Comparison of Machine learning Algorithms for early prediction of Diabetes

10.05.2023

By

Aleksander Vanberg Eriksen

Table of contents

Abstract

The prediction of early diabetes a crucial when it comes to preventing the disease to harm patients. If undiscovered, it will result in complications in the body that might turn fatal. The hospitals monitor useful variables from current diabetic patients that is useful for other undiagnosed patients. These variables can be used with machine learning to find the correlation, and to start early treatment for those who haven't been diagnosed yet. This paper uses machine learning techniques such as Long short-term memory (LSTM) and Recurrent Neural network (RNN) to classify diabetics and would be diabetics using a dataset called "Early-stage diabetes risk prediction dataset" that has been conducted in a hospital located in Sylhet, Bangladesh and approved by a doctor and contained 520 samples. The models resulted in that the RNN had the best performance at an accuracy of 97 %, over the LSTM at an accuracy of 94%.

1. Introduction

Diabetes is a chronic disease that occurs either when the body cannot effectively use the insulin it produces or when the pancreas doesn't produce enough for the body to use [1,2,3]. This paper will focus on Type 1 diabetes, which is characterized by deficient insulin production and resultant hyperglycemia [1,2]. The symptoms of Type 1 include excessive excretion of urine, thirst, constant hunger, weight loss, vision change, and fatigue [1,2]. Over time, diabetes can damage the heart, blood vessels, eyes, kidneys, and nerves [1]. Early predictions are therefore important to prevent these kinds of damage and can be done through relatively inexpensive testing of blood glucose [1].

A diagnosis of diabetes is based on a fasting blood glucose concentration above 7·0 mmol/L (126 mg/dL), a random blood glucose concentration above 11·1 mmol/L (200 mg/dL) with symptoms, or an abnormal result from an oral glucose tolerance test [2,3]. There is also a different diagnosis which can be made of the basis of a glycated haemoglobin concentration, which refers to a measure of average blood sugar levels over a period of weeks/months, above 48 mmol/mol (6-5%) [2]. For children with diabetes type 1, the commonly symptoms that is present are polyuria (the passage of large volumes of diluted urine), polydipsia (chronic or intermittent ingestion of large volumes of water), and weight loss [1,2,3]. It is one of the most common chronic diseases of childhood even though it can be diagnosed at any age [3]. It happens the most at around the age of 5-7 and at or close to puberty [3]. The disease is more common in men and boys than women, and more cases where the child has diabetes type 1 happens if they are born in autumn or winter rather than spring or summer [3]. Type 1 diabetes accounts for only 5-10% of all cases of diabetes, though it continues to grow worldwide, and it will have serious short-term and long-term implications [4]. The management of the disease requires continues attention which means insulin administration, blood glucose monitoring and meal planning [4].

2. Background

Since these are clinical data that can be extracted and studied, it can be used in conjunction with machine learning to automate the study so that it is easier to identify the variables that might have a bigger impact which will lead to diabetes type 1 [5]. Since the data can be measured over time, we will use machine learning techniques such as LSTM and RNN to predict, and also compare to see which one performs the best in terms of accuracy.

2.1. Recurrent Neural Network

RNN stands for recurrent neural network and is a type of artificial neural network that can process sequence data [6]. It consists of an input layer, hidden layer, and an output layer [7]. In the neuron of the hidden layer is an RNN unit that consist of a hidden state which holds the short-term memory, an input path, a tanh activation function, and an output path [8].

