Cumhuriyet Sci. J., 45(4) (2024) 769-776
DOI: https://doi.org/10.17776/csj.1441313

Cumhuriyet Science Journal

| csj.cumhuriyet.edu.tr | Founded: 2002 ISSN: 2587-2680 e-ISSN: 2587-246X Publisher: Sivas Cumhuriyet University

Analyzing Diabetic Dynamics with MRK4, and LSTM Techniques with Multiplicative Calculus

Buğçe Eminağa Tatlıcıoğlu [1,a,*]

[1] *Department of Computer Engineering, Faculty of Engineering, Final International University, Kyrenia, Northern Cyprus, via Mersin 10, Türkiye.*

**Corresponding author*

Research Article

History
Received: 22/02/2024
Accepted: 24/10/2024

ABSTRACT

This study compares the use of Long Short-Term Memory (LSTM) networks for predictive modeling with multiplicative calculus. We evaluate and quantitatively analyze both methodologies to determine their prediction performance. While LSTM networks are investigated for them power to learn and generalize patterns, the multiplicative calculus technique is analyzed for its ability to grasp complex connections within the data. This study attempts to shed light on the efficacy of each approach by carefully analyzing error measures including mean squared error (MSE), root mean square error (RMSE), and mean absolute percentage error (MAPE). The results aid in the comprehending of the subtleties related to LSTM networks and multiplicative calculus, assisting practitioners and researchers in choosing the best method for tasks involving predictive modeling.

Keywords: Neural network, Runge kutta, Multiplicative calculus, Long Short-Term Memory, Dabetes.

[a] *bugce.tatlicioglu@final.edu.tr* *https://orcid.org/0000-0001-8854-4464*

Introduction

The utilization of big data in computational biology research is on the rise due to the quick and practical creation of many data due to improvements in high-throughput computing and biotechnology. The main objective is to analyze the growing corpus of biological data and offer a basis for tackling significant biological and medical problems. Although these techniques can precisely identify patterns and create models from data, they are dependable and efficient. One important use of this study is in the detection and treatment of life-threatening conditions, such as diabetes mellitus (DM). [1-3].

Diabetes is a well-known and serious issue that affects both industrialized and developing nations in the modern world [4]. The hormone responsible for facilitating the body's absorption of glucose from diet is insulin. This deficit, which is often brought on by pancreatic dysfunction, can lead to several significant symptoms, including coma, renal [5], heart disease [6], retinal failure, and cardiovascular problems [7]. Research has shown that the number of adult (18 and older) cases of diabetes increased significantly between 1980 and 2014 [1], and by 2045, more cases are expected. Global estimates of diabetes patients in 2017 were 451 million; by 2045, that number is expected to climb to 693 million [8].

An extra statistical analysis [6] indicates that approximately 500 million individuals globally suffer from diabetes, highlighting the severity of the issue. Furthermore, the study projects a notable rise in diabetes prevalence, with estimations of 25% by 2030 and 51% by 2045. However, there is still a significant need for research focused on improving the health and quality of life (QoL) of persons with diabetes, as well as reducing the start of disease complications and premature mortality. It's critical to remember that, even in the absence of long-term treatment, effective management and prevention are still achievable, particularly in the case that accurate and early forecasts can be established.

Recent technological developments, particularly the use of artificial intelligence methods, have shown to be extremely advantageous for the healthcare sector. The literature offers a variety of methods and approaches to enhance diabetes accuracy. In recent years, several of diabetes prediction techniques have been developed and published. A framework based on machine learning was introduced in [9]. Numerous classification algorithms were employed by the authors, including Naive Bayes (NB) [10], Support Vector Machine (SVM) [11], AdaBoost (AB) [12], Decision Tree (DT) [13], Random Forest (RF) [14], Logistic Regression (LR) [15], Gaussian Process Classification (GPC) [10] and Artificial Neural Network (ANN) [11]. In addition, a logistic regression model was presented by Qawqzeh et al. [15] for the categorization of type 2 diabetes, and it achieved an impressive 92% accuracy.

Nevertheless, a comparison study with known procedures was absent from their research. Further research that focused on the effectiveness of linear support vector machines (SVMs) also used SVMs to classify individuals with diabetes mellitus. However, this study only included scant information on parameter selection and lacked a comprehensive comparison with state-of-the-art technology [16]. Furthermore, although the study did not specify accuracy standards, the PIMA

Indian Diabetes Study was used as the basis for the investigation of the categorization of diabetes using naïve Bayes (NB) and Support Vector Machines (SVMs). Numerous additional studies have used a variety of machine learning methods and datasets to predict diabetes; some of these studies have produced encouraging results. Some of these research, too, lacked direct comparisons with cutting-edge techniques [17, 18].

