Developing A Deep Learning Solution to Estimate Instantaneous Glucose Level from Heart Rate Variability

E. Shaqiri*, M. Gusev**, L. Poposka***, M. Vavlukis***, I. Ahmeti****
* Innovation DOOEL, Skopje, North Macedonia
** Sts. Cyril and Methodius University in Skopje, Faculty of Computer Science and Engineering, Skopje, North Macedonia
*** Sts. Cyril and Methodius University in Skopje, Clinic of Cardiology, Skopje, North Macedonia
**** Sts. Cyril and Methodius University in Skopje, Clinic of Endocrinology, Skopje, North Macedonia
E-mail: ervin.shaqiri@innovation.com.mk, marjan.gushev@finki.ukim.mk,
lidijapoposka@gmail.com, marija.vavlukis@gmail.com, iahmeti@yahoo.com

***Abstract*—A great deal of studies address the use IoT devices coupled by machine learning in order to predict and better detect health problems. Diabetes is an issue that society is struggling for a very long time. The ease with which ECG signals can be recorded and interpreted provides an opportunity to use Deep Learning techniques to predict the estimated Sugar Levels of a patient. This research aims at describing a Deep Learning approach to provide models for different short term heart rate variability measurements.**

Our approach is based on a special method to calculate heart rate variability with identification of segments, then averaging and concatenating them to exploit better feature engineering results.The short-term measurements are used for determination of instantaneous plasma glucose levels. Deep Learning method is based on Autokeras, the neural architectural search provided the best results for the 15 minute measurements.

Our research question is to develop a solution to estimate the Instantaneous glucose value from heart rate variability with sufficient quality. The evaluated test set gave the following results: RMSE(0.368), MSE(0.193), R square(51.281), and R squared loss(54.128).

***Index Terms*—*ECG*; *HRV*; *Deep Learning*; *Glucose Short Term*; *Diabetes*;**

I. Introduction

Diabetics is a health condition that affects people of all age groups, and in a lot of cases, people are not aware they are diabetic. Known methods to detect raised glucose levels include invasive, minimally invasive and non-invasive techniques [1]. Invasive techniques are based on finger pricking and measuring the chemical properties of a blood drop, while the minimally invasive techniques mainly use a set of very small needles and test interstitial body liquids or other parameters. Non-invasive techniques use electromechanical properties to scan the effects that raised glucose levels produce on the skin, eyes or other interstitial body liquids. Our approach is based on using wearable non-invasive ECG sensors to scan he heart condition and determine a set of heart rate variability parameters which will be processed by a new Deep Learning (DL) method.

DL-methods are subset of the field Machine Learning (ML), utilizing artificial neural networks. The technology has been around for quite some time, the first recognized use dating back to 1873 where Alexander Bain introduced Neural Groupings as the earliest models of neural network, inspired Hebbian Learning Rule. [2]. However, in that period machines were slow and the calculations quite expensive both in terms of computation and finance wise. The last decade in computer science has seen an explosion of DL-methods, especially with the availability of cloud computing. Machines have enhanced their performance so much that today a person can run a DL process at home or even on a raspberry PI. [3]

Taking into account the current state of DL, we set a research hypothesis to explore and find out whether it is possible to develop a DL solution which detects the glucose level from HRV. This endeavour brings about an inquiry for a DL method of estimated instantaneous blood glucose level from simple Time Domain HRV parameters. The approach begins with collecting electrocardiogram data from which HRV parameter are calculated for different short term time frames. The initial steps are followed by correlation analysis and conclude with the training of DL models for each time frame. The data and resources are provided under the umbrella of the Glyco project [4]

The measurement were conducted within the Glyco project, in which a group a patients carry a small apparatus that records the persons ECG signal and then relays that to a proprietary application which processes and annotates the appropriate information. These processed annotations comprise the adequate information for calculating Heart Rate Variability (HRV). HRV is known as a good parameter for predicting different health conditions [5]. From previous research we have identified two methods for calculating HRV for a certain annotation file segmenting the signal into clean Normal-to-Normal intervals [6]:

- Averages - taking the average HRV for each segment.
- Combined - concatenating all segments into one long segment and calculating the HRV on that long segment.

