

Original Article

Predicting Thyroid Disease using Optimized Multi-Layered Long Short-Term Memory (LSTM) Neural Network with Residual Attention Mechanism

R. Nagalakshmi¹, R. Priya²

^{1,2}Department of Computer Application, VISTAS, Chennai, Tamilnadu, India.

¹Corresponding Author : lakshmikumar@live.com

Received: 22 March 2023

Revised: 18 June 2023

Accepted: 29 June 2023

Published: 21 July 2023

Abstract - Thyroid disease is a common ailment that arises when the thyroid gland produces an inadequate amount of thyroid hormone. The challenge in diagnosing thyroid disease is that its symptoms resemble those of other conditions, making it difficult to detect. One of the most effective ways to diagnose thyroid problems is through blood tests, which measure the amount of thyroid hormones in the bloodstream. However, interpreting the complex data generated by these tests can be challenging. Early detection of thyroid issues is crucial to prevent complications and reduce mortality rates. To improve decision-making systems, medical data mining is increasingly using deep learning models. This study proposes a novel stacked residual long-short memory architecture (SR-LSTM) with an attention mechanism for predicting thyroid disease. The model employs an attention mechanism and layer stacking to enhance the accuracy of prediction. Additionally, this work is optimized using the Giant Trevally Optimizer (GTO) metaheuristic optimization algorithm. The performance results of the proposed model achieve outstanding performance in precision, recall, F1, and accuracy, with scores of 99.836%, 99.839%, 99.837%, and 99.931% than traditional methods.

Keywords - Thyroid, GTO, LSTM, Prediction.

1. Introduction

Thyroid disease is a widespread health condition globally, affecting a significant proportion of the population. In India alone, it has been estimated that approximately 42 million individuals suffer from thyroid-related illnesses such as iodine deficiency, thyroid goiter, and thyroid cancer [1][2]. The prevalence of thyroid diseases highlights the urgent need for advanced diagnostic technologies in the medical field to identify and prevent the onset of these conditions.

The thyroid gland is an endocrine gland found in vertebrates that produces two important hormones: total serum thyroxine (T4) and total serum triiodothyronine (T3). These hormones play a crucial role in regulating the body's metabolism, influencing processes such as body temperature, heart rate, and bone growth. When the thyroid gland malfunctions, it can lead to thyroid disease. Thyroid-related diseases are classified into three categories based on hormone levels: euthyroidism, hyperthyroidism, and hypothyroidism. Euthyroidism refers to the normal functioning of the thyroid gland. Hyperthyroidism occurs when the thyroid gland is overactive, producing excessive amounts of hormones. On the other hand, hypothyroidism is a condition characterized by an underactive thyroid gland that fails to produce an adequate amount of hormones.

The timely prediction and diagnosis of thyroid disease is of utmost importance to patients, as it can help them receive prompt treatment and minimize medical expenses

[3]. In recent times, machine learning and deep learning algorithms have gained popularity for their intelligent approach towards the early diagnosis and classification of thyroid disease. Additionally, the Internet of Things (IoT) has brought forth innovative solutions to the healthcare industry, such as data mining and big data analytics. Data mining techniques have played a crucial role in assessing patient risks and supporting health-related decision-making. As a result, healthcare professionals strive to categorize diseases in their early stages to ensure that patients receive suitable treatment as soon as possible. This proactive approach can go a long way in minimizing the negative impact of the disease and improving patient outcomes.

The problem addressed in this work is the data mining-based thyroid prediction presented. Thyroid disease is a common medical condition that can have serious health consequences if not detected and treated early. However, diagnosing thyroid disease is challenging due to its complex symptoms and the need for specialized blood tests to confirm the diagnosis. The use of data mining techniques can help analyze the large and complex data sets involved in thyroid disease diagnosis, leading to more accurate predictions and earlier interventions. The proposed approach in this work aims to improve the accuracy of thyroid disease prediction using a deep learning model with an attention mechanism and the Giant Trevally Optimizer for parameter tuning. The paper is structured as follows:



Section II provides an overview of existing machine learning and deep learning-based prediction models. Section III introduces the proposed work. The experimental results and information about the dataset used in Section IV are presented. Finally, Section V summarizes the conclusions drawn from the study.

