LSTM Algorithm in Predicting Chronic Kidney Disease Optimized Using Genetic Algorithm

Brillyando Magathan Achmad[1*], Siti Sa'adah[2], Isman Kurniawan[3]

[1,2,3]*Faculty of Informatics, Telkom University, Indonesia*

*corr_author: brillyando@student.telkomuniversity.ac.id

Abstract - Chronic Kidney Disease is a health condition in which the kidneys experience a progressive decline in function. Kidneys are vital organs that filter waste and excess blood fluids. CKD can lead to excess products in the body and cause various health issues, so early detection of CKD is necessary. While traditional machine learning techniques have performed well in predicting CKD in existing studies, this study investigates the potential of long short-term memory (LSTM) optimized with Genetic Algorithm to enhance predictive accuracy and efficiency by optimizing its hyperparameters, including number of units, hidden layers, activation function, recurrent activation, and dropout rate. The result demonstrates that the optimized LSTM slightly performs better than without optimization, achieving higher precision, recall, accuracy, and f1 score by 100% respectively. This outstanding result can be attributed to several key factors, such as ensuring rigorous data preprocessing and utilizing k-fold cross-validation to make the model more reliable. This indicates the hybrid approach can be a powerful method for the early detection of CKD, leading to better patient outcomes. Despite the promising performance, further research is suggested, specifically using a larger dataset to ensure applicability to more general population and exploring other optimization methods to reduce computational cost.

Keywords: Long Short-Term Memory, Genetic Algorithm, chronic kidney disease

I. INTRODUCTION

Chronic kidney disease (CKD) is a significant health issue with considerable impacts worldwide and has become a leading cause of death in the 21st century [1]. Various factors contribute to the rising occurrence of CKD, such as age, obesity, and diabetes. CKD can result in challenges in removing excess fluid from the bloodstream effectively. In advanced stages of CKD, complications such as hypertension, anemia, and nerve damage may arise due to the accumulation of fluids, electrolytes, and hazardous waste within the body [2]. A Global Burden of Disease study highlighted CKD as a significant cause of global mortality. The mortality rate due to CKD has increased by 82.3% in the last two decades, making it the third largest increase in mortality rates, following HIV and diabetes [3]. Additionally, the Global Burden of Disease reported a 90% increase in fatalities from 1999 to 2013, ranking CKD as the 13th leading cause of death worldwide [4]. A 2017 study estimated that the number of CKD patients could reach 843.6 million, with a global prevalence of 11.1% (10.4% in men and 11.8% in women) [5]. By the year 2040, CKD is projected to become the fifth leading cause of death worldwide [6]. To diagnose whether someone is affected by CKD, hospital visits for examination are necessary. However, this process requires significant time and additional costs. Therefore, to simplify the detection of CKD with minimal testing and low costs, an approach known as machine learning technology is utilized in the development of effective and optimal CKD predictor models.

Many studies have been done to examine the prediction of CKD using machine learning. Referring to previous research, such as the study by Neves et al. in 2015 [7], which constructed a model employing artificial neural network (ANN) for the diagnosis of chronic kidney disease (CKD), encompassing 558 patients with an average age of 51.7 years and observations of chronic kidney disease in 175 cases, this study obtained sensitivity values ranging from 93.1% to 94.9% and specificity values ranging from 91.9% to 94.2%. The research conducted by Vasquez-Morales et al. [8] in 2019 demonstrated that the accuracy achieved through the utilization of neural networks surpassed support vector machine (SVM) [9] and random forest [10] in predicting chronic renal disease. The accuracy attained using neural networks was 95%, whereas SVM and random forest achieved accuracies of only 61% and 92%, respectively. In the year 2020, Bhaskar et al. [11] compared the performance of machine learning and deep learning methods in predicting CKD. The best performance was achieved by CorrNN-SVM with an accuracy of 98.67%. However, the LSTM method utilized demonstrated a relatively high accuracy of 91.64%. In 2021, Krishnamurthy S. et al. [12] researched predicting CKD using Taiwan's National Health Insurance Research Database (NHIRD). They developed a CKD prediction model using convolutional neural network and bi-directional long short-term memory (BLSTM) techniques, achieving respective accuracies of 95.7% and 93%. In 2021, Hamida et al. [13] aimed to compare the random forest algorithm with the J48 decision tree for CKD prediction. Their findings showed that the J48 decision tree had a higher accuracy of 85.5% compared to random forest. Furthermore, Arroyo et al. [14] 2022 showed that the accuracy of using artificial neural network algorithms optimized with genetic algorithms (GA) is higher compared to neural networks without optimization. This study focused on predicting cardiovascular disease in patients. The accuracy achieved by utilizing the genetic algorithm optimization for artificial neural networks was 73.43%. In contrast, artificial neural networks without optimization only attained an accuracy of 68.35%.

Based on the research above [7], [8] and [11]-[14], the optimization of LSTM using GA is chosen for CKD prediction because optimization using GA has been proven to give better accuracy compared to no optimization. Additionally, LSTM will be used for CKD prediction because LSTM is an optimal version of recurrent neural network (RNN), which can address issues related to gradient disappearance and explosion when learning from large sequential data [15]. Moreover, LSTM can detect underlying patterns in non-sequential data. This makes LSTM a powerful method for analyzing data and uncovering important features for predictions [16]. This research aims to increase its accuracy by 1% to 5% with the application of GA optimization and is expected to improve the LSTM model's performance, leading to more reliable prediction of CKD. Hence, this research will compare the accuracy outcomes of LSTM optimization using GA with LSTM without optimization.

