'tflite'**) format that was used in the development of the mobile application.

$$Precision = \frac{TP}{(TP + FN)} \tag{1}$$

$$Recall = \frac{TN}{(TN + FP)} \tag{2}$$

$$F1\ Score = \frac{2TP}{(2TP + FP + FN)} \tag{3}$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{4}$$

Where:
TP is True Positive
TN is True Negative
FP is False Positive
FN is False Negative

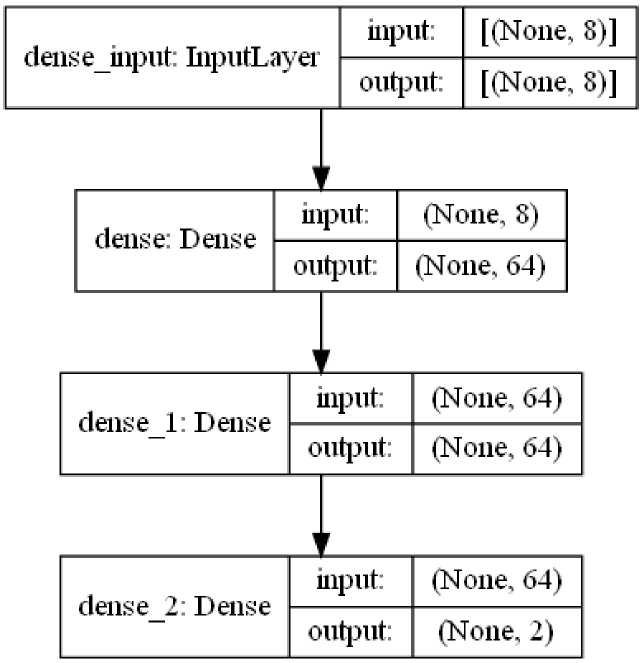

Fig 3. The Deep Neural Network Model Architecture Plot

C. *Mobile Application Development*

The deep learning-based mobile application for diabetes mellitus prediction was developed using the Android Studio, which is Google's official IDE for the smartphone with Android operating system. The resulting model from the second phase employed in the development of the application.

To design the mobile application, Test-Driven Development was performed. This method has a number of benefits, the most important of which is a commitment to improving the software product's performance and programmers' productivity.

The measures in the procedure are as follows: 1) select a user story; 2) create a failed test by writing a test that fulfills a small task from the user story and running it; 3) rewrite all of the production code required to implement the feature; 4) re-run any failed tests, once the code is right, go to the refactoring stage; and 5) when the refactoring stage is complete, the correct production code is produced, and the user is able to choose a new user story. Fig. 4 shows the flowchart of the mobile application development model.

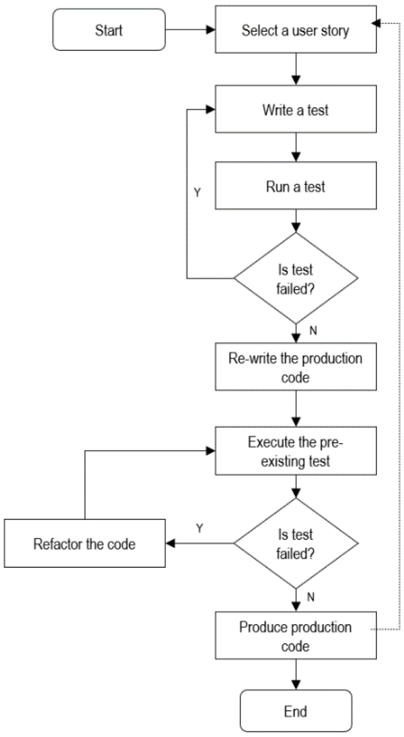

Fig 4. Mobile Application Development Model

The mobile application will be available on the Google Play Store, which comes pre-installed on all Android-certified smartphones.

