EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 5 No. 1 (2024) 267-275
https://publisher.uthm.edu.my/periodicals/index.php/eeee

Multivariate Modelling for Prediction of Time-Series Blood Glucose Level using Vector Autoregression (VAR)

Nurul Aliss Abdul Monir[1], Farhanahani Mahmud[2]*

[1] *Faculty of Electrical and Electronic Engineering, Universiti Tun Hussien Onn Malaysia, Batu Pahat, 86400, MALAYSIA*
[2] *Microelectronics and Nanotechnology Shamsuddin Research Centre (MiNT-SRC) Universiti Tun Hussien Onn Malaysia, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: farhanah@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2024.05.01.034

Article Info

Received: 14 January 2024
Accepted: 28 February 2024
Available online: 30 April 2024

Keywords

Blood glucose, VAR, Machine learning, RMSE, MAE, MSE

Abstract

Predicting glucose levels remains a significant challenge in diabetes management, with various factors influencing regulation. Modern technologies like AI and ML offer potential solutions by implementing prediction systems. This study focuses on utilising the Vector Autoregression (VAR) method to make accurate predictions, considering factors such as insulin and meal intake. Ten datasets, including blood glucose levels, carbohydrate intake, and insulin intake, were collected using MATLAB Simulink simulations. Python was then used to build predictive models with a 70:30 and 80:20 ratio for training and testing. The VAR model's prediction performance was evaluated using metrics like MAE, RMSE, and MSE. The 80:20 data split with binary insulin values yielded better results for blood glucose prediction, with MAE of 11.25808, RMSE of 12.36846, and MSE of 184.3054. This study offers insights into time series prediction of blood glucose using the VAR machine learning model, potentially enhancing diabetes care.

1. Introduction

The regulation of blood glucose levels is crucial for maintaining overall health, and the pancreas plays a key role by secreting hormones to control these levels [1]. Diabetes mellitus, a metabolic disorder, disrupts this balance, with Type 1 diabetes characterized by insufficient insulin secretion and Type 2 diabetes by a diminished response to insulin [2]. Managing diabetes involves careful monitoring of blood glucose levels, often requiring external insulin administration [3]. However, the complexity of insulin dosage adjustments and the risk of hypoglycemia pose significant challenges [4]. Continuous glucose monitoring (CGM) and machine learning techniques offer promising solutions [5]. While traditional methods involve manual adjustments and routine checks, machine learning algorithms, such as the Vector Autoregression (VAR) model, have shown effectiveness in real-time blood glucose prediction. By integrating various factors like insulin sensitivity, diet, and exercise into personalized algorithms, these models have the potential to revolutionize diabetes care, easing the burden of self-management and improving overall patient outcomes [6].

The global rise in diabetes diagnoses has underscored the need for innovative approaches in treatment. Machine learning, with its ability to adapt and improve through data-driven algorithms, is reshaping diabetes management [7]. The VAR model, particularly effective in multivariate time series analysis, is a promising tool for predicting blood glucose levels [3]. Integrating real-time data and predictions through closed-loop systems or CGM devices offers a more automated and precise insulin delivery mechanism [8]. By providing personalized

recommendations based on individual factors like insulin sensitivity and lifestyle habits, these models aim to enhance glycemic control, reduce the risk of hypoglycemia, and ultimately improve the quality of life for individuals with Type 1 diabetes. Blood glucose prediction models, driven by machine learning, emerge as critical tools in optimising treatment plans and empowering individuals to manage their condition more effectively.

2. Design and Methods

2.1 Study Framework and Diabetic Model in Simulation using an Open-Loop Insulin Delivery System

The study methodology as shown in Fig. 1 is the process flow of the work starts with a review of relevant literature to provide context and background for this study and to better comprehend the current state of knowledge and concepts in the field of machine learning. The blood glucose dataset was generated using a simulation of a type 1 diabetic model, specifically the Cobelli diabetes model, through MATLAB Simulink simulation. Cobelli's glucose-insulin model is approved by the Food and Drug Administration approved as an alternative to animal testing for glucose-insulin interactions. Following this, a cointegration test will be performed. Subsequently, the dataset will undergo preprocessing before being utilised for designing the machine learning model for time series blood glucose prediction.

