ISSN: 0067-2904

Evaluating the Performance and Behavior of CNN, LSTM, and GRU for Classification and Prediction Tasks

Hasanen S. Abdullah[1*], Nada Hussain Ali [1], Nada A.Z. Abdullah [2]
[1] *Department of Computer Sciences, University of Technology, Baghdad, Iraq*
[2] *Department of Computer Science, College of Sciences, University of Baghdad, Baghdad, Iraq*

Received: 29/4/2023 Accepted: 19/8/2023 Published: 30/3/2024

Abstract

Deep learning (DL) plays a significant role in several tasks, especially classification and prediction. Classification tasks can be efficiently achieved via convolutional neural networks (CNN) with a huge dataset, while recurrent neural networks (RNN) can perform prediction tasks due to their ability to remember time series data. In this paper, three models have been proposed to certify the evaluation track for classification and prediction tasks associated with four datasets (two for each task). These models are CNN and RNN, which include two models (Long Short Term Memory (LSTM)) and GRU (Gated Recurrent Unit). Each model is employed to work consequently over the two mentioned tasks to draw a road map of deep learning models for a variety of tasks under the control of a unified architecture for each proposed model.

Keywords: Deep learning (DL), Recurrent Neural Network (RNN), Convolution Neural Network (CNN), Classification and Prediction.

تقييم الاداء والسلوك لنماذج CNN, LSTM و GRU لمهام التصنيف والتنبؤ

حسنين سمير عبدالله[1] , ندى حسين علي[1*] , ندا عبد الزهرة عبدالله[2]
[1] قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق
[2] قسم علوم الحاسوب، كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصة

التعلم العميق له دور مهم في عدة مهام خصوصا مع التصنيف والتنبؤ. مهام التصنيف من الممكن انجازها بصورة كفؤة باستعمال الشبكات العصبية الالتفافية بالأخص اذا كانت البيانات حجمها كبير، بينما الشبكات العصبية المتكررة تستطيع ان تؤدي مهام التنبؤ بسبب قابليتها على تذكر البيانات التسلسلية. في هذا البحث ثلاث معماريات من شبكات التعلم العميق تم اقتراحها للمصادقة على مسار التقييم لمهام التصنيف والتنبؤ المقترنة بأربع مجاميع بيانات (مجموعتين لكل مهمة)، هذه المعماريات هي GRU, LSTM, CNN. كل معمارية وظفت لكي تعمل على المهمتين لرسم خارطة طريق لطرق التعلم العميق لمختلف المهام وتحت سيطرة المعمارية الموحدة لكل طريقة.

*Email: nada.h.ali@uotechnology.edu.iq

1. Introduction

Deep learning is a field that is driven by machine learning. Deep learning dives into deep texture, using many layers to learn fine-grained details [1]. This type of learning allows huge amounts of data to be processed in order to find relationships and patterns that are often undetected by humans. The term "deep" refers to the layers in the neural network; more layers mean a deeper network, in which they give strength and power to the deep networks [2]. For different fields in AI like image classification, prediction, pattern recognition, and many others, it has been proven that having deeper hidden layers gives an extraordinary result in comparison to the classical methods. Two of the most known deep neural networks are convolutional neural networks (CNN) and recurrent neural networks (RNN) [3]. Convolutional neural networks (CNNs) are considered a type of artificial neural networks (ANNs), a type that is very special and empowering. This type of ANN gains importance by presenting remarkable performance on different visual tasks. These tasks focus on image processing in their different applications and other learning tasks that can be handled using AI techniques. Convolutional neural networks are built to deal with data that is composed of multiple arrays. As an example, a color image composed of three color channels is composed of three 2D arrays consisting of pixel intensities. The convolutional filters are used to extract information from images; the first layers detect edges, while object parts are detected in the later layers, and the farther layers can detect complete objects, such as faces or some complex geometrical shapes [4]. Sequence models are models that deal with data in a sequential manner, and the sequence of all the entities is important. RNN works perfectly with sequence data because the neurons in the RNN have a memory that will use it to remember information about the steps before it. Each neuron takes the output from itself and feeds it back to the same neuron before making any predictions; the neuron takes input not only from the hidden layer before it but from itself too. The training depends on the time; the error is propagated from the last time stamp to the first time stamp in the hidden layers. The recurrent connections in RNN allow updating the weights based on the calculation of the errors for each time stamp.