Furthermore, a thorough assessment of machine learning algorithms for diabetes prediction from 2010 to 2019 was carried out by Hussain et al. (2021). Based on the Matthews correlation coefficient, they discovered that Random Forests and naïve Bayes performed better overall [19].

Complex-valued function solutions involving complex variables have been addressed by Bashirov and Riza [20, 21] and Uzer [22] through the expansion of geometric (multiplicative) calculus. Differential equation modeling has become a part of this discipline [23]. Due to its extensive applications in fields including applied and pure mathematics, engineering [24], applied mechanics, quantum physics, analytical chemistry, astronomy, and biology, nonlinear equations have attracted a lot of attention [24, 25]. Aniszewska et al. [26] investigated an alternate use of the Runge-Kutta technique in dynamical systems, and [27] created the bigeometric variant of the Runge-Kutta method. For approximating solutions to ordinary differential equations, temporal discretization makes considerable use of this approach, which belongs to an important family of implicit and explicit iterative methods in numerical analysis [28]. The Runge-Kutta family includes the popular RK4 technique, which is well-known in numerical analysis for its efficiency in approximating solutions to ordinary differential equations.

LSTM networks have been increasingly explored in the medical field for predicting and managing various diseases beyond diabetes. Researchers have applied LSTMs in the prediction and diagnosis of complex conditions such as heart disease [29], Alzheimer's [30], and skin diseases [31], where temporal patterns in patient data play a crucial role.

Proposed Method

According to this notion, individuals with diabetes are split into two categories based on their current health status: group C is for individuals with challenges, and group D is for those without issues. The paradigm treats acute and chronic disorders in the same way, without distinguishing between various types of consequences. Initially, individuals without difficulties in the diabetic population can advance to the complications group at the pace indicated by the symbol λ. This transition results in a reduction in group D's population by λ times the current number, while group C's population increases proportionately. For individuals with diabetes-related problems, three probable outcomes are taken into account: disability, death, or recovery. It is assumed that those who have recovered would remain in the diabetic category and keep their diabetes status. According to the diabetic complications model (DC), people with diabetes who encounter difficulties may be able to overcome them, pass away from them, or become disabled. It's noteworthy to notice that people who recover from diabetes are still considered to be diabetics, even if those who die or become crippled are no longer considered to be part of the community. The variables γ, δ, and ν, in that order, indicate the rates of recovery, disability, and mortality due to complications.

The number of people in the complications section fluctuates as a result. The number of people experiencing difficulties connected to their recovery, the number of fatalities, and the number of people with impairments all drop by γ times, δ times, and ν times, respectively, in the compartment. Patients with uncomplicated diabetes are becoming more prevalent at the same time as twice as many people are recovering from complications.

Natural fatalities that occur in both compartments are also taken into consideration by the model; in this case, ν represents the mortality rates. Natural mortality reduces the population by ν times the current number in the compartment (C) for complications and by ν times the current number in compartment (D) for simple diabetes.

Furthermore, it is predicted that the number of people with simple diabetes will rise along with the total incidence of diabetes, which is indicated by the letter.

A proposed formulation of this DC model is as follows, according to [29]. Give the mathematical equations or expressions that match the formulation of the model as it appears in the cited publication. For a more in-depth explanation, you can insert any specific equations or expressions from the text here.

$$\frac{dD}{dt} = I - (\lambda - \nu)D + \gamma C \tag{1}$$

$$\frac{dC}{dt} = \lambda D - (\gamma + \delta + \nu + \mu)C \tag{2}$$

where > 0 and λ, γ, μ, δ, and ν > 0.

Two different approaches were used in the examination of the DC model: long short-term memory (LSTM) technology and multiplicative Runge-Kutta. These techniques were evaluated and contrasted for their efficacy in modeling the dynamics of the comorbid diabetic population.

Multiplicative Runge Kutta

The fact that multiplicative calculus is limited to positive-valued functions of real variables is one of its drawbacks. Complex multiplicative calculus can effectively address this constraint, though. Uzer's groundbreaking work [22] provided an initial introduction to complex multiplicative calculus, which was further developed by Bashirov and Riza's in-depth mathematical analyses in [20] and [32]. The key motivation for expanding to the complex domain is the knowledge that the derivative is a local property. Functions can be transformed into complex-valued

functions of real variables to escape the restriction to positive values. This transformation also eliminates the need for the Cauchy-Riemann criteria, allowing the real and imaginary components to be clearly separated from one another.

$$\lim_{h\to 0}\frac{f(x+h)^{1/h}}{f(x)} \tag{3}$$

This definition becomes crucial since f (x) is a positive function on the interval A and has a well-defined derivative at x. In these cases, the relationship between the multiplicative and classical derivatives can be expressed as follows:

$$f^*(x) = exp\left(\frac{f'(x)}{f(x)}\right) \tag{4}$$

where (ln ◦f)(x) = ln f (x).

