Al-Salam Journal for Engineering and Technology
Journal Homepage: http://journal.alsalam.edu.iq/index.php/ajest
e-ISSN: 2790-4822 p-ISSN: 2958-0862

Utilizing Synthetic Tabular Data Method to Improve Heart Attack Prediction Accuracy

Randa shaker Abd-Alhussain[1] *, Hadab Khalid Obayes[2], Farah Al-Shareefi[3]

[1]College of Science for women, University of Babylon, Babylon, 51002, Iraq.
[2] College of Education for humanities studies, University of Babylon, Babylon, 51002, Iraq.
[3] College of Science for women, University of Babylon, Babylon, 51002, Iraq.

*Corresponding Author: Randa Shaker Abd-Alhussain

DOI: https://doi.org/10.55145/ajest.2024.03.01.002
Received June 2023; Accepted August 2023; Available online August 2023

ABSTRACT: Predicting heart attack possibility is a crucial step in saving human life because it is one of the leading causes of death. How improve and/or achieve as accurate as possible prediction value is a challenging task. Though Deep Learning networks, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs), have been widely used to predict medical events, the issue of providing perfect prediction outcomes is still progressing. The training data insufficiency is a major hindrance to these network's performance. Accordingly, to overcome that issue, this research develops a method for heart attack prediction by using the Conditional Tabular Generative Adversarial Network (CTGAN) model. By anticipating and warning the patient that he may be having a heart attack, he to take the appropriate preventative steps, the proposed method in this research can help to reduce deaths from heart attacks and preserve patients' lives. The suggested method uses the CTGAN model to expand the patient's laboratory test results to build a synthetic trained database. Then, the RNN and LSTM are trained on the generated dataset to enhance the prediction accuracy, and both RNN and LSTM are evaluated before and after the database expansion. The experimental results show that the accuracy of RNN and LSTM is increased to 100% and 99% from 80% and 65%, respectively and the precision of RNN and LSTM is increased to 100% for each of them from 68% and 0, respectively. Finally, sensitivity(recall) increased for RNN and LSTM from 77% and 0 to 100% and 98%, respectively.

Keywords: heart attack, RNN, LSTM, CTGAN

1. INTRODUCTION

More people die from cardiovascular illnesses than any other cause of death, making them the leading cause of death worldwide. Worldwide, cardiovascular diseases (CVDs) are regarded as one of the leading causes of death.

The World Health Organization estimates that 17.9 million individuals died from CVDs in 2016, which contributed to 31% of all deaths worldwide. Due to the high number of premature deaths during the prime of life, CVDs place a financial burden on low- and middle-income countries[1].

Heart attacks constitute a considerable cause of worldwide deaths nowadays. Prediction of a heart attack may dramatically reduce the chance of fatal events in the future. To attain that, the prediction should be precise and away from human error[2]. For these reasons, deep learning methods[3] have been extensively applied to assist clinicians in producing as accurate results as possible[4][5][6]. Common examples of such methods are Recurrent Neural Networks (RNNs)[7] and Long Short-Term Memory Networks (LSTMs)[8]. These methods have been profitably conducted to predict healthcare incidents[9]. Unfortunately, the accuracy of these methods is directly proportional to the size of the database that are training with. In other words, an inadequate training database is a drawback for their performance yields.

This research aims at addressing the above drawback by exercising a method from the deep learning field that is known as the Conditional Tabular Generative Adversarial Network (CTGAN) model[10]. CTGAN is a mathematical model that generates what is called synthetic data from a real tabular one. The synthetic term refers to the artificial and scalable generation of data instead of the real data that has been gathered from actual natural events. As a consequence, we employ CTGAN to generate an expanded trained database from the results of the patient's laboratory tests. The gained database is fed to the RNN and LSTM to enhance their accuracy rate.

*Corresponding author: randashaker1984@gmail.com

The following are the study's primary contributions:

1) Creating an automated process to predict the extent of cardiac arrest in patients with heart disease;
2) Improving the RNN and LSTM accuracy by employing the CTGAN method;
3) Implementing the CTGAN method to expand the training data set of heart patients;
4) Evaluating the effectiveness of the LSTM and RNN ways that make use of several performance measures.

The remainder of the project is structured as follows. The relevant literature is covered in Section 2. The primary approaches and procedures used in this paper are summarized in Section 3. The suggested system is described in Section 4 in great detail. While Section 6 provides a summary of this study, Section 5 describes our observations and results.

2. RELATED WORK

This section evaluates the literature that has been completed in fields that are similar to ours because part of our work involves determining the existence of heart issues and safeguarding patient information:

The authors of this study improved the internal forgetting gate input to present a unique paradigm. The irregular time interval is smoothed to provide the time parameter vector, which is then used as the input to the forgetting gate to overcome the prediction barrier. The experiment's findings support the model's efficacy, which was developed using medical information obtained from a hospital's HIS. The paper's proposed dynamic prediction model outperforms the conventional LSTM model in classification by a large margin. An accuracy of 89% was produced by the upgraded model[11].

To detect probable heart sickness, this study uses convolutional neural networks (CNN) with a cutting-edge dataset obtained from the UCI library. This dataset includes certain heart test parameters as well as usual human activities. The results show that the suggested model works better than the existing techniques stated in this study. The overall accuracy of the suggested model is 97%[12].