2. Related Works

This section carried the thyroid prediction based on various methodologies referred to by different scholars.

In A. R. Rao et al.'s study [4], thyroid disease classification was addressed using the ML algorithms Decision Tree ID3 and Naive Bayes. The authors treated the thyroid disease classification as a binary classification issue and the Decision Tree method to achieve better prediction.

In [5], the authors evaluated the K-Nearest Neighbors (KNN) classifiers, Support Vector Machine (SVM), and regression for thyroid disease classification. Among these algorithms, SVM exhibited higher accuracy. A thyroid disease classification model was developed in [6] by employing Adaboost and Bagging techniques. Additionally, the authors introduced a novel pre-processing technique to handle noise in the dataset.

In [7], the Ensemble model was compared is evaluated in this work in terms of classification metrics and proved its effectiveness.

S. Kurnaz et al. [8] explored a genetic method for thyroid disease prediction. The study highlighted the significance of feature selection in determining classification performance.

However, the stacked ensemble learning model is presented by M. Karmeni et al. [9] for thyroid classification. This model surpassed that of individual ML algorithms, contributing to improved classification outcomes. For cardiovascular disease prediction, a deep neural network model with an attention mechanism was introduced in [10]. This model, named cardiovascular disease, demonstrated accurate handling of medical data.

Y. An et al. [11] analyzed time-aware performance for medical data. This method incorporated factors such as patient age, elapsed time between visits, and disease progression patterns to predict risks.

A graph learning model for disease prediction was proposed in [12], effectively utilizing information from various modalities.

In [13], a hybrid ML method model is presented for thyroid prediction. This model enhanced the classification accuracy compared to using an individual model. Y. Shen et al. [14] presented a score selection method to identify a suitable boosting feature through score calculation and its evaluation on three public datasets related to human microbiome profiling.

X. Li [15] introduced a dual-ranking algorithm for complex disease analysis. This algorithm incorporated data heterogeneity to improve the performance of risk prediction models, considering the influence of heterogeneous medical data on disease diagnosis and treatment. X. Zhang et al. [16] applied the concept of recurrent neural networks (RNNs) to Electronic Health Records (EHRs) for health risk prediction. RNNs were advantageous in retaining temporal information from the data, and context-aware information was also incorporated for risk prediction.

3. SR-LSTM MODEL

LSTM networks are a type of recurrent neural network (RNN) that have shown great effectiveness in processing long-term temporal dependencies. However, LSTM networks can face challenges when it comes to handling temporal irregularities and dependencies.

To address these challenges, a novel model called SR-LSTM has been investigated. The SR-LSTM model builds upon the LSTM architecture by introducing multiple LSTM layers instead of a single layer. This stacking of LSTM layers allows for a more powerful representation of temporal dependencies and enhances the overall data-mining capability of the model.

Furthermore, the SR-LSTM model incorporates an attention mechanism, which establishes a connection between the front and back of every two layers. This attention mechanism enhances the ability based on relevant temporal information and selectively attends to important features. By leveraging this attention mechanism, the SR-LSTM model effectively addresses the limitations of traditional LSTM networks in handling temporal irregularities and dependencies.

3.1. LSTM Architecture

The LSTM model is designed to address two common issues in traditional RNNs: the vanishing gradient problem and overfitting. To overcome these challenges, LSTM incorporates three special gate functions: the input gate, output gate, and forget gate. These gates play a crucial role in regulating the flow of information within the LSTM model.

Figure 1 illustrates the architecture of an LSTM model, showcasing these gate functions.

Let's consider an input data sequence, denoted as $x = x_1, x_2, \dots, x_T$, and a corresponding hidden state sequence, denoted as $h = h_1, h_2, \dots, h_T$, with a period T . The network generates a response sequence, represented as $y_o = y_{o1}, y_{o2}, \dots, y_{oT}$.

