# Enhanced Recurrent Neural Network (RNN) For Heart Disease Risk Prediction Using Framingham Datasets

Surenthiran Krishnan[1], Pritheega Magalingam[1], Roslina Ibrahim[1]

[1]*Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, 54100 Kuala Lumpu*

*surenthiran@mod.gov.my, mpritheega.kl@utm.my, iroslina.kl@utm.my*

**Article history**

Received:
20 April 2023

Received in revised form:
18 May 2023

Accepted:
23 June 2023

Published online:
27 June 2023

*Corresponding author

mpritheega.kl@utm.my

***Abstract***

*Heart disease is one of the leading causes of death globally, which takes 17.9 million lives each year. The existing heart disease prediction techniques have a gap that does not consider the smoking attributes from the heart disease data. So, the accuracy is based on the limited number of medical data and the deep learning model. The existing deep learning models which use the Recurrent Neural Network (RNN) for heart disease prediction consume more processing and analysing time, mainly due to the delay of data retrieval. This delay causes the prediction process to become slower and leads to a moderate prediction only. The backpropagation of the RNN with an update gate and internal memory to carry the updated data cause a minor data glitch that leads to lower accuracy. Therefore, an efficient heart disease prediction model is very crucial to provide early detection among patients. This research proposes a Heart Disease Risk Prediction Model (HDRPM) with an enhanced RNN to improve the prediction accuracy using Framingham heart disease datasets. The specificity and sensitivity are imposed to improve the quality of the predictions. Sensitivity measure is used for detecting patients with heart disease perfectly and specificity measure is used for detecting patients without the disease perfectly. Besides the accuracy and quality of the prediction problem, the imbalance of minority classes in the dataset occurred in most deep learning prediction fields. This research aims to improve the quality of imbalanced Framingham datasets using Synthetic Minority Over-sampling Technique (SMOTe), which will synthetic instances in a small class to be equalized. The existing RNN model faces vanishing gradients that impede the learning of long data sequences. These gradients that carry information in the RNN cells will become smaller gradually till it minimises the parameter updates and leads to poor learning. For this purpose, the presence of multiple Gated Recurrent Unit (GRU) is used to overcome the vanishing gradients and ensure the hidden layers. The neurons of RNN rapidly cater for the essential information during the training and validation phase of the HDRPM. The integration of multiple GRU with the RNN, operating on the Tensorflow as back-end and Keras as the core for the neural network library has increased the performance of the proposed model. The proposed model provides up to 98.78%, the highest accuracy achieved compared to related previous work, which is a quantum neural network model with 98.57. This HDRPM is expected to significantly contribute to early detection of heart disease patients.*

***Keywords:*** *Framingham, Recurrent Neural Network, SMOTe, Gated Recurrent Unit, Tensorflow*

* *Corresponding author. mpritheega.kl@utm.my*

## 1. Introduction

Cardiovascular disease (CVD) is one of the leading causes of mortality worldwide[1]. Exhaustive risk factors are associated with CVD and managing these risk factors is complex but could prevent deaths if detected earlier. In recent studies, various prediction models were developed to predict a high risk in developing CVD [2] by using machine learning and deep learning [3]. The deep learning approach provides higher accuracy and precision over machine learning techniques in heart disease prediction [4], [5]. Most of the heart disease datasets in previous research are imbalance in the classes and even have some irrelevant and null data [6], [7]. The vast imbalance of dataset distribution is another problem for healthcare analysis and particularly for heart disease classification [8]. This imbalanced data will affect the performance of a prediction model and its output. This imbalanced data uses a small instance of class and goes through a deep learning process with a dominant data of class that might be overwritten by the dominant data [9]. Deep learning using neural network that has outstanding performance capabilities for classification of data [10] that can make heart disease risk prediction more efficient. Datasets that are deemed to be imbalanced may affect the deep analysis process [11]. Deep learning models for heart disease prediction are based on the dependent variables and factors of heart disease [12]. Thus, backpropagation is essential in deep learning techniques for heart disease prediction rather than feed-forward neural network [13]. The RNN has backpropagation but faces time delay when the data passes through between the layers of the neural network. This delay will gather a small amount of data in the built-in memory of RNN and will lead to data glitch. Therefore, GRU is introduced and implemented between the GRU in order to process faster the data between layers to avoid data glitch in the overall neural network [6].