Researchers created a non-intrusive robot in this study. Using the most common ECG, a system based on deep learning networks may conduct basic categorization of certain ECG data, such as whether it belongs to a normal or abnormal EKG (arrhythmia present). The MIT-BIH provides access to the arrhythmia database. For LSTM and 1D-CNN, they compared performance using a range of deep learning architectures. The results can be further enhanced by adopting a more complex deep learning architecture with an interest in computing cost. The researchers also talked about how deep learning methods like 1D-CNN can be used to classify arrhythmias. The fact that the proposed method does not require any noise filtering feature engineering mechanisms is its most significant feature. The collected results demonstrate that the researcher's performance in effectively classifying an ECG as normal or arrhythmic, where accuracy is 99%, is superior to other published data[13].

This study demonstrates the potential for additional investigation into the generation of synthetic EEG data using deep learning techniques like TGAN and CTGAN. The EEG data from CTGAN exhibits higher similarity than TGAN through visualization and similarity score. The researchers attempted to use the synthesized dataset as input data for several machine learning algorithms, unlike the related research. However, this study has a problem in that machine learning models do not perform better than the real data when the synthetic data from our trials is utilized as the input data. Using data from the website www.kaggle.com, the accuracy value for all methods employed in this study ranged from 49.1% to 49.8%[14].

This study's main goal was to categorize cardiac disease using various models and a real-world dataset. To forecast the presence of the condition, A dataset of people with heart disease was subjected to the k-modes clustering algorithm. Bins of 10 intervals were created using the diastolic and systolic blood pressure data, while the age attribute of the dataset was translated to years and divided into bins of 5 years. To account for the differences in the progression and features of heart disease between men and women, the data were further divided based on gender. For both the male and female datasets, the elbow curve approach was used to estimate the right number of clusters. The figures showed an accuracy of 87.23% for the MLP model. These results show that k-modes clustering can reliably predict cardiac disease, and the method may aid in the development of focused diagnostic and therapeutic approaches for the disease. The 70,000-item Kaggle dataset on cardiovascular disease was used in the study, and Google Colab was used to build all of the algorithms. All algorithms have accuracy levels above 86%, with decision trees having the lowest accuracy (86.37%) and multilayer perceptrons having the maximum accuracy (as already indicated). Limitations Despite the positive outcomes, there are a few limits that should be taken into account. First of all, the study may not apply to other demographics or patient groups since just one dataset was employed. Additionally, the study neglected other potential risk factors for heart disease, such as genetic predispositions or features of lifestyle, and only considered a small number of clinical and demographic information. Additionally, the model's performance on a test dataset was not assessed, which may have provided insight into how well the model generalizes to fresh, untested data. Finally, the interpretability of the findings and their capacity to explain the clustering that the approach generated were not assessed. Further study is encouraged to overcome these problems and comprehend the potential of k-modes clustering for cardiac disease prediction in light of these restrictions[15].

3. MATERIALS AND METHODS

The primary methods and supplies used in this work are summarized in this section.

3.1 FEATURE STANDARDIZATION

Work was done on feature scaling, which entails altering values using one of two main techniques: normalization or standardization, before entering the data for the algorithms utilized. The input values are altered during normalization such that they fall between 0 and 1. Standardization techniques change the values to have a standard deviation of 1 and a zero-centered value. This work employed the normalization technique[16].

3.2 LONG SHORT-TERM MEMORY (LSTM)

A special case of RNNs is LSTM. It is made to deal with the issue of gradients that burst or disappear. The memory cell and several gates make up the LSTM's fundamental design. It was initially introduced in 1997 that these memory cells and gates were added to each neuron in the network [6]. Google and Apple's Siri both use recurrent neural networks (RNNs), the most complex technique for processing sequential data. It is the first algorithm to recall its input thanks to its internal memory, which makes it perfect for machine learning problems needing sequential data. RNNs work on the principle that they can predict an output by maintaining and reinforcing the output of a previous layer. Whether or not the information from the prior timestamp should be recalled is decided in the first section. This cell attempts to learn new information using the input from the second part. The cell then sends the updated data from the current timestamp of the third segment to the following timestamp. These three LSTM cell constituents are referred to by Gates. The first, second, and third components are referred to as the forget gate, input gate, and output gate, respectively. The hidden state of an LSTM is identical to that of a traditional RNN, with H(t-1) denoting the hidden state of the prior timestamp and the present timestamp, respectively, to represent a cell state for the LSTM, and H(t) denoting the current timestamp's hidden state. The timestamps C(t-1) and C(t), which stand for the prior and current timestamps, respectively, are used to represent a cell state for the LSTM [17]. LSTM includes two types of activation functions: sigmoid and its range are between (0 and 1), as it is used with the forgetting gate, because it tends more to forget unimportant words, while tanh has its range between (1 and -1), as it works to remember important words.

The following stages are used to implement this algorithm:

Step 1: The forget gate is used to determine what should be forgotten based on knowledge from a prior time step;

Step 2: New information is sought via input gate and tanh for updating cell state;

Step 3: The information from the two gates above is used to update the cell state;

Step 4: Information is usefully provided by the squashing operation and the output gate.