The recurrent connections in the recurrent networks in the hidden units are able to read a sequence of data and produce output based on that data. If the model is too slow to learn, then it is suffering from a problem called the "vanishing gradient," which can be solved by long-short-term memory (LSTM) [5].

2. Related Work

As the growth of deep learning is moving rapidly, several research studies tend to use more than one approach or model of deep learning and compare the results in order to better find the best model for a particular problem. In this section, a few works will be discussed that used more than one deep learning model. In [6], the authors used three models of deep learning for sign language recognition: time-LeNet as the first model, multi-channel deep CNN as the second model, and a modification to the time-LeNet model as the third model. These models achieved 79.7%, 83.9%, and 81.6%, respectively, in classification accuracy. The authors in [7] proposed a pathological diagnosis system for speech diseases. They used an SVD dataset; a feature extraction process was performed on the dataset as a first step, and then these features were inserted into the RNN model for the diagnosing process. The authors also used the CNN model for diagnosing purposes. The RNN and CNN models achieved accuracy of 86.5% and 87.1%, respectively, for the diagnosing task.

In [8], the authors used CNN, Gated Recurrent Unit (GRU), and RNN models for a comparative study of natural language processing. The models achieved approximate results, and the RNN outperformed the CNN and GRU except in some tasks like key phrase recognition and question answering, in which the CNN outperformed the RNN and GRU. The authors used four datasets

to achieve the purpose of the research. The models achieved approximate results for the four datasets used, i.e., the CNN model achieved accuracy of 82%, 77%, 71%, and 94%, while the GRU reached accuracy of 86%, 78%, 69%, and 93%, and the LSTM achieved 84%, 77%, 71%, and 93% accuracy. The authors in [9] proposed two deep learning approaches for opinion mining from long textual documents extracted from e-newspapers. The authors used CNN and RNN models for the task. The CNN model was used with the document-to-vector preprocessing step to boost performance, and the RNN model with document-to-vector conversion. The CNN model outperformed the RNN model with a slight advantage; the CNN model achieved 97% accuracy while the RNN model achieved 94% accuracy on a dataset that was collected by the authors from web pages and e-newspapers. In [10], the authors used several deep learning models for sentiment analysis tasks: three CNN models and five RNN models with different input structures (word-base input and character-based input). These models were implemented on 13 datasets. The results showed that the deeper the CNN model is, the better the results will be; nevertheless, using a complex RNN model proved that the results would be more accurate than simple RNN. The best results in word-based input structure were achieved by the Bidirectional-LSTM model, which reached 91% using the Trip dataset, while the best results in character-based input structure were achieved by the Bidirectional-GRU model, which reached 89% using the Trip dataset.

3. Theoretical Background

Classification and prediction are two forms of machine learning approaches that can be used to describe important data classes, extract models, or predict future data trends. The classification goal is to predict categorical labels (classes), while the prediction goal is to model continuous-valued functions [11]. In the context of prediction, RNN takes temporal input data to be trained on in order to produce the desired temporal output. The output can be any time-series data that is related to the input data. Gradient-based is the most common training technique; however, it is not the only technique, and other techniques have been proposed too, based on convex optimization or derivative-free approaches. The loss function is the objective function to be reduced, which depends on the calculated error between the estimated output and the real output of the network. An interesting aspect of RNNs is that they can be executed in a generative mode when suitable training is achieved, as they have the ability to reproduce temporal patterns that are similar to those they have been trained on [12]. Two common models of RNN are presented: long-short-term memory (LSTM) and gated recurrent units (GRU). LSTM adds two gates when compared with RNN; these gates are the input gate and the forget gate. These gates solve the problems of gradient disappearance and gradient explosion, so they can capture long-term information and achieve better performance in long-sequence text. GRU is similar to ordinary RNN with regard to the input and output structures, but it is similar to the internal structure of LSTM [13].

LSTM is one of the most popular deep learning models nowadays because of its superior performance in modeling both short- and long-term correlations in data. LSTM tries to solve the problem of vanishing gradients by not forcing any bias against recent observations, but a constant error is kept owing back through time. The LSTM layer consists of three gates; each gate has unique and special functionality. The forget gate decides the information that will be discarded (forgotten) from the cell state before the current state. After being modified by the forget gate, the input gate works on the previous state and decides the amount of effect that should be enforced on the new state h[t], using a new candidate ~ h[t] to produce the output y[t]. The third gate is the output gate, which selects the part of the state that will be returned as output. Each gate in the LTSM model depends on the current external input x[t] and the output y[t-1] from the previous cells [12].