II. METHOD

Fig. 1 shows the research methodology, including CKD dataset input, data preprocessing, dataset split, LSTM model building, LSTM hyperparameter optimization using GA, testing, and evaluation. Long Short-Term Memory (LSTM) is used to do the CKD prediction and its hyperparameter is optimized by Genetic Algorithm (GA). The CKD dataset is observed to understand its characteristics. Any anomalies in the dataset are addressed during preprocessing data. Splitting and selecting the most relevant features was utilized to enhance the effectiveness and robustness of data analysis and modelling. Next, the LSTM model and GA optimization are built to train the dataset and make CKD predictions. The model's accuracy, precision, recall, and F-1 score are calculated with the help of the confusion matrix to highlight the performance of the model.

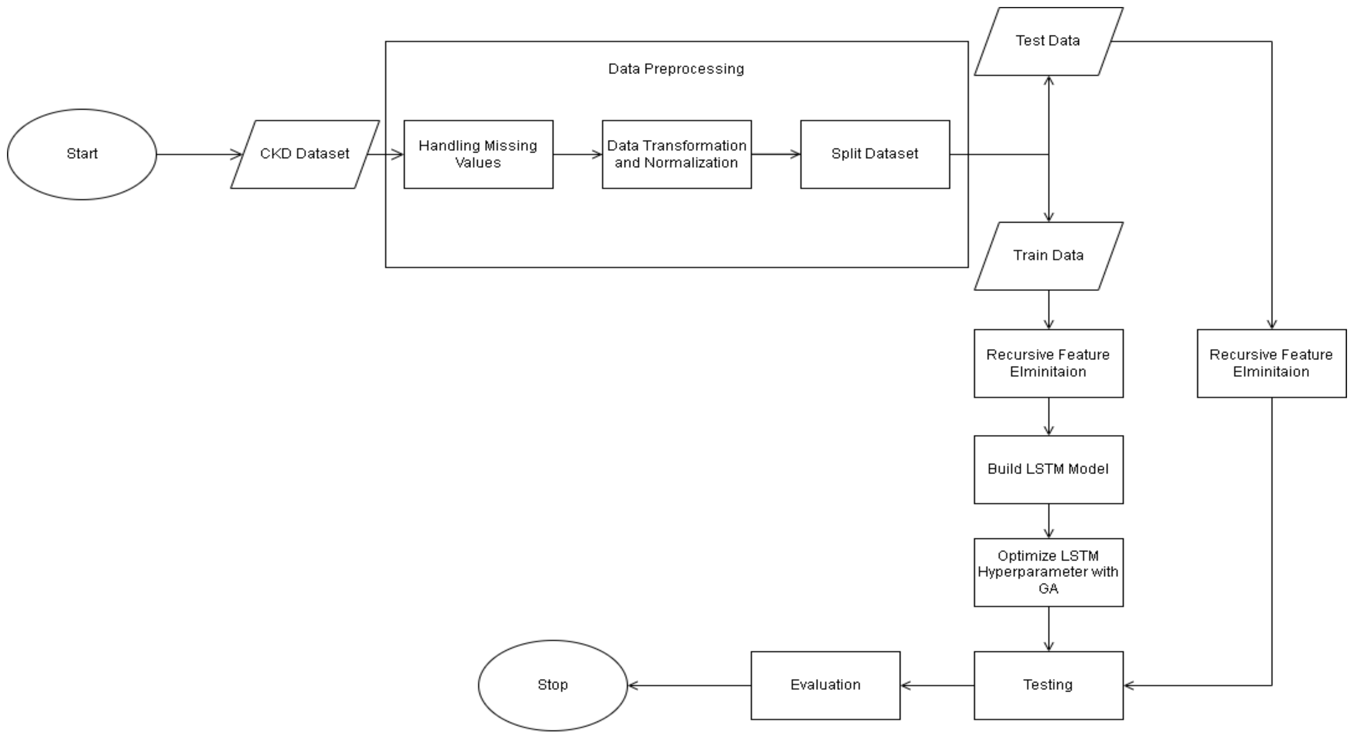


Fig. 1 Research methodology

A. Chronic Kidney Disease (CKD) Dataset

California University, Irvine Machine Learning Repository [17] has collected patient data from the hospital in Karaikudi, India. It is available on the UCI repository or Kaggle. This dataset is tabular data with 400 records, where 250 records indicate CKD and the rest are not. There are 25 attributes or features, including their class (ckd and notckd). More information about the features can be seen in Table I. The "Feature Name" consists of each feature in the dataset, the "Description" column explains what each feature measures, the "Attribute Type" column classifies features as either numerical or categorical and the "Units" column indicates the measurement unit for numerical attributes.

B. Data Preprocessing

Preprocessing data is an important task to do to make proper readable data for the model to obtain maximum accuracy in the training phase. The benefit of implementing data preprocessing is to ensure that data is clean and optimized for analysis, ultimately leading to a more balanced dataset structure. Several stages will be applied, such as observing the dataset, handling missing values, data transformation, data normalization, splitting the dataset, and then doing feature selection.

After some observations on the dataset, it was revealed that some cells contained many missing values. This case must be handled appropriately. The size of the dataset is quite small, so ignoring or deleting these values is not a suitable approach. Therefore, the mode is used to fill in the missing values in the categorical columns. Meanwhile, for the numerical columns, the missing values are filled by the mean or average value of each column.