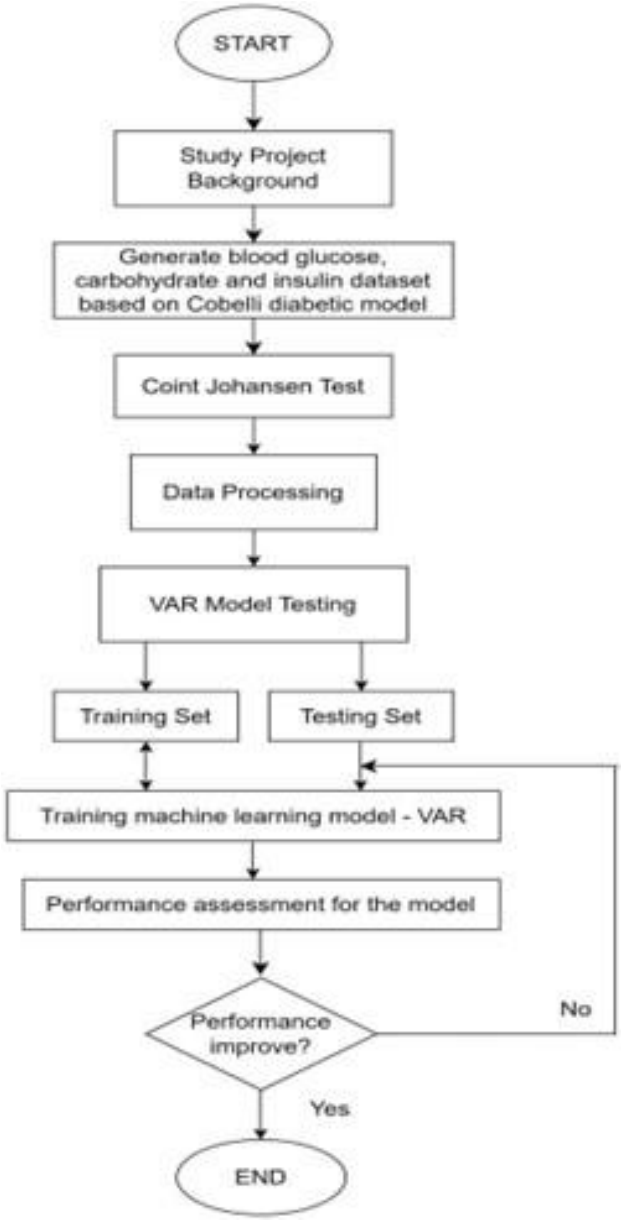


Fig. 1 *Process flow of the work*

Evolution in Electrical and Electronic Engineering Vol. 5 No. 1 (2024) p. 267-275

The variables are presumed stationary in VAR models, meaning their statistical characteristics do not alter over time. The variables are tested for stationary behavior using techniques like the Augmented Dickey-Fuller (ADF) test. If the variables are discovered to be nonstationary, stationarity can be induced using differencing or other modification techniques. Then, the data will split into training and testing sets to train and evaluate the machine learning model. This training and testing set is carried out to assess its performance to ensure that it meets the required criteria before reaching the end of the flowchart process. To develop a time- series blood glucose prediction model for this work, the VAR method will be used. The data will be cleaned and analysed to determine the optimal values where Yt represents a vector of variables at time t, α is the intercept, a constant term, and β1, β2 until βp are the coefficients of the lags of Y until the p order. Then, the VAR model will be trained on the data and utilized to predict future blood glucose values using multivariate models. The model's performance will be evaluated through regression performance metrics such as MAE, MSE, and RMSE.

To obtain the required data, the Simulink block diagram feature within the MATLAB Simulink software has been utilized. Fig. 2 depicts the comprehensive Simulink model that is created using MATLAB to simulate the model in an open-loop configuration. The simulated system model encompasses a duration of 24 hours, which is equivalent to 1440 minutes. Numerous blocks are employed within the MATLAB Simulink model, and the diabetic block is specifically constructed utilizing the S-function block to represent the mathematical equations of the model. These equations are subsequently solved using the MATLAB ODE solver function ode 45. The main block involved in this system is a virtual diabetic model, a mathematical model that explains the dynamics of blood glucose levels. This block is represented by this block, an insulin input block based on the selected dosing regimen. This block determines the profile of insulin administration; for the meal disturbance input block, this block and blood glucose plot simulate the impact of meals or carbohydrate intake on blood sugar levels represented in the graph.