## 2. Recurrent Neural Network (RNN)

RNN is a neural network in deep learning where it uses the previous state data as input to determine the next state of data as the output. RNN has built-in memory that holds a small amount of data from the previous state that can study the past behaviour pattern of data, enabling it to determine the next state of data more efficiently [14]. In traditional neural networks, all the inputs and outputs are independent of each other, but for RNN it is unique where it is capable of predicting next state of data based on the past behaviour pattern and data that has been stored in its hidden memory [15]. RNN can solve the prediction issue in time sequence with the hidden layers and built-in memory. The primary and most crucial feature of RNN is a hidden state, which remembers some information about a sequence [16]. The main drawback of RNN is its exploding and vanishing gradients during model training.

### 2.1 Process in RNN

RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers in the proposed model consisting of RNN and GRU. The previous state is stored in the RNN's memory by giving each output as input to the following hidden layer [17]. RNN training performs a frequent update of the input in its neural network. Then, it estimates the current state based in the latest input and the previous state [18].

RNN can perform huge training sets to generalise the model and increase the probability of exact predictions. The predicted output of RNN is compared to the actual output, and the gap of the predicted result and the actual result will be calculated and updated in the RNN. This makes the RNN to get update frequently and reduces the gap of errors, minimising the repetitions of the same mistakes. Back-propagation technique is used to send the updates to RNN and update the weights of the RNN [19].

## 2.2 Advantages of Recurrent Neural Network

RNN records most information in time sequence. It is useful in time series prediction only because of the feature to remember previous inputs as well [17]. RNN can purge irrelevant data that is unused from its memory to allow other important data processing in RNN. Essential memory stored in RNN's memory makes it quickly and efficiently retrieve the past behaviour pattern of data.

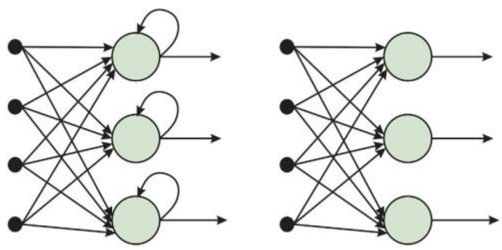

**Figure 1. RNN Architecture with Back Propagation (Left) and Feed Forward Neural Network (Right)**

## 2.3 Disadvantages of Recurrent Neural Network

RNN is facing gradient vanishing and exploding problems due to the built in memory in the RNN itself [17]. The data retrieval from the RNN's memory may be complicated and cause some delay for a long sequence of the data queue. It cannot process very long sequences if using tanh or relu as an activation function [18]. The exploding gradient is an issue that occurred in RNN and in most of neural networks during the model's training. Gradients are used during training to update the network weights where the best gradients are the small updates on the model generalisation. The instable gradients can lead to a decreased learning rate of the RNN and its final output as well. Gradient clipping and weight regularization are used to overcome the exploding gradients issues. Exploding gradients can reduce the model's accuracy drastically during training and the learning cannot be completed. The values of the weights can also become so large as to overflow and result in something called NaN values. NaN values, which stands for not a number, are values that represent undefined or unrepresentable values.

A vanishing gradient problem occurs with the sigmoid and tanh activation function because the derivatives of the sigmoid and tanh activation functions are between 0 to 0.25 and 0–1. So, the small updated weight values do not affect the whole weight because there is only a slight difference between the old and new weight values. This leads to vanishing gradient problem [20].

## 3. Research Methodology

The model design of the proposed model consists of 3 main components which are data collection, data processing, and data visualization as shown in Figure 2. First component will collect the standard format of .csv from Framingham dataset and overcome the imbalance issues with SMOTe. The second component is the core of the proposed model which has the improvised RNN with GRU supported by Tensorflow and Keras. Keras constructed the neural network while Tensorflow runs the deep analysis process between all the layers between the RNN and GRU. This RNN with perfectly balanced with GRU able to reduce the complexity of increasing parameters and memorizing each previous output. Data visualization shows the accuracy and other processing parameters as the third component. The entire model is developed in Python 3 and supported by various external libraries. The proposed model is operated in two operating system, macOS and Ubuntu 18.04LTS. SMOTe creates synthetic observations based upon the existing minority observations.