The next step is doing data transformation for the categorical columns. Data transformation is a method to transform, convert, or change to be more suitable for analysis and model training. Using "LabelEncoder" from sklearn automatically encode the string in each categorical column to an integer starting from 0. For instance, the "rbc" column with values "normal" and "abnormal" will be encoded to 1 and 0 respectively. Data normalization is carried out to handle the outliers in numerical columns, which can interfere the model training process later and affect negatively the estimation of statistical measures [17]. So, the "RobustScaler" from sklearn is used to remove the median and scale the data according to its quartile range between the first quartile and third quartile. The formula of "RobustScaler" is shown in (1) with X: original data; Q3(x): 3rd quartile; Q1(x): 1st quartile.

$$X = \frac{X_i - X_{median}}{Q_3(x) - Q_1(x)} \tag{1}$$

TABLE I
INFORMATION OF CKD DATASET

Feature Name	Description	Attribute Type	Units
age	Age	Numerical	Year
bp	Blood pressure in mm/Hg	Numerical	mm/Hg
sg	Specific gravity	Numerical	
al	Albumin	Nominal	
su	Sugar	Nominal	
rbc	Red blood cells	Nominal	
pc	Pus cells	Nominal	
pcc	Pus cells clumps	Nominal	
ba	bacteria	Numerical	
bgr	Blood glucose Random	Numerical	mgs/dl
bu	Blood urea	Numerical	mgs/dl
sc	Serum creatinine	Numerical	mgs/dl
sod	Sodium	Numerical	mEq/L
pot	Potassium	Numerical	mEq/L
hemo	Hemoglobin	Numerical	gms
pcv	Packed cell volume	Numerical	
wc	White blood cell count	Numerical	cells/cmm
rc	Red blood cell count	Numerical	millions/cmm
htn	Hypertension	Nominal	
dm	Diabetes mellitus	Nominal	
cad	Coronary artery disease	Nominal	
appet	Appetite	Nominal	
pe	Pedal edema	Nominal	
ane	Anemia	Nominal	
class	Target class (ckd or notckd)	Nominal	

The dataset was then split into two sets, 75% for the training set and 25% for the test set. But the training set will be split again into the training set and validation set. This validation set is used for K-fold cross-validation, with the value of K is 4. The final step is doing a feature selection using recursive feature elimination (RFE). RFE will select the most relevant features and correlate with the target by removing the least important features recursively.

C. Long Short-Term Memory (LSTM) Model

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) specifically designed to address the issue of vanishing gradients when learning long-term dependencies [18]. The primary role of LSTM is to analyze complete sequences of data, making it widely used in deep learning. In the case of CKD prediction, the chosen dataset is tabular, so LSTM's ability to handle sequential data problems is not fully utilized. However, the temporal correlations between features can be revealed by LSTM [19]. Even though the dataset is not sequential, LSTM's ability to detect interaction patterns among features is reliable.

In this research, the structure of LSTM is consisting of 1 input layer, 2 hidden layers, and 1 output layer. Sigmoid is applied to the recurrent activation, while the activation function is using tangent hyperbolic (tanh). To reduce overfitting, one dropout layer is inserted into the structure with a rate of 0.4. The dropout layer will temporarily remove units from the LSTM network randomly, including all its connections. This means any updates on network weights will not be applied on the backpropagation. So, the networks become less sensitive to weights and generalize better to unseen data [20]. Furthermore, l2 regularization is applied to all layers except the output layer, adding penalty term into the loss function that matches the sum of the squared values of the model's coefficients (2). This prevents overfitting and helps maintain the model weights remain small.

$$L2(x, y) = \lambda \sum_{i=0}^{n} W_i^2 \tag{2}$$

where λ is the regularization parameter, W_i are the model weights, and n is the number of model parameters.

D. Genetic Algorithms

Genetic algorithms (GAs) are a type of evolutionary algorithms inspired by biological principle such as inheritance, mutation, selection, and crossover. The main goal of GAs is to find approximate solutions to complex problems. They begin by initializing a population of potential solutions. Each member of the population will have its performance calculated, known as the fitness score. After that, a new population will be generated through various genetic operations, such as crossover and mutation. This process continues until the end of the iteration, aiming to achieve an accurate fitness value [21].

In this research, genetic algorithms are applied to optimize hyperparameters, which will be integrated into LSTM networks. Hyperparameters that will be optimized are units, activation functions, hidden layers, recurrent activations, and dropout rates. The process flow from it is presented in Fig. 2.