IV. Results and Discussion

After several tests on each parameter of building the prediction model using Sequential function, a 5-fold cross validation was performed as shown in Fig. 5.

```
Train Index: [   0    1    2 ... 1997 1998 1999]

Test Index: [  23   29   30   32   44   45   49   56   59   63   65   67   69   70
   73   76   78   99  100  109  111  115  120  123  124  128  135  162
  163  168  173  175  185  188  194  196  203  210  211  212  218  220
  231  233  237  239  247  251  254  256  261  266  270  275  281  289
  297  298  300  303  305  306  307  316  322  324  331  342  344  350
  351  352  353  354  361  366  367  368  374  382  383  393  394  411
  414  416  422  427  429  432  433  438  450  453  462  464  471  478
  479  480  482  485  494  495  507  514  519  526  527  529  530  534
  535  538  543  544  552  554  555  570  572  579  581  582  583  584
  585  591  599  602  607  610  611  613  617  618  620  628  630  637
  651  654  670  674  678  679  693  701  712  720  730  743  744  746
  755  757  759  764  771  780  782  785  787  788  792  802  807  808
  824  829  832  834  855  857  862  874  879  886  887  888  889  905
  906  907  909  916  930  937  938  944  949  964  965  973  974  978
  987  990  993 1004 1027 1033 1036 1041 1053 1054 1063 1075 1078 1080
 1083 1084 1089 1091 1100 1103 1105 1107 1116 1118 1120 1134 1137 1164
 1173 1178 1179 1182 1185 1190 1193 1196 1198 1223 1225 1229 1233 1242
```

Fig 5. Training and Testing Set in the First Iteration of 5-Fold CV

The results of prediction model evaluation through precision, recall, and f1 score, as well as its accuracy, are presented in a classification report shown in Table 2. The highest level of accuracy achieved was 93%.

TABLE II. Classification Report

	Precision	Recall	F1 Score	Support
0	0.95	0.94	0.95	269
1	0.88	0.91	0.89	131
Accuracy			0.93	400
Macro Average	0.92	0.92	0.92	400
Weighted Average	0.93	0.93	0.93	400

The result of the confusion matrix is shown in Table 3.

TABLE III. Confusion Matrix

	Predicted Non-Diabetic	Predicted Diabetic
Actual Non-Diabetic (0)	253	16
Actual Diabetic (1)	12	119

Also, the validation accuracy of the model is presented in Fig. 6, and the validation loss of the model is presented in Fig. 7. The figures show good performance of the prediction model.

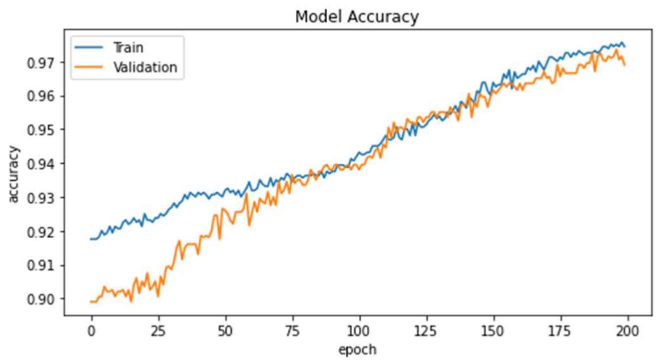

Fig 6. Epoch and Validation Accuracy

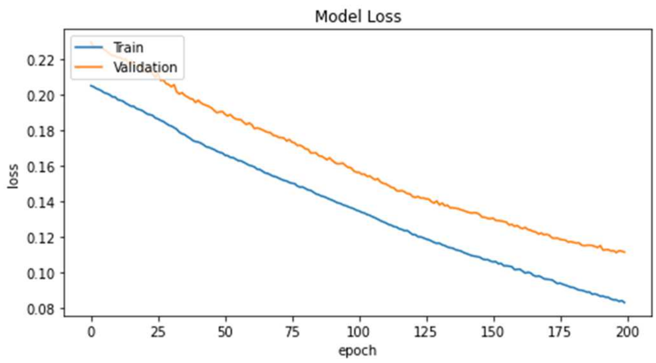

Fig 7. Epoch vs Validation Loss

Also, the result of the Area Under Receiver Operating Characteristic (ROC) Curve is also considered which is shown in Fig. 8. The score is 0.925 (near 1) which means the model distinguishes the classes better.