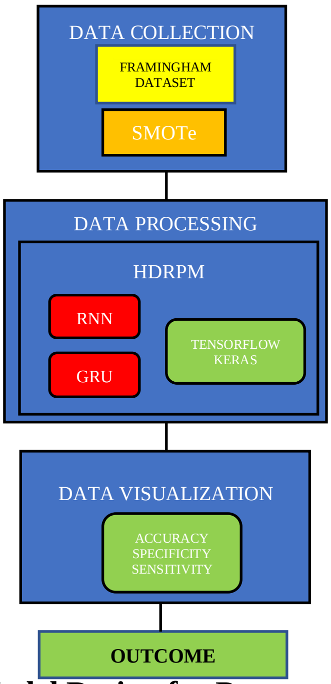


**Figure 2. Model Design for Proposed Model**

### 3.1 Datasets

Framingham dataset has 16 features of 4240 individuals for heart disease study and has 1% missing values. 3596 are in the majority class which is negative and 644 in the minority class which is positive. So, the positive patients over the negative patients are smaller, about 15.19% [21]. The patients' age is from 28 to 62 years old, that were lived in Framingham. The Framingham datasets clearly show the environmental factors that influence the development of CVD in men and women. Examination of participants has taken place every two years and the cohort has been followed for morbidity and mortality over that time period.

### 3.2 SMOTe

SMOTe is an improvised technique to solve the imbalanced data issues in machine learning and deep learning research [22]. The SMOTe algorithm works by drawing a random sample from the minority class. It performs data augmentation by creating synthetic data points based on the original data points [23], [24]. It is performs oversampling or undersampling using a specific algorithm in data augmentation. The advantage of SMOTe that it is not generating duplicates, but rather creating synthetic data points that are slightly different from the original data points [25] as shown in Figure 3.

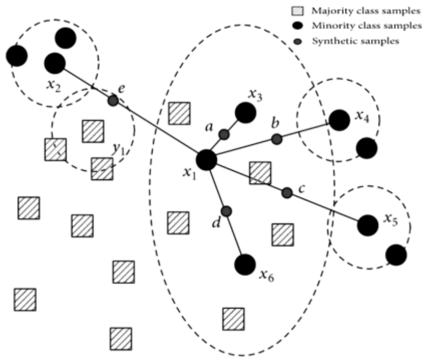


**Figure 3. SMOTe Process to Create Synthetic Data Points**

SMOTe works using a k-nearest neighbour algorithm to create synthetic data points. The step of SMOTe algorithm is to identify the minority class vector which are x1 until x6 while y1 is majority class. The second step is to decide the number of nearest numbers (k), to consider. The third step is to compute a line between the minority data points and any of its neighbours and place a synthetic point, a until d. Finally, the process will go until all minority data points and their k neighbours, till the d is balanced. These minority points, a until d, fill the gap between 2 minority classes, and make them balance compared to the previous data distribution without change its nature.

### 3.3 Imbalanced Datasets

Imbalanced datasets occur when the instance in a dataset is not balanced by its classes. Data used in healthcare have various features and classes. These datasets are high-dimensional datasets that dynamic and volatile. Sometimes, some classes of datasets are not distributed well, and even cause an uncertain on the quality of the data [26]. Several instances from one class might be significantly low, which is below 5% when compared to other classes in the same datasets. So, the datasets could portray biased behaviour patterns of datasets during model training and will drive the deep analysis towards the majority class. So, the model's accuracy would be high, not because the model is good but simply because the data is imbalanced since the features of minority class being treated as noise are often ignored. Framingham datasets are found to be imbalanced where the instances are not balanced with its classes as shown in Figure 4. These imbalances issues need to be overcome before the data is sent into the proposed model for deep analysis process in determining the heart disease risk.