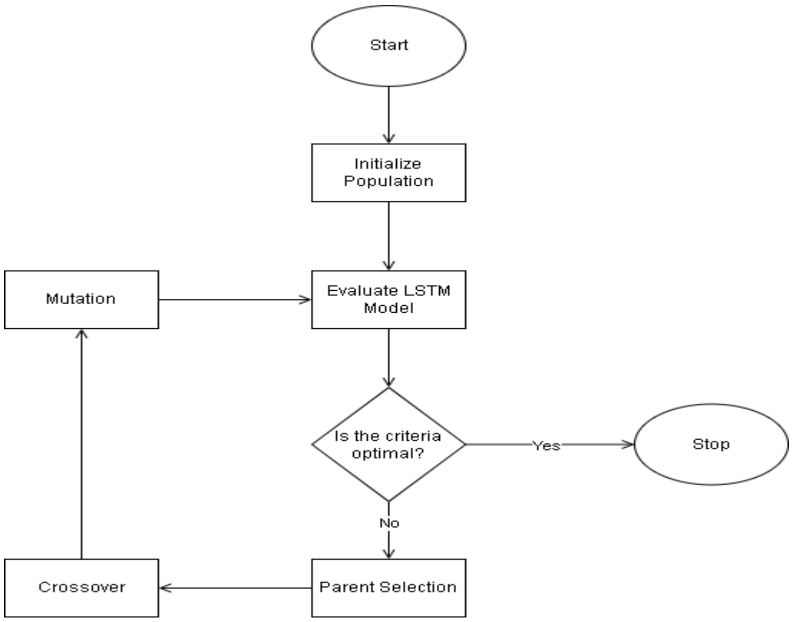


Fig. 2 Flowchart of genetic algorithm

1) Population: In the first generation, the population is generated randomly, consisting of individuals or chromosomes with various genes, such as the number of units, hidden layers, activation function, recurrent activation, and dropout rate, as illustrated in Table II. Units is a cell that consists of an input gate, forget gate and an output gate to process the input data and capture temporal dependencies. It's crucial to adjust for better output, just like hidden layers that affect model development. The activation function and recurrent activation are chosen as parameters to be optimized because the activation function and recurrent activation control information flow and are encoded as an integer within the range [0, 7], then encoded again into binary number (Table III). Lastly, the reason the dropout rate was selected is to reduce the model getting overfitting.

2) Evaluate Fitness Score: The model evaluation is performed using the population that is initialized with the CKD dataset. The validation accuracy produced by the LSTM model in this study is referred to as the fitness score. To calculate the validation accuracy, here K-fold cross-validation is conducted. This method splits the training data to k number of subsets, which in this research is using 4 folds. The model will train using different folds or subsets as validation data with 4 iterations. The accuracy of each subset will be recorded and the mean of the recorded accuracy across all subsets is calculated to get the fitness score.

3) Parent Selection: Parent selection is performed among individuals with the highest fitness values. They are retained, while those not selected are removed or discarded from the population. The roulette wheel method will be used to select parents based on probabilities proportional to their fitness scores. Generally, the calculation to determine these probabilities is as follows in (3), with P_i is the probability of an individual at index-i; f_i is the fitness score of an individual at index-i; N is population size.

$$P_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \tag{3}$$

TABLE II
SAMPLE OF POPULATION WITH 2 INDIVIDUALS

	Units	Hidden layers	Activation function	Recurrent activation	Dropout rate
Individual A	16	2	010	000	0.8
Individual B	20	3	100	001	0.2

4) Crossover: A crossover is conducted between selected parents to produce new offspring. These offspring will have more varied genes. The selection of genes for crossover is done randomly. However, not all parents in the population will do the crossover; a probability is required to determine whether crossover occurs. Fig. 3 presents the single-point crossover scheme applied in this research with a probability of 0.7.

TABLE III
FUNCTION NAME AND ITS ENCODING RESULT

Function	Integer	Binary
Sigmoid	0	000
RelU	1	001
Tanh	2	010
Softmax	3	011
Elu	4	100
Hard_sigmoid	5	101
Linear	6	110
Gelu	7	111

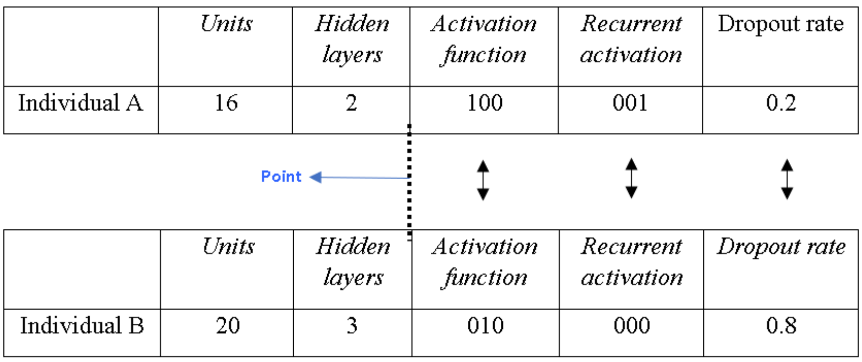


Fig. 3 The scheme of crossover with single point

5) Mutation: Mutation is needed by swapping genes or attributes among individuals. Mutation simulates the impact of errors occurring during duplication, with a low probability. This ensures that only a small portion of the population experiences mutation in each generation or iteration. A probability of 0.3 and the random resetting method, which refers to the set of all possible learning model parameters for units, hidden layers, and dropout rate, as shown in Table IV is used. A random value from Table IV will be assigned to randomly chosen genes. Meanwhile, the activation function and recurrent activation are used bit flip mutation, in which random bits will be selected and flipped. The schema of mutation is presented in Fig. 4.

E. Result Evaluation

The performance of LSTM will be compared to the performance of LSTM optimized using GAs. This comparison will observe the impact of GA optimization. Recall (4), precision (5), F1 score (6), and accuracy (7) of the test set will be calculated using a confusion matrix. The performance of the algorithms will then be visualized with plots to illustrate the differences clearly.

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

$$F1\ Score = 2 * \frac{precision*recall}{precision+recall} \tag{6}$$

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{7}$$

where TP: true positive; TN: true negative; FP: false positive; FN: false negative

III. RESULT AND DISCUSSION

A. LSTM Performance

LSTM is a powerful approach for identifying interaction patterns between features because LSTM is good at capturing temporal connections [20]. The LSTM model in this research was built with various parameter settings, aimed to find the maximum accuracy in predicting CKD before comparing it to the LSTM with genetic algorithm optimization. However, several parameters did not change throughout the experiment, including type of optimizer, number of epochs, learning rate value, batch size, and data split proportion. The optimizer used was Adam with a learning rate of 0.001. Applied epochs of 20 and batch size of 32. Lastly, the proportion of data split was 75% for training and 25% for testing. Table V presents the LSTM performances.

