Developing an Efficient Predictive Model Based on ML and DL Approaches to Detect Diabetes

Said Gadri
Laboratory of Informatics and its Applications of M'sila LIAM, Department of Computer Science
Faculty of Mathematics and Informatics, University Mohamed Boudiaf of M'sila, M'sila, 28000, Algeria
E-mail: Said.kadri@univ-msila.dz

Keywords: diabetes classification, machine learning, deep learning

Received: January 26, 2020

During the last decade, some important progress in machine learning ML area has been made, especially with the apparition of a new subfield called deep learning DL and CNN networks (Convolutional Neural Networks). This new tendency is used to perform much more sophisticated algorithms allowing high performance in many disciplines such as; pattern recognition, image classification, computer vision, as well as other supervised and unsupervised classification tasks. In this work, we have developed an automatic classifier that permits the classification of a large number of diabetic patients based on some blood characteristics by using ML and DL approaches. Initially, we have proceeded to the classification task using many ML algorithms. Then we proposed a simple DNN model composed of many layers. Finally, we established a comparison between ML and DL algorithms, as well as our model with other existing models. For the programming task, we have used Python, Tensorflow, and Keras which are the most used in the field.

Povzetek: V tem delu smo razvili avtomatski klasifikator, ki omogoča klasifikacijo več bolnikov s sladkorno boleznijo na podlagi nekaterih značilnosti krvi z uporabo ML in DL pristopov.

1 Introduction

Diabetes is one of the chronic diseases which causes an increase in blood sugar. Many serious complications can occur within the patient's body if diabetes remains untreated or unidentified. The best way to identify diabetes results in visiting a diabetes diagnostic center or consulting a specialist doctor. The advance in machine learning ML and deep learning DL can help efficiently in solving this critical problem. The main task to perform using these approaches is to design a predictive model that helps to predict the presence of diabetes in patients with maximum accuracy [1, 2]. In simple words, the machine learning area is based on a set of methods that allow a machine to learn meaningful features (patterns) from data without the interaction of the human. During the last years, many IA tasks like pattern recognition, image classification, computer vision, etc, are based essentially on Machine Learning approaches that gave a high performance. Several algorithms have been developed in this area, including; logistic regression, k-nearest neighbors, naïve Bayes, decision trees, support vector machine, artificial neural networks, etc. Researches on Artificial Neural Networks ANN can be considered as the oldest discipline in ML and AI that dates back to J. McCulloch and W. Pitts in 1943, but these researches were quickly interrupted due to their high requirements in terms of hardware, software, and running time. Many years later, they were revived in parallel with the apparition of a new subfield called deep learning DL. This new approach greatly simplifies the feature engineering process in many vital areas such as; medical imaging. Among the DL methods, CNNs are of special importance. When exploiting local connectivity patterns efficiently which is the case of those used in the ImageNET competition [3]. Many works are trying to apply CNNs on image analysis [4, 5] using a variety of methods like the rectified linear unit [6] and deep residual learning [7].

The present paper is organized as follows: section 1 is a short introduction presenting the area of our work and its advantages and benefits. Section 2 presents the background of deep learning as a new sub-field of ML and AI. Section 3 is a detailed overview of the related works in the same area. In the fourth section, we described our proposed model. The fifth section presents the experimental part we have done to validate our proposed model. In section 6, we illustrated the obtained results when applying the new model. In section 7, we discussed the results obtained in the previous section. In section 8, we established a quick comparison between the ML algorithms and the proposed ANN model. In section 9, we established a large comparison to the existing models. In the last section, we summarized the realized work and suggested some perspectives for future researches.

