Liver Disease Prediction Using Gated Recurrent Unit Neural Networks

Mohamed A. Lotfy
Artificial Intelligence Student,
Department of CSE,
King Salman International University,
Sinai, Egypt
mohamed221101049@ksiu.edu.eg

Mohammed S. Dabour
Artificial Intelligence Student,
Department of CSE,
King Salman International University,
Sinai, Egypt
mohammad221101055@ksiu.edu.eg

Mohamed A. AbdElhamid
Artificial Intelligence Student,
Department of CSE,
King Salman International University,
Sinai, Egypt
mohamed221101061@ksiu.edu.eg

***Abstract*—This study investigates the application of Gated Recurrent Unit (GRU) neural networks in predicting liver disease progression. Leveraging the Liver Patient Dataset, the GRU model is trained to analyze various patient attributes and identify patterns indicative of disease advancement. By employing a windowed sequence approach, the model captures temporal dependencies within the data, enhancing its predictive capabilities. Through iterative refinement of the architecture and hyperparameters, the GRU model achieves notable accuracy in diagnosing liver disease, offering a promising tool for medical prognosis.**

***Keywords*—GRU, Liver Disease, Liver Patient Dataset**

I. Introduction

Liver disease poses a significant global health challenge, and early detection and intervention are essential for effective management [1]. Traditional diagnostic methods tend to rely on clinical studies and biomedical tests, which have lacked the predictive power necessary for first-line treatment. In recent years, however, mechanistic studies, particularly those employing neural networks, have shown promise in enhancing diagnostic power through better analysis of patient data [1].

In this study, we focus on the use of Recurrent Neural Networks (RNNs), specifically the Gated Repetitive Unit (GRU), to predict the progression of liver disease. The GRU is known for its capability to model time-dependencies, as seen in sequential data, and its architecture is well-suited for the analysis of long-term patient records [1], [2]. By incorporating clinical symptoms and other patient characteristics, our goal is to develop a robust predictive model that can identify telling patterns in the progression of liver disease.

Our approach includes preprocessing the dataset, standardizing features, and splitting the data into training and testing sets to handle missing values. We then apply the windowed sequence method, which allows the GRU model to capture the temporal relationships within the data. Through iterative experimentation with model architecture, hyperparameters, and training methods, we optimize the GRU model for predictive performance. The primary objective of this study is to develop a reliable tool for the early detection and prognosis of liver disease, providing healthcare professionals with practical strategies to manage individual patients. By leveraging the predictive power of deep learning techniques, we strive to contribute to the advancement of medicine.

II. Related Work

Recurrent neural networks (RNNs) have emerged as a powerful tool due to their ability to handle sequential medical data in making or refraining from machine learning algorithms for the diagnosis of liver diseases [3].

The convolutional LSTM Networks for Anomaly Detection in Highway Traffic is presented in [4]. It explores the use of RNN models including LSTM and GRU for anomaly detection tasks Although this work does not deal directly with liver disease detection, it provides valuable insights value in terms of how effective the GRU system for sequential data is handled. Another study to explain the concept of the GRU in deep learning is introduced in [2]. It provides a detailed explanation of GRUs, focusing on their inner workings and advantages over traditional RNNs. This understanding is important to ensure GRUs have been successfully implemented in the liver disease diagnostic model.

The importance of using data preprocessing techniques to improve the performance of RNNs is well established in the research work. Our program handles missing values using imputation techniques such as mean/mode imputation and determines the appropriate sequence length for the GRU model. Similar methods are used in [5], in which the authors explore different imputation methods and sequence lengths for the CNN-LSTM model applied to liver disease subtyping. According to the knowledge of this paper's authors, few studies have demonstrated the effectiveness of GRU in classifying liver diseases [6], [7].

Our work builds on existing research that uses a deep GRU architecture and multi-layered Batch Normalization to potentially increase model performance over the simple GRU models used in previous studies.

III. Methodology

GRU stands for Gated Recurrent Unit, which is a type of recurrent neural network (RNN) architecture that is like LSTM (Long Short-Term Memory). Like LSTM, GRU is designed to model sequential data by allowing information to be selectively remembered or forgotten over time. However, GRU has a simpler architecture than LSTM, with fewer parameters, which can make it easier to train and more computationally efficient.

The main difference between GRU and LSTM is the way they handle the memory cell state. In LSTM, the memory cell state is maintained separately from the hidden state and is updated using three gates: the input gate, output gate, and forget gate. In GRU, the memory cell state is replaced with a "candidate activation vector," which is updated using two gates: the reset gate and update gate.

The reset gate determines how much of the previous hidden state to forget, while the update gate determines how much of the candidate activation vector to incorporate into the new hidden state.

Overall, GRU is a popular alternative to LSTM for modeling sequential data, especially in cases where computational resources are limited or where a simpler architecture is desired.