The LSTM parameters that have the best performance consist of 40 number of units, 3 hidden layers, hard sigmoid activation function, tanh recurrent activation function, and 0.9 dropout rate, resulting in an accuracy of 0.99. For further information on its performance, Fig. 5 displays the training process and the result obtained during testing on the test data.

TABLE IV
LIST OF PARAMETERS AND THEIR VALUES

Parameter	Value
Unit	8, 16, 20, 32, 40, 56, 64, 80, 114, 128
Hidden layer	1, 2, 3
Activation function	000, 001, 010, 011, 100, 101, 110, 111
Recurrent activation	000, 001, 010, 011, 100, 101, 110, 111
Dropout rate	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9

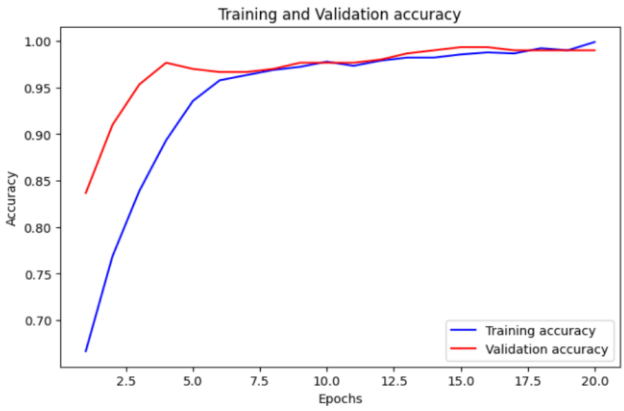
	Units	*Hidden layers*	*Activation function*	*Recurrent activation*	*Dropout rate*
Individual A	**16**	2	100	001	0.2

Individual A	**32**	2	100	001	0.2

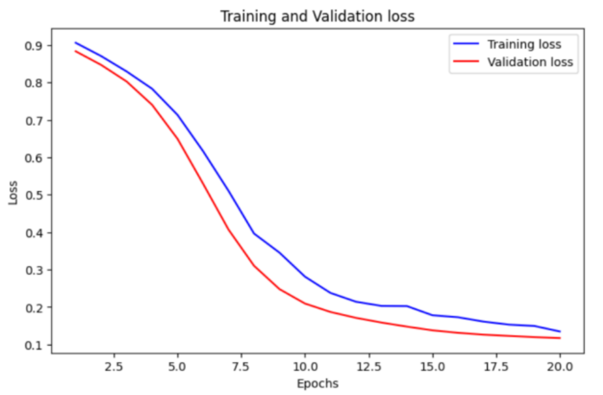
Fig. 4 The scheme of mutation

TABLE V
VARIOUS LSTM PARAMETER RESULTS

Units	Hidden Layers	Activation Function	Recurrent Activation	Dropout Rate	Precision	Recall	Accuracy	F-1 Sccore
15	3	sigmoid	gelu	0.2	0.65	1.0	0.65	0.79
10	2	sigmoid	tanh	0.2	0.9843	0.9692	0.97	0.9767
10	4	tanh	tanh	0.1	0.65	1.0	0.65	0.79
40	**3**	**hard sigmoid**	**tanh**	**0.9**	**1.0**	**0.9846**	**0.99**	**0.992**
25	2	elu	sigmoid	0.5	1.0	0.9682	0.98	0.9838



(i) LSTM accuracy



(ii) LSTM loss

```
4/4 [==============================] - 1s 4ms/step
              precision    recall  f1-score   support

           0       0.97      1.00      0.99        35
           1       1.00      0.98      0.99        65

    accuracy                           0.99       100
   macro avg       0.99      0.99      0.99       100
weighted avg       0.99      0.99      0.99       100

[[35  0]
 [ 1 64]]
```

(iii) LSTM result on test data

Fig. 5 LSTM model performance

In the accuracy plot, the training accuracy remained stable and increased steadily from the range 0.6 – 0.7 to 0.9-1.0. It aligned with validation accuracy that follow from bottom to top and did not exceed the training accuracy by the end of epoch. This indicated that the model consistently performed well on both training and validation data, showing no sign of model overfitting or underfitting. But that was debunked by what happened in the loss plot. Although the loss pattern from training and validation data aligning, there was a significant gap between them, with the value from training loss being higher. The model did not reduce errors efficiently in training data compared to validation data. As a result, there was an error in the model's prediction of the test dataset. In other words, the model could be overfitting.

B. LSTM Optimized using GA Performance

The application of Genetic Algorithm to optimize hyperparameters machine learning models, especially neural networks has been proven to provide more optimal results [14]. Experiments have been done to determine GA's performance and its impact. Different GA parameters explored the search space to identify the optimal chromosome for maximum accuracy in the LSTM model. Even though larger parameters will provide more desired solutions, but they will lead to significantly higher computational costs. To balance the solution quality and computational expenses is a critical attention of the optimization process. In this experiment, the main platform used is Google Colab, which has 50Gb total RAM, limiting GA execution to 30 generations. The results of GA optimization and the accuracy of best chromosome, which in this research the best chromosome was at the end of generations when applied to the LSTM model are shown in Table VI. Moreover, the best results from several trials of GA will be shown in Table VII.