3.3 RECURRENT NEURAL NETWORK (RNN)

Recurrent neural networks (RNNs), Apple's Siri, and Google voice search are built on the most complex algorithm analyzing sequential data. Due to its internal memory, it is the first algorithm to recall its input, which makes it ideal for machine learning issues requiring sequential data. RNN works (as shown in Fig.1) on the principle that it can predict a layer's output by preserving that layer's output and feeding it back into the input[18]. To create a single layer of recurrent neural networks, the nodes from various layers of the neural network are condensed[19]. RNN uses the sigmoid or tanh function for hidden layers. The tanh function has better performance. Only the identity activation function is considered linear. All other activation functions are non-linear.

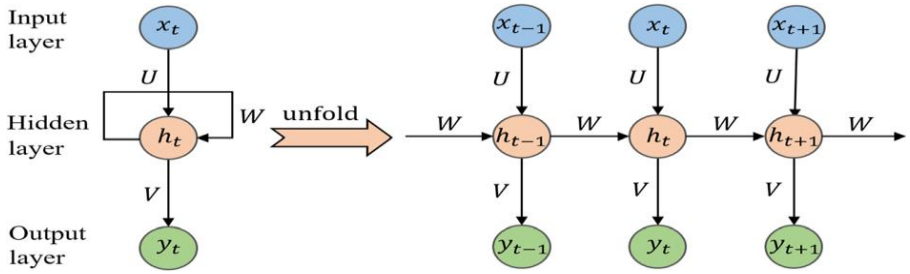

FIGURE 1. - Structure of Recurrent Neural Networks[20]

The following stages enable this algorithm to function:

Step 1: Initialize. We first determine the dimensions of the various parameters U, V, W, b, and c before implementing the fundamental RNN cell;

Step 2: Forward pass;

Step 3: Compute Loss;
Step 4: Backward pass;
Step 5: Update weights;
Step 6: Repeat steps 2–5;

3.4 DATA AUGMENTATION

Data augmentation is a series of techniques used to add additional data to a machine learning model, either in the form of copies of current data that have been significantly adjusted or newly produced synthetic data from existing data. The machine learning model is smoothed out, and data overfitting is reduced. The approaches for data augmentation can be applied to a variety of data kinds, including tabular data, photos, audio, video, text, and other sorts of data[21][22].

3.5 GENERATIVE ADVERSARIAL NETWORK (GAN)

Let's dissect "Generative Adversarial Networks" into their parts to better understand them. Because of this, the first word in the statement, generative, denotes the existence of a network that continuously generates new data, while the second term, adversarial, denotes the existence of two networks that conflict with one another. A network is nothing more than an arrangement of data that is always changing and producing new data. The first used was a GAN, a deep learning model[23]. It consists of two neural networks that compete with one another: the discriminator, which is trained to distinguish between real and fake data, and the generator, which creates fictitious data. To trick the discriminator, the generator makes adjustments throughout training, which comprises learning to produce fake data that is indistinguishable from the real data[24].

3.6 PERFORMANCE METRICS

Four evaluation metrics_accuracy, precision, sensitivity, and F1-score—are employed in this study. The accuracy measure counts the number of times a classifier correctly classified data throughout the entire dataset. It is presented as the proportion of correctly classified instances to all cases that were correctly classified. Precision is the ratio of accurately labeled positive instances to all cases that were either correctly or mistakenly classified as positive. In other words, accuracy measures the frequency of instances that can be positively detected as affirmative. By dividing the total number of positive instances, including the incorrectly categorized negative cases, by the total number of positive cases, sensitivity calculates how well the system can classify positive cases. The F1-score combines sensitivity and precision measurements. which is used to assess a classifier's accuracy. The following equations produce the four metrics listed above:[25][26][27]

$$ACCURACY = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

$$PRECISION = \frac{TP}{(TP + FP)} \quad (2)$$

$$SENSITIVITY = \frac{TP}{(TP + FN)} \quad (3)$$

Where the acronyms mentioned above can be clarified as follows:

- True Positive (TP) is the positive states that are correctly labeled as positive states.
- False Positive (FP) denotes the negative states that are incorrectly labeled as positive states.
- True Negative (TN) represents the right classification of negative diagnosis.
- False Negative (FN) indicates the positive cases that are incorrectly classified as negative

$$F1\_score = \frac{2 * Precision * Sensitivity}{Precision * Sencitivity} \quad (4)$$

To evaluate the method used to increase the data for the data set used in this paper, the equations shown below were used :

$$SCALED(X) = \frac{X - \min(r)}{\max(r) - \min(r)} \quad (5)$$

where (r) represents all the values in the real data and (x) represents all the values in the synthetic data.

$$SCORE = 1 - \frac{MatchingSyntheticRows}{TotalSyntheticRows} \quad (6)$$

4. PROPOSED SYSTEM

This study demonstrates the creation of a secure and efficient healthcare system by leveraging existing knowledge of cardiac conditions. By utilizing a collection of clinical markers, the system aims to assist in the diagnosis and classification of cardiac patients. It is crucial to improve performance accuracy, especially when compared to previous similar studies. In this study, we compare the performance of long short-term memory (LSTM) algorithms and recurrent neural network (RNN) algorithms.