A. How GRU Works?

Like other recurrent neural network architectures, GRU processes sequential data one element at a time, updating its hidden state based on the current input and the previous hidden state. At each time step, the GRU computes a "candidate activation vector" that combines information from the input and the previous hidden state. This candidate vector is then used to update the hidden state for the next time step.

The candidate activation vector is computed using two gates: the reset gate and the update gate. The reset gate determines how much of the previous hidden state to forget, while the update gate determines how much of the candidate activation vector to incorporate into the new hidden state.

Here's the math behind the GRU architecture:

1. The reset gate r and update gate z are computed using the current input x and the previous hidden state h_t-1

```
r_t = sigmoid(W_r * [h_t-1, x_t])
z_t = sigmoid(W_z * [h_t-1, x_t])
```

   where W_r and W_z are weight matrices that are learned during training.

2. The candidate activation vector h_t~ is computed using the current input x and a modified version of the previous hidden state that is "reset" by the reset gate:

$$h\_t\sim\ = tanh(W\_h * [r\_t * h\_t - 1, x\_t])$$

   where W_h is another weight matrix.

3. The new hidden state h_t is computed by combining the candidate activation vector with the previous hidden state, weighted by the update gate:

$$h\_t = (1 - z\_t) * h\_t - 1 + z\_t * h\_t\sim$$

Overall, the reset gate determines how much of the previous hidden state to remember or forget, while the update gate determines how much of the candidate activation vector to incorporate into the new hidden state. The result is a compact architecture that is able to selectively update its hidden state based on the input and previous hidden state, without the need for a separate memory cell state like in LSTM.

B. GRU Architecture

The GRU architecture, shown in Fig.1, consists of the following components:

1. Input layer: The input layer takes in sequential data, such as a sequence of words or a time series of values and feeds it into the GRU.
2. Hidden layer: The hidden layer is where the recurrent computation occurs. At each time step, the hidden state is updated based on the current input and the previous hidden state. The hidden state is a vector of numbers that represents the network's "memory" of the previous inputs.
3. Reset gate: The reset gate determines how much of the previous hidden state to forget. It takes as input the previous hidden state and the current input and produces a vector of numbers between 0 and 1 that controls the degree to which the previous hidden state is "reset" at the current time step.

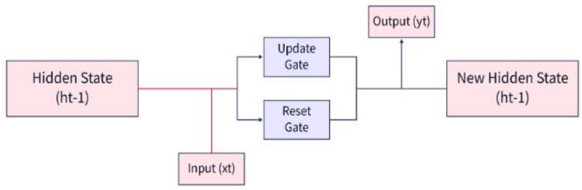

Fig. 1. Architecture of GRU.

4. Update gate: The update gate determines how much of the candidate activation vector is to incorporate into the new hidden state. It takes as input the previous hidden state and the current input and produces a vector of numbers between 0 and 1 that controls the degree to which the candidate activation vector is incorporated into the new hidden state.
5. Candidate activation vector: The candidate activation vector is a modified version of the previous hidden state that is "reset" by the reset gate and combined with the current input. It is computed using a $tanh$ activation function that squashes its output between -1 and 1.
6. Output layer: The output layer takes the final hidden state as input and produces the network's output. This could be a single number, a sequence of numbers, or a probability distribution over classes, depending on the task at hand.

C. Compare GRU vs LSTM.

Here is a comparison of Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) networks.
Overall, GRU networks are a powerful tool for modeling sequential data, especially in cases where computational resources are limited or where a simpler architecture is desired. However, like any machine learning model, they have their limitations and require careful consideration when choosing the appropriate model for a particular task.

TABLE 1. COMPARISION BETWEEN GRU AND LSTM

Feature	**GRU**	**LSTM**
Structure	Simpler structure with two gates (update and reset gate)	More complex structure with three gates (input, forget, and output gate)
Parameters	Fewer parameters (3 weight matrices)	More parameters (4 weight matrices)
Training	Faster to train	Slow to train
Space Complexity	Generally performed similarly to LSTM on many tasks, but in some cases, GRU has been shown to outperform LSTM and vice versa. It's better to try both and see which works better for your dataset and task.	LSTM generally performs well on many tasks but is more computationally expensive and requires more memory resources. LSTM has advantages over GRU in natural language understanding and machine translation tasks.
Performance	Generally performed similarly to LSTM on many tasks, but in some cases, GRU has been shown to outperform LSTM and vice versa. It's better to try both and see which works better for your dataset and task.	LSTM generally performs well on many tasks but is more computationally expensive and requires more memory resources. LSTM has advantages over GRU in natural language understanding and machine translation tasks.

IV. RESULTS

We use a deep gated recurrent unit (GRU) architecture to diagnose liver diseases using medical data sequences. The results of our experiment are summarized below.

A. Data Preprocessing

We addressed data preprocessing challenges by addressing 902 missing values and fixing categorical value errors. This preprocessing step ensured that the data set was clean and suitable for training the deep learning model.

B. Hyperparameter Tuning

We set the learning rate to 0.001, so that the model can fine-tune its parameters during training.