TABLE VI
GA PARAMETER SETTINGS AND LSTM ACCURACY

Population	Generation	Crossover Probability	Mutation Probability	Best Fitness Score	Accuracy in LSTM Model
10	20	0.7	0.3	0.9904761910438538	0.98
15	30	0.8	0.2	0.9904761910438538	0.99
8	25	0.9	0.2	0.9733333438634872	0.97
8	**25**	**0.7**	**0.3**	**0.996666669845581**	**1.0**

TABLE VII
BEST CHROMOSOME FITNESS SCORE IN OPTIMAL PARAMETER GA SETTINGS

Generation	Best Fitness Score	Best Chromosome
1	0.9900000095367432	{'unit': 80, 'n_hidden': 3, 'a_func': '101', 'r_func': '110', 'do_rate': 0.4}
5	0.9933333396911621	{'unit': 80, 'n_hidden': 3, 'a_func': '101', 'r_func': '110', 'do_rate': 0.1}
10	0.9933333396911621	{'unit': 56, 'n_hidden': 3, 'a_func': '000', 'r_func': '110', 'do_rate': 0.9}
15	0.9866666793823242	{'unit': 40, 'n_hidden': 3, 'a_func': '101', 'r_func': '110', 'do_rate': 0.7}
20	0.9933333396911621	{'unit': 114, 'n_hidden': 3, 'a_func': '101', 'r_func': '100', 'do_rate': 0.2}
25	0.996666669845581	{'unit': 114, 'n_hidden': 3, 'a_func': '101', 'r_func': '100', 'do_rate': 0.1}

Table VI shows the comparison of different genetic algorithm parameter settings applied to LSTM model. Each configuration specifies the population size, number of generations, crossover probability, and mutation probability. It includes the best chromosome fitness score and the accuracy of the LSTM model to calculate configuration effectiveness. For example, 8 population size, 25 generations, crossover probability of 0.7, and mutation probability of 0.3 achieved the best LSTM accuracy of 1.0 with the fitness score of 0.996, indicating the most effective parameters. The best chromosome and its fitness score across generations during the most optimal GA run are shown in Table VII, starting from a fitness score of 0.990. Fitness score unstable, but, at the end of generation, the maximum fitness score of 0.996 was achieved. The best chromosome included 114 units, 3 hidden layers, activation function 101 or hard sigmoid, recurrent activation 100 or ELU, and a dropout rate of 0.1. The optimized parameters will be applied into the LSTM model to predict CKD and its performance is shown in Fig. 6.

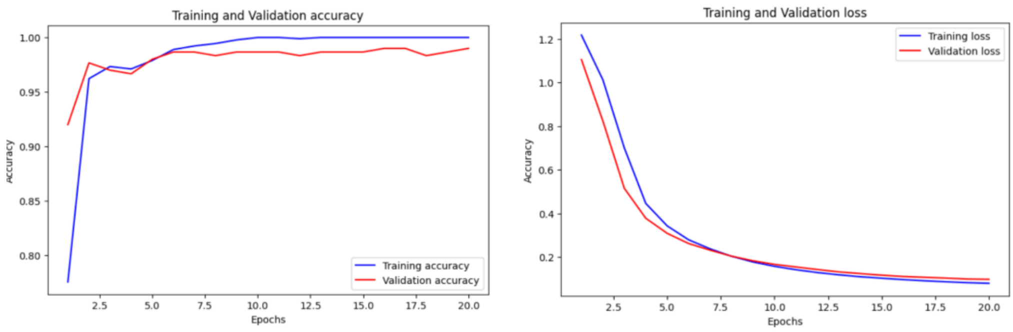
During the training, the goal of the model was to maximize accuracy and minimize loss. The training accuracy increased quickly at the beginning of the epoch from 0.775 to 0.962, and then the training accuracy stably moved to 1.0 until the last epoch. It means the model rapidly learnt relationships within the training dataset. The validation accuracy also improved from 0.92 to 0.99. Although there was instability in the middle of the epoch, the validation accuracy displayed occasional slight increases after the 17th epoch, indicating that the model could generalize well to new data. Both training and validation loss improved, indicating the model's capability to minimize errors. Notably, there were no signs of overfitting during the training process. Consequently, when the model was tested with test data or unseen data, the accuracy reached 1.0.

C. *Comparative Analysis*

In our test of LSTM and optimized LSTM using test data, both models had good performance across various metrics as shown in Fig. 7. They both achieved a precision of 1.0, indicating they perfectly avoided false positive errors. When it comes to recall, the optimized LSTM got a recall of 1.0, while LSTM achieved 0.98. The slightly higher value of the optimized LSTM's recall indicates that it is better at reducing false negative errors compared to LSTM without optimization.

Regarding accuracy, LSTM and optimized LSTM achieved an accuracy of 0.99 and 1.0 respectively, the same as the F1 score. This slight difference brings out that the optimized LSTM generalizes effectively from the training data to unseen data or test data than LSTM without optimization. Furthermore, the higher F1 score for the optimized LSTM shows that it had stronger performance because it balanced better between precision and recall, which are important metrics.

The comparison with the existing research is shown in Table VIII. The proposed model, which is optimized LSTM using GA (GA-LSTM) reached highest accuracy on test data in chronic kidney disease prediction with accuracy of 100%.