Some basic rules of differentiation are:

$$\begin{aligned}
&(cf)^*(x) = f^*(x)\\
&(fg)^*(x) = f^*(x)g^*(x)\\
&\left(\frac{f}{g}\right)(x) = \frac{f^*(x)}{g^*(x)}\\
&\left(f^h\right)^* = f^*(x)^{h(x)}\cdot f(x)^{h'(x)}\\
&\left(foh\right)^*(x) = f^*\left(h(x)\right)^{h'(x)}
\end{aligned} \tag{5}$$

The multiplicative Runge-Kutta techniques are applied in the above context, where c is a positive constant and functions f, g, and h are differentiable, to estimate solutions to the multiplicative differential equations of the following form:

$$y^*(x) = f(x,y), \quad y(x_0) = y_0 \tag{6}$$

An equivalent derivation of the ordinary Runge-Kutta (RK) techniques may be made for the multiplicative Runge-Kutta (MRK) methods. In particular, the fourth-order Runge-Kutta technique (RK4) is frequently used because it strikes a compromise between accuracy and simplicity. The Runge-Multiplicative obtained in [33]

$$\begin{aligned}
&y(x+h) = y(x).f_0^{h/6}.f_1^{h/3}.f_2^{h/3}.f_3^{h/6}\\
&f_0 = f(t,y)\\
&f_1 = f\left(t+\frac{h}{2}, yf_0^{h/2}\right) \qquad (7)\\
&f_2 = f\left(t+\frac{h}{2}, yf_1^{h/2}\right) \qquad (8)\\
&f_3 = f\left(t+h, yf_2^{h}\right) \qquad (9)
\end{aligned}$$

By using the properties of multiplicative calculus we have to change the form of DC-model to multiplicative by using equation (4). The multiplicative version of DC-Model will be:

$$\left(\frac{dD}{dt}\right)^* = \exp\left(\frac{dD/dt}{D}\right) = \exp\left(\frac{I-(\lambda-\nu)D+\gamma C}{D}\right) \tag{10}$$

$$\left(\frac{dC}{dt}\right)^* = \exp\left(\frac{dC/dt}{C}\right) = \exp\left(\frac{\lambda D-(\gamma+\delta+\nu+\mu)C}{C}\right) \tag{11}$$

Long Short-Term Memory (LSTM)

Long input sequences may be processed efficiently by recurrent neural networks (RNNs) containing feedback connections, as the Long Short-Term Memory (LSTM) architecture created by Hochreiter and Schmidhuber in 1997 [34]. A conventional LSTM system consists of an input gate, an output gate, a forget gate, and a cell. This enables it to manage the information flow via the gates and preserve values for arbitrary lengths of time [35]. This design is particularly well-suited for tasks requiring time series data because of its ability to handle delays in unforeseen events [36]. In many situations, LSTMs are preferable to RNNs, hidden Markov models, and other sequence learning approaches because they were specifically designed to address the vanishing gradient problem that typically happens during RNN training. Moreover, LSTMs are advantageous for handling sequences with varying length gaps [37].

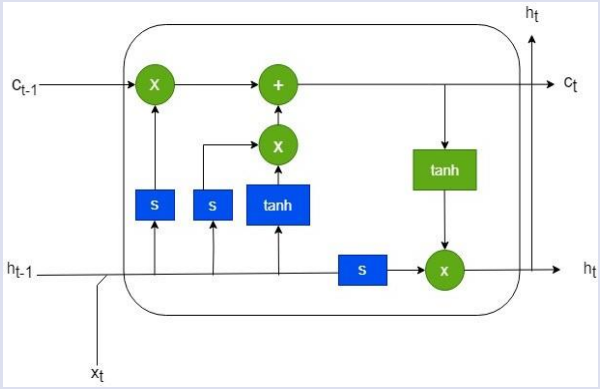


Figure 1. The Architecture of LSTM Network

Tensors of particular forms are essential to the construction of an LSTM, as seen in Figure 1, in order to facilitate effective information flow throughout the network. The dimensions of the cell state tensor ( ) often show up as (batch size, num units), which indicates the number of LSTM units and the simultaneous processing of samples. Potential values for addition to the cell state are stored in the candidate cell state tensor, which has a form similar to the cell state tensor. Similarly, the output of the LSTM cell, the hidden state ( ) tensor, is consistent in shape (batch size, num units) and guarantees smooth information transfer across the network. These tensor configurations are pivotal in facilitating the effective processing and retention of information across lengthy sequences, rendering LSTM networks highly suitable for diverse sequential data tasks.