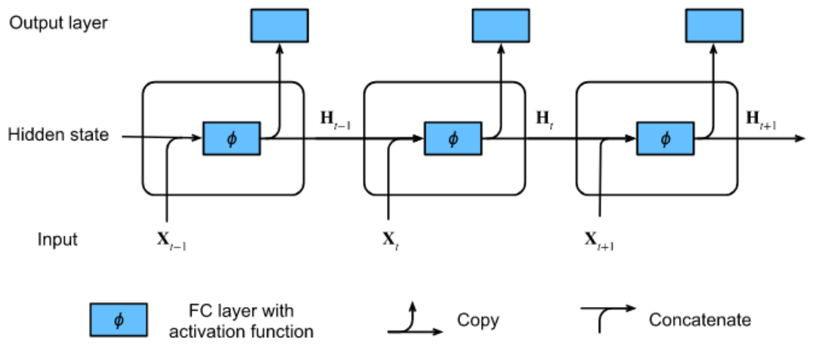


Figure 1: an overview of the RNN unit

The short-term memory works that it remembers the previous inputs through the state, which allows it to affect the next input's output [9]. Since the RNN only uses short-term memory, it struggles to find the local optima when the dataset used is large [8,9]. This is due to the vanishing/explosive gradient, which means that when the weight is updated through backpropagation, the update is so small or too big that it takes forever to converge [9, 11]. LSTM on the other hand can solve this problem [9, 10, 11].

2.2. Long short-term memory

LSTM stands for long-short term memory and is an enhanced version of the RNN model [9, 10, 11]. It additionally consists of a long-term memory called the cell state and uses three gates which prevents the vanishing/exploding gradient [9, 10, 11]. Therefor the LSTM is quite efficient when it comes to large datasets with long length sequence data, since it can capture the long-term dependence and modeling nonlinear dynamics [9, 10, 11]. The gates that are used in the LSTM can be considered as filter functions, as it discards unwanted variables so that it does not reach vanishing/explosive gradient problem [9, 10].

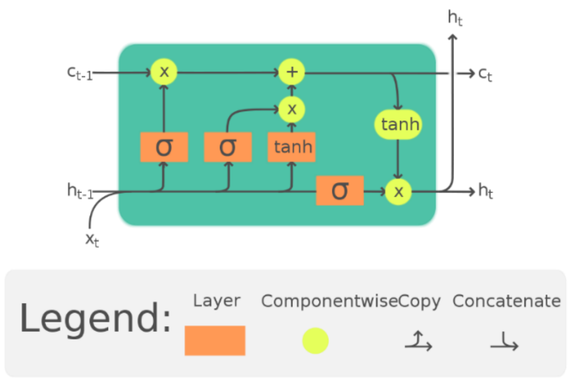


Figure 2: Overview of an LSTM unit

The name of the gates is forget gate, input gate, and output gate [9, 10, 11]. The forget gate determines if the input value is worth keeping through a sigmoid function that outputs between 0-1 [9, 10]. If the output is 0 it will discard the input value, if its 1, it will keep the input value, and if it's something in between then it will keep that percentage of that input value [9, 10, 11]. The result gets multiplied with the current long-term memory [9,10,11]. The input gate determines how much of the input value is to be used in the new long-term memory [9, 10, 11]. The gate also uses a sigmoid function for filtering reasons, and additionally a tanh function [9, 10, 11]. The tanh function outputs a value between -1 and 1and is used to determine the importance of the input value [9, 10, 11]. The result is added to the current long-term memory [9,10,11]. After the LSTM unit has calculated a new long-term memory value, the output gate is used to determine a new short-term memory [9, 10, 11]. The new long-term memory is feed through a tanh function, and the output is multiplied together with the output of a sigmoid function that is fed the input value [9, 10, 11]. After this process, a new short-term memory is calculated and sent to the next time-step and to the output layer for backpropagation [9, 10, 11].

3. Related work

A study done by Swapna G, Vinayakumar R and Soman K.P used deep learning to detect diabetes. They used techniques such as Recurrent neural network(RNN), Long short-term memory(LSTM), Convolutional neural network(CNN), Hybrid network(CNN-LSTM) and support vector machine (SVM). These techniques were applied on a dataset containing twenty electrocardiograms each on people with diabetes and people without it. The proposed architecture where a CNN-LSTM network that handled feature extraction that was passed into SVM for classification. They tried with one trough five fold cross validation, and compared it with regular CNN with SVM. The accuracy obtained with CNN and SVM ranged from 0.684 to 0.939, starting with one to five in fold-cross-validation. The

CNN-LSTM with SVM ranged from 0.743 to 0.957 accuracy, starting with one to five in fold-cross-validation. From these results the CNN-LSTM with five-fold-cross-validation and SVM was superior with an accuracy of 95.7 % [12].