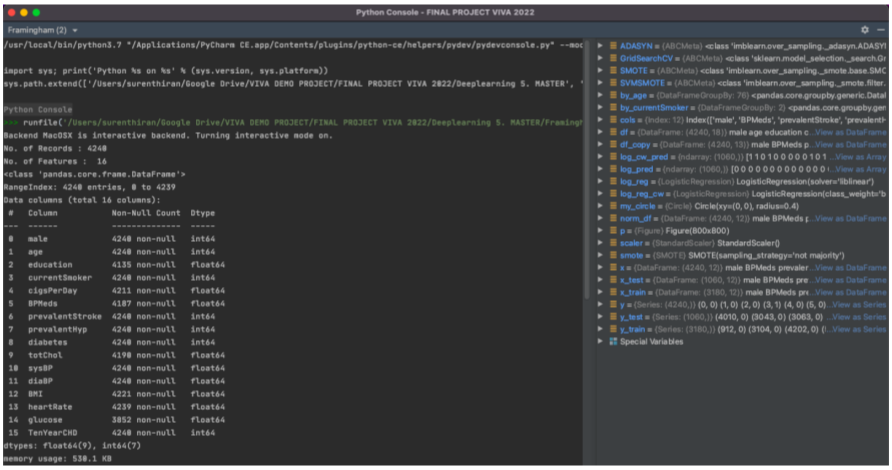

**Figure 4. Framingham Dataset in PyCharm**

Female patients without Coronary Heart Disease (CHD) are higher than male patients, as shown in Figure 5. Positive patients by gender are less than 20% overall for both genders. Therefore, the model's probability of training with the negative CHD is higher by eight times where it would update the enormous weights of negative patients. Apart from that, some other features are null values that create imbalance issues.

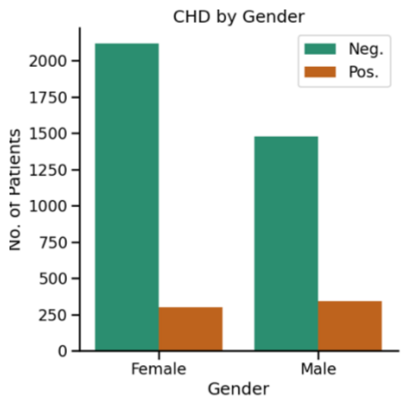


**Figure 5. Patients by Gender in Framingham Dataset**

The cigarettes consumed by patients per day with their respective age groups as shown in Figure 6. The maximum number of cigarettes a patient consumes is 70 a day. These 70 is very high for a day and could be irrelevant value for a model during its training. Normalization and oversampling are crucial to produce a quality dataset that fits for a deep learning model.

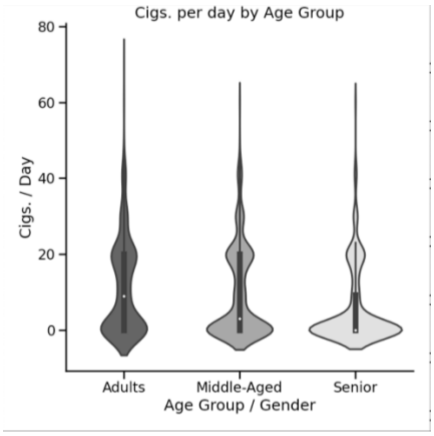


**Figure 6. Smokers' Patients in Framingham Dataset**

Nearly 90% of the patients have the normal heart rate, and only 2% have a high data rate, as shown in Figure 7.

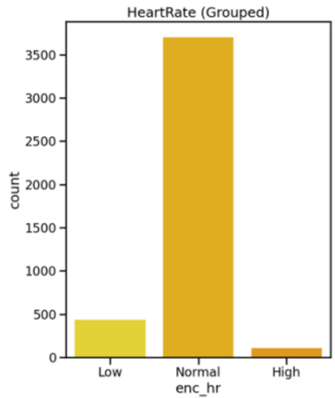


**Figure 7. Patients' Heart Rate in Framingham Dataset**

A deep learning model could not perform the training model well and was not able to generalise its model by including the data patterns have the noise and irrelevant data. A good model can generalise and adapt to the dynamic and volatility of new datasets. A model will end up learning irrelevant information during model training if a model trains with a higher number of trainings that is not suitable with the model size and the datasets. The high variance will show the state of overfitting of a model while bias shows the difference between predicted and the target value. If the model is oversimplified, the predicted value will have a huge gap from the exact value. This is important in determining the accuracy and precision of a model deep learning. The variance shows the inconsistency of the predictions on different datasets. A model's performance tested on various datasets will produce lower variance if there is consistency with its predictions.