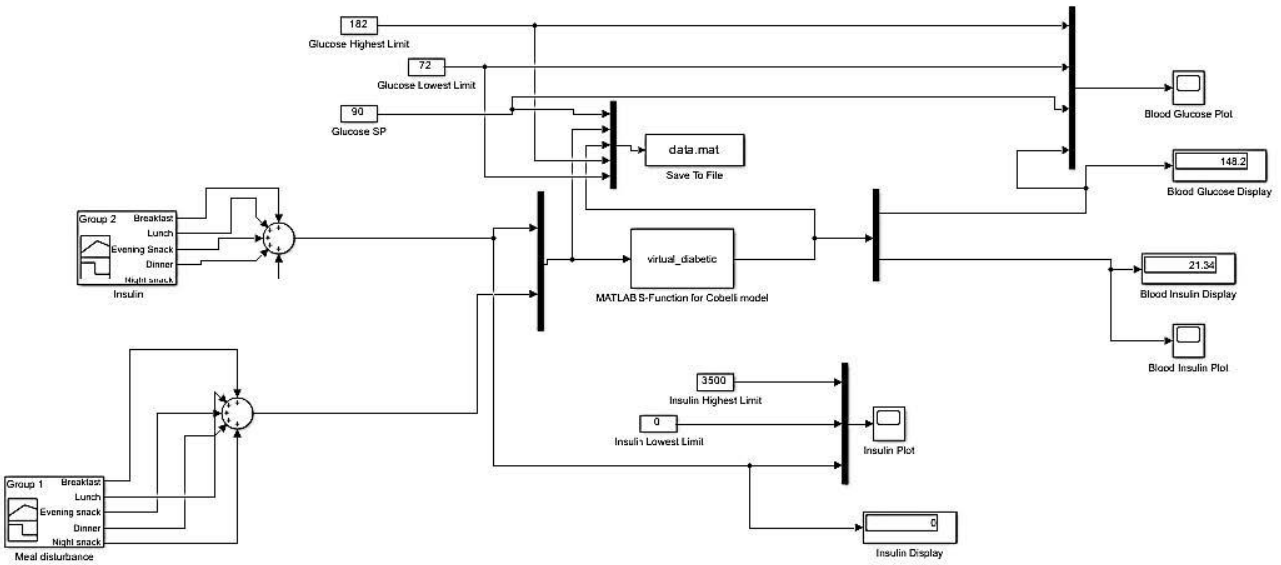


Fig. 2 *Open-Loop insulin delivery system for MATLAB Simulink design*

2.2 Vector Autoregression (VAR)

An approach for time series forecasting called vector autoregression (VAR) expands autoregression (AR) to include multiple variables. Each variable in a VAR is represented as a linear combination of its historical values and the historical values of every other variable in the system. It is extensively employed in finance and econometrics to simulate the dynamic interdependencies between several time series.

2.2.1 Importing Libraries and Coing Integration Test

Google Colab comes pre-loaded with libraries, eliminating the need for extra installations. Importing necessary libraries at the beginning ensures access to required functions and classes, making coding easier. Google Drive authentication is essential for data access, followed by uploading datasets to it. Importing libraries like pandas, numpy, matplotlib, and statsmodels facilitates data processing, numerical calculations, and statistical operations. Carbohydrate and insulin data in actual and binary values are used for VAR predictive model analysis. This allows comprehensive examination and validation under different conditions. Binary values represent presence (1) or absence (0) of insulin or carbohydrate intake in the datasets.

The Python code utilises the coint_johansen function from statsmodels.tsa.vector_ar.vecm to perform Johansen's Cointegration Test. This function accepts a panda DataFrame of time series variables and conducts the

test, identifying significant columns at a specified significance level. Test statistics and critical values are extracted, and significant columns are determined by comparing these values. The function then returns a list of significant columns, which are retained, while non-significant columns are dropped from the dataset.

2.2.2 VAR Model Testing and Data Processing, and Dataset Splitting for Training and Testing

Initially, before employing a Vector Autoregression (VAR) model, confirming the stationarity of time series variables is crucial. Stationary series are easier to predict and lead to more accurate predictions. Time series analysis validity relies on stationarity, and deviating from this assumption can result in unreliable data and inaccurate model predictions. The Augmented Dickey-Fuller (ADF) test is commonly used to verify stationarity by evaluating the presence of unit roots in variables. The p-value from the ADF test is compared to a significance level (e.g., 0.05) to decide whether to reject the null hypothesis [1]. Insulin appears stationary based on the ADF test, while blood glucose requires further investigation or differencing to achieve stationarity, as indicated by its p-value nearing the significance level. Once the VAR model has been developed and trained, the dataset is divided into training and testing sets.

The testing set is a subset used to assess the model's predicted performance on untested data. To make it easier to split a dataset into training and testing sets for later uses, such training and assessing a model, the provided code creates the split_and_display function. The function for dataset splitting requires two parameters: "nobs" indicating the number of observations for the testing set, and "data" representing the dataset. The dataset was divided into 70:30 and 80:20 ratios for ease of comparison. Subsequently, the VAR model will be developed using the training data frames, and its performance will be tested and evaluated.