The equations governing the behavior of the gates in the network can be expressed as follows:

Forward gate (f_t):

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f) \quad (1)$$

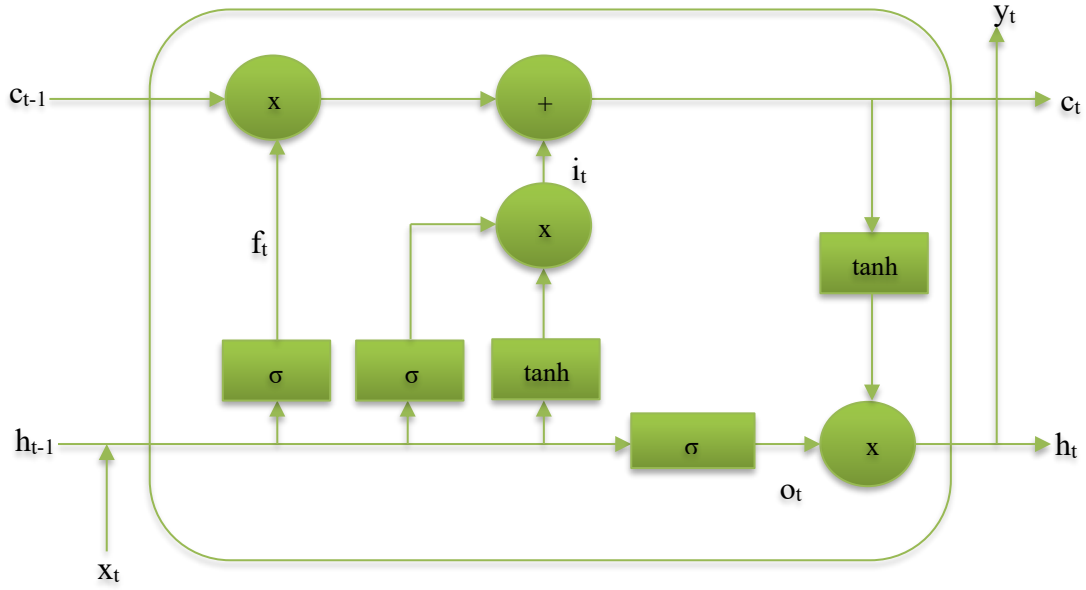


Fig. 1 The structure of LSTM

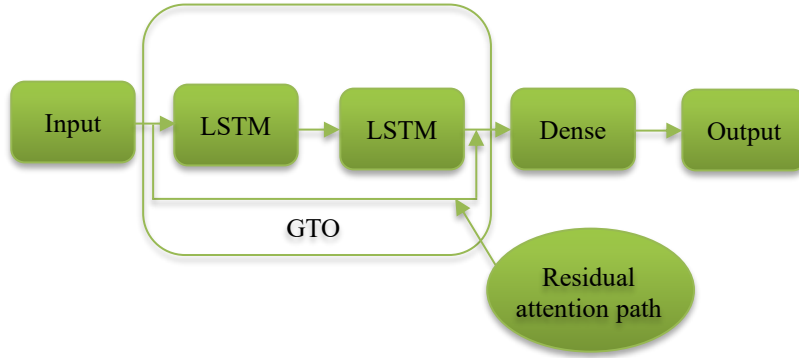


Fig. 2 SR-LSTM model

Input gate (i_t):

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i) \quad (2)$$

Candidate memory cell (\bar{c}_t):

$$\bar{c}_t = \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (3)$$

Memory cell (c_t):

$$c_t = f_t * c_{(t-1)} + i_t * \bar{c}_t \quad (4)$$

Output gate (O_t):

$$O_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o) \quad (5)$$

Hidden state (h_t):

$$h_t = O_t * \tanh(c_t) \quad (6)$$

Output response (yo_t):

$$yo_t = \sigma(W_{hy} * h_t + b_y) \quad (7)$$

Where $\sigma(x)$ represents the sigmoid function given by

$$\sigma(x) = 1/(1 + e^{-x}) \quad (8)$$

Equations (1) to (7) define the computations performed by the gates and the activation functions involved. The weights (W) and biases (b) are the parameters that determine the behavior of these functions within the network.