The concept behind our idea is that the autonomous nerve system controls the heart and reacts on blood glucose levels,

setting a lead that short-term HRV are correlated to instantaneous plasma glucose levels. Further on, we are eager to find answers on the following questions:

- How do these different HRV measurements hold up against the fluctuation of sugar levels in the blood?
- Do these calculations have predictive capabilities?
- Which calculation method performs the best at predictions?

Answers to all these questions lead to specification of the research question which is addressed in this paper: Are HRV parameters able to predict Sugar Level of a person?

The article begins with presentation of related work in Section II and description of the Methods in Section III. Section IV shows the results with their evaluation, and Section V discusses the applicability and performance of achieved results. Finally, the conclusions and future work are elaborated in Section VI.

II. Related Works

There is a solid amount of studies on Correlating HRV parameters to Glucose levels, and in some cases there are attempts at implementing ML or even DL techniques in order to predict whether a person is diabetic. However, finding works on predicting an actual Sugar Level value proves to be extremely difficult. To the authors knowledge no such works were found.

In December 2018 Swapna et al [7] published a paper describing their attempts at detecting diabetes using DL algorithms. Their dataset consists of 20 diabetic and 20 healthy patients. Each patient had a 10min recording of their ECG in a lying down relaxed supine position. The ECG signal is sampled at 500 Hz from which 71 datasets for each group of patients are generated. Each dataset consists of 1000 samples. The authors conclude that the maximum accuracy they achieved through DL-methods is 95,70%.

In a more recent study Aggrawal et al [8], explore the possibilities of detecting diabetes using artificial neural networks and support vector machines from HRV parameters. The study is conducted on a population of 10 rats, where they are evenly split into a experimental and control group. The difference between the groups were their diets which for the experimental group would induce diabetes. After collecting the ECG signals and calculating HRV parameters the authors then implement Artificial Neural Networks (ANN) and Support Vector Machines (SVM) to predict diabetes. They conclude that ANN had an accuracy of 86.30, while SVM had an accuracy of 90.50%.

Another recent study by Rahman et al [9] who aim to see the predictive capabilities of diabetes through different DL techniques: Convolutional Long Short-term Memory (Conv-LSTM), Convolutional Neural Network (CNN), Traditional LSTM (T-LSTM), and CNN-LS. The dataset contains records of 768 female patients aged atleast 21 years among them 268 are diabetes positive and the rest arediabetes negative. The dataset has eight predictor variables like Preg-nancies, Glucose, Blood Pressure, BMI, Skin Thickness, Insulin, DiabetesPedigree Function, and Age for diagnostically predicting whether apatient has diabetes or not and one target variable named as outcome. The authors conclude that the best model was generated by the Conv-LSTM with an accuracy of 97.26%.

DeepHeart is another study worth mentionig [10]. This study cohort encompasses 14,011 users of a specific Apple Watch app. Each participant's data was then split into week-long chunks, and any weeks $\leq$ 30 minutes of continuous heart rate recordings were omitted this resulted in a total of 57,675 person-week data. The authors conclude that a multi-task long short term memory (LSTM) yielded high accuracy results at detecting multiple medical conditions, including diabetes (0.8451), high cholesterol (0.7441), high blood pressure (0.8086), and sleep apnea (0.8298)[10].

III. Methods

In this section we described the methods and materials used, First we describe the used dataset, along with dataset splitting techniques for creation of training and testing subsets, then we continue to elaborate the feature engineering process. Special focus is set on the specification of the experimental setup and evaluation metrics.

A. Dataset

The Dataset consists of subjects that take part in the Glyco Study. There are a total of 155 subjects however, from those 155 only 138 are valid subjects that pose both clean and continuous ECG measurements alongside manual Glucose measurement through finger pricking. It is also important to note that the subjects in this dataset are known to have health problems, more precisely heart problems. The dataset contains 94 male and 44 female patients consisting of 75.442 observations, the age in males is 59.7 ± 9.6 years of age while in females 63.2 ± 11.3 years of age. Where males have a weight of 84.4 ± 17.6 cm and females 76.6 ± 11.1 cm, with height being 161.9 ± 46.5 kg and 164.1 ± 4.7 respectively. Finally, males have a BMI of 26.1 ± 8.6 while females 28.5 ± 4.1. The distribution are visualized in Fig.1.