Since these algorithms perform better with larger data sets, we initially augmented the patient data set (training data only) using the CTGAN method. The results obtained for the newly generated data, measured using the New Row Synthesis metric, were promising. Subsequently, we compared the performance of LSTM and RNN algorithms in predicting the likelihood of cardiac patients experiencing a heart attack.

The system is composed of two phases, as illustrated in Fig.1data augmentation of the original dataset and diagnosis of the patient's condition. Below is a detailed explanation of these phases.

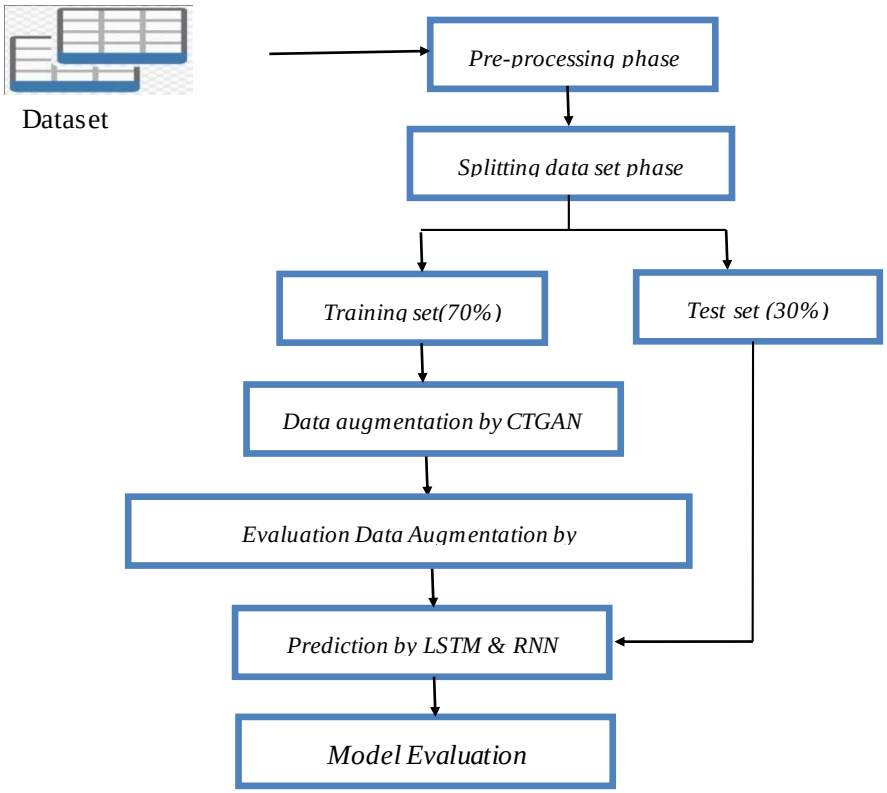

FIGURE 2. - The Architecture of the Proposed System

4.1 PRE-PROCESSING PHASE

Likely, the data values in this collection were manually entered, obtained from many sources, and subsequently made public by numerous government entities because our system uses an available dataset. This means that these values must be preprocessed before classification. Although there are several ways to perform the pre-processing task, we use the Feature Scaling method in our suggested solution. It is a technique for spreading independently occurring features in the data uniformly throughout a predetermined range. It has a broad range of magnitudes or values under control. A machine learning algorithm would often rely on values regardless of the units in the absence of feature scaling, therefore big values would be preferred over tiny values even if they were bigger. As part of the standardization procedure, a feature value is rescaled using Equation (7), resulting in a distribution with a mean of 0 and a variance of 1.

$$V_{new} = \frac{V_i - V_\mu}{V_\sigma} \quad (7)$$

Where V_i is a value for a feature from the dataset, V_{new} is a scaled value for a feature, V_μ is the average of the feature values, and V_σ is the feature value standard deviation.

4.2 SPLITTING DATA PHASE

After the necessary preprocessing, The data will be divided into two sets after the appropriate preprocessing: the training set, whose main goal is to allow machine learning algorithms to produce exact findings, and the testing set., which is used to evaluate the system's performance. In essence, a training set makes up 70% of the data, whereas a testing set makes up 30%.

4.3 DATA AUGMENTATION PHASE

Data augmentation, a collection of techniques, can be utilized to artificially increase the amount of data by creating new data points from existing data. Deep learning models are employed to either add new data points or make minor adjustments to the existing data.

Data augmentation enhances the performance and output of machine learning models by introducing more diverse instances to the training datasets. Machine learning models perform better and achieve higher accuracy when trained on extensive and diverse datasets. However, collecting and labeling such data can be time-consuming and costly. Implementing data augmentation strategies can help businesses reduce these operational expenses by modifying existing datasets.

In this system, the CTGAN technique is used. CTGAN utilizes a generative adversarial network (GAN) to model the distribution of tabular data and selects representative rows from that distribution. To address CTGAN's non-Gaussian and multimodal distribution, we employ mode-specific normalization. A GAN consists of two components: the generator, which produces synthetic data, and the discriminator, which learns to differentiate between real and generated data by utilizing the instances created by the generator.