The provided TensorFlow/Keras code describes a Sequential model architecture with two Long Short-Term Memory (LSTM) layers and a Dense layer, designed for sequence prediction tasks. Each LSTM layer contains 64 memory units and handles input sequences of shape (None, 2), offering flexibility in managing sequences of varying lengths while capturing complex temporal relationships. Efficient training and the introduction of non-linearity are crucial for learning intricate patterns. This can be achieved by using the 'relu' activation function, along with Glorot uniform initialization for both the kernel and recurrent weights. The way the returning sequences are configured in both LSTM layers makes it easier to generate output sequences at every time step, which improves the predictive power of the model. The Dense layer with two units then uses a linear activation function to provide predictions based on the sequences that have been analyzed. The model remains stable and consistent throughout thanks to the Glorot uniform initialization of kernel weights and the zero initialization of bias terms. This well designed architecture is a strong answer for a wide range of sequential data processing jobs because it makes use of the advantages of both the Dense layer's flexibility for precise predictions and the capabilities of LSTM layers for capturing temporal correlations. The offered code uses TensorFlow/Keras to assemble and train a sequential model. The mean squared error (MSE) loss function and the Adam optimizer are used to construct the model, and accuracy is included as a metric for assessment during training. Using a batch size of 32, the model is fitted to the training data for 200 epochs during the training phase. Furthermore, validation data is supplied to evaluate the model's performance on data that was not encountered during training. The 'history' variable contains the training history for further examination.

Table 1. Model Summary

Layer (type)	Output Shape
LSTM	(None, None, 64)
LSTM	(None, None, 64)
Dense	(None, None, 2)
Total params	50,306
Trainable params Non-trainable params	50,306 0

Two LSTM layers and a Dense layer make up the model. With 64 units in the output tensor, each LSTM layer's output form of (None, None, 64) denotes variable-length sequences. The output form of the Dense layer is (None, None, 2), which represents predictions in two units. The model has 50,306 total parameters, all of which may be trained. The model does not contain any non-trainable parameters.

Error Analysis and Model Comparison

This section conducts a comprehensive study of the model's performance, including a forecast for 992 time steps and a full error analysis. Two key error measures are used to assess the efficacy of the models: Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). These metrics quantify the variation between the expected values and the actual values (xi). For MSE, average squared differences are computed, providing a measure of overall prediction accuracy. On the other hand, MAPE gives information about the relative accuracy by computing the percentage difference between the actual and projected values. In order to determine the specific benefits and drawbacks of the models, it is necessary to have a complete understanding of the predictive capability of the models, which is provided by both measurements.

The MSE formula and MAPE formula can be summarized as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \tilde{x}_i)^2$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{x_i - \tilde{x}_i}{x_i}\right|$$

The predicted values are indicated by $\tilde{x}_i$, the actual values are denoted by x_i, and the total number of data points in the dataset is indicated by n. The error analysis that follows will offer a detailed comparison of the models and clarify how well each one captures the underlying patterns in the dataset.

Simulation and Comparison Results

The effectiveness of the Long Short-Term Memory (LSTM) networks and the conventional Runge-Kutta method of order 4 (RK4) has been assessed and contrasted with the Multiplicative Runge-Kutta method of order 4 (MRK4) utilizing three important error metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE).

The Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE) for both D(t) and C(t) were computed in order to assess how well the RK4 and MRK4 approaches replicated the dynamics of the diabetic population model.

The dynamics of two variables, D(t) and C(t), with their rates of change controlled by certain parameters, are modeled in this system of differential equations. The following is a definition of the parameters used in this model: With respect to λ=0.1, μ=0.2, γ=0.05, δ=0.3, and v=0.1, I=1.0 denotes a constant input. Important processes including growth, decay, and the relationships between the two variables are controlled by these factors. The system describes the evolution over time of D(t), which may correspond to a factor connected to diabetes, and C(t), which may represent a control mechanism. Since both variables begin at the same value, D (0) = 1.0 and C (0) = 1.0 are the beginning conditions. The simulation has a step size of 0.1 and runs from 1 to 200 units. The numerical solution to the issue is obtained using the 4th-order Runge-Kutta (RK4) approach, which computes the

development of D(t) and C(t) at each step in time. This approach helps us study the relationships and changes in the variables over time, giving us important information about the behavior of the system, given these initial conditions and the parameters that have been provided.