Model generalisation will adapt the model weight and its learning towards the Framingham data which has 16 features. The Framingham datasets will be split into 70:30, respective to training and testing datasets. So, the model generalisation only applies to 705 of the Framingham datasets. Another 30% will remain unknown and untested for the model and will be tested for the first time during the testing phase. Shows that the model is trained to extract helpful data patterns and classify unseen data samples. Feature selection is imposed in the model to choose the vital features and prioritise them during the deep analysis of the proposed model. Underfitting occurs when the model has a high bias during the deep analysis. Oversimplifying and the model not capable of performing a good analysis on the Framingham datasets. The proposed model could end up overfitting and cause exploding gradient with a high variance where the training produces a good learning rate but the proposed model does not perform accurately in the testing set. The main reason is the data patterns during training were stored by the model but not able to generalize and perform its deep analysis when it comes to new datasets. The flowchart of designing the proposed model as shown in Figure 8.

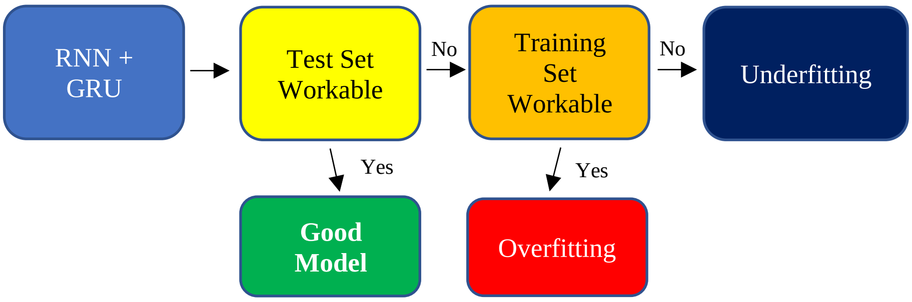


**Figure 8. Flowchart for Good Model**

The model has several neural layers stacked together. The model of RNN and GRU often lead to overfitting during the model training because the adjustment of its learning rate for new datasets. The learning rate is influenced by the batch size of the datasets and the number of epochs. Overfitting is solved by imposing the early stopping during the model training where the training is halted immediately when the learning rate decreases gradually. Dropout is used to reduce unused nodes I. model during training to make the model more flexible and simpler. This will make the transmission of data across the layers in the neural network become faster and more efficient. It simplifies the proposed model by neglecting irrelevant or less impact nodes without affecting the accuracy of the prediction. The best total recurrent layers for the model consists of 6 GRUs and 6 RNNs as per shown in Figure 9. These combinations of RNN and GRU can construct a good model that is able to generalize with Framingham datasets.

The learning rate for the model increased until the recurrent layers of 8, 4 RNN and 4 GRU. After the recurrent layers is increased, the learning rate is reduced immediately but able to overcome the reduction and increase back with the increase of GRU numbers. The main reason because the backpropagation in RNN is not strong enough to purge the storing data, make the RNN process become slower. So, the presence of GRU at this point could clear the queuing data in the RNN's layer faster and frequently. The peak of the model training achieved 98.78% with 6 RNN and 6 GRU for the Framingham datasets. The model with more than 12 recurrent layers is large to perform deep analysis for Framingham dataset. So, they could not produce a good accuracy in heart disease prediction.