The SR-LSTM model incorporates a multi-LSTM layer instead of individual LSTM layers. This design choice enhances the model's ability to handle data irregularities. By establishing connections from the input layer to the final layer, the model improves its capacity to handle diverse data patterns.

The distribution of attention in the SR-LSTM model can be calculated as follows:

$$\alpha_i = p(z = i|X, q) = \frac{\exp(s(X_i, q))}{\sum_{j=1}^N \exp(s(X_j, q))} \quad (9)$$

Where, q is a query vector with the index of selected information by attention $z = 1, 2, \dots, N$,

$$\alpha_i = \text{softmax}(s(X_i, q)) \quad (10)$$

Where s is an attention score which can be expressed as follows

$$(s(X_i, q)) = \frac{x_i^T q}{d^2} \quad (11)$$

Then, the attention function is calculated as follows:

$$att((K, V), q) = \sum_{i=1}^N \alpha_i V_i = \frac{\exp(s(k_i, q))}{\sum_j \exp(s(k_j, q))} V_i \quad (12)$$

Where k denotes pair of key values.

3.1.1. GTO

The Giant Trevally Optimizer (GTO) is based on a metaheuristic algorithm introduced by H. T. Sadeeq et al. [21]. Inspired by the hunting behavior of the giant Trevally in nature, GTO models the mathematical strategy employed by the fish to solve real-world engineering problems.

The hunting process of the Trevally is characterized by three steps: forage moving, selecting a suitable hunting area, and attacking a target. These steps are mathematically formulated to address optimization problems effectively.

Forage Moving

In the moving step of the optimization process, the giant Trevally, represented by D , undertakes long-distance travel in search of food. This foraging movement can be mathematically described as follows:

$$D(t + 1) = B_p * RAND + ((HIGH - LOW) * RAND + LOW) * Levy(Dim) \quad (13)$$

Here, $D(t+1)$ denotes the future position vector of the Giant Trevally. B_p represents the best position obtained thus far. $Levy(Dim)$ refers to the Levy flight distribution, which introduces randomness to the searching behavior. $RAND$ is a random number ranging from zero to one.

The equation combines the current best position (B_p) with two random components. In the first term, $B_p * RAND$ accounts for the influence of the best position on Trevally's movement. The second term, $((HIGH-LOW) * RAND + LOW) * Levy(Dim)$, incorporates a random element and the Levy flight distribution to enhance the exploration capabilities of the algorithm.

Selecting a Suitable Area

In this stage, the optimization algorithm aims to identify the area with the highest concentration of food (seabirds). The process of finding a suitable area is mathematically modeled as follows:

$$D(t + 1) = B_p * a * RAND + ((MeanI - Di(t)) * RAND) \quad (14)$$

In Equation (14), $D(t+1)$ represents the future position vector of the Giant Trevally. B_p is the best position obtained thus far, and a is the position controlling coefficient that typically ranges from 0.3 to 0.4. $RAND$ is a random number between 0 and 1.

The equation combines the current best position (B_p) with two random components. In the first term, $B_p * a * RAND$ incorporates the influence of the best position and the position-controlling coefficient on Trevally's movement. The second term, $((MeanI - Di(t)) * RAND)$, incorporates the difference between the average of past best points ($MeanI$) and the current candidate solution ($Di(t)$) to guide the search process.

To calculate the average of past best points ($MeanI$), equation (15) is used:

$$MeanI = \frac{1}{M} \sum_{i=1}^M Xi(t) \quad (15)$$

Here, $MeanI$ represents the average of the past M best points, $Xi(t)$ denotes the i -th best point at time t , and M represents the number of GTO members.

Attacking Target

The behavior of Giant Trevally jumping from the water and catching its target can be mathematically represented as follows:

$$D(t+1) = LS + VD + JS \quad (16)$$

In equation (16), $D(t+1)$ represents the future solution vector of the Giant Trevally. LS denotes the launch speed required to attack the target, and it can be calculated using the following equation:

$$LS = Xi(t) * \sin(\Theta_2) * OBJ(Xi(t)) \quad (17)$$

VD represents visual distortion, which can be expressed as:

$$VD = \sin(\Theta_2) * D \quad (18)$$

JS stands for the jumping slope, which is calculated as:

$$JS = RAND * (2 - t - 2/T) \quad (19)$$

Here, t represents the current iteration, and T is the maximum number of iterations.