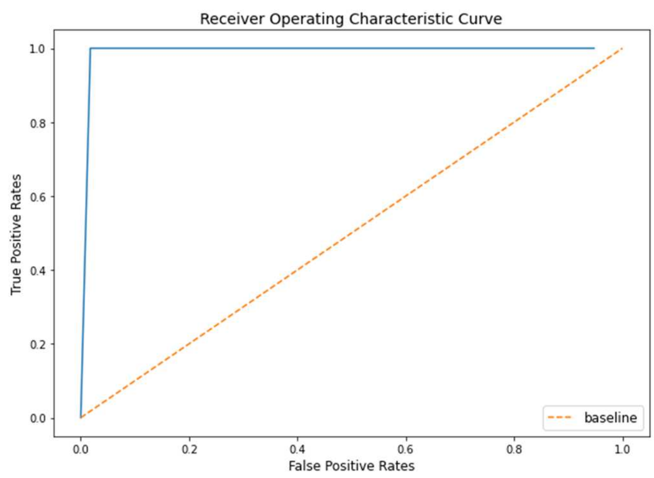

Fig 8. Area Under ROC Curve

The developed prediction model, which was converted to the 'tflite' file format, was deployed in the Android Studio IDE for the development of the deep learning-based mobile application. The GUI of the deep learning-based mobile application for predicting diabetes mellitus is presented in Fig. 9 shows the menu of the mobile application which is consists of four buttons.

The 'Prediction' button displays the required inputs from the user consists of the eight attributes which is shown in Fig. 10. The 'How it Works' button displays the procedures on how to use the application which is shown in Fig. 11. Also, the 'Diabetes Mellitus Facts' button displays the different facts about diabetes mellitus which is shown in Fig. 12.

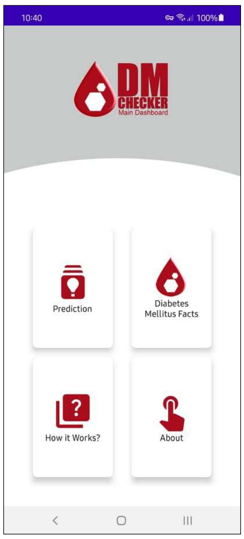

Fig 9. The menu of the Mobile Application

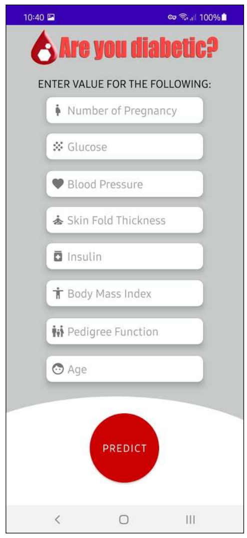

Fig 10. The Input Interface of the Mobile Application

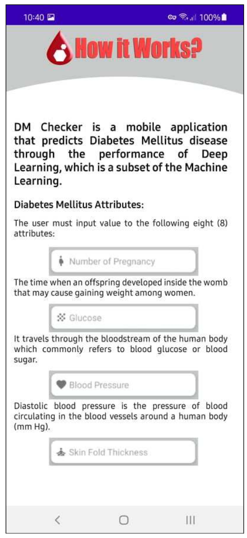

Fig 11. How-it-works Screen of the Mobile Application

Fig 12. Facts About Diabetes Screen of the Mobile Application

V. Conclusion and Future Plans

This research focused on how to predict the diabetes mellitus using the performance of the deep neural network algorithm embedded in a mobile application. The predicting model was created through a Sequential, a function in Keras library of TensorFlow platform. The accuracy of the model was 93% which is considered a very good result for deploying the model in a 'tflite' format into a mobile application for predicting diabetes mellitus. The Android Studio was utilized to developed a user-friendly interface application. However, additional instances in the dataset were recommended to create an excellent performance of the predicting model. This deep learning-based mobile application is an important new method for detecting diabetes mellitus early. If the prognosis is positive, the lifestyle may be altered, and a severe complication may be prevented.

For the future actions, the evaluation of the mobile application will be performed through Unified Theory of Acceptance and Use of Technology (UTAUT) which is derived from [17]. Whereas, the core determinants of behavioral intention are the performance expectancy, effort expectancy and social influence. While facilitating conditions and behavioral intentions are direct determinants of use behavior.

Acknowledgment

The authors would like to thank the Faculty and Staff Development Program of Cavite State University for granting the scholarship. Also, extending the deep gratitude to the professors and classmates in Doctor in Information Technology program at the Technological Institute of the Philippines-Manila.

References

[1] "Diabetes." https://www.who.int/health-topics/diabetes#tab=tab_1 (accessed Apr. 24, 2021).