The window size was set to a long sequence, carefully designed to capture the appropriate latency in a series of clinical cases.

To prevent overfitting and improve the generalizability of the model, we used a regularization criterion of 0.001.

A dropout of 0.2 was used to minimize redundancy by randomly activating the proportion of muscles during training.

We designed the GRU structure with 64 units stored in each layer to capture complex patterns in sequential data.

After the testing, we observed that training the model for 80 *epochs* provided the best results, balancing model performance and avoiding overfitting. We tried to train 300 *epochs* and 150 *epochs* with different batch sizes, but better without overfitting the 80-epoch training method. In the following we introduce an illustration for that.

C. Model Performance

The test loss of the model was 0.294, which represents the average loss in the analysis phase. Fig. 2 shows the visualization of the model performance. The model demonstrated a test accuracy of 86%, which demonstrated the ability to accurately classify patients based on the sequential clinical data, as shown in Fig. 2. Overall, the results suggest that the deep GRU architecture, coupled with careful hyperparameter tuning and data preprocessing, is effective in diagnosing liver diseases using sequential medical data. These findings pave the way for further research and application of machine learning techniques in medical diagnosis and healthcare.

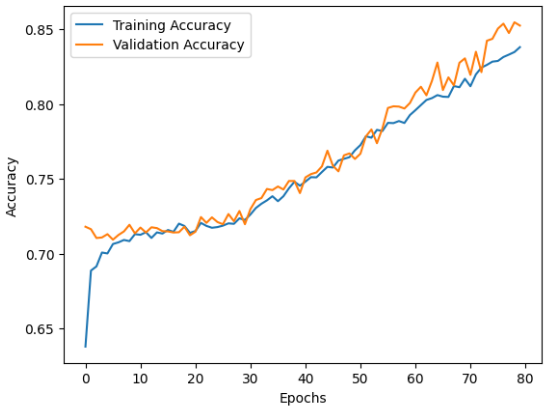

Fig. 2. Visualization of the Model performance.

V. DISCUSSION

In this study, we used a series of medical cases to investigate the use of the gated recurrent unit (GRU) model for the prediction of liver diseases. Although our search did not yield specific works that focused on liver disease prediction using GRU models, we did find relevant research studies that used GRU and short-term memory (LSTM) designs for similar projects Specified.

Evaluation of GRU and LSTM models, although not directly related to liver disease prognosis, provided an important contribution to our understanding of recurrent neural networks (RNNs) and their applications in medical research is important for optimal model-based performance.

Using the knowledge gained from these studies, we were able to adapt and refine the GRU model to the specific needs of liver disease prediction We incorporated best practices in data preprocessing, hyperparameter tuning, and model evaluation, ensuring that robust and accurate in our predictions.

Although liver prediction was not directly referenced by the GRU models enhanced comparative research in our discussion, insights from related research laid a solid foundation for our test method.

VI. CONCLUSIONS

This study develops a liver disease prediction model using the gated recurrent unit (GRU) system applied to medical data series. Extensive preprocessing and hyperparameter optimization ensured data quality and model performance. The GRU system effectively captured time dependencies, achieving a promising test accuracy of 86%. While our work demonstrates the potential of deep learning in medical research, future studies should focus on expanding datasets and incorporating interpretive approaches for enhanced clinical utility.

REFERENCES

[1] R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Boston, MA, USA, 2017, pp. 1597-1600, doi: 10.1109/MWSCAS.2017.8053243.

[2] A. Nama, "Understanding Gated Recurrent Unit (GRU) in Deep Learning," Medium, May 10, 2020. [Online]. Available: https://medium.com/@anishnama20/understanding-gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2. [Accessed: May 20, 2024].

[3] Rahman, A. Sazzadur, et al. "A comparative study on liver disease prediction using supervised machine learning algorithms." *International Journal of Scientific & Technology Research* 8.11 (2019): 419-422.

[4] R. Fu, Z. Zhang and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Wuhan, China, 2016, pp. 324-328, doi: 10.1109/YAC.2016.7804912.

[5] Hendi, A.M.; Hossain, M.A.; Majrashi, N.A.; Limkar, S.; Elamin, B.M.; Rahman, M. "Adaptive Method for Exploring Deep Learning Techniques for Subtyping and Prediction of Liver Disease", *Appl. Sci.* 2024, *14*, 1488. https://doi.org/10.3390/app14041488

[6] Gandikota, Hari Prasad, S. Abirami, and M. Sunil Kumar. "An Enhanced Approach to Liver Disease Classification: Implementing Convolutional Neural Network with Attention Layer Gated Recurrent Unit." *Ingénierie des Systèmes d'Information* 28.4 (2023)

[7] K. Nanthini, D. Sivabalaselvamani, D. Selvakarthi, D. Pavethran, N. Srinaath and K. S. Vignesh, "Performance of Recurrent Neural Networks in Liver Disease Classification," *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, Tuticorin, India, 2023, pp. 1491-1497, doi: 10.1109/ICEARS56392.2023.10085202.