The GTO algorithm is employed to optimize the hyperparameters of an LSTM model, thereby improving its prediction performance. The hyperparameters that are tuned include Batch Size (BS), Optimizer, Learning Rate (LR), Number of Epochs (NE), Number of Hidden Units Per Layer (NHUPL), and Length of Time Lags (NTL).

To accomplish this optimization, a fitness equation is derived that takes into account both accuracy and mean square error. A new model is constructed for each iteration based on the newly-tuned hyperparameters.

The GTO-based SR-LSTM model parameter tuning process can be outlined using the following pseudocode:

1. Input: population size, coefficient 'a', and model parameters
2. Output: optimized hyperparameters (BS, optimizer, LR, NE, NTL, and NHUPL)
3. For $t = 1$ to T :
4. Generate a random population

5. Estimate the objective function
6. Sort the population
7. Identify the best solution
8. For n = 1 to N:
9. Execute Forage Moving
10. Calculate new positions using Equation 13
11. Execute Selecting Suitable Area
12. Update means and new positions using an equation
13. Execute Attacking Target
14. Update jumping position
15. Update LS, JS, and VD using equations 17, 18, and 19
16. Estimate model accuracy and MSE
17. If criteria are satisfied, return the best hyperparameters
18. Else, repeat steps 8-17
19. End
20. End

The pseudocode describes the iterative process of the GTO algorithm applied to optimize the hyperparameters of the SR-LSTM model. It involves generating a random population, evaluating the objective function, and sorting the population to identify the best solution. The algorithm then performs various operations, such as Forage Moving, Selecting a Suitable Area, and Attacking Target, to update the positions and calculate LS, JS, and VD. The model accuracy and MSE are estimated, and if the predefined criteria are met, the best hyperparameters are returned. Otherwise, the process is repeated until convergence or the maximum number of iterations is reached.

In the hyperparameter tuning process, a population of a specific size is randomly generated, and the fitness function is calculated based on accuracy and error rate. The coefficient 'a' is assigned a value of 0.4. The iteration count is set to 1000, and the population size is 30. The optimizer is executed 50 times for each function. The solutions are updated using equations 13, 17, 18, and 19. By sorting the obtained solutions, the best solution is determined. The hyperparameter tuning process continues until the required criteria are satisfied; the best parameter values are returned at this point.

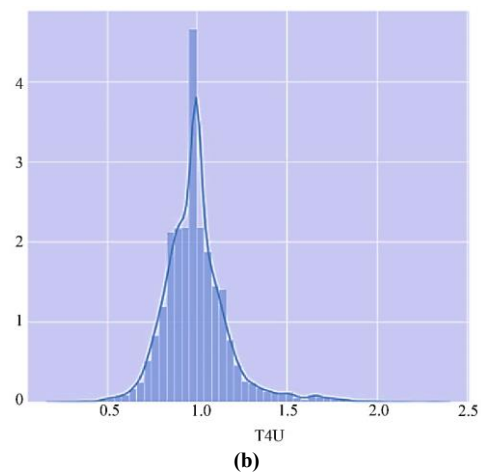
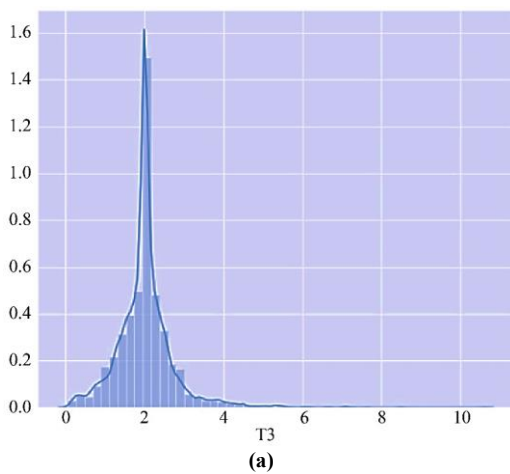
4. Performance Analysis

The SR-LSTM model is implemented in Python IDLE-3.7 using the pandas, numpy, and TensorFlow packages. The dataset used for training and evaluation is obtained from the Kaggle website (<https://www.kaggle.com/datasets/yasserhessein/thyroid-disease-data-set>). The dataset consists of 3 categories, 512 samples, and 5 attributes. To ensure proper evaluation, the dataset is split into training, validation, and test sets.