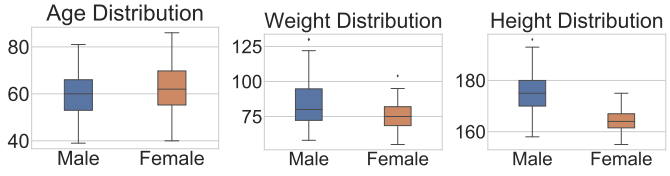

Fig. 1. Distribution of Age, Weight and Height by Gender

This article's scope is on short term recordings. The approach here is to divide the dataset into six distinct datasets, each dataset is treated as a separate experiment group which results in a classification model for each short term recordings. Each datasets consists of data collected on a sliding window concept. In the end we are left with the following short term datasets:

- D30S - 30s sliding window with a 30s offset
- D1M - 1m sliding window with a 1m offset
- D5M - 5m sliding window with a 1m offset
- D10M - 10m sliding window with a 5m offset
- D15M - 15m sliding window with a 5m offset
- D30M - 30m sliding window with a 10m offset

B. Features

The HRV parameters that are taken into consideration are the the Time Domain features. It was decided to take them because of the ease of calculation and the already standardized techniques which can be reproduced in other environments. Since we are dealing with 2 calculation techniques we are left with 15 features:

Parameter	Description
A-SDNN	Standard Deviation of NN intervals
A-ASDNN	The Average Standard Deviation of NN intervals (minimum 5 minutes)
A-SDANN	The Standard Deviation of the averages of NN intervals (minimum 5 minutes)
A-NN50	Adjacent NN interval pairs differing more than 50ms
A-pNN50	NN50 counts divided by total count of NN intervals
A-rMSSD	The square root of the mean of the sum of the squares of differences between adjacent NN intervals
C-SDNN	Standard Deviation of NN intervals
C-ASDNN	The Average Standard Deviation of NN intervals (minimum 5 minutes)
C-SDANN	The Standard Deviation of the averages of NN intervals (minimum 5 minutes)
C-NN50	Adjacent NN interval pairs differing more than 50ms
C-pNN50	NN50 counts divided by total count of NN intervals
C-rMSSD	The square root of the mean of the sum of the squares of differences between adjacent NN intervals
Distance	Total length of recording expressed in seconds
Timestamp	The time of the recording

An important aspect that requires attention is that some parameters require a min of 5min recordings in order to be calculated. This means that for the D30S, D1M, D5M datasets the parameters ASDNN, SDANN, NN50, pNN50 are removed from the feature list.

C. Experiments

As mentioned previously each dataset is considered as a single experiment group. In other words, each dataset represents a time frame for which HRV is calculated. We aim at creating a model for each time frame (dataset) which can be used in real world cases where there would be a need to predict on 30s, 1m, 5m, 10m, 15m or 30m time spans. Each model may only be used for the desired time frame, so the model trained and tested on 30s can not be used to make predictions on 30m data or vice versa.

The experiments or DL processes, are done through the utilization of AutoKeras[11]. Autokeras is an automated ML-based system based on Keras[12]. Keras in turn in a just a wrapper of Tensorflow which in turn is a DL-based framework developed by Google[13]. For these experiments we are attempting to predict the estimated sugar level of a patient from their ECG, which means we have a regression problem. Autokeras requires us to only set some parameters and provide the data, after that it handles the architectural search entirely automatically. The parameters we provide are:

- the structured data regression class
- epochs - number of epochs for each trial
- max number of trials - the number of different architectures to try (trial=architecture)
- batch size
- metrics - which metrics to report while training
- objective - which metric to track and develop upcoming architectures according to the metric maximization or minimisation
- callbacks - custom callback which the user wants to use, the same as manual Keras callbacks (optional)

D. Evaluation metrics

Since the problem at hand is a regression problem, the following metrics are tracked during model development and evaluation:

- Mean Squared Error
- Root Mean Squared Error
- R squared
- R squared loss

Mean Squared Error (MSE) measures the average of the squares of the errors and is calculated by (1). It is the average squared difference between the estimated value and the actual value.