2.2.3 Interpretation, Lag Order Selection, Training and Forecasting with Python VAR Model using Statsmodels

Before training the VAR predictive model, analysis of the Vector Autoregression (VAR) model was conducted using key metrics such as Bayesian Information Criterion (BIC), Hannan-Quinn Information Criterion (HQIC), Akaike Information Criterion (AIC), log likelihood value, Final Prediction Error (FPE), and asymptotic covariance matrix (Det(Omega_mle)). Additionally, determination of the optimal lag order, representing the historical time points considered for each variable, was necessary.

Using the VAR model, future values were forecasted based on recent lagged observations to find the optimal lag order. The provided Python code segment employs the VAR class from statsmodels.tsa.api to train the VAR model on train_data with the optimal lag order. It generates forecasts for both training and test datasets, displaying a summary of the model's key information. Forecasted values are presented for both datasets, underscoring the VAR model's utility in time series forecasting, especially with interrelated variables. An inverse transformation procedure is then applied to return predicted values to their original scale. Subsequently, predicted and actual values are compared to evaluate forecast accuracy, employing metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE) for variables like glucose and insulin. Then, the code defines a function, calculate_accuracy, to compute accuracy metrics for the VAR model. This function takes the fitted model, forecasted values, and actual data as inputs and returns accuracy metrics for 'Glucose' and 'Insulin'. The function is applied to both training and test sets, with the results printed, providing insights into the VAR model's forecasting accuracy for 'Glucose' and 'Insulin' variables in each dataset.

2.2.4 Durbin Watson Test

This analysis evaluates autocorrelation presence and direction in residuals for each variable in both training and test datasets, aiding in VAR model adequacy assessment and autocorrelation issue identification. The Durbin-Watson statistic is computed for each variable's residuals in VAR models fitted to training (train_data) and test (test_data) datasets. Utilising the durbin_watson function from statsmodels.stats.stattools, this statistic measures autocorrelation in regression model residuals, ranging between 0 and 4. A value near 2 indicates no autocorrelation, while values significantly less or greater than 2 suggest positive or negative autocorrelation, respectively. Residuals of VAR models are stored as residuals_train and residuals_test for training and test datasets, respectively. The code iterates through each dataset column, computes the Durbin-Watson statistic for each variable's residuals, and prints the results.

Evolution in Electrical and Electronic Engineering Vol. 5 No. 1 (2024) p. 267-275

3. Results and Discussion

3.1 Cobelli Model Simulation Results

The Cobelli model is a 24-hour simulation of an open-loop glucose-insulin regulation system, depicting the management of type-1 diabetes patients with insulin therapy. The model considers five meals with varying carbohydrate values and consumption rates, yielding results for blood insulin, glucose levels, and insulin. The simulation is initiated by configuring insulin input and meal disturbance parameters. It generates three types of graphs: blood insulin, insulin administration, and glucose concentration. Fig. 3 shows the glucose concentration with hyperglycemia, indicated by the red line's upper boundary, reflects elevated blood glucose due to insufficient insulin. Conversely, hypoglycemia, denoted by the blue line's lower boundary, signifies blood glucose levels below the normal range. The black line represents the Set Point value, aiming to maintain blood glucose levels within the desired range.