2 Deep learning background

Deep Learning DL becomes one of the most popular subfields of IA, and ML, especially in speech recognition, computer vision, and some other interesting topics [8]. Its success is motivated by three factors: the increased amount of available data especially with the massive connection on the internet, the improvement and the lowest cost of hardware and software [6, 7, 9, 8, 10], the increased processing abilities (e.g., GPU units) [11]. DL is based essentially on the use of ANNs with at least two layers or many hidden layers. The convolutional neural network CNN is a specialized feedforward neural network that was developed to process multidimensional data, such as images. Its origins refer to the neo-cognition proposed by Fukushima in 1980 [12]. The first model of CNN was proposed by LeCUN et al., [13] in 1998 for the purpose of character recognition. Furthermore, many other alternative ANN architectures have been developed later, including; recurrent neural networks RNNs, autoencoders, and stochastic networks [14, 15, 16]. Deep learning also provided an efficient solution to the problem of input data representation which is considered as the most critical phase in ML especially when the problem is enough complex such as image or speech recognition [17] where it is difficult to define the best features to process on. In this way, DNNs can learn high-level feature representations of inputs through their multiple hidden layers. The first DNNs had appeared in the 1960s but were abandoned, after that for a long time in favor of the ML approach, due to its high requirements in terms of difficulties in training and inadequate performance [18]. In 1986 RumelHart et al., [19] proposed the back-propagation method to update efficiently neural network weights using the gradient of the loss function through multiple layers. Despite the promising results given by DNNs in the late 1980s [12] and 1990s [20], they were abandoned due to many problems. In 2006, researches in DL were revived especially when researchers have developed new methods for sensibility initializing DNN weights using a supervised layer-wise pretraining procedure [21, 22, 23]. It is the case of deep belief networks DBNs which worked very well through supervised learning and gave high accuracy in image and speech tasks in 2009 and 2012 [11]. In 2012, Krizhevsky et al., [18] proposed a deep CNN for the large-scale visual recognition challenge (LSVRC) [24] reducing significantly the error rate. This CNN has been implemented on multiple graphics processing unit GPUs for the first time. This new technique has since become a common practice in DL work until our days because it allows the training of large datasets and increases significantly the speed of processing. Furthermore, the use of a new activation function RELU (Rectified Linear Unit) was the alternative solution of the gradient that allowed faster training of data. The dropout technique is also used as a regularization method to decrease overfitting in large networks with many layers. All these interesting improvements in DL let the leading technology companies increase the research efforts, producing many other advances in the field. Many DL frameworks using tensor computation [14, 15, 16, 17] and GPU compatibility libraries [18] have been developed and made available to researchers through open-source software [23] and cloud services [24, 25]. On the other hand, many companies have met the challenges of big data when exploring large amounts of data to predict value decisions [26]. The notion of big data refers to data that exceeds the capability of standard data storage and data processing systems [27]. This large volume of data requires also high-performance hardware and very efficient analysis tools [28]. Some other ML challenges appeared with big data including; high dimensionality, distributed infrastructures, real-time requirements. Najafabadi et al [26] discuss the use of DL to solve big data challenges and proved that the capacity of DNNs to extract meaningful features from large datasets is extremely important. The automatic extraction of features from heterogeneous data, e.g., image text, audio is very useful and difficult. But this task becomes easy with the use of DL methods. Other tasks including; semantic-based IRS like semantic indexing and hashing [27, 28] also became possible with these high-level features, furthermore, DL is also used to tag incoming data streams allowing to classify fast-moving data [26]. In general, high-level DNNs are suitable for learning from large data issued of big data sources. In conclusion, we can say that DL is currently growing faster than before.

3 Related work

Diabetes detection is one of the most important topics in the health care field. Many studies have been done in this field and have as a common goal the improvement of the accuracy as well as the speed of detection. Among these studies, we can note the following:

Kayaer and Yeldirim [29] proposed a general regression neural network to detect diabetes and achieved an accuracy of 80.21%. Goncalves et al. [30] developed a system based on a new hierarchical neuro-fuzzy binary space repartitioning method BSP, they achieved an accuracy of 80.08% for the training set and 78.26% for the test set. Polat and Gunes [31] used a generalized discriminant analysis method GDA and the least square support vector machine LS-SVM, they reported an acceptable accuracy of 79.16%. Kim et al. [32] applied SVM with CPONs to obtain the best classification accuracy, they used the following datasets: WISCONSIN Breast Cancer Dataset, Pima Indians Diabetes, BUPA Liver Disorder Dataset, Ionosphere Dataset, and the MNIST Digit dataset from the UCI machine learning repository. They obtained an accuracy of 83.11% for the Pima Indians Diabetes PID dataset. Caliskan et al. [33] developed a simple training strategy for deep neural network classifiers using L-BFGS algorithm on many datasets including the PID dataset, they achieved an accuracy of 77.09%. Dwivedi [34] used machine learning algorithms to predict diabetes mellitus. He used many algorithms including; SVM, ANN, LR, DT, KNN, NB. The highest obtained accuracy was 82% using the Naïve Bayes algorithm. Vijayashree et al. [35] proposed a new system based on the use of recursive feature elimination combined with an analysis component to predict diabetes. Two architectures have been used here: a deep neural network which gave an accuracy of 82.67% and an artificial neural network which gave an accuracy of 78.62%. Ashiquzzaman et al. [36] proposed a prediction framework for diabetes mellitus using a deep learning approach with diminution of the overfitting. They used the

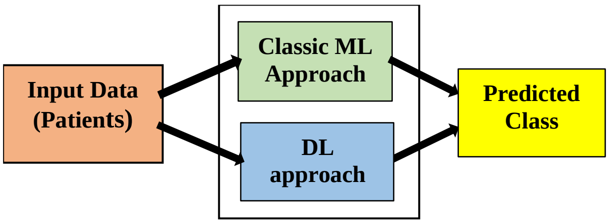

Figure 1: Diabetes Classification Process.