After applying data augmentation using the CTGAN method on a dataset comprising 13 features and 300 rows, we obtain a new dataset with 13 features and 5000 rows. Several metrics such as Memory Requirements, Machine Learning Efficacy, Statistical Similarity, Distance to Closest Record, CategoricalCAP, and NumericalLR (NLR) can be used to evaluate the performance. In our case, we measured the performance of CTGAN using the New Row Synthesis metric, where the optimal ratio was 1.0. This metric determines if each row in the generated data is unique or if it duplicates an existing row in the real data. Consequently, the resulting GAN can emulate a synthetic dataset that is entirely fabricated but maintains the same structure as the actual data.

4.4 EVALUATION DATA AUGMENTATION PHASE

This metric aims to identify rows that are identical between the real and synthetic datasets. For a match to occur, all individual values in the real row must exactly match those in the synthetic row. The specific matching requirements depend on the type of data:

Boolean/categorical data: The value in the real data and the value in the synthetic data must be identical.

Numerical or date-time scale data: All values must match between the real and synthetic data, or if the synthetic value is within a certain percentage of the real value, it is considered a match. By default, this percentage parameter is set to 0.01 (1%).

The next step is to calculate the percentage of rows in the synthetic data that correspond to a row in the real data. The complement of this score ensures that 1 represents the best score (each row is unique) and 0 represents the poorest score (each row has a match).

4.5 PREDICTION PHASE

We analyze two well-known algorithms (LSTM and RNN) to identify which is superior based on the assessment metrics because the suggested system has the potential to directly impact human life and the diagnosis of health conditions.

5. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed work consists of two phases: data augmentation for the medical dataset and estimation of the likelihood of cardiac patients experiencing a cardiac arrest. This section focuses on examining and evaluating the outcomes of these phases. It should be noted that the system is implemented in Python, and the dataset1 used in these stages is publicly available, containing various features such as age, gender, anemia, diabetes, high blood pressure, and other factors that can significantly impact a person's risk of developing heart disease.

The first phase addresses the main issue of the small size of the dataset used in our work. To address this, we propose increasing the size of the dataset using the CTGAN method. The performance of this method was evaluated

using the New Row Synthesis measure, which is a well-known performance metric. We achieved a ratio of 1.0, indicating the highest performance rate for the chosen scale.

The second phase involves two steps to accomplish our objectives:

In the first step, we used the LSTM and RNN algorithms to predict the extent of the possibility of heart patients having a heart attack by applying them to the data set of heart patients before augmentation as shown in the following summary:

([1])The data set can be found online at (https://www.kaggle.com/datasets/cherngs/heart-disease-cleveland-uci)

Summary of RNN

Layer (type)	Output Shape	Param #
simple_rnn_10 (Simple RNN)	(None, 12, 10)	120
dropout_13 (Dropout)	(None, 12, 10)	0
simple_rnn_11 (Simple RNN)	(None, 10)	210
dropout_14 (Dropout)	(None, 10)	0
dense_8 (Dense)	(None, 1)	11

Total params: 341
Trainable params: 341
Non-trainable params: 0

Summary of LSTM

Layer (type)	Output Shape	Param #
lstm_9 (LSTM)	(None, 12, 10)	480
lstm_10 (LSTM)	(None, 12, 10)	840
lstm_11 (LSTM)	(None, 10)	840
dropout_15 (Dropout)	(None, 10)	0
dense_9 (Dense)	(None, 1)	11

Total params: 2,171
Trainable params: 2,171
Non-trainable params: 0

In the second step, we applied the same algorithm above, but after data augmentation for the same set of data mentioned above as shown in the following summary:

Summary of RNN

Layer (type)	Output Shape	Param #
simple_rnn_4 (SimpleRNN)	(None, 12, 10)	120
dropout_7 (Dropout)	(None, 12, 10)	0
simple_rnn_5 (SimpleRNN)	(None, 10)	210
dropout_8 (Dropout)	(None, 10)	0

dense_5 (Dense) (None, 1) 11

Total params: 341
Trainable params: 341
Non-trainable params: 0

Summary of LSTM

Layer (type)	Output Shape	Param #
lstm_6 (LSTM)	(None, 12, 10)	480
lstm_7 (LSTM)	(None, 12, 10)	840
lstm_8 (LSTM)	(None, 10)	840
dropout_6 (Dropout)	(None, 10)	0
dense_4 (Dense)	(None, 1)	11

Total params: 2,171
Trainable params: 2,171
Non-trainable params: 0

The results were according to the tables and figures listed below:

A: Performance of RNN and LSTM algorithms before data augmentation:

Table 1. - The performance results of RNN and LSTM prediction techniques

Applied method	Accuracy	Precision	Sensitivity	F1_score
RNN	80%	68%	77%	72%
LSTM	65%	0	0	0