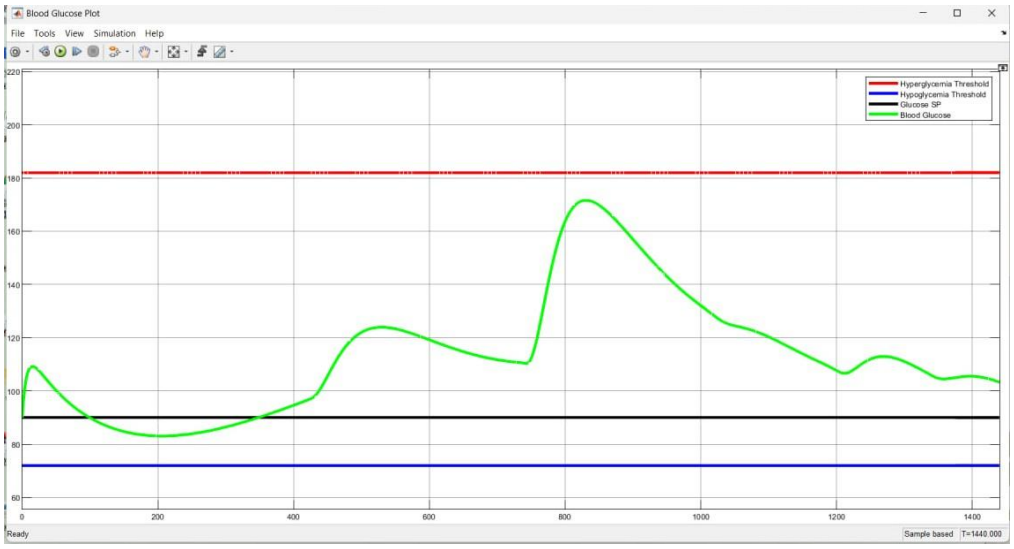


Fig. 3 *Simulation results of a normal blood glucose level*

3.2 Machine Learning Modelling using VAR

3.2.1 Interpreting VAR Model

In this study, a train-test split ratio of 70:30 and 80:20 is utilised for dataset partitioning, enabling comparative analysis of outcomes. Fig. 4 shows the VAR model summary. These ratios are well-established, facilitating evaluation of model performance across different training and testing proportions. Analysis of the Vector Autoregression (VAR) model using the 70:30 ratio reveals robustness, with 28,772 observations and key metrics like BIC, HQIC, and AIC indicating good performance. A log likelihood value of 308,946 signifies strong fit, while low FPE and Det(Omega_mle) values underscore prediction accuracy and model reliability. Regarding lag order parameter selection, coefficients reflect past values impact on current states, with significant contributions from Insulin to its current values. Optimal lag order at 29 highlights past 29 periods relevance in predictions, warranting ongoing diagnostics for model adequacy and reliability as shown in Fig. 5.

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Tue, 23, Jan, 2024
Time:                     13:38:36
--------------------------------------------------------------------
No. of Equations:         2.00000    BIC:                   -27.1091
Nobs:                     28772.0    HQIC:                  -27.1321
Log likelihood:           308946.    FPE:                1.62905e-12
AIC:                     -27.1430    Det(Omega_mle):     1.62239e-12
```

Fig. 4 *VAR model summary*

L25.Glucose	-118325.501557	NAN	NAN	NAN
L25.Insulin	-0.000004	0.008321	-0.001	1.000
L26.Glucose	119446.466182	NAN	NAN	NAN
L26.Insulin	-0.000026	0.008321	-0.003	0.997
L27.Glucose	-94937.389909	5705.082328	-16.641	0.000
L27.Insulin	0.000027	0.008321	0.003	0.997
L28.Glucose	45530.139292	8756.034568	5.200	0.000
L28.Insulin	-0.000024	0.008321	-0.003	0.998
L29.Glucose	-9573.124303	13311.371251	-0.719	0.472
L29.Insulin	-0.005933	0.005901	-1.005	0.315