(i) Optimized LSTM accuracy **(ii) Optimized LSTM loss**

```
4/4 [==============================] - 1s 4ms/step
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        35
           1       1.00      1.00      1.00        65

    accuracy                           1.00       100
   macro avg       1.00      1.00      1.00       100
weighted avg       1.00      1.00      1.00       100

[[35  0]
 [ 0 65]]
```

(iii) Optimized LSTM result on test data

Fig. 6 Optimized LSTM performance

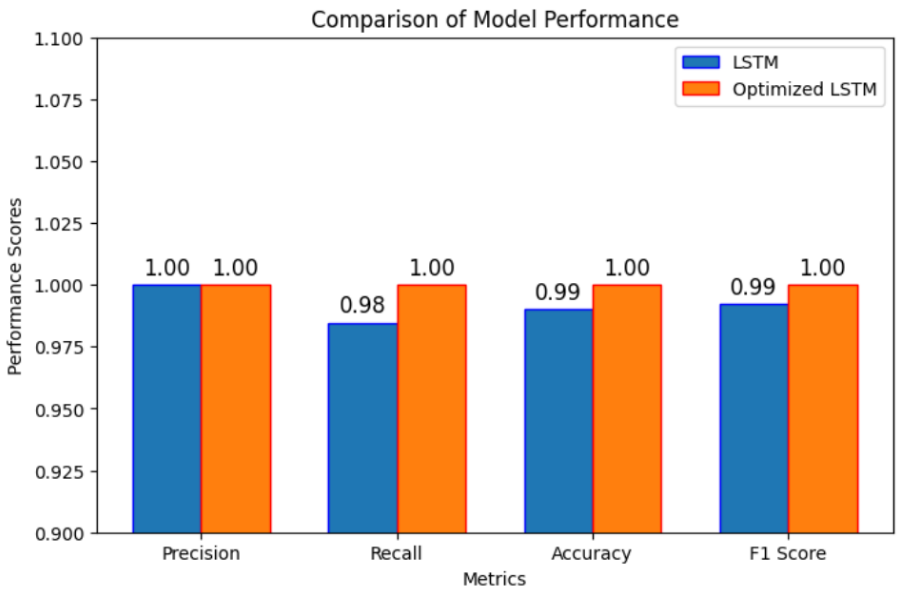


Fig. 7 Performance of both models on test data

TABLE VIII
COMPARISON OF THE PROPOSED MODEL WITH THE EXISTING STUDY

Author	Model	Disease Prediction	Accuracy
Vasquez-Morales et al. (2019) [8]	Neural networks	Chronic renal disease	95%
Bhaskar et al. (2020) [11]	CorrNN-SVM	Chronic kidney disease	98.67%
Hamida et al. (2021) [13]	J48 decision tree	Chronic kidney disease	85.5%
Arroyo et al. (2022) [14]	GA-ANN	Cardiovascular disease	73.43%
Terlapu et al. (2023) [23]	PCA+GA+MLP	Chronic kidney disease	98.58%
Present Research	**GA-LSTM**	**Chronic kidney disease**	**100%**

IV. CONCLUSION

The LSTM optimized using a genetic algorithm has been proven to perform better in predicting chronic kidney disease than LSTM without optimization. The model was built with a CKD dataset from Kaggle that consists of 25 features and 400 records. Each feature or column will undergo data preprocessing before training the LSTM model. This process includes handling missing values, transforming the dataset, normalizing data, splitting data, and selecting the most important feature. The LSTM model and the optimized LSTM model will be trained using various parameter settings to achieve maximum results. To find out which model performs better, both models will be tested using test data, consisting of 100 records of unseen data, and then the results will be compared. The result revealed that the optimized LSTM achieved an accuracy of 1.0, while the LSTM without optimization achieved an accuracy of 0.99. It means that optimized LSTM pointed to a slight improvement in this case.

REFERENCES

[1] C. P. Kovesdy, "Epidemiology of chronic kidney disease: an update 2022," Kidney Int Suppl (2011), vol. 12, no. 1, pp. 7–11, Apr. 2022, doi: 10.1016/j.kisu.2021.11.003.

[2] V. Singh, V. K. Asari, and R. Rajasekaran, "A Deep Neural Network for Early Detection and Prediction of Chronic Kidney Disease.," *Diagnostics (Basel)*, vol. 12, no. 1, Jan. 2022, doi: 10.3390/diagnostics12010116.

[3] J. Radhakrishnan, G. Remuzzi, R. Saran, D.E. Williams, N. Rios-Burrows, N. Powe, CDC-CKD Surveillance Team, K. Brück, C. Wanner, V. S. Stel, European CKD Burden Consortium 6, S. K. Venuthurupalli, W. E. Hoy, H. G. Healy, A. Salisbury, R. G. Fasset, CKD.QLD group, D. O'Donoghue, P. Roderick, S. Matsuo, A. Hishida, E. Imai, and S. Iimuro, "Taming the chronic kidney disease epidemic: a global view of surveillance efforts," *Kidney Int*, vol. 86, no. 2, pp. 246–250, Aug. 2014, doi: 10.1038/ki.2014.190.

[4] J. Radhakrishnan and S. Mohan, "KI Reports and World Kidney Day," *Kidney Int Rep*, vol. 2, no. 2, pp. 125–126, Mar. 2017, doi: 10.1016/j.ekir.2017.01.014.

[5] K. J. Jager, C. Kovesdy, R. Langham, M. Rosenberg, V. Jha, and C. Zoccali, "A single number for advocacy and communication—worldwide more than 850 million individuals have kidney diseases," *Kidney Int*, vol. 96, no. 5, pp. 1048–1050, Nov. 2019, doi: 10.1016/j.kint.2019.07.012.