Pima Indians Diabetes dataset and obtained an accuracy of 88.41%. Cheruki et al. [37] proposed an expert system based on PCA and an adaptive neuro-fuzzy inference system ANFIS on the PID dataset for diabetes diagnosis and obtained an accuracy of 89.47%. Kannadasan et al. [38] proposed a new predictive model for type 2 diabetes classification using stacked autoencoders cascaded with a softmax classifier and achieved an accuracy of 86.26%. Kowsher et al. [39] developed a deep ANN and ML classifier using many performance measures such as; accuracy and precision to specify the best DNN model and obtained an accuracy of 95.14% on the training dataset.

Many other studies have been done, but the classification accuracy was within 59.5% and 77.7%.

4 The proposed approach

In the present work, we have developed an automatic binary classifier that permits us to identify people affected by diabetes based on some characteristics (features). Thus, it is a binary classification into two (02) given classes (1: the patient has diabetes, 0: the patient does not have diabetes). All of the input variables that describe each patient are numerical. We note also that to build the best predictive model, we have used two approaches: the classic ML approach and DL approach. Initially, we have proceeded with the classification task using many ML algorithms including; LR, LDA, KNN, CART, NB, and SVM. Then we proposed a DNN model composed of many simple layers: one input layer (180 neurons), eight hidden layers (150, 120, 80, 50, 30, 18, 8, 4 neurons), and finally an output layer (01 neuron). In the last stage, we established a first comparison between the different algorithms and a second comparison between our model and the existing models. As programming tools, we have used Python, Tensorflow, and Keras which are the most used in this field. Fig 1 summarizes the classification task based on the two approaches, while Fig 2 presents a detailed architecture of the proposed DNN model to improve the performance of the classification task.

5 Experimental work

5.1 Used dataset

In our experiments on the scikit-learn toolkit and DNNs, we used the Pima Indians onset of diabetes dataset which is a standard machine learning dataset from the UCI Machine Learning repository. It describes patient medical record data for Pima Indians and whether they had an onset of diabetes within five years. Thus, our problem is a binary classification problem (onset of diabetes as 1 or not as 0). All of the input variables that describe each patient are numerical. This makes it easy to use directly with neural networks that expect numerical input and output values, and ideal to use the neural network in Python, Tensorflow, and Keras. The dataset includes data from 768 women with 8 characteristics, including:

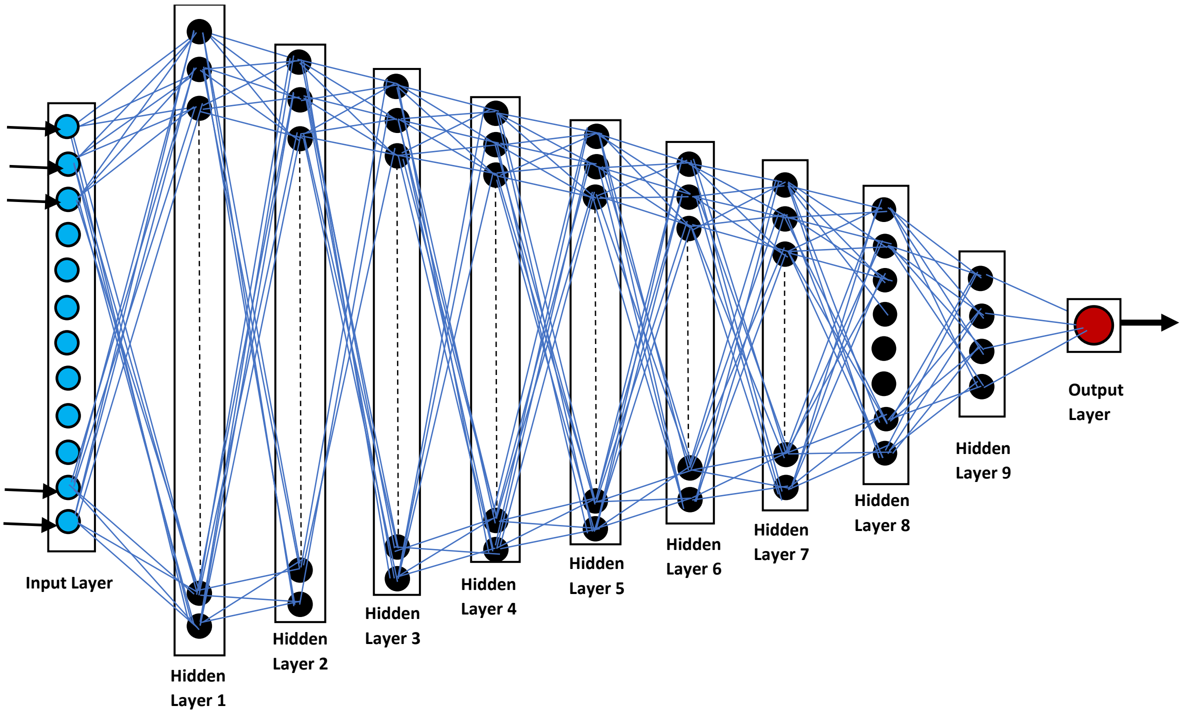

Figure 2: Architecture of the proposed CNN model.