```
The optimal lag order is: 29
```

Fig. 5 *Results on the lag order parameter selection*

3.2.2 Prediction Values

The results of predicting blood glucose levels and insulin intake using a Vector Autoregression (VAR) model with 70:30 and 80:20 training and testing data ratios based on actual insulin values are presented here. The dataset comprises real-world continuous observations, capturing the complexities of the phenomenon. Table 1 presents the graphical representations compare actual and predicted values for the glucose and the insulin with the actual values, illustrating trends for both training and testing datasets. The graph portrays the trends actual glucose levels (depicted by the blue line), predicted glucose values for training set (illustrated in red line) and predicted glucose values for the testing se (represented by the purple line, as well as actual insulin levels (depicted by the green line), predicted insulin values for training set (illustrated by the yellow line) and predicted insulin values for testing set (represented by the dark blue line). The analysis underscores the model's predictive performance on real-world datasets, revealing discrepancies between actual and predicted glucose levels, potentially attributed to the model's inability to precisely capture glucose trends.

Table 1 *Graphs for prediction and actual data for 70:30 and 80:20 ratio with the actual insulin values*

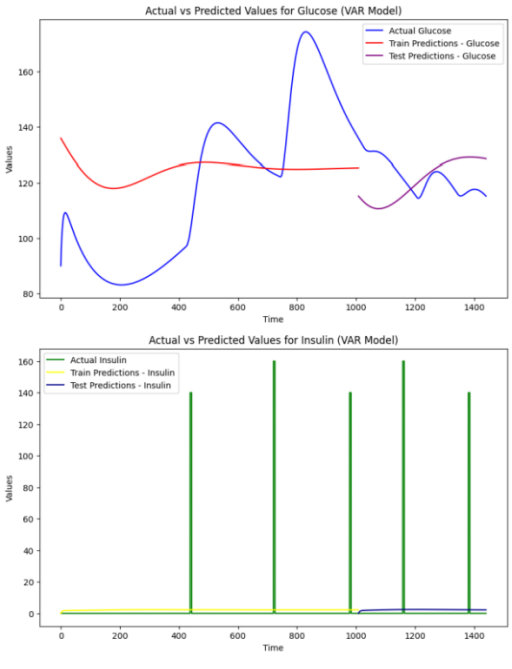
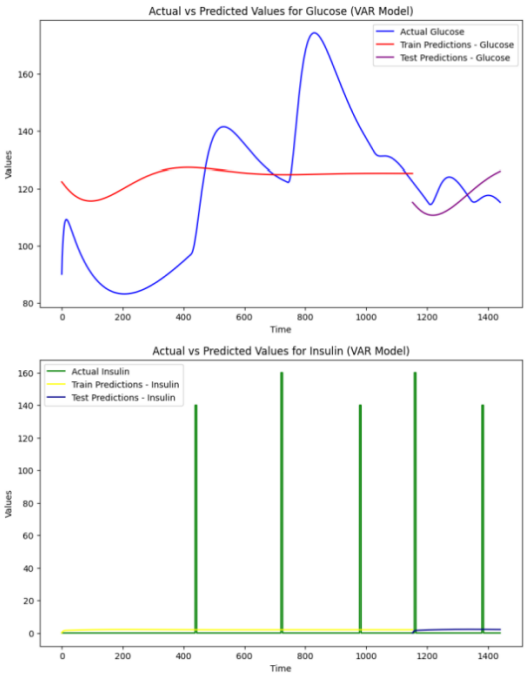
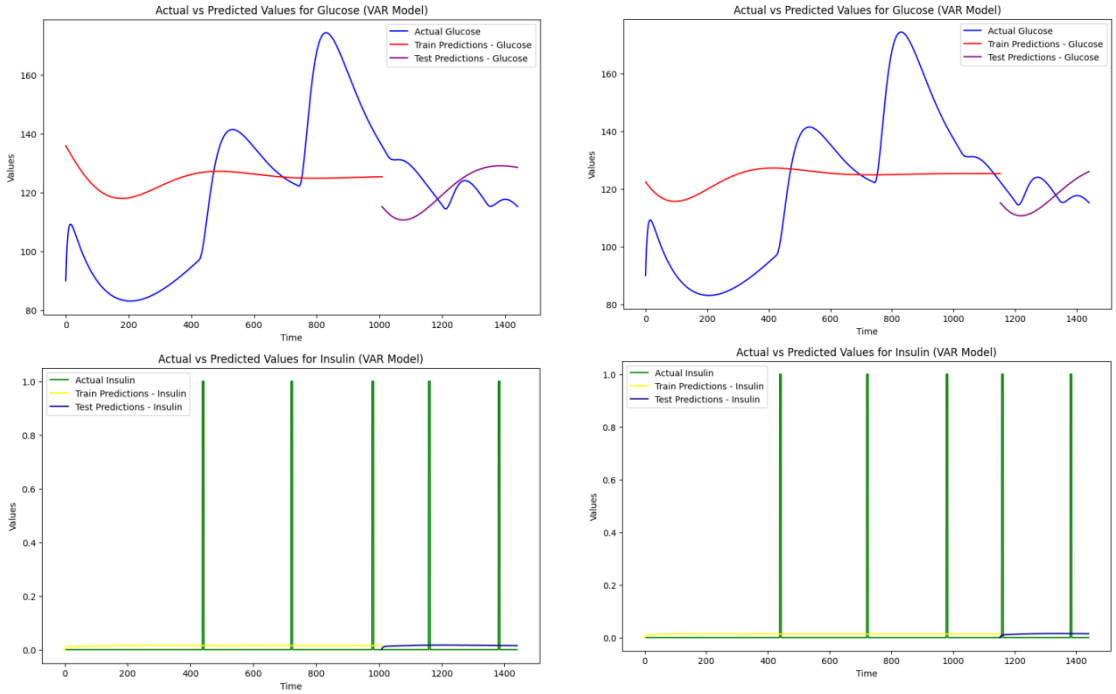
Dataset	70:30 ratio	80:20 ratio
Data 1		

Table 2 presents results of predicting blood glucose levels and insulin intake for 70:30 and 80:20 training and testing data ratios, using binary-transformed insulin values for comparison. Graphs depicting glucose and insulin differ from those of the standard method, reflecting binary insulin representation. Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) outcomes were compared with those from the standard method. Binary transformation allows distinct evaluation of data representation impact on both

graphs and performance metrics, providing insights into analytical outcomes. Overall, the model's performance indicates inaccuracies in predicting glucose levels as observed in Table 2.

Table 2 *Graph for prediction and actual data for 70:30 and 80:20 ratio with the binary insulin values*

Dataset	70:30 ratio	80:20 ratio
Data 1		

3.2.3 VAR Performance Evaluation

The Performance evaluation of machine learning models, crucial for effectiveness, involves data splitting to prevent overfitting, especially in Vector Autoregression (VAR) models. Key metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) quantify model accuracy by comparing predictions with actual values. MSE measures average squared differences, MAE calculates average absolute differences, and RMSE combines these while considering squared values. Discrepancies in MAE and MSE values may stem from factors beyond graph stability, including data scale, outliers, error distribution, model sensitivity, and data quality. Comparing actual and binary datasets, evaluation tables (3,4 for 70:30 ratios; 5,6 for 80:20 ratios) reveal superior performance of the binary dataset in predicting blood glucose levels using VAR models. Consistently lower MAE, RMSE, and MSE values are observed for the binary dataset across both training and testing sets. The 80:20 ratio generally outperforms the 70:30 ratio, indicating its superiority in predictive accuracy for this analysis.

Table 3 *Values of MAE, RMSE, and MSE for blood glucose level prediction of 70:30 ratio with actual insulin values*