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}\Big(\frac{d_i - f_i}{\sigma_i}\Big)^2 \tag{1}$$

Root Mean Squared Error (RMSE) is calculated by (2) as an absolute measure of the goodness for the fit. In other words the differences between value predicted by a model or an estimator and the value observed.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(\frac{d_i - f_i}{\sigma_i}\Big)^2} \tag{2}$$

R squared (R^2), calculated by (3), is the proportion of the variance in the dependent variable that is predictable from the independent variable. Sometimes it is called the coefficient of determination. In other words R^2 is the measure of how close the data is to the fitted regression line. It is expressed as a percentage. Having an R^2 value trending towards 100% shows that the regression line is a good fit for the data.

$$R^2 = 1 - \frac{RSS}{TSS} \tag{3}$$

IV. Results

The Experiments are done through the utilization of python packages such as Tensorflow, Keras, and Autokeras. The environment consists of python 3.8 where all DL processes are run through a CUDA enabled GPU. The training and automated DL time for each Dataset is roughly 24h.

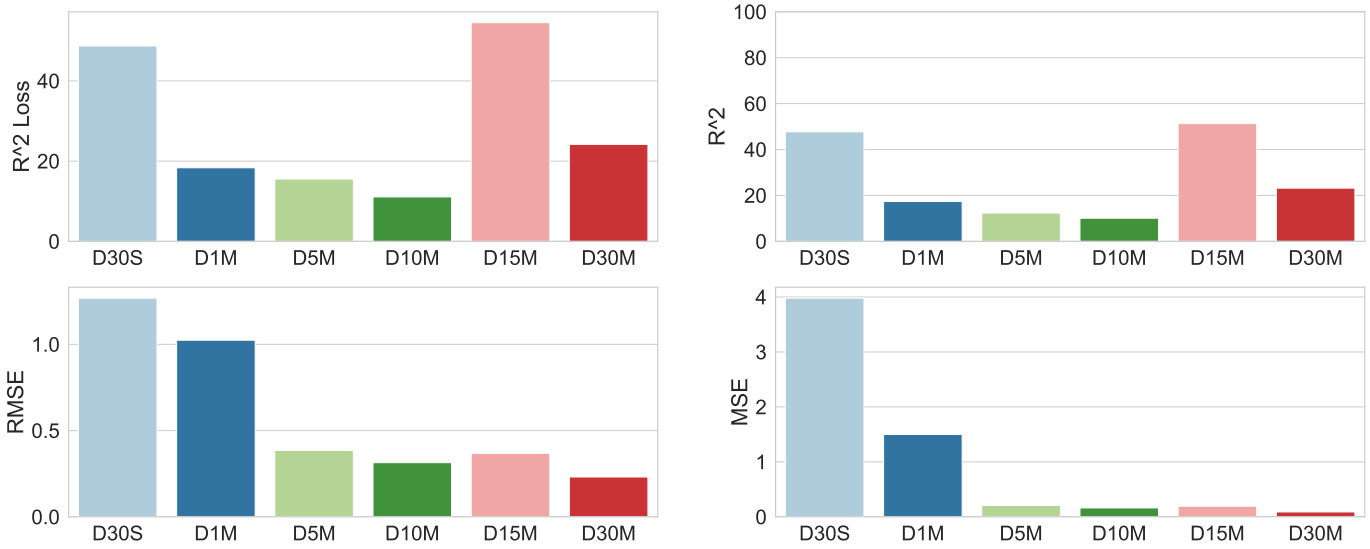

Fig. 2. Final Results.