1. Number of times pregnant
2. Plasma glucose concentration within 2hours oral glucose tolerance test.
3. Diastolic blood pressure (mm/Hg)
4. Triceps skinfold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)
9. The last column of the dataset indicates if the person has been diagnosed with diabetes (1) or not (0).

5.2 Programming tools

Python: Python is currently one of the most popular languages for scientific applications. It has a high-level interactive nature and a rich collection of scientific libraries which lets it a good choice for algorithmic development and exploratory data analysis. It is increasingly used in academic establishments and also in industry. It contains a famous module called scikit-learn tool integrating a large number of ML algorithms for supervised and unsupervised problems such as; decision trees, logistic regression, Naïve Bayes, KNN, ANN, etc. this package of algorithms allows to simplify ML to non-specialists working on a general-purpose.

Tensorflow: TensorFlow is a multipurpose open-source library for numerical computation using data flow graphs. It offers APIs for beginners and experts to develop for desktop, mobile, web, and cloud. TensorFlow can be used from many programming languages such as; Python, C++, Java, R,…, and Runs on a variety of platforms including; Unix, Windows, iOS, Android. We note also that Tensorflow can be run on single machines (CPU, GPU, TPU) or distributed machines of many 100s of GPU cards

Keras: Keras is the official high-level API of TensorFlow which is characterized by many important characteristics: Minimalist, highly modular neural networks library written in Python, Capable of running on top of either TensorFlow or Theano, Large adoption in the industry and research community, Easy productization of models, Supports both convolutional networks and recurrent networks and combinations of the two, Supports arbitrary connectivity schemes (including multi-input and multi-output training), Runs seamlessly on CPU and GPU.

5.3 Evaluation

To validate the different ML algorithms, and obtain the best model, we have used the cross-validation method consisting of splitting our dataset into 10 parts, train on 9, and test on 1 and repeat for all combinations of train/test splits. For the DNN model, we have used two parameters which are: loss value and accuracy metric.

1.Accuracy metric: This is a ratio of the number of correctly predicted instances divided by the total number of instances in the dataset multiplied by 100 to give a percentage (e.g., 90% accurate).

2.Loss value: used to optimize an ML algorithm or DL model. It must be calculated on training and validation datasets. Its simple interpretation is based on how well the ML algorithm or the DL model is doing in these two datasets. It gives the sum of errors made for each example in the training or validation set.

3.Precision: It is the number of real correct positive results divided by the total number of positive results predicted by the classifier.

4.Recall: It is the number of real correct positive results divided by the number of all relevant samples in the dataset (all samples that should have been identified as positive).

5.F1-Score: is the Harmonic Mean between precision and recall. The range for F1-Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is.

6.Confusion Matrix: The Confusion matrix is one of the easiest metrics used for finding the correctness and accuracy of the model. It is used for classification problem where the output can be of two or more types of classes and give the correctness for each class.

6 Illustration of obtained results

To build the best predictive model, we performed two tasks:

1. Applying many ML algorithms, including; Logistic Regression LR, Linear Discriminant Analysis LDA, K-nearest Neighbors KNN, Decision Tree (CART variant), Gaussian Naïve Bayes NB, Support Vector Machine SVM. For this purpose, we used the scikit-learn library of python containing the most known learning algorithms.

2. Designing a DNN (Deep Neural Network) model according to the following process:

- We proposed a model composed of ten (10) full connected layers described as follows:
- layer 1(180 neurons and expects 8 input variables), layer 2 (150 neurons), layer 3 (120 neurons), layer 4 (80 neurons), layer 5 (50 neurons), layer 6 (30 neurons), layer 7 (18 neurons), layer 8 (8 neurons), layer 9 (04 neurons), and finally, layer 10 or the output layer has 1 neuron to predict the class (onset of diabetes or not).
- The ten (10) fully connected layers are defined using the Dense class of Keras which permits to specify the number of neurons in the layer as the first argument, the initialization method as the second argument, and the activation function using the activation argument.
- We initialize the network weights to a small random number generated from a uniform distribution (‘Uniform‘), in this case between 0 and 0.05 which is the default uniform weight initialization in Keras. Or ‘normal’ for small random numbers generated from a Gaussian distribution.
- We use the rectifier (‘Relu’) activation function on most layers and the sigmoid function in the output layer.