A study by Motiur Rahman, Dilshad Islam, Rokeya Jahan Mukti and Indrajit Saha also used a deep learning approach to detect diabetes. They used convolutional LSTM to automate the detection of diabetes. They used clinical data from a dataset named Pima Indians Diabetes Database which was obtained from the UCI machine learning repository. The dataset contained records of 768 female patients where 268 of them was diabetes positive. They were at least 21 years, and the variables recorded was pregnancies, glucose, blood pressure, BMI, Skin thickness, Insulin, Diabetes Pedigree Function, age and a target variable which told whether they were diabetes positive or not. The CNN-LSTM was compared with a traditional LSTM (T-LSTM) on parameters like learning rate, batch size, hidden units, Epochs and Mean test score for hyperparameter optimization. The CNN-LSTM worked best at 0.01 learning rate, with 32 batch size, 100 hidden units, 300 epochs, 5*5 kernel size with a Mean Test score of 94.87. CNN also worked best with the same setup with a Mean Test score of 89.02. For the evaluation, the study compared T-LSTM, CNN, CNN-LSTM and Conv-LSTM with techniques such as Separate Training-Testing Set and Five-Fold Cross-Validation which was used with selected features. For the first technique Conv-LSTM had the highest accuracy of 91.38 %, while CNN had the lowest at 82.14 %. For the second technique, Conv-LSTM also had the highest accuracy of 97.26, while CNN had the lowest at 88.73 %. Using all features Conv-LSTM had the highest accuracy for both at 86.61 using Separate Training-Testing Set and 94.23 using Five-Fold Cross-Validation. Their conclusion based on these results was that the Conv-LSTM worked best since LSTM can capture long term data dependencies, and Convolution helps the algorithm further by adding extra connection called peephole to extract features [13].

Another study done by Swapna G., Soman Kp and Vinayakumar R, they were analyzing the heart rate variability obtained through ECG signals to diagnose diabetes. They used deep learning techniques such as CNN and a combination of CNN and LSTM called CNN-LSTM to automatically detect the abnormalities in the signals. The techniques used did not need feature extraction, so they only split the dataset into training and test set and preformed classification. With the use of 5-fold cross-validation, the deep learning technique CNN gave an accuracy of 93.7%, while the combination technique CNN-LSTM gave an accuracy of 95.1 %. With no 5-fold cross-validation, CNN-LSTM gave a maximum accuracy of 90.9%. From their knowledge, their paper is the first one to take deep learning techniques and employ them in diabetes and normal HVR, and their result is the best one gotten using cross-validation for automated detection of diabetes using HVR [14].

4. Methodology

Methods used to manage the chosen dataset and how it got implemented.

4.1. The dataset

The dataset used in this paper is a dataset from the Machine Learning Repository. It is named Early-stage diabetes risk prediction dataset and contains the sign and symptom data of newly diabetic or would be diabetic patients. It is multivariate which means there are multiple features and is described as a classification problem. This dataset was donated 12.07.2020 and collected using direct questionnaires from the patients of Sylhet Diabetes Hospital in Sylhet, Bangladesh and approved by a doctor [15].

4.2. Inspecting the dataset

This study done using google collab, which is an environment that uses python programming. The dataset was converted from a csv file to a data framework using the library *pandas*. Pandas was then further used to inspect the dataset, mostly to see what type of values were represented in the dataset and to see if the dataset had missing values or not. If it had any missing values, then the rows that had them got removed.