Dataset	Original	Cleaned	Pearson Original	Pearson Cleaned
D30S	2025	1674	0.003	0.1
D1M	2951	2414	-0.032	0.042
D5M	8139	6447	-0.080	0.023
D10M	156	93	-0.003	-0.190
D15M	856	554	0.030	-0.050
D30M	714	460	-0.464	-0.213

TABLE I
OUTLIER REMOVAL

A. Outlier Removal

The DL-based model generation process is similar to every other ML-based project. The initial steps involved collecting appropriate data and pushing that data into a statistical analysis phase. Through this analysis we come to the part of detecting outliers and their removal. The Outlier removal results are presented in Tab. IV-A

B. Autokeras

The AuoKeras process for each dataset took around 24 hours to complete. The code is the same only the input data and the input shape of the initial layer changed. The objective of the experiments was to track MSE score and minimize it as much as possible without falling into the trap of overfitting. Additionally close attentnion was payed to R^2 which served as a way to distinguish the overall fit of the model. This metric was chosen since it is the most popular metric when it comes to regression problems[14]. The only negative aspect of the process is that we do no know the rational behind which AutoKeras decides to increase layer sizes or number of electrons. With that said below are presented the results for each best model generated from AutoKeras tested on the test set after training Fig.2.

In order to get the bigger picture below is a description of the architectures of each best model Table.II:

V. DISCUSSION

As mentioned before the data is fed into an Auto ML process made possible through AutoKeras. Each dataset corresponding to a specific time span, goes through the same process. The only difference is that the time spans less than five minutes have a smaller amount of features. This is due to the fact that features such as ASDNN require a minimum of a five minute recordings, this for example would be impossible to calculate in 30s dataset. When it comes to metrics tracking, in the related works section II we see that all research papers have used MSE as the main metric to follow and track and then R^2 in order to asses the accuracy level of generalization on new data.

A. Comparison to other research

Comparing previous research, we notice that the datasets in use were relatively small compared to the Glyco datasets. Additionally, we see that the goal of these papers is to classify a person as diabetic or not, in other words they tackle a classification problem. To the authors knowledge there were no attempts at predicting the actual value of the Sugar Level.

B. Optimizing HRV for glucose estimation

When it comes to the architectures that AutoKeras came up with, for the 30s time span dataset the optimal combination of hyperparamater involves an input layer followed by three dense layers with 16,24, and 32 neurons. Additionally there are three batch normalization layers and finally the output layer. All activation functions are of type ReLu, with an Adam optimizer at a learning rate of 0.001, and for the loss function Mean Squared Error Loss. The model results an RMSE score of 1,267, MSE of 3,978, R^2 of 47,710, and R^2 of 48,710.

The 1m dataset best performed with a model consisting of 3 hidden layers with 32,32, and 16 neurons respectively. For the optimizer it chose again Adam with a learning rate of 0.001