Figure 3 displays a visualization of the dataset, providing an overview of its structure and distribution. Additionally, Figures 4a, 4b, 4c, 4d, 4e, and 4f illustrate the visualization of attributes such as age, T3, T4, T4UFTI, and TBG measured in the dataset.

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	Thyroid surgery	1131 treatment	query hypothyroid	...	TT4 measured	T
1	41	F	0	0	0	0	0	0	0	0	...	1	1
2	23	F	0	0	0	0	0	0	0	0	...	1	1
3	46	M	0	0	0	0	0	0	0	0	...	1	1
4	70	F	1	0	0	0	0	0	0	0	...	1	1
5	70	F	0	0	0	0	0	0	0	0	...	1	6
...
3767	30	F	0	0	0	0	0	0	0	0	...	0	?

Fig. 3 Data set



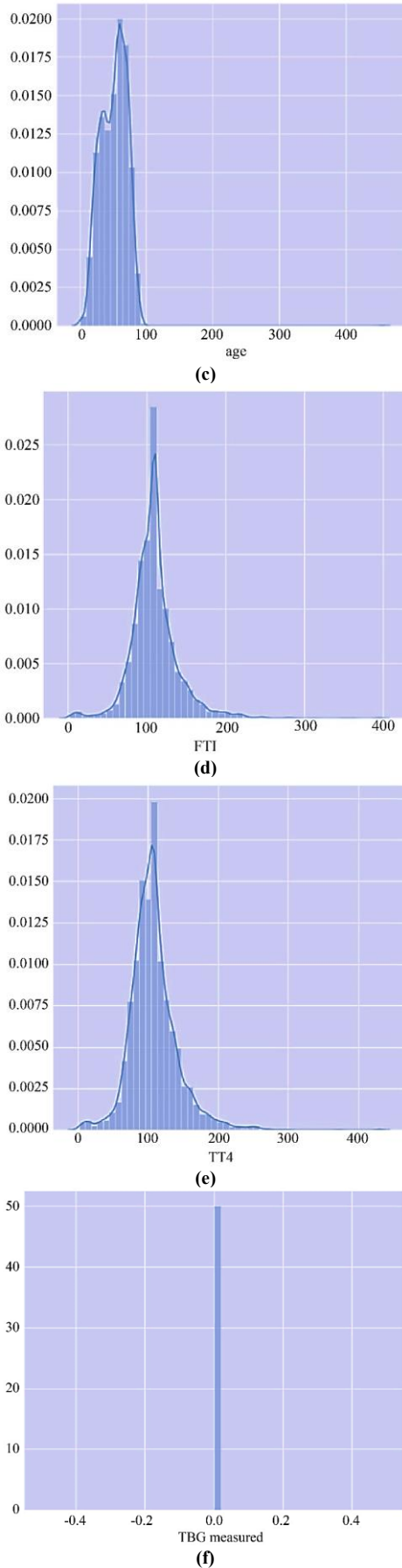


Fig. 4 (a) Visualization of age. (b) T3. (c) TT4. (d) T4U. (e) FTI algorithm. (f) TBG Measured

The performance evaluation of the SR-LSTM model is conducted using metrics such as Accuracy (Acc), Precision (Pre), Recall (Rec), and F1 score. These metrics assess the model's ability to classify instances and correctly provide insights into its overall performance.