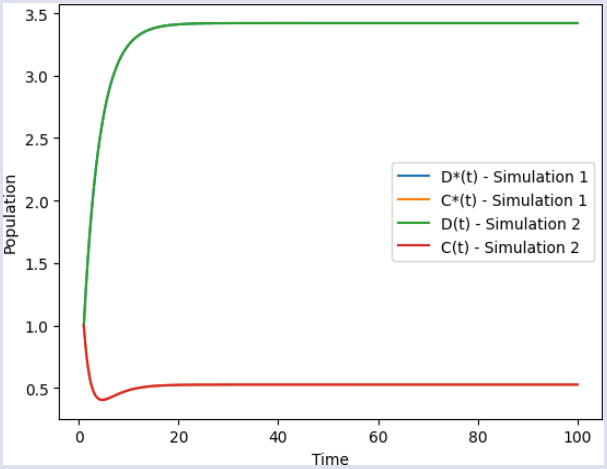


Figure 2. Comparision of MRK4 and RK4

These metrics are crucial indicators of how accurately and consistently each numerical method represents the model's behavior across time. Figure 2 offers critical benchmarks for assessing how accurate the simulations are in comparison to the actual data. This graph demonstrates exactly how MRK4 and RK4 work together and in a similar manner.

Table 2: Error Metrics for D(t) and C(t)

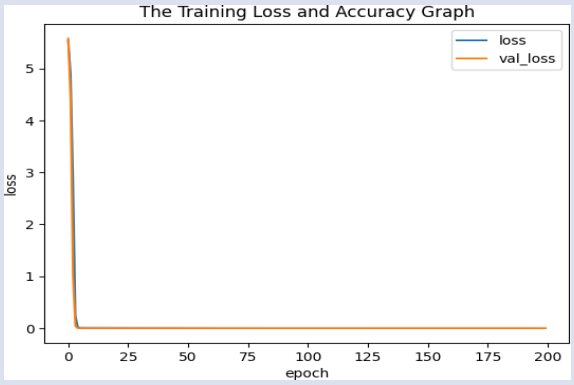
Metric	D(t)	R(t)
MSE	6.9523×10^{-19}	4.0598×10^{-18}
RMSE	8.3381×10^{-10}	2.0149×10^{-9}
MAE	1.08997×10^{-10}	3.5262×10^{-10}

All of the error metrics (MSE, RMSE, and MAE) in Table 2 for both components D and C are extremely close to zero, suggesting that RK4 and MRK4 are essentially producing results that are almost similar with very little fluctuation. This might imply that MRK4 is a tweak that maintains accuracy on par with traditional RK4.

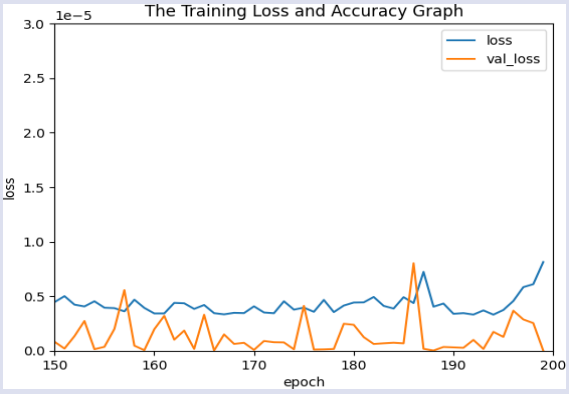
The model is configured using the mean squared error (MSE) as its loss function and leverages the well-known Adam optimizer for neural network optimization.

During a predetermined number of epochs (200 in this example), the training procedure involves regularly presenting the training set and its associated output to the model with a batch size of 32. Throughout this training process, vigilant monitoring is employed, and the progression of the validation loss as well as the training loss is graphically shown. An important tool for analyzing the convergence patterns and spotting any overfitting tendencies in the model is this chart.

Plotting the training loss and validation loss over the duration of epochs, the resulting graphical representation provides important insights into the learning dynamics of the model. Following the training phase, predictions are created on the validation set, allowing for a direct comparison of the projected values with the actual data. To completely evaluate the model's predictive capacity, standard evaluation metrics including mean squared error (MSE), root mean square error (RMSE), and mean absolute percentage error (MAPE) are produced. When combined, these measures offer a quantitative assessment of the model's accuracy and give academics enlightening guidance on the reliability and efficacy of the predictions.