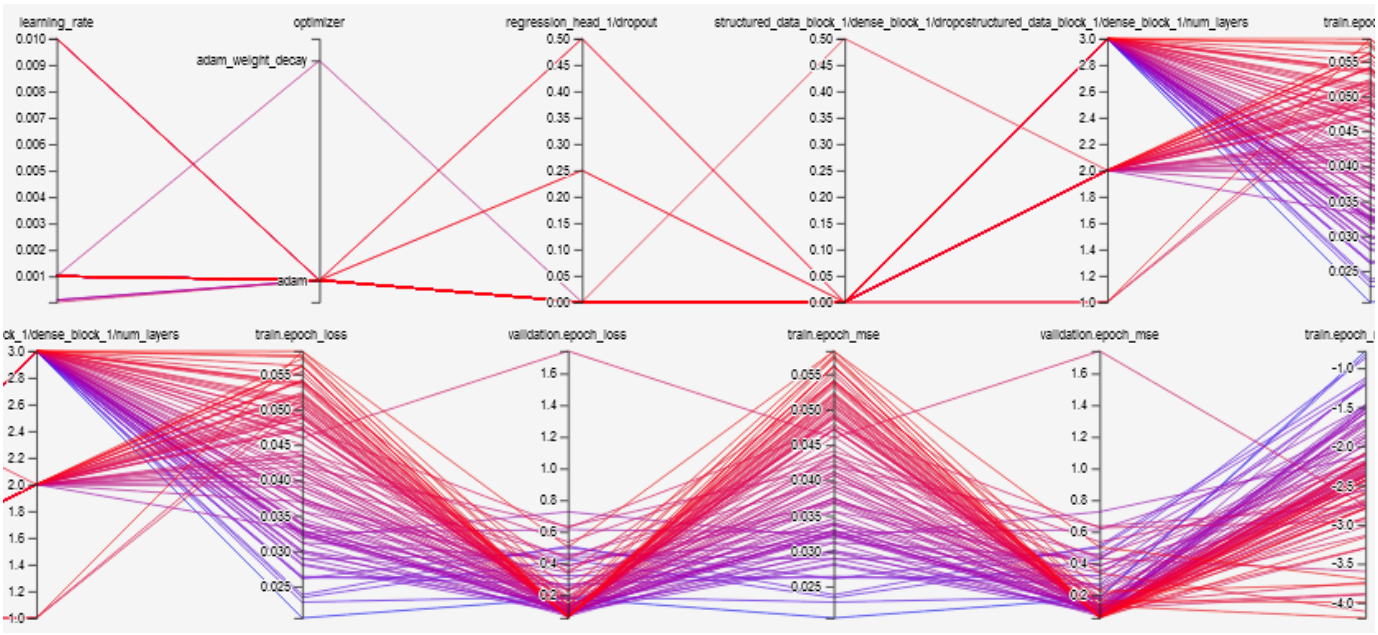

Fig. 3. AutoKeras training process on the D30S dataset.

TABLE II
BEST MODEL ARCHITECTURES

Dataset	Optimizer	Learning Rate	Decay	Loss	Input Layer	Dense Layers	Number Neurons	Activations Functions	Dropout Layers	Batch Layers
D30S	Adam	0.001	0.0	MSE	shape (0,6)	3	16,24,32	ReLu, ReLu, ReLu	0	3
D1M	Adam	0.001	0.0	MSE	shape (0,6)	3	32, 32, 16	ReLu, ReLu, ReLu	0	0
D5M	Adam	0.001	0.0	MSE	shape (0,6)	3	64, 16, 64	ReLu, ReLu, ReLu	0	0
D10M	Adam	0.001	0.0	MSE	shape (0,15)	2	32,32	ReLu, ReLu	0	2
D15M	Adam	0.001	0.0	MSE	shape (0,15)	3	512, 512, 128	ReLu, ReLu, ReLU	0	2
D30M	Adam	0.001	0.0	MSE	shape (0,15)	2	512, 256	ReLu, ReLu	0	2

MSEL = Mean Squared Error Loss

and Mean Squared Error Loss. The model results an RMSE score of 1,024, MSE of 1,498, R^2 of 17,341, and R^2 of 18,341.

Moving on to the 5m dataset, the optimizer learning rate and loss function stay the same as in the 1m dataset, The number of hidden layers is also the same, however the number of neurons changes to 64,16, and 64 respectively. The activation functions are the same and the usage of Batch Normalization Layers is also the same. The model results an RMSE score of 0,385, MSE of 0,206, R^2 of 12,334, and R^2 of 15,538.

The 10m dataset, consists of 2 hidden layers each with 32 neurons and ReLu activation functions. It has no dropout layers, but has two Batch Normalization layers. The loss is the same as in the previous models. The optimizer that best performed is Adam with a learning rate of 0.001 . The model results an RMSE score of 0,314, MSE of 0,162, R^2 of 10,080, and R^2 of 11,080.

The 15m dataset, consists of 3 hidden layers with 512, 512, and 128 neurons all of which have a ReLu activation function. It has no dropout layers, but has two Batch Normalization layers. The loss is the same as in the previous models. The optimizer that best performed is Adam with a learning rate of 0.001 . The model results an RMSE score of 0,368, MSE of 0,193, R^2 of 51,280, and R^2 of 54,513.

Finally, AutoKeras decided that the best architecture for the 30m dataset to have two hidden layers of 512, and 256 neurons with ReLu activation functions and 0 dropout layer. The loss function is the same like in previous experiments, and same goes for the optimizer with the learning rate. The model results an RMSE score of 0,232, MSE of 0,090, R^2 of 23,182, and R^2 of 24,183.