4.3. Encoding the dataset

For the values of the dataset, the inspection was done in regarding if there were any categorical data, which it was. To solve this problem with, a library called OneHotEncoder from *sklearn.preprocessing* was used to convert the categorical values to numerical values based on logic. Example if one feature had two unique values, then they would convert to 0 and 1 respectively.

4.4. Transforming and preprocessing

After the encoding of the dataset, the entire dataset needed to be transformed from a 2D tensor to a 3D tensor, because when feeding it to the LSTM and RNN, we need to consider the timesteps between the rows. To solve this, an extraction of each feature is done from the data frame to an array, where the number of values extracted from each feature is the timesteps. This means that when combining each feature again to a single row, that row will result in a timestep. Example wise will sample number one have a shape like this (1, 1, 15).

For the models to have an easier time learning the dataset, I applied a preprocessing technique called MinMaxScaler that was imported from the *sklearn.preprocessing* library. This technique scales all variables to a number between 0 and 1 and is scaled according to how many unique values are in the respective feature.

4.5. Model and training

The data is then split into a training and test set using the library *sklearn.model_selection* and with its function *train_test_split*. The training set contains 64% of the samples, while the test set contains 33%. The LSTM model starts with an input layer with a shape of (1,15). Then an LSTM layer is added with 128 units, and *return_sequence* set to True. Next is another LSTM layer with 64 units, but *return_sequence* is now set to False. Then a Dense layer is added with 10 units, and lastly another Dense layer is added with 1 unit that acts as the output layer. The loss function used is *BinaryCrossentropy* with *from_logits* set to True. The optimizer used is called Adam and its learning rate is set to 0.001. The RNN model is set up the same as the LSTM mode regarding number of layers, and parameters, but uses SimpleRNN layers instead of LSTM ones. The LSTM and RNN model both uses a batch size of 16, with 500 epochs when training. The validation split is set to 0.25.

5. Experiments

Results of the implemented methods. This will also contain the analysis of the different machine learning algorithms used.

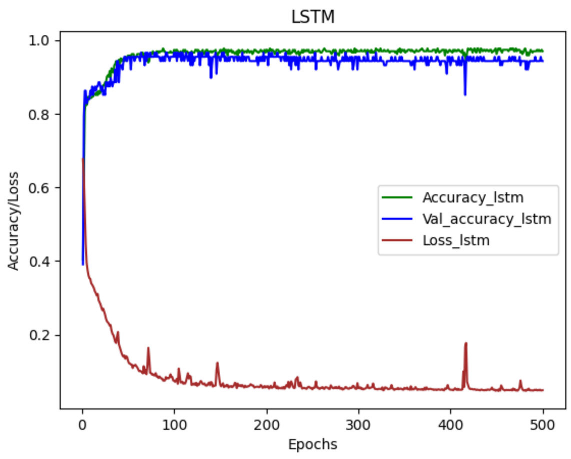


Figure 3 - LSTM acc, loss and val acc

The result from Figure 3, shows how the LSTM model preformed during the training period. The graph contains three metrics which is as follows: Accuracy, Val_accuracy and Loss. The accuracy shows the performance on the model in terms on how well the model predicted during the training period. The Val accuracy shows how well the model preformed when the validation set was used on the model during the training period. The loss describes how the model preformed in terms of going towards convergence. The model got an accuracy of 0.9692, a val accuracy of 0.9425 and a loss value of 0.0494.

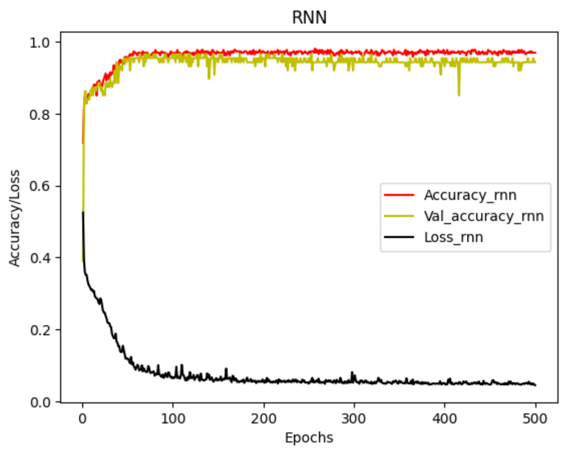


Figure 4 - RNN acc, loss and val acc

The RNN model had very similar results as the LSTM model with an accuracy of 0.9692, val accuracy of 0.9540 and a loss value of 0.0450.