Accuracy (Acc) is calculated as the ratio of the sum of true positive (TP) and true negative (TN) rates to the total number of instances, including true positives, false positives (FP), false negatives (FN), and true negatives. The formula for accuracy is given by equation (20).

$$ACC = \frac{TP+TN}{(TP + TN+FP + FN')} \quad (20)$$

Precision (Pre) measures the proportion of correctly predicted positive instances (TP) out of the total predicted positive instances (TP + FP). It helps evaluate the model's precision in identifying positive cases. Equation (21) represents the calculation of precision.

$$PRE = \frac{TP}{(TP + FP')} \quad (21)$$

Recall (Rec), also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances (TP) out of the total actual positive instances (TP + FN). It assesses the model's ability to capture positive cases. Equation (22) represents the calculation of recall.

$$REC = \frac{TP}{(TP + FN')} \quad (22)$$

The F1 score is a metric that combines precision and recall into a single value, providing a balanced measure of the model's performance. It is calculated as the harmonic mean of precision and recall using equation (23).

$$F1 \text{ Score} = \frac{2 \cdot PRE \cdot REC}{PRE + REC} \quad (23)$$

During the optimization process, the SR-LSTM model initially uses the Adam optimizer with a learning rate (LR) of 0.01. The number of epochs (NE) is set to 150, and the number of hidden units per layer (NHU) is set to 50. The batch size (BS) is set to 32. Through the application of the GTO algorithm, suitable hyperparameters are identified to improve the model's performance.

Table 1 provides a comprehensive comparison of six different optimizers based on their performance metrics, including Accuracy (Acc), Precision (Pre), Recall (Rec), and F1 score (F-Score) rates. The results indicate that the Adam optimizer achieved the highest scores across all metrics, with an Acc of 99.32%, a Pre rate of 96.84%, a Rec rate of 94.05%, and an F1 score of 95.12%.

The RMSprop optimizer is the second-best performer, with an Acc of 98.88%, a Pre rate of 97.2%, a Rec rate of 80.1%, and an F1 score of 84.3%. The Adadelta optimizer ranks third, achieving an Acc of 98%, a Pre rate of 91.72%, a Rec rate of 92.12%, and an F1 score of 91.65%. On the

other hand, the Adagrad optimizer exhibited the poorest performance among the compared optimizers, with an Acc of 63.88%, a Pre rate of 10.77%, a Rec rate of 16.78%, and an F1 score of 13.05%. Figure 5 presents a graphical representation of these results, providing a visual comparison of the performance of the different optimizers.

Table 1. The performance analysis of different optimizers

Optimizer	Acc	Prec	Rec	F1-score
Adam	99.32	96.84	94.05	95.12
RMSprop	98.65	95.55	92.42	93.73
Adadelata	63.88	10.77	16.78	13.05
Adagrad	63.88	10.77	16.78	13.05

Table 2. Summary of different hypermeters results

Hyper parameters	Type	Suggested parameters offer higher accuracy
HUL	20,40,60,90,120	60
BS	6,12,24,48,96,192	24
LR	0.0001,0.001,0.01,0.1	0.001
NE	25,50,75,100,125,150,175,200,225,250,275,300	250
Optimizer	Adam, RMSprop, Adagrad and Adadelata	Adam