The Gated Recurrent Unit is considered a special type of LSTM. The internal structure of the GRU is similar to the internal structure of the LTSM, except that the LTSM has three gates, while the GRU merges the input gate and the forget gate into a single gate called the update gate, while the other gate is called the reset gate. Even though the GRU is similar to and based on the LTSM, it is thought to be a simpler model. The calculations, training, and updating of the internal state are easier to do in the GRU than in the LSTM, but both are immune to the vanishing gradient problem [14].

The update gate controls the extent of information that will be returned from the previous state to the current state, while the reset gate determines whether the previous information from the previous state should be combined with the current state [14].

A CNN is a neural network that consists of one or several convolutional layers [15]. CNN obtains a huge amount of information and enables learning from raw data abstraction levels [16]. A convolutional layer implements a convolution process on the input data to obtain features. A convolution is an operation on two functions of a real-valued argument; it is a dot product between the input values and the kernel values [17]. In CNN models, several activation functions are used, such as the Rectified Linear Unit function (ReLu), which is used in CNN more frequently in comparison with other functions [18].

$$\text{ReLu}(x) = \max(0, x) \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad 1$$

While the other activation function that is also used in CNN models is the SoftMax activation function, which is a combination of several sigmoid activation functions, In the sigmoid function, the output values range from 0 to 1. These values can be considered probabilities of a certain class’s data points. Sigmoid is used in binary classification, while SoftMax can be implemented for problems with multiple classifications. For every data point, SoftMax returns the probability of classes for all the individuals [19]. In CNN models, there is a pooling layer. This layer chooses a subset of the vectors in order to reduce the feature map size that will be passed to the next convolution layer. The traditional methods used are average pooling and max pooling. In max pooling, the largest value in each pool region is selected [20]. The final layer in the CNN model is the fully connected layer. In the final feature map, each feature is connected to a neuron; these neurons are in the hidden state of the first layer of the fully connected layer. This layer works in the same manner as a conventional feed-forward network. One fully connected layer is used in most cases, but sometimes if there is a need for more than one fully connected layer to increase the power of the computations, then the connections between these layers are structured like a conventional feed-forward network [21].

The stochastic gradient descent (SGD) optimizer is a way to reduce an objective function. It works by changing the model's parameters in the opposite direction of the gradient of the objective function [22]. The Adam optimizer is a method based on adaptive estimates of lower-order moments. It also has lower memory requirements, is computationally efficient, and is invariant to diagonal rescaling of the gradients [23].

4. Methodology

In this study, four datasets were used to accomplish the goal of the study: two datasets for classification and two datasets for prediction. The aim of this study is to obtain results from three deep learning models that are best known for classification and prediction purposes. As it’s known, CNN gives the best results for classification purposes, while RNN models operate better with prediction tasks because of their nature for handling series data.

In this paper, three models were constructed, trained, and tested for a classification task as well as a prediction task implemented on four datasets.

4.1Dataset description

Four datasets were used in this paper; the first two are for classification tasks, and the other two are for prediction tasks.

- Breast cancer dataset (BCD): This dataset contains 570 records in 33 columns of different values like malignant benignant, texture, radius, perimeter, and area of the tumor [24].
- Multi-class images for weather classification (MCIWC): this dataset contains 1125 files of different weather conditions like cloudy, sunny, rainy, and shiny [25].
- LSTM-multivariate air pollution prediction (MAPP): this dataset contains several columns and 43801 records that represent weather conditions like dew, temperature, pollution, snow, rain, and wind speed to predict the air pollution for the next few hours [26].
- Heart Disease Prediction (HDP): This dataset contains 18 columns and 319796 records. The columns represent features like smoking, alcohol drinking, physical health, age, and other diseases like kidney or skin cancer to predict the possibility of having heart disease in the future [27].
-

4.2 Preprocessing

For each dataset that is fed to the models of the presented approach, it is divided into 75% training data and 25% testing data. The Min-Max data normalization is also used to reshape the data in the 0–1 range.

4.3 Feature Extraction

Feature extraction is a significant stage in the learning process; the features are extracted by utilizing vector space representation. The feature extraction process is utilized for extracting features from any dataset, including formats that are not suitable for machine learning.