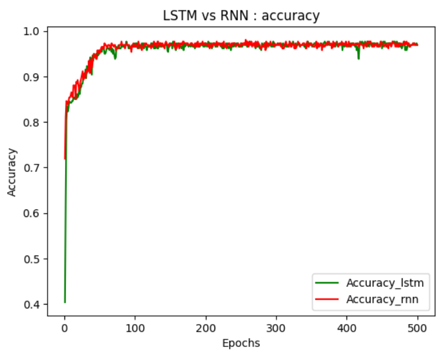


Figure 5 - LSTM vs RNN acc

For comparison of the two models, Figure 5 shows that the LSTM is a bit more unstable at during training but ends up getting the same value in the end as the RNN.

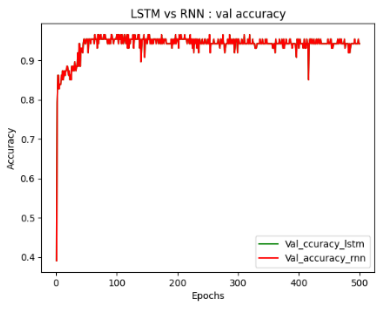


Figure 6- LSTM vs RNN: val acc

Looking at the val accuracy for both models, both has the exact same result.

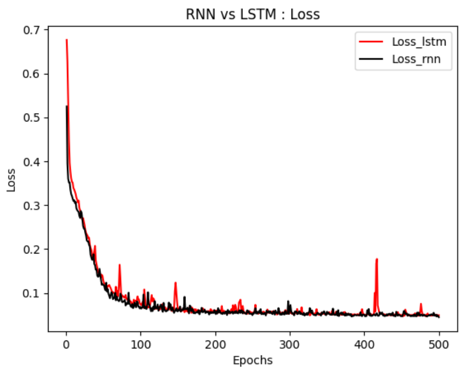
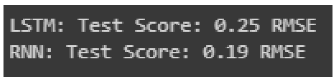


Figure 7- RNN vs LSTM: Loss

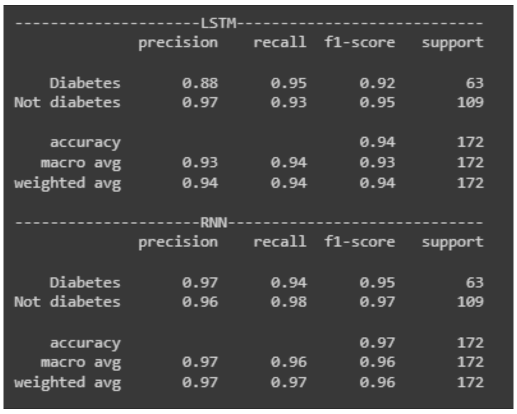
The loss function shows that for the LSTM, it starts a bit higher than the RNN model and is a lot more unstable for all epochs. The RNN model have some jumps, but a lot more consistency than the LSTM.



```
LSTM: Test Score: 0.25 RMSE
RNN: Test Score: 0.19 RMSE
```

Figure 8 - RMSE score for LSTM and RNN

When testing the RMSE for the models, the LSTM a value of 0.25 while the RNN got a value of 0.19.



```
--------------------LSTM--------------------------
              precision    recall  f1-score   support

    Diabetes       0.88      0.95      0.92        63
Not diabetes       0.97      0.93      0.95       109

    accuracy                           0.94       172
   macro avg       0.93      0.94      0.93       172
weighted avg       0.94      0.94      0.94       172

--------------------RNN---------------------------
              precision    recall  f1-score   support

    Diabetes       0.97      0.94      0.95        63
Not diabetes       0.96      0.98      0.97       109

    accuracy                           0.97       172
   macro avg       0.97      0.96      0.96       172
weighted avg       0.97      0.97      0.96       172
```

Figure 9 Classification report

The classification report for both models shows that the overall best model was the RNN with an average precision, recall and f1-score at 0.97. The LSTM had an average of 0.94.