[2] V. N. Shah and S. K. Garg, "Managing diabetes in the digital age," *Clin. Diabetes Endocrinol.*, vol. 1, no. 1, pp. 1–7, 2015, doi: 10.1186/s40842-015-0016-2.

[3] J. Wang, B. Cao, P. S. Yu, L. Sun, W. Bao, and X. Zhu, "Deep Learning Towards Mobile Applications ∗."

[4] S. Wei, X. Zhao, and C. Miao, "A comprehensive exploration to the machine learning techniques for diabetes identification," *IEEE World Forum Internet Things, WF-IoT 2018 - Proc.*, vol. 2018-Janua, pp. 291–295, 2018, doi: 10.1109/WF-IoT.2018.8355130.

[5] S. Islam Ayon and M. Milon Islam, "Diabetes Prediction: A Deep Learning Approach," *Int. J. Inf. Eng. Electron. Bus.*, vol. 11, no. 2, pp. 21–27, 2019, doi: 10.5815/ijieeb.2019.02.03.

[6] P. Prabhu and S. Selvabharathi, "Deep Belief Neural Network Model for Prediction of Diabetes Mellitus," *2019 3rd Int. Conf. Imaging, Signal Process. Commun. ICISPC 2019*, pp. 138–142, 2019, doi: 10.1109/ICISPC.2019.8935838.

[7] H. Naz and S. Ahuja, "Deep learning approach for diabetes prediction using PIMA Indian dataset," *J. Diabetes Metab. Disord.*, vol. 19, no. 1, pp. 391–403, 2020, doi: 10.1007/s40200-020-00520-5.

[8] M. Rahman, D. Islam, R. J. Mukti, and I. Saha, "A deep learning approach based on convolutional LSTM for detecting diabetes," *Comput. Biol. Chem.*, vol. 88, no. July, p. 107329, 2020, doi: 10.1016/j.compbiolchem.2020.107329.

[9] M. Kowsher, M. Y. Turaba, T. Sajed, and M. M. Mahabubur Rahman, "Prognosis and treatment prediction of type-2 diabetes using deep neural network and machine learning classifiers," *2019 22nd Int. Conf. Comput. Inf. Technol. ICCIT 2019*, no. December, pp. 18–20, 2019, doi: 10.1109/ICCIT48885.2019.9038574.

[10] 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.

[11] N. L. Fitriyani, M. Syafrudin, G. Alfian, and J. Rhee, "Development of Disease Prediction Model Based on Ensemble Learning Approach for Diabetes and Hypertension," *IEEE Access*, vol. 7, pp. 144777–144789, 2019, doi: 10.1109/ACCESS.2019.2945129.

[12] K. Kantawong, S. Tongphet, P. Bhrommalee, N. Rachata, and S. Pravesjit, "The Methodology for Diabetes Complications Prediction Model," *2020 Jt. Int. Conf. Digit. Arts, Media Technol. with ECTI North. Sect. Conf. Electr. Electron. Comput. Telecommun. Eng. ECTI DAMT NCON 2020*, pp. 110–113, 2020, doi: 10.1109/ECTIDAMTNCON48261.2020.9090700.

[13] N. Nai-arun and R. Moungmai, "Comparison of Classifiers for the Risk of Diabetes Prediction," *Procedia - Procedia Comput. Sci.*, vol. 69, pp. 132–142, 2015, doi: 10.1016/j.procs.2015.10.014.

[14] V. K. Priya and A. Monika, "Android Application to Predict and Suggest Measures for Diabetes Using DM Techniques," vol. 4, no. 4, pp. 291–295, 2015.

[15] M. K. Sowjanya, "MobDBTest : A machine learning based system for predicting diabetes risk using mobile devices," pp. 397–402, 2015.

[16] N. S. Khan, M. H. Muaz, A. Kabir, and M. N. Islam, "Diabetes Predicting mHealth Application Using Machine Learning," *2017 IEEE Int. WIE Conf. Electr. Comput. Eng.*, no. December, pp. 237–240, 2017.

[17] Y. Zhang *et al.*, "Factors Influencing Patients' Intentions to Use Diabetes Management Apps Based on an Extended Unified Theory of Acceptance and Use of Technology Model: Web-Based Survey," *J. Med. Internet Res.*, vol. 21, no. 8, p. e15023, Aug. 2019, doi: 10.2196/15023.