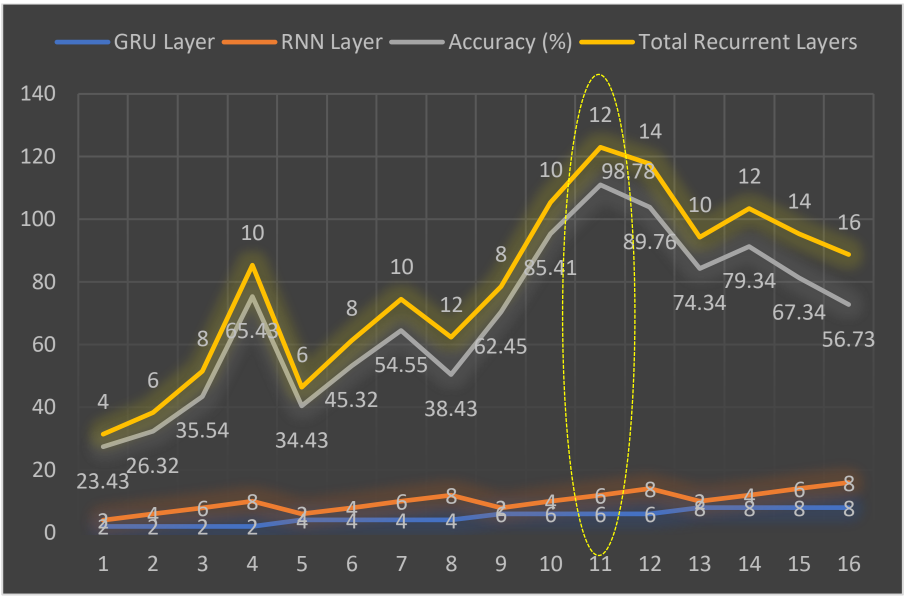


**Figure 9. Stacked Line Graph for Combination of Recurrent Layers in Heart Disease Risk Prediction Model**

## 4. Results

SMOTe was used to implement sampling on not majority classes, it is similar to Minority Over-sampling using imblearn's. Imbalanced-learn which is used as imblearn is running on scikit-learn and is responsible with classification with imbalanced classes. SMOTe successfully generated 2688 samples of each class, making the total sampling of 7,328. The optimum number of trainings can reduce losses and increase the proposed model's accuracy as shown in Figure 10. The accuracy increases gradually with the increase of epochs, where the model can generalise well during the model training. During the epochs of 750, the model training was at the peak with the highest learning rate until it touched the epochs of 860 and achieved the highest accuracy of 98.78%. The loss increases after the epochs of 860 where the accuracy reduces tremendously to 31.98% at epochs of 1250. This shows the model was facing overfitting where the optimum model training has exceeded, and the model is not able to provide a good prediction.

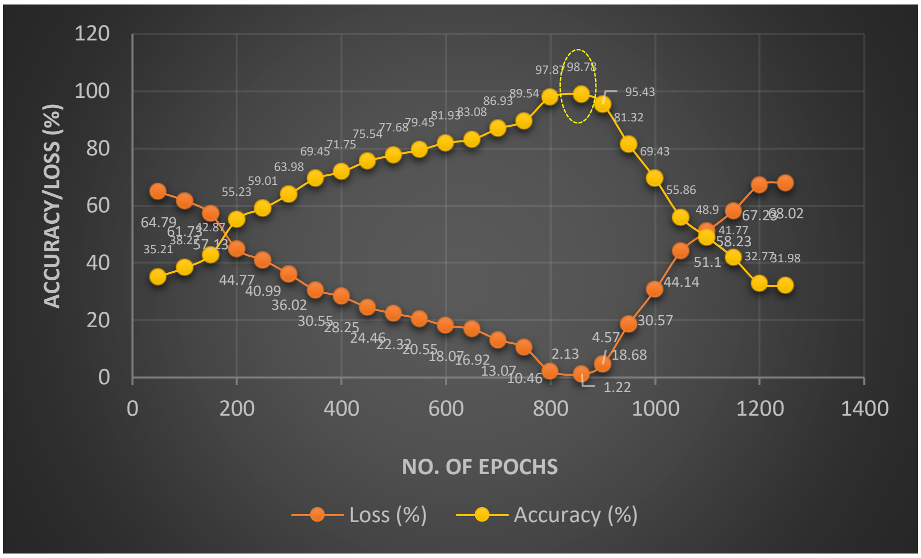


**Figure 10. Line Graph for Loss and Accuracy in Heart Disease Prediction Model**

Five different types of RNN algorithms with two datasets are used, a simple RNN that takes an input sequentially and iteratively updates an output, a GRU that decides to take a current input for generating output, a LSTM that embeds a carrier that delivers the former value to the future, RNN with LSTM, and RNN with GRU. In the heart disease prediction model, RNN with multiple GRU outperformed the other algorithms, as shown in Table 1. Framingham datasets with RNN and GRU able to achieve 81.43%. This accuracy able to be increased up to 98.78% by improving the quality of the Framingham datasets using SMOTe. This result implies that RNN with multiple GRU shows a better deep analysis and process in the heart disease prediction model.