- We use a sigmoid function on the output layer to ensure our network output is between 0 and 1 and easy to map to either a probability of class 1 or 0.
- We compile the model using the efficient numerical libraries of Keras under the covers (the so-called backend) such as TensorFlow. The backend automatically chooses the best way to represent the network for training and making predictions to run on your hardware (we have used CPU in our application).
- When compiling, we must specify some additional properties required when training the network. We note that training a network means finding the best set of weights to make predictions for this problem.
- When training the model, we must specify the loss function to evaluate a set of weights, the optimizer used to search through different weights of network, and any optional metrics we would like to collect and report during training. Since our problem is a binary classification, we have used a logarithmic loss, which is defined in Keras as “binary_crossentropy“.
- We will also use the efficient gradient descent algorithm “adam” because it is an efficient default.
- Finally, since it is a classification problem, we report the classification accuracy as the performance metric.
- Execute the model on some data.
- We can train or fit our model on our loaded data by calling the fit() function on the model, the training process will run for a fixed number of iterations through the dataset called epochs, which we must specify using the n-epochs argument. We can also set the number of instances that are evaluated before a weight update in the network is performed, called the batch size, and set using the batch_size argument. For our case, we fixed the following values: Nb-iter=350, batch-size=10. These are chosen experimentally by trial and reducing the error.
- We trained our DNN on the entire dataset (training set) and evaluated its performance on a part of the same dataset (test set) using the evaluate () function. This will generate a prediction for each input and output pair and collect scores, including the average loss and any metrics you have configured, such as accuracy.

The following tables summarize the obtained results on ML algorithms and the CNN model.

7 Discussion

Table 1 summarizes the obtained results when applying the different ML algorithms including; LR, LDA, KNN, DT (CART), NB, SVM. We observe that LR, LDA, KNN, CART, and NB give a high classification accuracy (> 70%), while SVM gives a relatively low value of accuracy (65%) compared to the previous algorithms.

Table 2 gives the performance of KNN with Minkowski similarity by class. The performance is given in terms of three measures: precision, recall, and F1-score for each class. Class 0 (no diabetes appears on a patient) has high values for the three measures, but class 1 (onset diabetes) not. This can be interpreted by the fact that we meet a con-flict in detecting a diabetic patient. Finally, table 2 calculates the micro-average, the macro-average and the weighted average of the different performance measures. These new metrics represent a kind of harmonic mean that summarizes the different values.

Table 3 presents the obtained results when applying the proposed DNN model on the training set and the test set. Two performance measures are considered in this case, the loss value which calculates the sum of errors after training the model, and the accuracy value which gives the rate of correctness. It is clear, that the loss value is very low against the accuracy which is very high and depends on the size of the used set. It is the reason for which the accuracy of the training set is higher than the accuracy of the test set.

In the same way, Figure 3 shows the evaluation of training loss and validation loss over time and in terms of the number of epochs. It begins very high for the training set and ends very low because of the large number of samples, but its variation for the validation set is not very quick and appears relatively stable.

Similarly, Figure 4 plots the evolution of training accuracy and validation accuracy in terms of the number of epochs. Contrary to the loss value, the accuracy starts very low and ends very high. This property is clearer with the training set because of its large size.

Algorithm	Accuracy
LR	77,09%
LDA	77,61%
KNN (k=5, metric = minkowski)	71,36%
CART	70,49%
NB	75,87%
SVM	65,27%

Table 1: The accuracy average after applying different ML algorithms.

Class	Precision	Recall	F1-score
0	0.75	0.84	0.79
1	0.62	0.49	0.55
Micro Avg	0.71	0.71	0.71
Macro Avg	0.69	0.66	0.67
Weighted Avg	0.70	0.71	0.70

Table 2: Performance report for KNN (k=5, metric = Minkowski).

DNN (180, 150, 120, 80, 50, 30, 18, 8, 4, 1)	
Training set	Loss: 0.0763 ; Accuracy: 97,39%
Test set	Loss: 0.26 ; Accuracy: 94.27%

Table 3: Loss and accuracy values obtained when applying the proposed model.

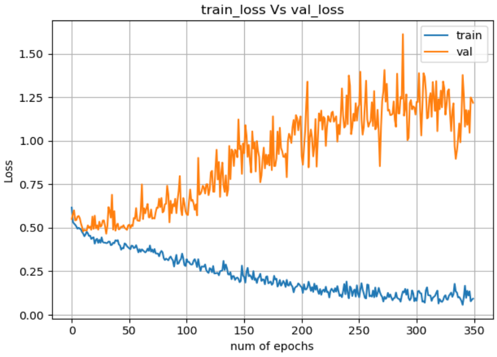

Figure 5: Training loss Vs Validation loss of the DNN model.