Linear discriminant analysis (LDA) is a method that identifies linear combinations of features that recognize or distinguish two or more classes. The combination result is used as a linear classifier or, more importantly, for dimensionality reduction prior to subsequent classification. The basic principle of LDA is to find a linear transformation that makes feature clusters most separable after the transformation, which can be done by evaluating the scatter matrix. The main goal of this operation is to optimize inter-class scatter matrix estimation while lowering within-class scatter matrix calculation. This step is used for the MCIWC dataset, which contains image data.

4.4 The CNN model

consist of 10 convolution layers with a kernel size of 5*5, 5 max pooling layers of size 2*2, a ReLu activation function after each convolution layer, a drop-out layer with a 0.25 value, a fully connected layer of size 512 nodes, and a fully connected layer, which comply with the SGD optimizer with a learning rate of 0.001 and a momentum term of 0.9. The model was trained for 100 epochs with a batch size of 1. Figure 1 illustrates the structure of the CNN model.

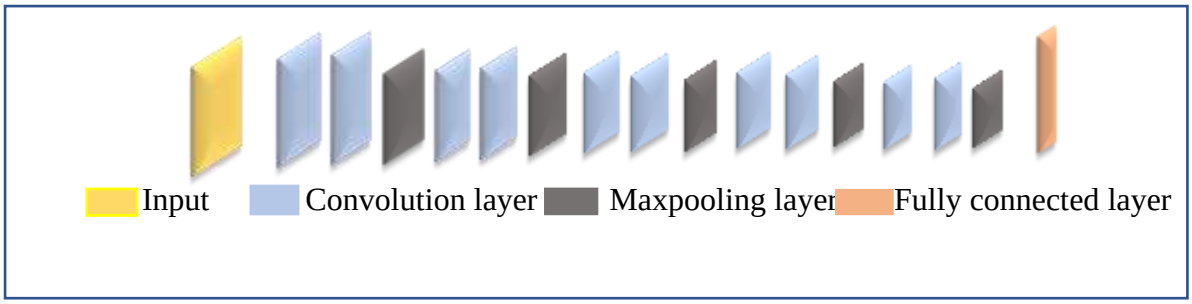


Figure 1: CNN model architecture

4.5 RNN Models

Two RNN models were constructed: GRU and LSTM. The GRU is used for the classification task because of its short memory, while the LSTM is used for the prediction task because of its longer memory than the GRU, which makes it more convenient for the prediction task than the GRU model.

I. The LSTM model has one input layer that takes the features of the dataset as input. Three LTSM layers were used, with two time steps (the number of previous steps the model saved to be used for predicting the next step), one dens layer, and an Adam optimizer. The model was trained for 100 epochs with a batch size of 1. The mean square error is used as an evaluation metric, as illustrated in Figure 2.

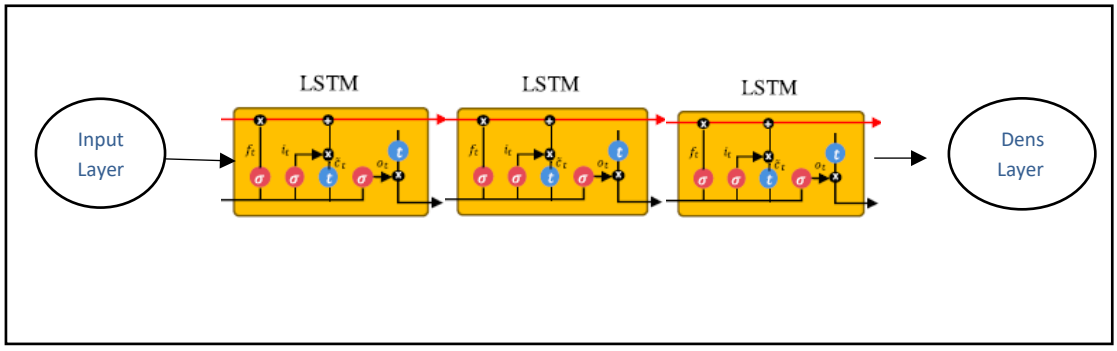


Figure 2: LSTM model architecture

II. The GRU model has one input layer, three hidden layers (GRU) with Tanh activation functions and sigmoid functions for the recurrent process, and one output layer with a linear activation function. The model was trained using the Adam optimizer for 100 epochs with batch size 1 and time step 1. The mean square error is used as an evaluation metric, as illustrated in Figure 3.