(a) Validation loss

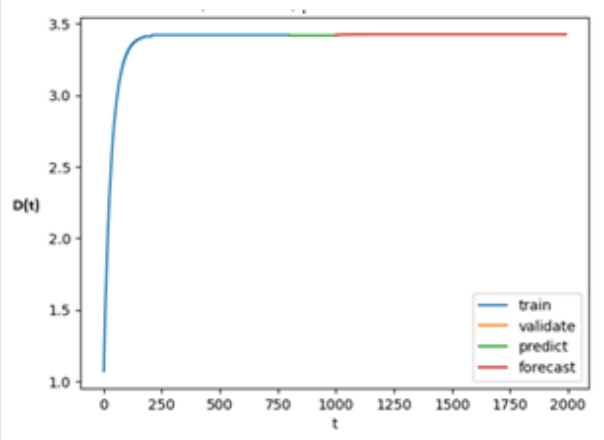


(b) Validation loss with zoom in

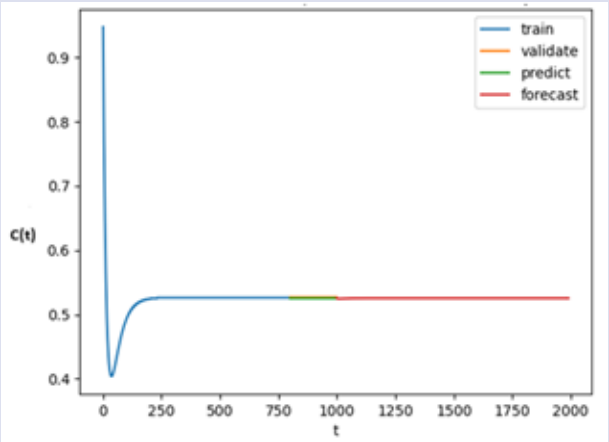
Figure 3.Validation loss during the training process

The validation loss's evolution during the training process is seen in Figure 3. The graph shows how the model's performance changed throughout several epochs on the validation set. The y-axis displays the corresponding loss values, while the x-axis represents the training epochs. Examining the validation loss's convergence over epochs is crucial since it tells us something about how effectively the model adapts to new data. A consistent decrease in the validation loss indicates better performance; oscillations or a plateau, on the other hand, can signal potential issues like overfitting or convergence issues. The model's learning phase is shown by the early decline in validation loss in Figure 3, and the

subsequent stability points to a balanced convergence. Researchers may determine how long to train the model for, what regularization strategies to employ, and whether the model is typically dependable in detecting the underlying patterns in the dataset by closely examining this graph.



(a) Trained, validated, predicted and forecasted for D



(b) Trained, validated, predicted and forecasted for C

Figure 4. Comparision Of LSTM and Multiplicative Runge-Kutta

Figures 4a and 4b provide a comprehensive comparison between the prediction performance of LSTM approaches and the numerical approximation for the system's state functions D(t) and C(t). The figure illustrates how effectively an LSTM model works when predicting system behavior using the MRK4 integration strategy. The blue line, which represents the training data, shows how the system's behavior evolved over the model's training phase and shows how well the LSTM was able to recognize the underlying dynamics. The yellow line represents the validation data, which was not used for training but helps determine how well the model can generalize to new data. The model's ability to predict the behavior of the system based on inputs from the validation or test set is demonstrated by the green line, which represents the predicted data. Last but not least, the LSTM model's estimate of the system's future behavior over a certain period is shown by the red line, which is the anticipated data.

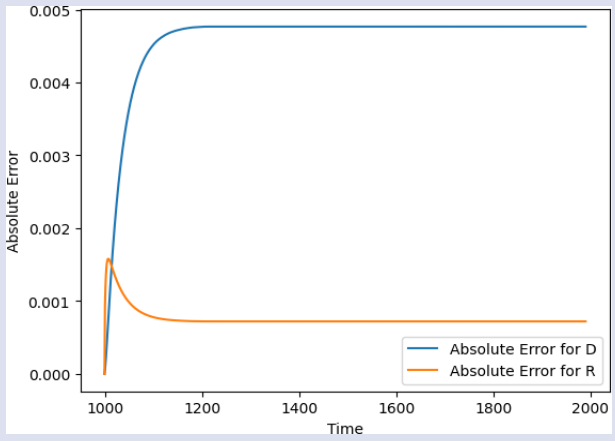


Figure 5. Absolute Error for Each Element

The effectiveness of the LSTM in predicting the behavior of the system when an MRK4 is present is seen in Figure 5. show the absolute error resulting from the LSTM prediction results using the numerical approximation of the state functions D(t) and C(t), respectively. The computation of these errors is based on the absolute difference between the matching data obtained from the LSTM algorithm and the numerical approximation data.

The numerical solution of our RC diabetes model, which serves as validation data, is compared with the absolute errors obtained via the Long Short-Term Memory (LSTM) technique in Figure 5. Particularly in terms of how well the LSTM performs on data that wasn't used for training, these errors serve as quantitative metrics that reveal information about the prediction accuracy of the model. The figures provide useful information on how effectively the LSTM approximates and captures the behavior of our RC diabetes model.

The absolute errors shown in Figure 4c may be used to calculate a quantitative evaluation of the differences between the deep learning approaches (LSTM) predictions and the actual numerical solution obtained with Multiplicative Runge-Kutta (MRK). The results of this comparative research show that the MRK-based numerical approach and the LSTM method accord quite well. The findings show a substantial relationship between the reliability of the LSTM model and its accuracy in capturing and reproducing the dynamics of the RC diabetes model.