**Table 1. Accuracy with Different RNN Variants for Different Datasets**

| Type Neural Network | Cleveland Datasets | Cleveland Datasets (SMOTe) | Framingham Datasets | Framingham Datasets (SMOTe) |
|---|---|---|---|---|
| RNN | 60.12 | 73.25 | 40.13 | 49.87 |
| LSTM | 40.54 | 71.34 | 35.65 | 59.13 |
| GRU | 70.23 | 78.85 | 56.71 | 61.84 |
| RNN + LSTM | 73.00 | 83.38 | 65.83 | 77.18 |
| Proposed Model (RNN + GRU) | 100% | 99.18% | **81.43%** | **98.78%** |

The novelty of this research is the highest prediction accuracy achieved for the enhanced RNN model for heart disease risk prediction which is **98.78% for Framingham datasets**. This research provides a HDRPM, a model constructed with RNN and GRU with Keras and performs the deep processing with Tensorflow. SMOTe can improve the data quality and eventually increase the model's accuracy.

## 6. Conclusion

The SMOTE algorithm is a solution for imbalanced data in classification problems. It can overcome imbalanced issues in datasets such as performing an oversampling by creating synthetic data points that are relatively similar to the original ones. GRU can rapidly solve the lagging in RNN and deliver data across RNN layers. The proposed model has achieved the highest prediction accuracy using the Framingham dataset. Datasets for this research have limitations where it has limited features and noisy data. Future work can be done using different datasets and neural network models.

## Acknowledgements

We want to express our highest gratitude and appreciation to the Creator, our family members who support us throughout the completion of this research. To all the great souls who keep inspiring us to move forward in this research. We acknowledge the motivations and inspirations of all the Universiti Teknologi Malaysia's lecturers and staffs.

## References

[1] J. M. Sung *et al.*, "Development and verification of prediction models for preventing cardiovascular diseases," *PLoS One*, vol. 14, no. 9, pp. 1–12, 2019, doi: 10.1371/journal.pone.0222809.

[2] S. Gao, Y. Zheng, and X. Guo, "Gated recurrent unit-based heart sound analysis for heart failure screening," *Biomed. Eng. Online*, vol. 19, no. 1, pp. 1–17, 2020, doi: 10.1186/s12938-020-0747-x.

[3] S. Tuli *et al.*, "HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments," *Futur. Gener. Comput. Syst.*, vol. 104, pp. 187–200, 2020, doi: https://doi.org/10.1016/j.future.2019.10.043.

[4] K. Harimoorthy and M. Thangavelu, "Multi-disease prediction model using improved SVM-radial bias technique in healthcare monitoring system," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 3, pp. 3715–3723, 2021, doi: 10.1007/s12652-019-01652-0.

[5] M. Raihan *et al.*, "Risk Prediction of Ischemic Heart Disease Using Artificial Neural Network," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Feb. 2019, pp. 1–5, doi: 10.1109/ECACE.2019.8679362.

[6] S. Dutta and S. K. Bandyopadhyay, "Early Detection of Heart Disease Using Gated Recurrent Neural Network," vol. 3, no. 1, pp. 8–15, 2020.

[7] S. Sajeev *et al.*, "Deep Learning to Improve Heart Disease Risk Prediction," in *Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting*, 2019, pp. 96–103.

[8] A. Baccouche, B. Garcia-Zapirain, C. C. Olea, and A. Elmaghraby, "Ensemble deep learning models for heart disease classification: A case study from Mexico," *Inf.*, vol. 11, no. 4, pp. 1–28, 2020, doi: 10.3390/INFO11040207.

[9] L. Yan *et al.*, "An interpretable mortality prediction model for COVID-19 patients," *Nat. Mach. Intell.*, vol. 2, no. 5, pp. 283–288, 2020, doi: 10.1038/s42256-020-0180-7.

[10] F. Jiang *et al.*, "Artificial intelligence in healthcare: Past, present and future," *Stroke Vasc. Neurol.*, vol. 2, no. 4, pp. 230–243, 2017, doi: 10.1136/svn-2017-000101.