[6] K. J. Foreman, N. Marques, A. Dolgert, K. Fukutaki, N. Fullman, M. McGaughey, M. A. Pletcher, A. E. Smith, K. Tang, C. Yuan, J. C. Brown, J. Friedman, J. He, K. R. Heuton, M. Holmberg, D. J. Patel, P. Reidy, A. Carter, K. Cercy, A. Chapin, D. Douwes-Schultz, T. Frank, F. Goettsch, P. Y. Liu, V. Nandakumar, M. B. Reitsma, V. Reuter, N. Sadat, R. J. D. Sorensen, V. Srinivasan, R. L. Updike, H. York, A. D. Lopez, R. Lozano, S. S. Lim, A. H. Mokdad, S. E. Vollset, and C. J. L. Murray, "Forecasting life expectancy, years of life lost, and all-cause and cause-specific mortality for 250 causes of death: reference and alternative scenarios for 2016–40 for 195 countries and territories," *The Lancet*, vol. 392, no. 10159, pp. 2052–2090, Nov. 2018, doi: 10.1016/S0140-6736(18)31694-5.

[7] J. Neves, M. R. Martins, J. Vilhena, J. Neves, S. Gomes, A. Abelha, J. Machado, and H. Vicente, "A Soft Computing Approach to Kidney Diseases Evaluation," *J Med Syst*, vol. 39, no. 10, p. 131, Oct. 2015, doi: 10.1007/s10916-015-0313-4.

[8] G. R. Vasquez-Morales, S. M. Martinez-Monterrubio, P. Moreno-Ger, and J. A. Recio-Garcia, "Explainable Prediction of Chronic Renal Disease in the Colombian Population Using Neural Networks and Case-Based Reasoning," *IEEE Access*, vol. 7, pp. 152900–152910, 2019, doi: 10.1109/ACCESS.2019.2948430.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Mach Learn*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[10] L. Breiman, "Random forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[11] N. Bhaskar and M. Suchetha, "A Computationally Efficient Correlational Neural Network for Automated Prediction of Chronic Kidney Disease," *IRBM*, vol. 42, no. 4, pp. 268–276, Aug. 2021, doi: 10.1016/j.irbm.2020.07.002.

[12] S. Krishnamurthy, K. Ks, E. Dovgan, M. Luštrek, B. G. Piletič, K. Srinivasan, Y. J. Li, A. Gradišek, and S. Syed-Abdul, "Machine Learning Prediction Models for Chronic Kidney Disease Using National Health Insurance Claim Data in Taiwan," *Healthcare*, vol. 9, no. 5, p. 546, May 2021, doi: 10.3390/healthcare9050546.

[13] H. Ilyas, S. Ali, M. Ponum, O. Hasan, M. T. Mahmood, M. Iftikhar, and M. H. Malik, "Chronic kidney disease diagnosis using decision tree algorithms," *BMC Nephrol*, vol. 22, no. 1, p. 273, Dec. 2021, doi: 10.1186/s12882-021-02474-z.

[14] J. C. T. Arroyo and A. J. P. Delima, "An Optimized Neural Network Using Genetic Algorithm for Cardiovascular Disease Prediction," *Journal of Advances in Information Technology*, vol. 13, no. 1, 2022, doi: 10.12720/jait.13.1.95-99.

[15] X. Zang, J. Du, and Y. Song, "Early Prediction of Heart Disease via LSTM-XGBoost," in *Proceedings of the 2023 9th International Conference on Computing and*

Artificial Intelligence, New York, NY, USA: ACM, Mar. 2023, pp. 631–637. doi: 10.1145/3594315.3594383.

[16] P. Dileep, K. N. Rao, P. Bodapati, S. Gokuruboyina, R. Peddi, A. Grover, and Anu, "An automatic heart disease prediction using cluster-based bi-directional LSTM (C-BiLSTM) algorithm," *Neural Comput Appl*, vol. 35, no. 10, pp. 7253–7266, Apr. 2023, doi: 10.1007/s00521-022-07064-0.

[17] Eswaran, L. R. P. S. (2015). Chronic kidney disease [Data set]. UCI Machine Learning Repository. doi: 10.24432/C5G020.

[18] S. K. Kwak and J. H. Kim, "Statistical data preparation: management of missing values and outliers," *Korean J Anesthesiol*, vol. 70, no. 4, p. 407, 2017, doi: 10.4097/kjae.2017.70.4.407.

[19] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif Intell Rev*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.

[20] D. Saif, A. M. Sarhan, and N. M. Elshennawy, "Deep-kidney: an effective deep learning framework for chronic kidney disease prediction," *Health Inf Sci Syst*, vol. 12, no. 1, p. 3, Dec. 2023, doi: 10.1007/s13755-023-00261-8.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. https://dl.acm.org/doi/10.5555/2627435.2670313.

[22] S. Koçer and M. R. Canal, "Classifying Epilepsy Diseases Using Artificial Neural Networks and Genetic Algorithm," *J Med Syst*, vol. 35, no. 4, pp. 489–498, Aug. 2011, doi: 10.1007/s10916-009-9385-3.

[23] P. R. V. Terlapu, D. Jayaram, S. Rakesh, M. V. Gopalachari, B. V. Ramana, N. Tangudu, and K. R. Kalidindi, "Optimizing Chronic Kidney Disease Diagnosis in Uddanam: A Smart Fusion of GA-MLP Hybrid and PCA Dimensionality Reduction," Procedia Comput Sci, vol. 230, pp. 522–531, 2023, doi: 10.1016/j.procs.2023.12.108.