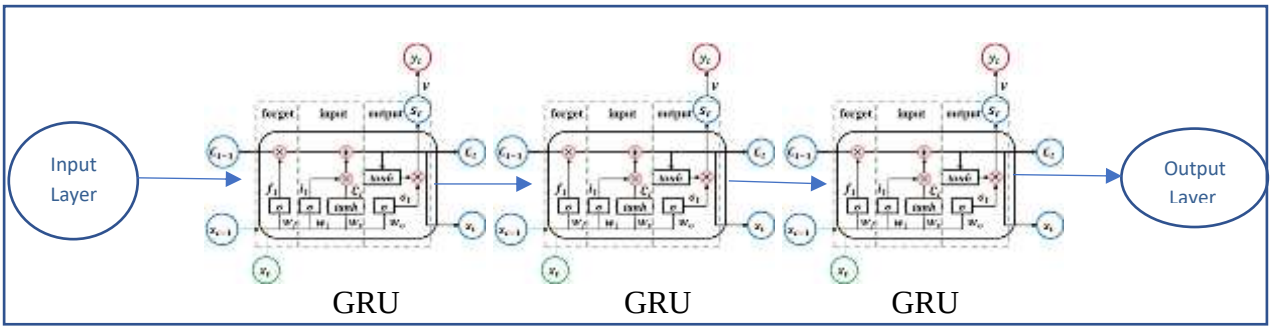


Figure 3: GRU model architecture

5. **Results**

The results were obtained from 3 different models and 4 different datasets, two for classification and the other two for prediction. The results are divided according to the model used.

5.1 Feature extraction

Using the LDA method, the proposed models of LSTM and GRU did a great job of improving accuracy. They also did a good job of extracting and choosing the most important features, like the color of the sky, the shadow, the snowflakes, the contrast, and the saturation. Less important features, like shapes in pictures that aren't affected by the weather, were not taken into account in this process. The efficiency and accuracy of LSTM and GRU models are clearly improved with the LDA method due to its ability to extract and select only the features that have a direct effect on the process, which leads to focusing on the important features.

5.2 CNN model

The CNN model gave remarkable results for the classification task with the BCD and MCIWC datasets, but modest results were achieved for the prediction task with the MAPP and HDP datasets compared with the other models (LSTM and GRU). Figures 4 and 5 show the experimental results for the CNN model.

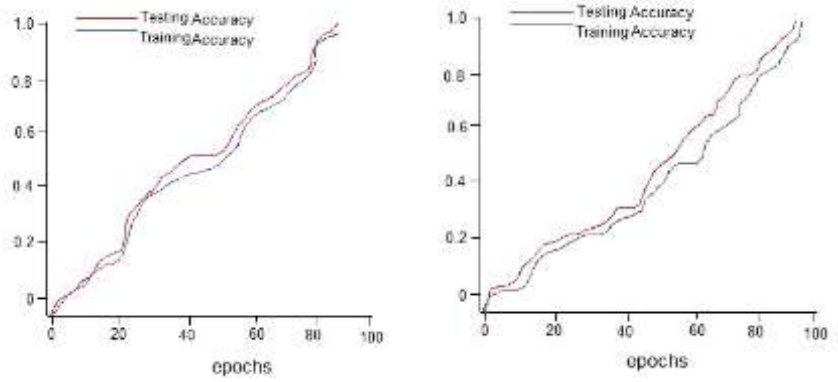
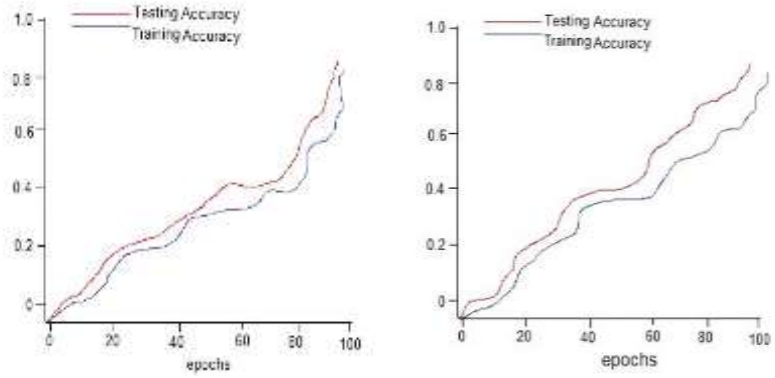


Figure 4: CNN model classification accuracy: (a) classification accuracy for the BCD dataset; (b) classification accuracy for the MCIWC dataset

From the above figure, it is obvious that the CNN achieved excellent results in the classification task for both datasets, exceeding 97% accuracy for both training and testing accuracy. The increase in accuracy was stable along the epochs, exceeding 90% around the 85th epoch. While the CNN model reached almost 98% accuracy in the classification task, it didn't exceed 85% accuracy in the prediction task. That is due to the structured nature of the CNN, in which the model does not take into consideration the data from the previous state and only depends on the current data. In order to make precise predictions, the model must take into consideration a series of data points to predict the next action. CNN does not operate accordingly, which makes it less suitable for prediction tasks and more convenient for classification tasks. Figure 5 illustrates the prediction accuracy of the CNN model.