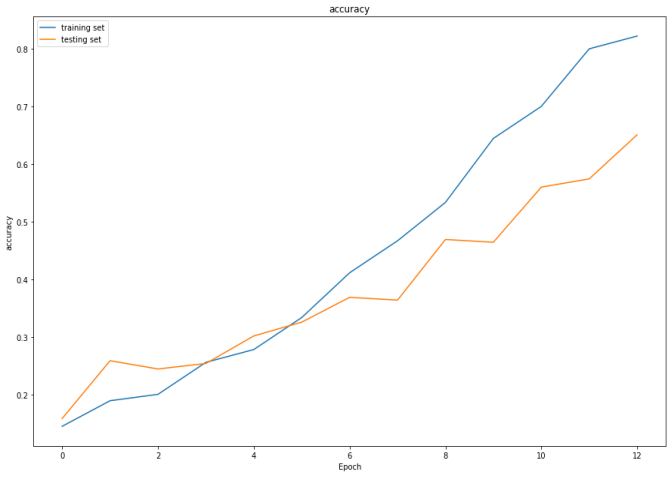

FIGURE 3. - The accuracy of RNN

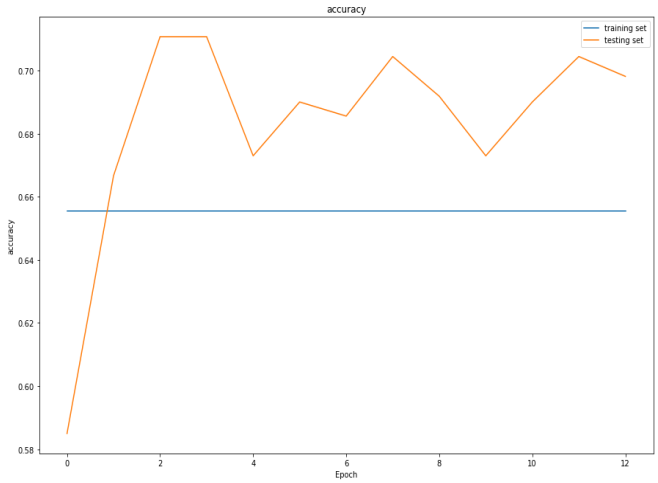

FIGURE 4. - The accuracy of LSTM

B: Performance of RNN and LSTM algorithms after data augmentation:

Table 2. - The performance results of RNN and LSTM prediction techniques

Applied method	Accuracy	Precision	Sensitivity	F1_score
RNN	100%	100%	100%	100%
LSTM	99%	100%	98%	99%

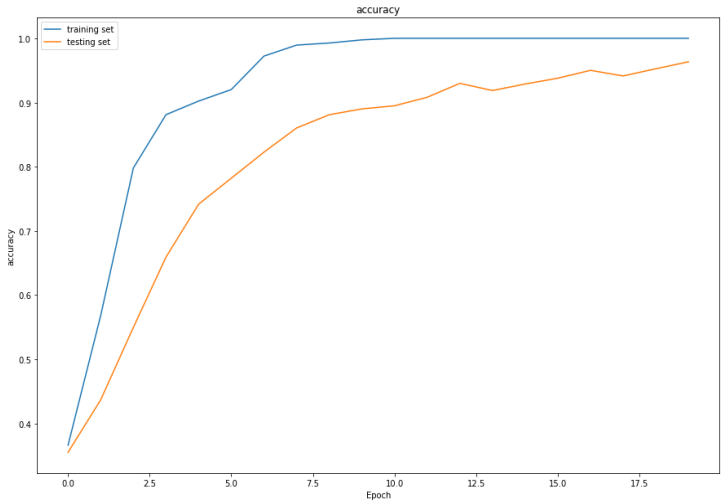

FIGURE 5. - The accuracy of RNN

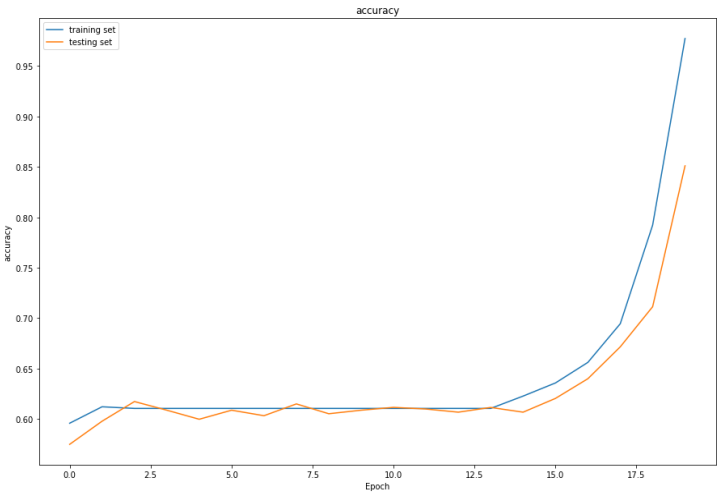

FIGURE 6. - The accuracy of LSTM

Fig. 7 shows a comparison between the LSTM algorithm before and after data augmentation depending on the metrics employed in this work.