Table 3: Performance Metrics

Metric	Value
Mean Absolute Error (MAE)	0.004594215327693641
Mean Squared Error (MSE)	2.151950235700345e-05
Root Mean Squared Error (RMSE)	0.004638911764304582
Mean Absolute Percentage Error (MAPE)	0.13433378151151

The performance evaluation of the model reveals positive results for several metrics. With mean absolute error (MAE) of 0.0046 and average deviations from true values averaging at a minimal scale, our model demonstrates a high degree of accuracy in forecasting values. The Mean Squared Error (MSE) of around 2.15×10−5, which indicates that the forecasts exhibit minute squared errors, further demonstrates the correctness of the model.

The Root Mean Squared Error (RMSE) of 0.0046, which indicates how frequently projections differ from real data, supports these findings. Furthermore, the Mean Absolute Percentage Error (MAPE) of 0.1343 indicates an average percentage error of around 13.43%, which is rather low given the circumstances of our research but not insignificant. These results corroborate each other and show how well our model predicts outcomes. The final method employed is outstanding; it demonstrates an amazing ability to find intricate patterns in the data, leading to low error projections that are accurate.

Conclusion

In the end, this study thoroughly compared two distinct methodologies: multiplicative calculus and long short-term memory (LSTM) networks. We assessed the prediction performance of both methods using both quantitative analysis and close inspection. It was thoroughly examined if the proposed multiplicative calculus technique could extract complicated connections from the data, and the study looked into the learning and generalization capacities of LSTM networks. The comparison was facilitated by in-depth analyses of error metrics, including mean squared error (MSE), root mean square error (RMSE), and mean absolute percentage error (MAPE). By comparing the benefits and drawbacks of LSTM networks and multiplicative calculus, we hoped to provide significant insights into the relative effectiveness of each in predictive modeling.

Conflict of interest

There are no conflicts of interest in this work.

Ethical Approval Statement

There is no Ethical Approval needed for this work.

References

[1] Emerging Risk Factors Collaboration, Diabetes mellitus, fasting blood glucose concentration, and risk of vascular disease: a collaborative meta-analysis of 102 prospective studies, *The lancet*, *375*(9733) (2010) 2215-222

[2] Ramsingh J., & Bhuvaneswari V. (2021). An efficient map reduce-based hybrid NBC-TFIDF algorithm to mine the public sentiment on diabetes mellitus–a big data approach, *Journal of King Saud University-Computer and Information Sciences*, *33*(8) 1018-1029.

[3] Artzi N. S., Shilo, S., Hadar E., Rossman H., Barbash-Hazan S., Ben-Haroush A., ... , Segal E., Prediction of gestational diabetes based on nationwide electronic health records. *Nature medicine*, 26(1) (2020), 71-76.

[4] Misra A., Gopalan H., Jayawardena R., Hills A. P., Soares M., Reza-Albarrán A. A., Ramaiya K. L., Diabetes in developing countries. *Journal of diabetes*, 11(7) (2019) 522-539.

[5] Edwards M. S., Wilson D. B., Craven T. E., Stafford J., Fried L. F., Wong T. Y., ... , Hansen K. J., Associations between retinal microvascular abnormalities and declining renal function in the elderly population: the Cardiovascular Health Study. *American journal of kidney diseases*, *46*(2) (2005) 214-224.

[6] Saeedi P., Petersohn I., Salpea P., Malanda B., Karuranga S., Unwin N., et all., IDF Diabetes Atlas CommitteeGlobal and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation Diabetes Atlas, *Diabetes research and clinical practice*, 157 (2019) 107843.

[7] Vaishali R., Sasikala R., Ramasubbareddy S., Remya S., Nalluri SGenetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset. In *2017 international conference on computing networking and informatics (ICCNI)*, (2017) 1-5.

[8] Cho N. H., Shaw J. E., Karuranga S., Huang Y., da Rocha Fernandes J. D., Ohlrogge,A. W., Malanda, B. I. D. F., IDF Diabetes Atlas: Global estimates of diabetes prevalence for 2017 and projections for 2045, *Diabetes research and clinical practice*, 138 (2018) 271-281.

[9] Maniruzzaman M., Rahman M. J., Al-MehediHasanM., Suri, H. S., Abedin, M. M., El-Baz A., Suri J. S., Accurate diabetes risk stratification using machine learning: role of missing value and outliers. *Journal of medical systems*, 42 (2018) 1-17.

[10] Brahim-Belhouari S., Bermak A., Gaussian process for nonstationary time series prediction, Computational Statistics & Data Analysis, 47(4) (2004) 705-712.

[11] Cortes C., Vapnik V., Support-vector networks, *Machine learning*, 20 (1995) 273-297.

[12] Kégl B., The return of AdaBoost. MH: multi-class Hamming trees, (2013) *arXiv preprint arXiv:1312.6086*.

[13] Tabaei B. P., Herman W. H., A multivariate logistic regression equation to screen for diabetes: development and validation, *Diabetes Care*, 25(11) (2002) 1999-2003.