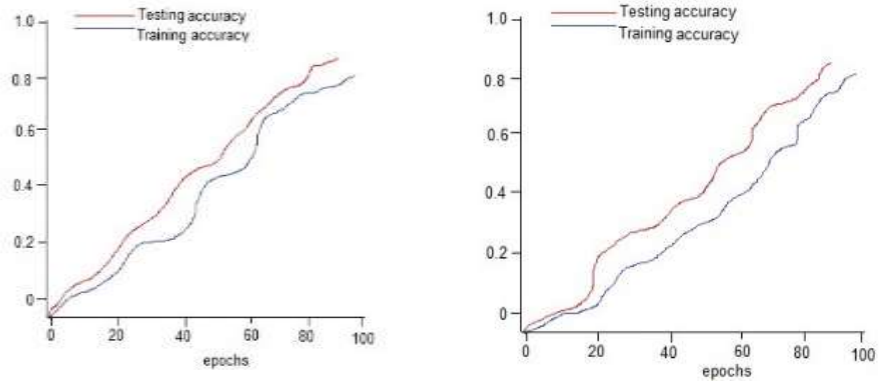


(a) (b)

Figure 5: CNN model prediction accuracy: (a) prediction accuracy for the MAPP dataset; (b) prediction accuracy for the HDP dataset

5.3 LSTM model

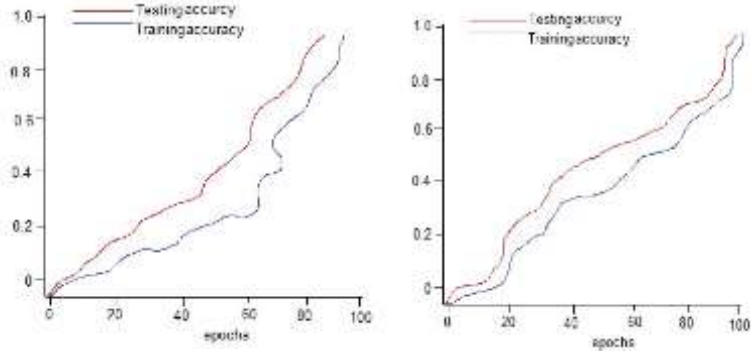
As expected, the LSTM model did well in the prediction task for the MAPP and HDP datasets, outperforming the CNN model with 97% and 96% accuracy for the MAPP and HDP datasets, respectively. This is because the model can remember the previous data (series data) and predict the next data based on the history of the data, as shown in Figure 6.



(a) (b)

Figure 6: RNN model classification accuracy (a) classification accuracy for the BCD dataset; (b) classification accuracy for the MCIWC dataset

The superior performance of the LSTM with the prediction task didn't comprehend the performance of the classification task of the model; the model achieved very modest results regarding the classification task for the two datasets (BCD and MCIWC), reaching 83% and 86% accuracy for the BCD and MCIWC datasets, respectively, for the classification task, as shown in Figure 7.



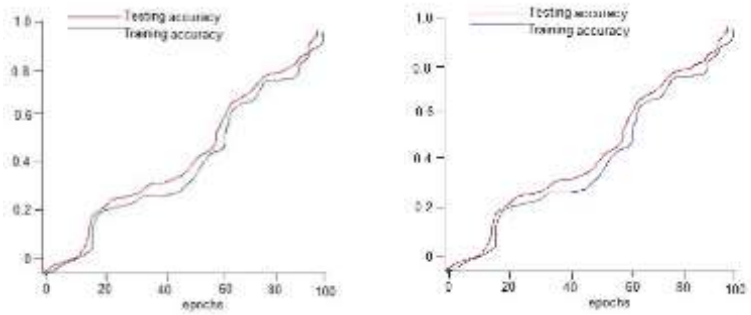
(a) (b)

Figure 7: RNN model prediction accuracy: (a) prediction accuracy for the MAPP dataset; (b) prediction accuracy for the HDP dataset