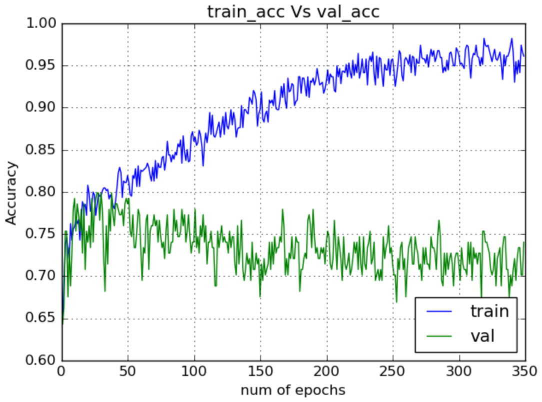

Figure 6: Training Accuracy Vs Validation Accuracy of the DNN model.

We can see also in Figures 3 and 4 representing loss value and accuracy value successively that curves are not continuous but there exist some pics, that because the two values of loss and accuracy don't progress continuously over epochs, but increase and decrease until stabilizing on a specific value.

8 Comparison between ML, and the DNN approaches

In this section, we try to establish a comparison between different algorithms, ML algorithms, and the DNN model. The result of this comparison is illustrated in Table 4.

According to Table 4 and Figure 5, The comparison favors the DNN model over ML algorithms. And we can confirm that the DL approach gives always the best values of accuracy if we choose a suitable architecture.

9 Comparison with other works

In this section, we will establish a comparative study between our proposed model and other models of the state-of-the-art, as illustrated in Table 5 and shown in Figure 6.

10. Conclusion and future suggestions

In the last years, object recognition is based essentially on the ML approach that gives high performance. Many years later, some important progress on the ML area has been

Algorithm	Accuracy rate
LR	77,09%
LDA	77,61%
KNN (k=5, metric = minkowski)	71,36%
CART	70,49%
NB	75,87%
SVM	65,27%
CNN model	Train-Acc: 97,39% Test-Acc: 94,27%

Table 4: Comparison between ML approach and DL Approach.

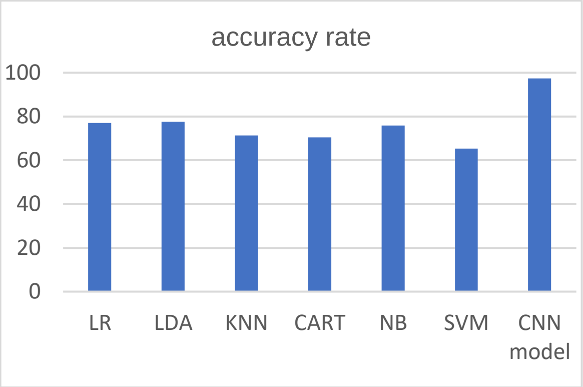

Figure 3: Comparison between different Algorithms.

Author	Year	Accuracy
Kayaer and Yeldirim [29]	2003	80.21%
Goncalves et al. [30]	2006	80.08%
Polat and Gunes [31]	2007	79.16%
Kim et al. [32]	2015	83.11%
Dwivedi [34]	2017	82%
Vijayashree et al. [35]	2017	82.67%
Ashiquzzaman et al. [36]	2017	88.41%
Cheruki et al. [37]	2017	89.47%
Caliskan et al. [33]	2018	77.09%
Kannadasan et al. [38]	2018	86.26%
Kowsher et al.[39]	2020	95.14%
My proposed model	2021	97.39%

Table 5: Comparison with other existing models.

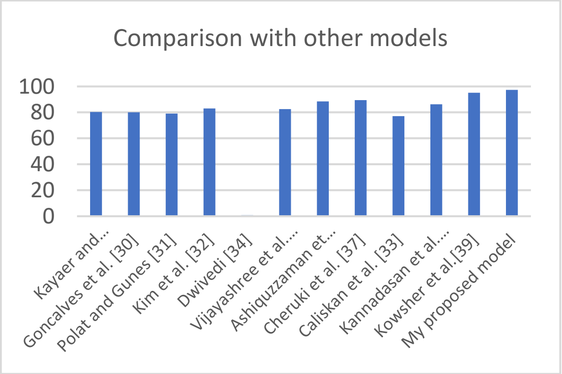

Figure 4: Comparison with other existing models.

made especially with the apparition of a new subfield called deep learning. It is mainly based on the use of many neural networks of simple interconnected units to extract meaningful patterns from a large amount of data to solve some complex problem such as; medical image classification, fraud detection, character recognition, etc. currently, we can use larger datasets to learn powerful models, and better techniques to avoid overfitting and underfitting. Until our days, the obtained results in this area of research are very surprising in different domains. We talk about very high values of accuracy which often exceed the threshold of 90%. For example, the accuracy rate on the digits set is over 97%. In the present paper, we have performed a task of classification on the Pima Indian dataset PID. We have used in the first stage many ML algorithms including; LR, LDA, KNN, DT (CART), NB, SVM. We obtained a good result of accuracy especially, on LR, LDA, KNN. In the second stage, we have built a DNN model to perform the same task of classification. The achieved performance is very surprising. We concluded our work by establishing a large comparison between different algorithms. The result of this comparison was in favor of the DL approach through the DNN model we have built. Furthermore, we have conducted a large comparison of our model with the existing models, our proposed model gives the best performance (a high value of accuracy). As a perspective of this promising work, we propose to improve these results by improving the architecture of the DNN model by studying other ANN architectures, changing some model parameters such as; the number of layers, the number of neurons in each layer, the number of training epochs and the size of data batches. Another suggestion that seems important, is to use other types of DNN or combining CNN with recurrent neural networks RNN.