Dataset	**MAE**		**RMSE**		**MSE**	
	Train	Test	Train	Test	Train	Test
1	24.36264	10.50467	28.01810	12.32716	785.01404	151.95894
2	26.17147	8.55065	30.29566	10.67967	917.82718	114.05546
3	21.28252	14.63334	25.55630	16.04498	653.12458	257.44143
4	24.59856	14.19516	28.98936	16.70034	840.38326	278.90147
6	39.17604	13.95423	45.31954	14.37934	2053.8607	206.76547
7	33.69495	17.51296	38.34486	21.19715	1470.3287	449.31924
8	26.87649	11.68037	31.31135	14.57714	980.40071	212.49319
9	35.30920	26.57007	41.26480	32.96705	1702.7840	1086.8267
10	42.77852	24.27208	47.93155	24.46536	2297.4340	598.55418
Mean	30.47227	15.76373	35.22572	18.14869	1300.12857	372.92401

The analysis of results from Table 3, Table 4, Table 5 and Table 6 highlights that predictions from the binary dataset consistently outperform those from the actual dataset. Transforming insulin values into a binary format led to the improved model accuracy. Across metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE), the binary dataset consistently shows lower error values, indicating enhanced predictive capabilities. This suggests a better balance between fitting training data and generalizing to unseen data with the binary dataset. Overall, employing the binary representation for insulin values improves accuracy and efficiency in the prediction model.

Table 4 *Values of MAE, RMSE, and MSE for blood glucose levels prediction of 70:30 ratio with binary insulin values*

Dataset	MAE		RMSE		MSE	
	Train	Test	Train	Test	Train	Test
1	24.36278	10.50521	28.01843	12.32741	785.03259	151.96527
2	26.17156	8.55066	30.29562	10.67975	917.82478	114.05711
3	21.28272	14.63376	25.55637	16.04537	653.12835	257.45407
4	24.59846	14.19415	28.98923	16.69973	840.37556	278.88119
6	39.17552	13.95569	45.31957	14.38070	2053.86370	206.80455
7	33.69452	17.51220	38.34474	21.19685	1470.31957	449.30678
8	26.87782	11.68136	31.31188	14.57751	980.43442	212.50389
9	35.30651	26.57453	41.26344	32.97200	1702.67185	1087.15303
10	42.77961	24.27020	47.93189	24.46347	2297.46659	598.46152
Mean	30.47217	15.7642	35.22569	18.1492	1300.12416	372.95416

Table 5 *Values of MAE, RMSE, and MSE for blood glucose levels prediction of 80:20 ratio with actual insulin values*

Dataset	MAE		RMSE		MSE	
	Train	Test	Train	Test	Train	Test
1	21.45182	6.55221	26.02893	7.10703	677.5055	50.50998
2	23.36802	5.59965	28.09522	7.06475	789.3418	49.91073
3	18.80541	9.40050	24.07436	10.48447	579.5750	109.92423
4	21.65390	10.66852	26.97644	11.53813	727.7287	133.12860
6	36.88045	9.85641	43.17460	11.56464	1864.0461	133.74090
7	29.88284	12.45492	35.43472	13.45941	1255.6196	181.15586
8	24.21584	7.68065	29.23370	8.20196	854.6092	67.27227
9	35.69185	24.11729	41.14647	26.20514	1693.0326	686.70954
10	42.59234	14.98205	46.66417	15.67784	2177.5452	245.79467
Mean	28.2825	11.25691	33.4254	12.36704	1179.8893	184.23853

Table 6 *Values of MAE, RMSE, and MSE for blood glucose levels prediction of 80:20 ratio with binary insulin values*

Dataset	MAE		RMSE		MSE	
	Train	Test	Train	Test	Train	Test
1	21.45201	6.55178	26.02868	7.10676	677.49237	50.50614
2	23.36701	5.59798	28.09558	7.06330	789.36195	49.89033
3	18.80529	9.40017	24.07424	10.48392	579.56908	109.91275
4	21.65379	10.66901	26.97665	11.53871	727.73974	133.14188
6	36.88061	9.85622	43.17457	11.56453	1864.04417	133.73840
7	29.88340	12.45539	35.43489	13.45998	1255.63188	181.17122
8	181.17122	7.68131	29.23393	8.20254	854.62274	67.28171
9	35.68855	24.12440	41.14307	26.21378	1692.75264	687.16237
10	42.59021	14.98642	46.66268	15.68259	2177.40570	245.94379
Mean	45.72134	11.25808	33.42492	12.36846	1179.8467	184.3054