5.4 GRU model

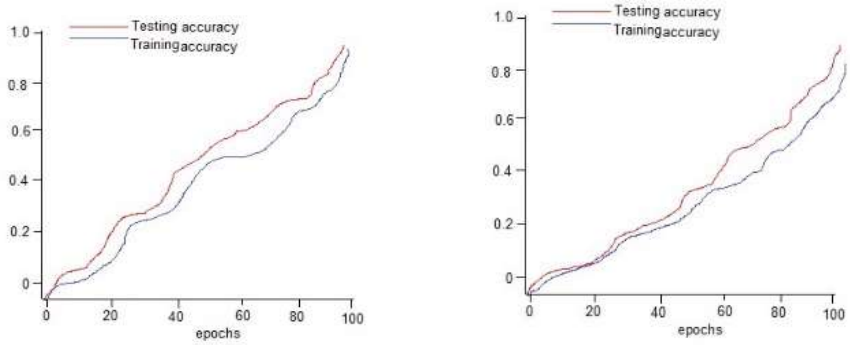
The performance of this model is acceptable for both task prediction and classification due to its structured nature and the amount of data that it remembers. It has the ability to outperform the LSTM model in classification tasks, achieving 89% and 90% accuracy for the BCD and MCIWC datasets, respectively. It also outperforms the CNN model in prediction tasks, achieving 94% and 95% accuracy for the MAPP and HDP datasets, respectively. From the experimental results of the three models, the GRU is the only model that reached acceptable and very close results for the four datasets and for both tasks.

Figures 8 and 9 show the results for the GRU model.



(a) (b)

Figure 8: RNN model classification accuracy: (a) classification accuracy for the BCD dataset; (b) classification accuracy for the MCIWC dataset



(a) (b)

Figure 9: GRU model prediction accuracy: (a) prediction accuracy for the MAPP dataset (b) prediction accuracy for the HDP dataset

5.5 Result Comparison

This section includes a comparison among the results of the proposed models, in addition to a comparison with related works. Table 1 illustrates the results of the proposed models for each dataset that was used in this research compared with related works.

Table 1: A Comparison Among the Proposed Models and Related Works

Model / Dataset	CNN	LTSM	GRU	RNN
BCD	97%	83%	89%	
MCIWC	98%	86%	90%	
MAPP	85%	97%	94%	
HDP	83%	95%	95%	
SVD[7]	87%			86%

SentiC[8]	82%	84%	86%	
TE[8]	77%	77%	78%	
QRM[8]	71%	71%	69%	
POS tagging[8]	94%	93%	93%	
Self-built DS[9]	94%			97%

The first four rows represent the results of the proposed models as a comparison with related works in terms of CNN, LSTM, and GRU, except for the works in [7] and [9], which used CNN and RNN. Also, it must be noted that each model has its own architecture for designing and training.

6. Conclusion

Deep learning has a great impact on many tasks, such as classification and prediction; the achievement of each task depends on a predefined DL model. That is, CNN outperforms classification tasks, while RNN, in terms of LSTM, outperforms prediction tasks better than the rest of the DL models.

Many conclusions can be drawn from the following behavior of the proposed and presented models: Firstly, all three proposed models succeed in the classification task, with the CNN model having a relative advantage. Secondly, all three proposed models succeed in the prediction task, with the RNN model having a relative advantage. Thirdly, the GRU model overcame reality and itself by producing outstanding results in both classification and prediction tasks for all four datasets.

References

[1] Nada H. Ali, Matheel Emad Abdulmunem, and Akbas Ezaldeen Ali, "Constructed model for micro-content recognition in lip reading based deep learning," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2557-2565, 2021.

[2] Tom Taulli, Artificial Intelligence Basics: A Non-Technical Introduction, Apress, 1st edition, Monrovia, CA, USA, 2019.

[3] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186

[4] T. Bezdan, N. Bačanin Džakula, "Convolutional Neural Network Layers and Architectures," in *Sinteza 2019 - International Scientific Conference on Information Technology and Data Related Research*, Belgrade, Singidunum University, Serbia, 2019, pp. 445-451. doi:10.15308/Sinteza-2019-445-451

[5] Akshay Kulkarni and Adarsha Shivananda, Natural Language Processing Recipes, Unlocking Text Data with Machine Learning and Deep Learning using Python, *Apress Berkeley*, CA, USA, 2019.