It is interesting to notice how the datasets that have longer time spans have better predictive capabilities. We believe this is due to the fact that longer time spans allowed for the other features to be present in the dataset, in other words the features that required a minimum of 5 minute recordings are not present in the 30s, 1m and 5m datasets which could be the reason that the rest of the datasets are better and generating models for prediction.

With that said following is a list of each dataset with its R^2 score:

- D30S - 47,710
- D1M - 17.341
- D5M - 12,334
- D10M - 10.080

- D15M - 51,280
- D30M - 23.181

The best result here is the D15M dataset where the R^2 is 51,280. The value is not ideal but the results can be improved. In order to make improvements more data would definitely help out, but also a longer training session would yield better results.

VI. Conclusion

To conclude, the architectural neural search capabilities of AutoKeras seem to differ greatly between datasets even though the correlations in some were stronger then the rest. There are however some common grounds, such as the loss function being the mean squared error, the Adam optimizer, the learning rate 0.001 and the activation function being ReLu for each dataset. We also notice that the maximum number of hidden layers does not exceed 3 layers and the number of neurons per layer is quite sporadic from 16 to 512.

The authors conclude that the research question is addressed and answered positively. Where the best architecture consists of three layers with 512,512, and 128 neurons, Adam optimizer, learning rate of 0.001, loss of MSE and two Batch Normalization Layers.

Regarding future endeavours, it would be interesting to see the comparison of manual DL-based experiments alongside a more controllable architectural search through the utilization of the Grid Search technique. Furthermore, the current results are results of a 24h training, thus a more longer training period might yield better architectures.

References

[1] M. Gusev, L. Poposka, G. Spasevski, M. Kostoska, B. Koteska, M. Simjanoska, N. Ackovska, A. Stojmenski, J. Tasic, and J. Trontelj, "Noninvasive glucose measurement using machine learning and neural network methods and correlation with heart rate variability," *Journal of Sensors*, vol. 2020, 2020.

[2] H. Wang and B. Raj, "On the origin of deep learning," *arXiv preprint arXiv:1702.07800*, 2017.

[3] O. Dürr, Y. Pauchard, D. Browarnik, R. Axthelm, and M. Loeser, "Deep learning on a raspberry pi for real time face recognition." in *Eurographics (Posters)*, 2015, pp. 11–12.

[4] Innovation Dooel, "A doctor in your pocket measure ECG & glucose levels with a small, non-invasive, wearable monitor," 2018. [Online]. Available: http://glyco.innovation.com.mk/

[5] E. Shaqiri and M. Gusev, "Deep learning method to estimate glucose level from heart rate variability," in *2020 28th Telecommunications Forum (TELFOR)*. IEEE, 2020, pp. 1–4.

[6] E. Shaqiri, M. Gusev, L. Poposka, and M. Vavlukis, "Correlating heart rate variability to glucose levels," in *International Conference on ICT Innovations, web proceedings*, 2020, pp. 1–10.

[7] G. Swapna, R. Vinayakumar, and K. Soman, "Diabetes detection using deep learning algorithms," *ICT Express*, vol. 4, no. 4, pp. 243–246, 2018.

[8] Y. Aggarwal, J. Das, P. M. Mazumder, R. Kumar, and R. K. Sinha, "Heart rate variability features from nonlinear cardiac dynamics in identification of diabetes using artificial neural network and support vector machine," *Biocybernetics and Biomedical Engineering*, 2020.

[9] M. Rahman, D. Islam, R. J. Mukti, and I. Saha, "A deep learning approach based on convolutional lstm for detecting diabetes," *Computational Biology and Chemistry*, vol. 88, p. 107329, 2020.

[10] B. Ballinger, J. Hsieh, A. Singh, N. Sohoni, J. Wang, G. Tison, G. Marcus, J. Sanchez, C. Maguire, J. Olgin *et al.*, "Deepheart: semi-supervised sequence learning for cardiovascular risk prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[11] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1946–1956.

[12] F. Chollet *et al.*, "keras," 2015.

[13] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.

[14] A. Botchkarev, "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology," *arXiv preprint arXiv:1809.03006*, 2018.