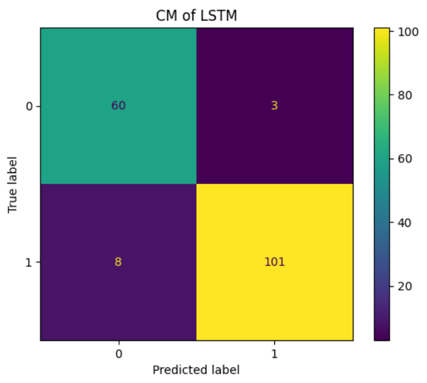


Figure 10 - CM LSTM

The confusion matrix of the LSTM shows that when classifying label 0 which is no diabetes, it got 60 right and 3 wrong. When classifying label 1 which is diabetes, it got 101 right and 8 wrong.

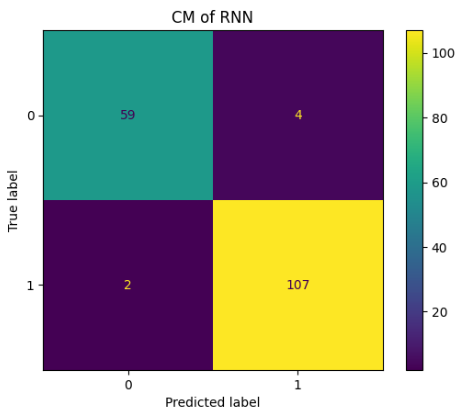
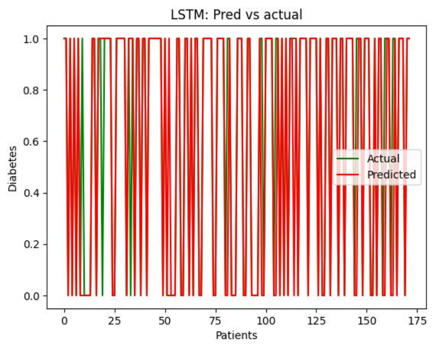


Figure 11 - CM RNN

The confusion matrix of the RNN model shows that when classifying label 0, it got 59 right and 4 wrong. When classifying label 1 it got 107 right and 2 wrong.



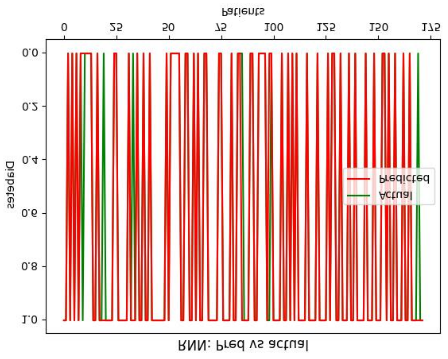


Figure 12 - LSTM: pred vs act

Figure 13 RNN: pred vs act

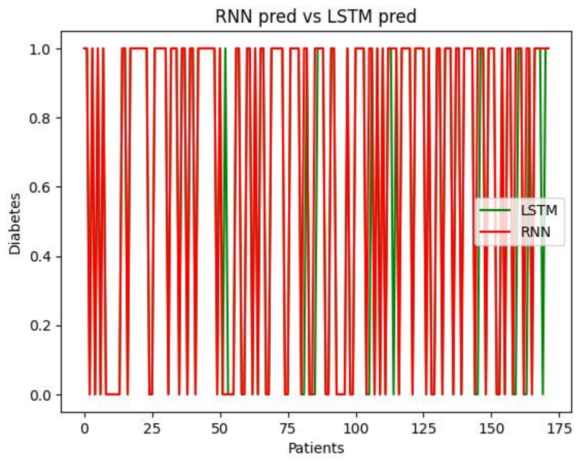


Figure 14 – RNN pred vs LSTM pred

Figure 12, 13 and 14 shows the results of the models when applying the test set to predict whether the samples classify as a diabetic or not. They are very similar but has some differences in terms of classification. From Figure 14, the differences are in the pure green or red line. The red line represents the classification of the RNN model, while the green line represents the classification of the LSTM model.