[6] Rinki Gupta and Sreeraman Rajan, "Comparative Analysis of Convolution Neural Network Models for Continuous Indian Sign Language Classification," *Procedia Computer Science*, vol. 171, pp. 1542-1550, 2020. https://doi.org/10.1016/j.procs.2020.04.165.

[7] Sidra Abid Syed, Munaf Rashid, Samreen Hussain, and Hira Zahid, "Comparative Analysis of CNN and RNN for Voice Pathology Detection," *BioMed Research International*, vol. 2021, Article ID 6635964, 8 pages, 2021. https://doi.org/10.1155/2021/6635964

[8] W. Yin , K. Kann , M. Yu, H. Schutze, "Comparative Study of CNN and RNN for Natural Language Processing," arXiv:1702.01923v1,7 Feb 2017. https://doi.org/10.48550/arXiv.1702.01923

[9] Yousfi, Siham, Rhanoui, Maryem, and Mikram, Mounia, "Comparative Study of CNN and LSTM for Opinion Mining in Long Text," *Journal of Automation, Mobile Robotics and Intelligent Systems*, Vol. 14, No. 3, pp. 50-55, 2021. Doi:10.14313/JAMRIS/3-2020/34

[10] S. Seo, C. Kim, H. Kim, K. Mo and P. Kang, "Comparative Study of Deep Learning-Based Sentiment Classification," in *IEEE Access*, vol. 8, pp. 6861-6875, 2020, doi: 10.1109/ACCESS.2019.2963426.
[11] P. Pundir, V. Gomanse , and N. Krishnamacharya, 'Classification and Prediction techniques using Machine Learning for Anomaly Detection," *International Journal of Engineering Research and Applications*, vol. 1, no. 4, pp. 1716-1722, 2006.
[12] Filippo Maria Bianchi, Enrico Maiorino, Michael C. Kampffmeyer, Antonello Rizzi, and Robert Jenssen, Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis, Springer Cham, 2017.
[13] S. Yang, X. Yu and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, Shanghai, China, 2020, pp. 98-101, doi: 10.1109/IWECAI50956.2020.00027.
[14] B. C. Mateus, M. Mendes, J. T. Farinha, R. Assis, and A. M. Cardoso, "Comparing LSTM and GRU Models to Predict the Condition of a Pulp Paper Press," Energies, vol. 14, no. 21, p. 6958, Oct. 2021, doi: 10.3390/en14216958. [Online]. Available: http://dx.doi.org/10.3390/en14216958
[15] Sandro Skansi, Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence, *Springer Cham*, 2018.
[16] W. M. Salih, I. Nadher, A. Tariq, "Modification of Deep Learning Technique for Face Expressions and Body Postures Recognitions", *International Journal of Advanced Science and Technology*, vol. 29, No. 3s, pp. 313-320, Mar. 2020.
[17] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, *MIT Press*, 2016.
[18] O. Sharma, "A New Activation Function for Deep Neural Network," *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, 2019, pp. 84-86, doi: 10.1109/COMITCon.2019.8862253.
[19] Siddharth Sharma, Simone Sharma, and Anidhya Athaiya, "Activation functions in neural networks," *International Journal of Engineering Applied Sciences and Technology*, vol. 4, no. 12, pp. 310-316, 2020.
[20] T. N. Nguyen, F. Dernoncourt, and T. H. Nguyen, "On the Effectiveness of the Pooling Methods for Biomedical Relation Extraction with Deep Learning," *Proceedings of the 10th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pp. 18–27, 2019.
[21] C. C. Aggarwal, Neural Networks and Deep Learning, *Springer Cham*, 2018.
[22] S. Ruder, Sebastian, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747, Spet*. 2016. https://doi.org/10.48550/arXiv.1609.04747
[23] D. P. Kingma, J. L. Ba, "Adam: A Method for Stochastic Optimization," *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.
[24] N. Yadav, Breast Cancer Classification, 2021, Can Be Found At Https://Www.Kaggle.Com/Code/Niteshyadav3103/Breast-Cancer-Classification/Notebook
[25] S. Sharma, multiclass-images-for-weather-classification, 2020, can be found at https://www.kaggle.com/datasets/somesh24/multiclass-images-for-weather-classification
[26] R. ROY, LSTM- multivariate air pollution prediction, 2021, can be found at https://www.kaggle.com/code/rupakroy/lstm-multivariate
[27] M. ELSAYED, Heart Disease Prediction, 2022, can be found at https://www.kaggle.com/code/andls555/heart-disease-prediction