[11] T. Johnson, L. Zhao, G. Manuel, H. Taylor, and D. Liu, "Approaches to therapeutic angiogenesis for ischemic heart disease," *J. Mol. Med.*, vol. 97, no. 2, pp. 141–151, 2019, doi: 10.1007/s00109-018-1729-3.

[12] M. A. Alim, S. Habib, Y. Farooq, and A. Rafay, "Robust Heart Disease Prediction: A Novel Approach based on Significant Feature and Ensemble learning Model," in *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2020, pp. 1–5, doi: 10.1109/iCoMET48670.2020.9074135.

[13] H. Apaydin, H. Feizi, M. T. Sattari, and M. S. Colak, "Comparative Analysis of Recurrent Neural Network," pp. 1–18, 2020.

[14] P. P. Prasad and F. S. Francis, "Medical Data Classification Based on SMOTE and Recurrent Neural Network," no. June, 2020, doi: 10.35940/ijeat.C5444.029320.

[15] J. Zhao, D. Zeng, S. Liang, H. Kang, and Q. Liu, "Prediction model for stock price trend based on recurrent neural network," *J. Ambient Intell. Humaniz. Comput.*, 2020, doi: 10.1007/s12652-020-02057-0.

[16] S. Krishnan, P. Magalingam, and R. B. Ibrahim, “Advanced recurrent neural network with tensorflow for heart disease prediction,” *Int. J. Adv. Sci. Technol.*, vol. 29, no. 5 Special Issue, 2020.
[17] A. H. Ribeiro and T. B. Schön, “Beyond exploding and vanishing gradients : analysing RNN training using attractors and smoothness,” 2020.
[18] T. Das and G. Saha, “Addressing big data issues using RNN based techniques,” *J. Inf. Optim. Sci.*, vol. 40, no. 8, pp. 1773–1785, 2019, doi: 10.1080/02522667.2019.1703268.
[19] L. Sun, Y. Wang, J. He, H. Li, D. Peng, and Y. Wang, “A stacked LSTM for atrial fibrillation prediction based on multivariate ECGs,” *Heal. Inf. Sci. Syst.*, vol. 8, no. 1, 2020, doi: 10.1007/s13755-020-00103-x.
[20] I. Javid, A. K. Z. Alsaedi, and R. Ghazali, “Enhanced accuracy of heart disease prediction using machine learning and recurrent neural networks ensemble majority voting method,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 3, pp. 540–551, 2020, doi: 10.14569/ijacsa.2020.0110369.
[21] R. Narain, S. Saxena, and A. K. Goyal, “Cardiovascular risk prediction: A comparative study of framingham and quantum neural network based approach,” *Patient Prefer. Adherence*, vol. 10, pp. 1259–1270, 2016, doi: 10.2147/PPA.S108203.
[22] M. Mahin *et al.*, “Selectively Oversampling Difficult Positive Samples from Imbalanced Data for Preprocessing,” in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, 2019, pp. 1–6, doi: 10.1109/ICCIT48885.2019.9038575.
[23] T. Sasada, Z. Liu, T. Baba, K. Hatano, and Y. Kimura, “A resampling method for imbalanced datasets considering noise and overlap,” *Procedia Comput. Sci.*, vol. 176, pp. 420–429, 2020, doi: 10.1016/j.procs.2020.08.043.
[24] T. E. Tallo and A. Musdholifah, “The Implementation of Genetic Algorithm in Smote (Synthetic Minority Oversampling Technique) for Handling Imbalanced Dataset Problem,” *Proc. - 2018 4th Int. Conf. Sci. Technol. ICST 2018*, vol. 1, pp. 1–4, 2018, doi: 10.1109/ICSTC.2018.8528591.
[25] S. Krishnan, P. Magalingam, and R. Ibrahim, “Hybrid deep learning model using recurrent neural network and gated recurrent unit for heart disease prediction,” *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, pp. 5467–5476, 2021, doi: 10.11591/ijece.v11i6.pp5467-5476.
[26] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, “The impact of class imbalance in classification performance metrics based on the binary confusion matrix,” *Pattern Recognit.*, vol. 91, pp. 216–231, 2019, doi: 10.1016/j.patcog.2019.02.023.