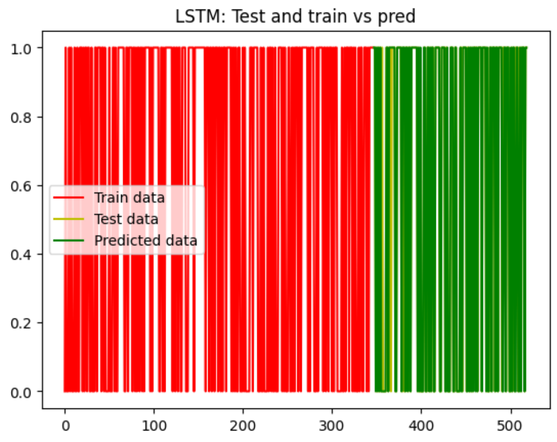


Figure 15 – LSTM: Test and train vs Pred

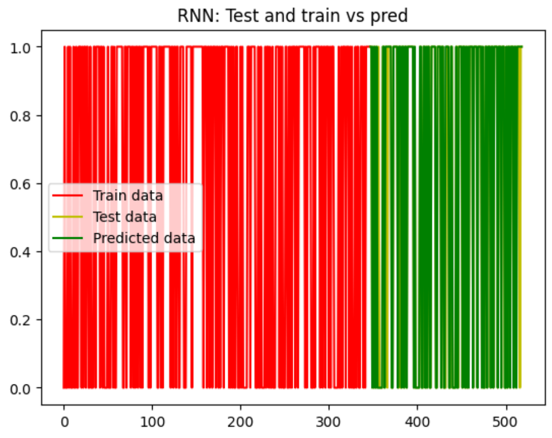


Figure 16 – RNN: Test and train vs pred

When comparing the true values with the predicted values for both the models and inspecting the yellow line, the LSTM have more visible errors than the RNN.

6. Conclusion

Both the LSTM and RNN model did good at classifying early diabetes on patients with different traits or symptoms that would result in them getting diabetes or not. They both were very similar when predicting using the test set, but the RNN model was better. Even though the LSTM technique is made to be better than the RNN technique by solving the vanishing/exploding gradient descent problem, this was not the case. This might be due to the amount of data used in this research. Total there was only 520 cases of diabetics or non-diabetics patients, and the LSTM is better when handling more data. Furthermore, the RNN had the best performance at an accuracy of 97% but the LSTM still had a decent of accuracy of 94%. Further research would be using a bigger dataset so that the LSTM could show its potential and it can show how the vanishing/exploding gradient is handled, and at the same time see how it affects the performance of the RNN technique.

7. References

[1] World Health Organization. (2023, March 5). *Diabetes*. Retrieved from https://www.who.int/news-room/fact-sheets/detail/diabetes.

[2] Linda A DiMeglio, Carmella Evans-Molina, Richard A Oram, Type 1 diabetes, The Lancet, Volume 391, Issue 10138, 2018, Pages 2449-2462, ISSN 0140-6736, https://doi.org/10.1016/S0140-6736(18)31320-5. (https://www.sciencedirect.com/science/article/pii/S0140673618313205)

[3] Mark A Atkinson, George S Eisenbarth, Aaron W Michel's, Type 1 diabetes, The Lancet, Volume383, Issue 9911,2014,Pages 69-82,ISSN 0140-6736, https://doi.org/10.1016/S0140-6736(13)60591-7. (https://www.sciencedirect.com/science/article/pii/S0140673613605917)