[14] Jenhani I., Amor N. B., Elouedi Z., Decision trees as possibilistic classifiers, *International journal of approximate reasoning*, *48*(3) (2008) 784-807.

[15] Qawqzeh Y. K., Bajahzar A. S., Jemmali M., Otoom M. M., Thaljaoui A., Classification of diabetes using photoplethysmogram (PPG) waveform analysis: logistic regression modeling. *BioMed Research International*, (2020).

[16] Pethunachiyar G. A., Classification of diabetes patients using kernel based support vector machines. In *2020 International Conference on Computer Communication and Informatics (ICCCI),* (2020) 1-4).

[17] Abdu-Allah Z. M., Mahmood O. T., AL-Naib A. M. I., Photovoltaic Battery Charging System Based on PIC16F877A Microcontroller, *International Journal of Engineering and Advanced Technology*, 3(4) (2014) 55-59.

[18] Kumari S., Kumar D., Mittal M., An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. *International Journal of Cognitive Computing in Engineering*, 2 (2021) 40-46.

[19] Hussain A., Naaz S., Prediction of diabetes mellitus: comparative study of various machine learning models, In *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2020* Springer Singapore, 2 (2021) 103-115.

[20] Hussain A., Naaz, S., Prediction of diabetes mellitus: comparative study of various machine learning models, In *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2020* Springer Singapore, 2 (2020) 103-115.

[21] Bashirov A. E., Mustafa R. I. Z. A., On complex multiplicative differentiation, *TWMS Journal of applied and engineering mathematics*, *1*(1) (2011) 75-85.

[22] Bashirov A. E., Norozpour S., On complex multiplicative integration, *TWMS Journal of Applied and Engineering Mathematics*, *7*(1) (2017) 82-93.

[23] Uzer A., Multiplicative type complex calculus as an alternative to the classical calculus, *Computers & Mathematics with Applications*, *60*(10) (2010) 2725-2737.

[24] Babacan Y., Kaçar F., Memristor emulator with spike-timing-dependent-plasticity, *AEU-International Journal of Electronics and Communications*, *73*(2017) 16-22.

[25] Bashirov A. E., Bashirova G., Dynamics of literary texts and diffusion, *Online Journal of Communication and Media Technologies,* 1(3) (2011) 60-82.

[26] Wang Y., Liao X., Stability analysis of multimode oscillations in three coupled memristor-based circuits, *AEU-International Journal of Electronics and Communications*, *70*(12) (2016) 1569-1579.

[27] Aniszewska D., Multiplicative runge–kutta methods, *Nonlinear Dynamics*, 50(1-2) (2007) 265-272.

[28] Aniszewska D., Multiplicative runge–kutta methods, *Nonlinear Dynamics*, 50(1-2) (2007) 265–272,.

[29] Dileep P., Rao K. N., Bodapati P., Gokuruboyina S., Peddi R., Grover A., Sheetal A., An automatic heart disease prediction using cluster-based bi-directional LSTM (C-BiLSTM) algorithm, *Neural Computing and Applications,* 35(10) (2023) 7253-7266.

[30] Srinivasu P. N., SivaSai J. G., Ijaz M. F., Bho, A. K., Kim W., Kang J. J., Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM, *Sensors*, 21(8) (2021) 2852.

[31] Dua M., Makhija D., Manasa P. Y. L., Mishra P., A CNN–RNN–LSTM based amalgamation for Alzheimer's disease detection, *Journal of Medical and Biological Engineering*, 40(5) (2020) 688-706.

[32] Bashirov A. E., Mustafa R. I. Z. A., On complex multiplicative differentiation, *TWMS Journal of applied and engineering mathematics*, 1(1) (2011) 75-85.

[33] Riza M., Aktöre H., (The Runge–Kutta method in geometric multiplicative calculus, *LMS Journal of Computation and Mathematics*, 18(1) (2015) 539-554.

[34] Hochreiter S., Schmidhuber J., Long short-term memory, *Neural computation*, 9(8) (1997) 1735-1780.

[35] Gers F. A., Schmidhuber J., Cummins F., Continual prediction using LSTM with forget gates, In *Neural Nets WIRN Vietri-99: Proceedings of the 11th Italian Workshop on Neural Nets, Vietri Sul Mare, Salerno, Italy, 20–22 May 1999,* Springer London, (1999) 133-138.

[36] Gers F. A., Schmidhuber J., Cummins F., Learning to forget: Continual prediction with LSTM, *Neural computation*, *12*(10) (2000) 2451-2471.

[37] Gers F. A., Schmidhuber E., LSTM recurrent networks learn simple context-free and context-sensitive languages, *IEEE transactions on neural networks*, *12*(6) (2001) 1333-1340.