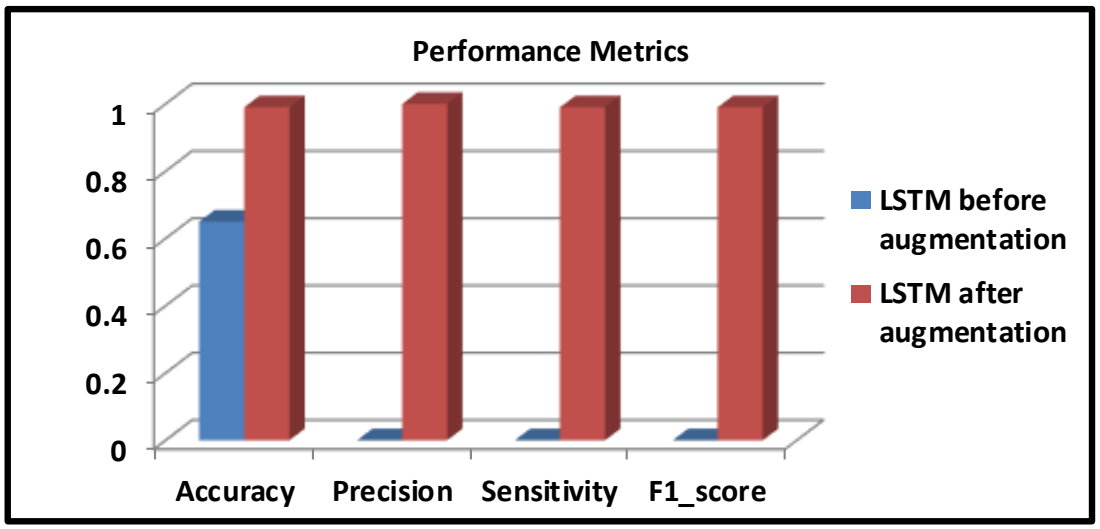

FIGURE 7. - The performance comparison between LSTM before and after data augmentation

Fig.8 based on the selected metrics specified in this work, compares the Recurrent Neural Network algorithm before and after data augmentation.

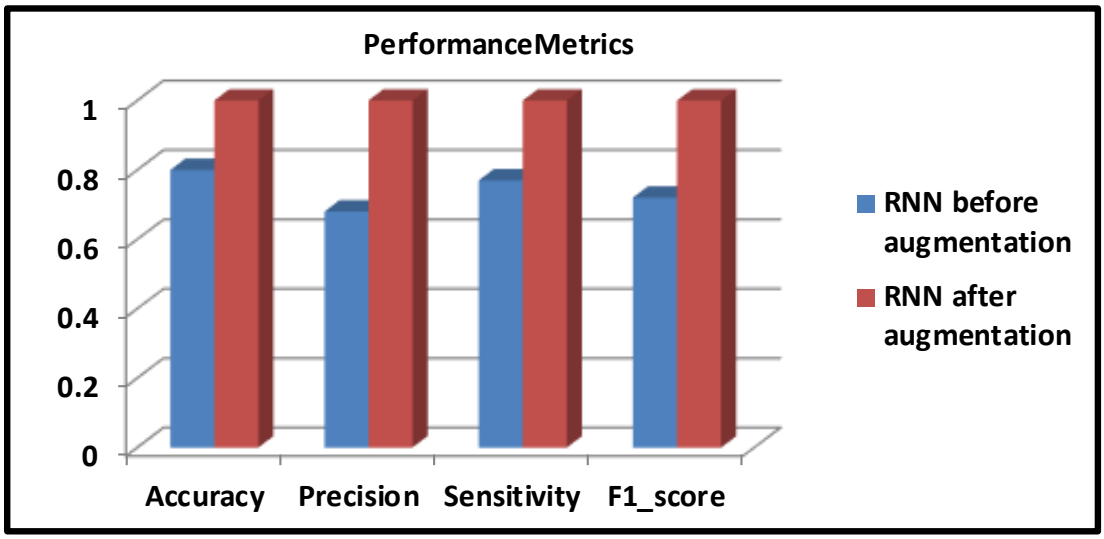

FIGURE 8. - The performance comparison between RNN before and after data augmentation

These figures show the superiority of the RNN algorithm over the LSTM algorithm.

Table 3 shows a comparison of our research's findings with those of various relevant studies. This table has five columns: the reference number, the year this reference was published, the dataset this reference utilized, the technique this reference used, and the accuracy this reference attained.

Table 3. - Comparison between the results of certain related research and the suggested approach

Reference	Year	Dataset	Method(s)	Accuracy
[1]	2021	UCI	convolutional neural networks (CNN)	97%
[2]	2021	www.kaggle.com	CTGAN, TGAN methods Random Forest, XGBoost, LightGMB, and Catboost algorithms	49.8%
[3]	2023	MIT BH	LSTM CNN	LSTM = 78% CNN = 99%
[4]	2023	www.kaggle.com	Decision tree classifier, multilayer perceptron, random forest classifier, and XGBoost	XGBoost = 87%
Our Proposed work	2023	www.kaggle.com	CTGAN method LSTM method RNN method	LSTM = 99% RNN = 100%

6. CONCLUSION

In this paper, the CTGAN method was used for augmentation of the patient's data set, because the LSTM and RNN algorithms used in our work need a large data set to work efficiently and with higher accuracy. After implementing the above-mentioned algorithms to predict the extent of cardiac patients' possibility of cardiac arrest, we obtained an accuracy of 99% and 100% for each LSTM and RNN, respectively. In future research, it is possible to use other methods to augment the data set and compare them to choose the best one in terms of accuracy.