[4] Denis Daneman, Type 1 diabetes, The Lancet, Volume 367, Issue 9513,2006, Pages 847-858, ISSN0140-6736, https://doi.org/10.1016/S0140-6736(06)68341-4. (https://www.sciencedirect.com/science/article/pii/S0140673606683414)

[5] (Rahman et al., 2020) M Rahman, D Islam, R. J Mukti, I Saha, A deep learning approach based on convolutional LSTM for detecting diabetes, Computational Biology and Chemistry, Volume 88, 2020,107329, ISSN14769271, https://doi.org/10.1016/j.compbiolchem.2020.107329. (https://www.sciencedirect.com/science/article/pii/S1476927120304692)

[6] S. Yang, X. Yu and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), Shanghai, China, 2020, pp. 98-101, Doi: 10.1109/IWECAI50956.2020.00027.

[7] Z. Yang et al., "An Artificial Neural Network Based Attenuation Tomography in Free Space Optical Network," 2018 International Conference on Networking and Network Applications (NaNA), Xi'an, China, 2018, pp. 52-57, Doi: 10.1109/NANA.2018.8648769.

[8] Raheleh Khanduzi, Arun Kumar Sangaiah,An efficient recurrent neural network for defensive Stackelberg game,

Journal of Computational Science, Volume 67,2023,101970, ISSN 1877-7503,https://doi.org/10.1016/j.jocs.2023.101970. (https://www.sciencedirect.com/science/article/pii/S1877750323000303)

[9] Q. Ye, X. Yang, C. Chen, and J. Wang, "River Water Quality Parameters Prediction Method Based on LSTM-RNN Model," *2019 Chinese Control and Decision Conference (CCDC)*, Nanchang, China, 2019, pp. 3024-3028, Doi: 10.1109/CCDC.2019.8832885.

[10] Rian Dolphin, “LSTM Networks | A Detailed Explanation”, *Towards Data Science*, Oct 21 2020, Date accessed: 09.03.2023. https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9

[11] Algoscale, “What is Long-short-term memory (LSTM)”, *Algoscale*, February 8 2022, Date accessed: 09.03.2023. https://algoscale.com/blog/what-is-long-short-term-memory-lstm/

[12] Swapna G., Vinayakumar R., Soman K.P., Diabetes detection using deep learning algorithms, ICT Express, Volume 4, Issue 4, 2018, Pages 243-246, ISSN 2405-9595, https://doi.org/10.1016/j.icte.2018.10.005. (https://www.sciencedirect.com/science/article/pii/S2405959518304624)

[13] Motiur Rahman, Dilshad Islam, Rokeya Jahan Mukti, Indrajit Saha, A deep learning approach based on convolutional LSTM for detecting diabetes, Computational Biology and Chemistry, Volume 88, 2020, 107329, ISSN 1476-9271, https://doi.org/10.1016/j.compbiolchem.2020.107329. (https://www.sciencedirect.com/science/article/pii/S1476927120304692)

[14] Swapna G, Soman Kp, Vinayakumar R, Automated detection of diabetes using CNN and CNN-LSTM network and heart rate signals, Procedia Computer Science, Volume 132, 2018, Pages 1253-1262, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.05.041. (https://www.sciencedirect.com/science/article/pii/S1877050918307737)

[15] https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. Date accessed: 05.03.2023.

8. Figures

Figure 1: An image of the RNN structure, Date accessed: 09.03.2023, https://d2l.ai/chapter_recurrent-neural-networks/rnn.html

Figure 2: An image of the LSTM unit, Date accessed: 09.03.2023, https://en.wikipedia.org/wiki/Long_short-term_memory
Figure 3 - 16: Plotted using the library https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html

9. Code

https://github.com/AleksanderVEriksen/My_Projects/blob/main/Jupyter%20Notebook%20/Machine%20learning/LSTM_RNN_Prediction.ipynb