References

[1] Ali A., Fakhreldeen A. S. (2021). A comparative analysis of machine learning algorithms to build a predictive model for detecting diabetes complications. *Informatica journal*, vol 45(1), pp. 117-125, https://doi.org/10.31449/inf.v45i1.3111

[2] Gjoreski M. (2021). A method for combining classical and deep machine learning for mobile health and behavior monitoring. *Informatica journal*, vol 45(1), pp. 169-170, https://doi.org/10.31449/inf.v45i1.3482

[3] Lee H., Grosse R., Ranganath R., and Ng A.Y(2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *In Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM. https://doi.org/10.1145/1553374.1553453

[4] Pinto N., Doukhan D., DiCarlo J.J. , and Cox D.D. (2009) A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS computational biology*, 5(11):e1000579. https://doi.org/10.1371/journal.pcbi.1000579

[5] Turaga S.C., Murray J.F., Jain V. Roth F., Helmstaedter M., Briggman K., Denk W., and Seung H.S. (2010). Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, 22(2):511–538. https://doi.org/10.1162/neco.2009.10-08-881

[6] Abadi B., Agarwal M., Barham A, P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X. (2018) *TensorFlow: large-scale machine learning on heterogeneous systems 2015*. http://tensorflow.org/. Accessed 1 Nov 2018. https://doi.org/10.1145/3190508.3190551

Theano Development Team. (2016). Theano: *a Python framework for fast computation of mathematical expressions*. arXiv e-prints arXiv:1605.02688.https://doi.org/10.1016/b978-2-294-71225-8.00029-9

Sivaslioglu S, Ozgur F., Şahinbaş K. (2021). A generative model based adversarial security of deep learning and linear classifier models. *Informatica journal*, vol 45(1), pp. 33-64, https://doi.org/10.31449/inf.v45i1.3234

[7] Chollet F, et al (2015). *Keras*. Received from https://keras.io. Accessed 1 Nov 2018.

[8] Chetlur S, Woolley C, Vandermersch P, Cohen J, Tran J, Catanzaro B, Shelhamer, E. cudnn (2014). Efficient primitives for deep learning. arXiv:1410.0759 [cs.NE]

Krizhevsky A, Sutskever I, Hinton GE (2012). Imagenet classification with deep convolutional neural networks. *Neural information processing systems*. p. 25. https://doi.org/10.1145/3065386

[9] Fukushima K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern*. 1980;36(4):193–202. https://doi.org/10.1007/bf00344251

[10] LeCun Y, Bottou L, Bengio Y, Haffner P (1998). Gradient-based learning applied to document recognition. *Proc IEEE*. 86(11):2278–324. https://doi.org/10.1109/5.726791

[11] Witten IH, Frank E, Hall MA, Pal CJ (2016). *Data mining*, Fourth Edition: Practical machine learning tools and techniques. 4th ed. San Francisco: Morgan Kaufmann Publishers Inc. https://doi.org/10.1186/1475-925x-5-51

[12] Goodfellow I, Bengio Y, Courville A (2016). *Deep learning. Cambridge*: The MIT Press; 2016. ISBN 9780262035613

[13] Minar MR, Naher J. (2018). Recent advances in deep learning: an overview. arXiv:1807.08169 [cs.LG] http://dx.doi.org/10.13140/RG.2.2.24831.10403

[14] LeCun Y, Bengio Y, Hinton G (2015). *Deep learning. Nature*;521:436. https://doi.org/10.1038/nature14539

[15] Schmidhuber J. (2015). Deep learning in neural networks: an overview. *Neural Net*; 61:85–117. https://doi.org/10.1016/j.neunet.2014.09.003

[16] Rumelhart DE, Hinton GE, Williams RJ. (1986). Learning representations by back-propagating errors. *Nature*; pp. 323-33. https://doi.org/10.1038/323533a0

[17] Le Cun,, Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L.D. (1990) :Handwritten digit recognition with a back-propagation network. *In Advances in neural information processing systems*. https://doi.org/10.1109/ijcnn.1990.137801

[18] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Comput*;1(4):541–51. https://doi.org/10.1162/neco.1989.1.4.541

[19] Hinton GE, Osindero S, Teh Y-W. (2006). A fast learning algorithm for deep belief nets. *Neural Comput*;18(7):1527–54. https://doi.org/10.1162/neco.2006.18.7.1527.