FUNDING

No funding received for this work

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their efforts.

CONFLICTS OF INTEREST

The authors declare no conflict of interest

REFERENCES

[1] T. B. Sabir, S. B. Fakhara, M. Barani, M. Mukhtar, A. Rahdar, M. Cucchiarini, and M. N. Zafar, "Nanodiagnosis and Nanotreatment of Cardiovascular Diseases."
[2] F. A.-S. R. Shaker Abd-Alhussain and H. K. Obayes, "Secure Heart Disease Classification System Based on Three Pass Protocol and Machine Learning," p. 11, 2023.
[3] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
[4] S. H. S., Bat-Ireedui Bat-Erdene, H. Zheng, and J. Y. Lee, "Deep learning-based prediction of heart failure rehospitalization during 6, 12, 24-month follow-ups in patients with acute myocardial infarction."
[5] A. Kishore, A. Kumar, K. Singh, M. Punia, and Y. Hambir, "Heart Attack Prediction Using Deep Learning," Int. Res. J. Eng. Technol. (IRJET), vol. 5, no. 4, pp. 4420–4423, 2018.
[6] Y. J. Kim, M. S. Kang, and J. Y. Lee, "Deep learning–based prediction model of occurrences of major adverse cardiac events during 1-year follow-up after hospital discharge in patients with AMI using knowledge mining."
[7] M. I. Jordan, "Serial order: A parallel distributed processing approach," 1986.
[8] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
[9] S. Liu, X. Wang, Y. Xiang, H. Xu, H. Wang, and B. Tang, "Multi-channel fusion LSTM for medical event prediction using EHRs," J. Biomed. Inform, vol. 127, 2022, doi: 10.1016/j.jbi.2022.104011.
[10] K. V. Lei Xu, M. Skoularidou, and A. Cuesta-Infante, "Modeling Tabular Data using Conditional GAN."
[11] L. J. Kuang Junwei and Y. Z. Yang, "Dynamic prediction of cardiovascular disease using improved LSTM."
[12] A. M. Masood, M. I. Nazir, Z. M. Ahmed, A. Iqbal, M. N. Qureshi, and T. Nazir, "Prediction of Heart Disease Using Deep Convolutional Neural Networks."
[13] P. N. Subba Reddy, S. Srinidhi, and S. Bhavana, "Automated Detection of Cardiac Arrhythmia Using Recurrent Neural Network."
[14] O. L. Min Jong Cheon, D. H. Lee, J. W. Park, H. J. Choi, and J. S. Lee, "CTGAN vs. TGAN: Which One Is More Suitable for Generating."
[15] C. M. Bhatt, T. Ghosh, and P. L. Modi, "Effective Heart Disease Prediction Using Machine Learning Techniques," in Machine Learning and Its Applications, 2019, pp. 149–160, doi: 10.1201/9780429448782-9.
[16] A. T. Lokman and S. M. M. H. A. Bakar, "Blockchain-Based Image Sharing Application," Commun. Comput. Inf. Sci., vol. 1132 CCIS, pp. 46–59, 2020, doi: 10.1007/978-981-15-2693-0_4.
[17] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing LSTM for Time Series Prediction in Indian Stock Market," Procedia Comput. Sci., vol. 167, pp. 2091–2100, 2020, doi: 10.1016/j.procs.2020.03.257.
[18] M. I. Shuvo, M. S. Shahnewaz, and F. Nawaz, "An Early Detection of Heart Disease Using Machine Learning (Recurrent Neural Network)."
[19] R. I. Surenthiran and P. Magalingam, "Hybrid Deep Learning Model Using Recurrent Neural Network and Gated Mechanism."
[20] W. Zhang, H. Li, Y. Li, H. Liu, Y. Chen, and X. Ding, "Application of Deep Learning Algorithms in Geotechnical Engineering: A Short Critical Review," Appl. Intell., vol. 54, no. 8, Springer Netherlands, 2021, doi: 10.1007/s10462-021-09967-1.
[21] H. X. Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, and X. Wang, "Time series data augmentation for deep learning," 2020. [Online]. Available: https://arxiv.org/abs/2002.12478.
[22] E. C. Kumar, Varun, and A. Choudhary, "Data Augmentation Using Imagenet."

[23] G. Han, S. Liu, K. Chen, N. Yu, Z. Feng, and M. Song, "Imbalanced Sample Generation and Evaluation for Power System Transient Stability Using CTGAN," Lecture Notes in Networks and Systems, vol. 371, pp. 555–565, 2022, doi: 10.1007/978-3-030-93247-3_55.

[24] G. D. Shreyansh Singh, K. Kayathwal, and H. Wadhwa, "MeTGAN: Memory Efficient Tabular GAN for High Cardinality Categorical Datasets."

[25] H. K. Obayes, F. S. Al-Turaihi, and K. H. Alhussayni, "Sentiment classification of user's reviews on drugs based on global vectors for word representation and bidirectional long short-term memory recurrent neural network," Indones. J. Electr. Eng. Comput. Sci., vol. 23, no. 1, pp. 345–353, 202