4. Conclusion

In conclusion, the research on "Multivariate Modelling for Prediction of Time-Series Blood Glucose Level Using Vector Autoregression (VAR)" systematically addressed its objectives, beginning with the analysis of blood glucose data using the Cobelli model, providing insights into blood glucose dynamics. The developed Vector Autoregression (VAR) machine learning model demonstrated robust performance in predicting blood glucose levels and insulin intake. Evaluation metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) consistently favored the binary representation of insulin values. The preference

for the 80:20 ratio over 70:30 underscored the significance of data splitting for optimal accuracy. Despite moderate performance, the study highlights the importance of VAR modeling and appropriate data representation in predicting blood glucose levels in individuals with Type 1 diabetes.

Acknowledgement

This research is supported by Universiti Tun Hussein Onn Malaysia (UTHM) through Tier 1 Grant Research (Q361).

Conflict of Interest

The authors declare that there is no conflict of interest regarding the completion of the paper.

Author Contribution

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] Alhaddad, A. Y., H. Aly, H. Gad, A. Al-Ali, K. K. Sadasivuni, J.-J. Cabibihan and R. A. Malik (2022). "Sense and Learn: Recent Advances in Wearable Sensing and Machine Learning for Blood Glucose Monitoring and Trend-Detection." Frontiers in Bioengineering and Biotechnology **10**, https://doi.org/10.3389/fbioe.2022.876672

[2] Woldaregay, A. Z., E. Årsand, S. Walderhaug, D. Albers, L. Mamykina, T. Botsis and G. Hartvigsen (2019). "Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes." Artificial Intelligence in Medicine **98**: 109-134, https://doi.org/10.1016/j.artmed.2019.07.007

[3] Doike, T., K. Hayashi, S. Arata, K. N. Mohammad, A. Kobayashi and K. Niitsu (2018). A Blood Glucose Level Prediction System Using Machine Learning Based on Recurrent Neural Network for Hypoglycemia Prevention. 2018 16th IEEE International New Circuits and Systems Conference (NEWCAS), https://doi.org/10.1109/NEWCAS.2018.8585468

[4] Janez, A., C. Guja, A. Mitrakou, N. Lalic, T. Tankova, L. Czupryniak, A. G. Tabak, M. Prazny, E. Martinka and L. Smircic-Duvnjak (2020). "Insulin Therapy in Adults with Type 1 Diabetes Mellitus: a Narrative Review." Diabetes Ther **11**(2): 387-409, https://doi.org/10.1007/s13300-019-00743-7

[5] Tang, B., Y. Yuan, J. Yang, L. Qiu, S. Zhang and J. Shi (2022). "Predicting Blood Glucose Concentration after Short-Acting Insulin Injection Using Discontinuous Injection Records." Sensors (Basel) **22**(21), https://doi.org/10.3390%2Fs22218454

[6] Xie, J. and Q. Wang (2020). "Benchmarking Machine Learning Algorithms on Blood Glucose Prediction for Type I Diabetes in Comparison With Classical Time-Series Models." IEEE Transactions on Biomedical Engineering **67**(11): 3101-3124, https://doi.org/10.1109/tbme.2020.2975959

[7] Hassan, M. M., M. A. Mamun Billah, M. M. Rahman, S. Zaman, M. M. Hasan Shakil and J. H. Angon (2021). Early Predictive Analytics in Healthcare for Diabetes Prediction Using Machine Learning Approach. 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)**:** 01-05, https://doi.org/10.1109/ICCCNT51525.2021.9579799

[8] Afsaneh, E., A. Sharifdini, H. Ghazzaghi and M. Z. Ghobadi (2022). "Recent applications of machine learning and deep learning models in the prediction, diagnosis, and management of diabetes: a comprehensive review." Diabetol Metab Syndr **14**(1): 196, https://doi.org/10.1186/s13098-022-00969-9