[20] Bengio Y, Lamblin P, Popovici D, Larochelle H. (2007) Greedy layer-wise training of deep networks. *In: Proceedings of the 19th international conference on neural information processing systems. NIPS'06*. MIT Press, Cambridge, MA, USA. p. 153–60.

[21] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L. (2015). ImageNet large scale visual recognition challenge. *Int J Comput Vision (IJCV)*;115(3):211–52. https://doi.org/10.1007/s11263-015-0816-y. https://doi.org/10.1007/s11263-015-0816-y

[22] Kumar M. (2016). An incorporation of artificial intelligence capabilities in cloud computing. *Int J Eng Comput Sci*. https://doi.org/10.18535/ijecs/v5i11.63.

[23] Saiyeda A, Mir MA. (2017). Cloud computing for deep learning analytics: a survey of current trends and challenges. *Int J Adv Res Comput Sci*;8(2):68–72. https://doi.org/10.26483/ijarcs.v8i2.2931
Dumbill E. (2012). What is big data?: an introduction to the big data landscape. http://radar.oreilly.com/2012/01/what-is-big-data.html

[24] Hongbiao Ni. (2020). Face recognition based on deep learning under the background of big data. *Informatica journal*, vol 44(4), pp. 491-495, https://doi.org/10.31449/inf.v44i4.3390

[25] Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E. (2015). Deep learning applications and challenges in big data analytics. *J Big Data*; 2(1):1. https://doi.org/10.1186/s40537-014-0007-7.

[26] Hinton G, Salakhutdinov R. (2011). Discovering binary codes for documents by learning deep generative models. *Top Cogn Sci*.;3(1):74–91. https://doi.org/10.1111/j.1756-8765.2010.01109.x

[27] Salakhutdinov R, Hinton G. (2009). Semantic hashing. *Int.J.Approx Reason*;50(7):969–78. doi:10.1016/j.ijar.2008.11.006

[28] Kayaer K, Yıldırım T. (2003). Medical diagnosis on pima indian diabetes using general regression neural networks. Proceedings of the International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP):181–184 Istanbul, Turkey, June 26–29. https://doi.org/10.1007/3-540-44989-2_127

[29] Goncalves L. B and Bernardes M.M. (2006). "Inverted Hierarchical Neuro-Fuzzy BSP System: A Novel Neuro-Fuzzy Model for Pattern Classification and Rule Extraction in Databases," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 36, no. 2, pp. 236-248, Mar. 2006. https://doi.org/10.1109/tsmcc.2004.843220

[30] Polat K, Gunes S. (2007). An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease. *Digit Signal Process*, 17(4):702–710 https://doi.org/10.1016/j.dsp.2006.09.005.

[31] Kim S, Yu Z, Kil R, Lee M. (2015). Deep learning of support vector machines with class probability output networks. *Neural Netw*ork. 64, 19–28. https://doi.org/10.1016/j.neunet.2014.09.007

[32] Caliskan A, Yuksel ME, Badem H, Basturk A (2018). Performance improvement of deep neural network classifiers by a simple training strategy. *Eng Appl Artif Intell*; 67:14–23. https://doi.org/10.1016/j.engappai.2017.09.002.

[33] Kumar A Dwivedi. (2017). Analysis of computational intelligence techniques for diabetes mellitus prediction," *Neural Comput. Appl*., vol. 13, no. 3, pp. 1–9. DOI:10.1007/s00521-017-2969-9

[34] Vijayashree J and Jayashree, J. (2017). An Expert System for the Diagnosis of Diabetic Patients using Deep Neural Networks and Recursive Feature Elimination," *International Journal of Civil Engineering and Technology*, vol. 8, pp. 633-641. ISSN Print: 0976-6308 and ISSN Online: 0976-6316

[35] Ashiquzzaman A, Tushar A. K., Islam M,. Kim J.-M et al., ``Reduction of overfitting in diabetes prediction using deep learning neural network," arXiv:1707.08386 [cs.CV]. DOI: 10.5815/ijieeb.2019.02.03

[36] Cheruku R, Edla and DR, Kuppili V, Sm-ruleminer. (2017). Spider monkey-based rule miner using novel fitness function for diabetes classification. *Comput Biol Med*;81:79–92 https://doi.org/10.1016/j.compbiomed.2016.12.009

[37] Kannadasan K, Edla D.R., Kuppili V. (2018). Type 2 diabetes data classification using stacked autoencoders in deep neural networks. *Clin. Epidemiol. Glob. Health*. 7(4), 530–535. https://doi.org/10.1016/j.cegh.2018.12.004

[38] Kowsher M., Turaba M.Y., Sajed T et al., (2020). Prognosis and treatment prediction of type-2 diabetes using deep neural network and machine learning classifiers, in *International Conference on Computer and Information Technology (ICCIT)*. http://dx.doi.org/10.1109/ICCIT48885.2019.9038574