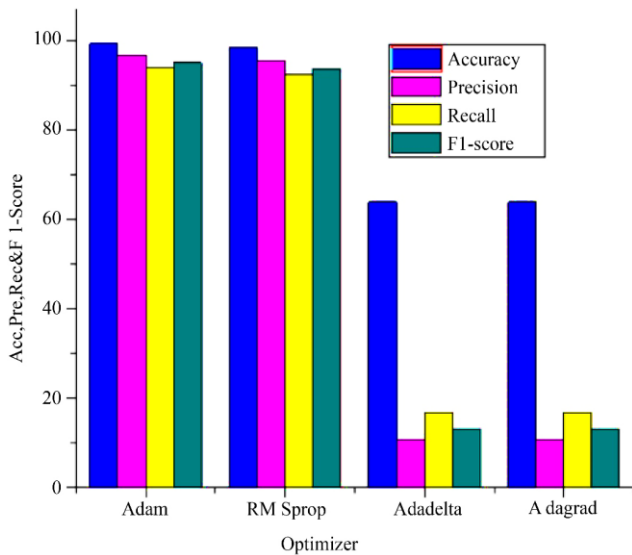


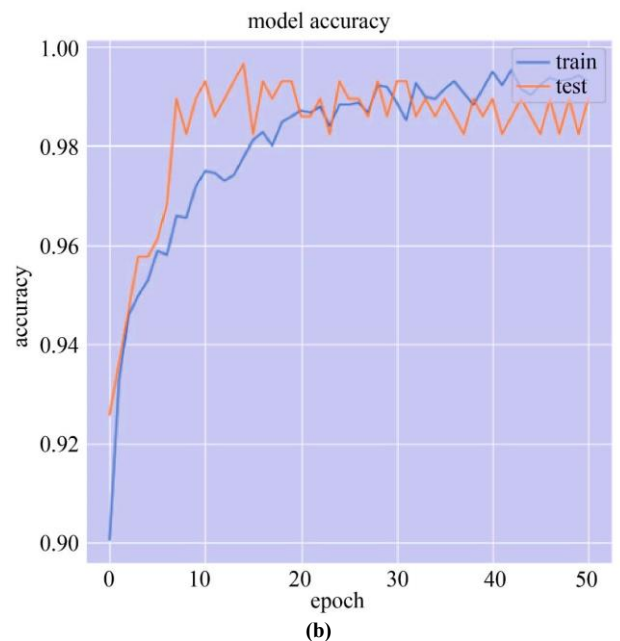
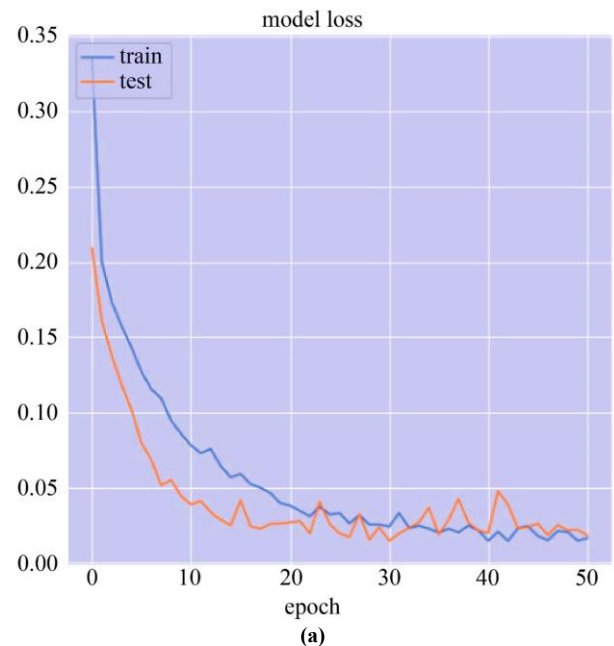
Fig. 5 Performance of the model for different optimizers

The performance of the SR-LSTM model is further analyzed by considering different hyperparameters, including Batch Size (BS), Learning Rate (LR), Number of Epochs (NE), Number of Hidden Units per Layer (HUL),

and Length of Time Lags (NTL). The analysis aims to identify suitable hyperparameters to enhance the model's performance.

Table 2 provides a summary of the recommended hyperparameters based on the analysis. It indicates that a batch size of 24 yields better results compared to other batch sizes. Additionally, a learning rate of 0.001 shows improved performance compared to other learning rates. Moreover, models with a higher number of hidden layers, specifically 60 hidden units per layer, demonstrate better performance.

Figure 6 presents visualizations related to the model training, accuracy, and loss analysis. Specifically, Figure 6a showcases the training progress of the model, while Figures 6b and 6c depict the accuracy and loss analysis, respectively.



```

Epoch 1/100
40/40 [=====] - 0s 7ms/step - loss: 0.3350 - accuracy: 0.9006 - val_loss:
0.2092 - val_accuracy: 0.9258
Epoch 2/100
40/40 [=====] - 0s 3ms/step - loss: 0.2003 - accuracy: 0.9328 - val_loss:
0.1609 - val_accuracy: 0.9364
Epoch 3/100
40/40 [=====] - 0s 3ms/step - loss: 0.1732 - accuracy: 0.9458 - val_loss:
0.1376 - val_accuracy: 0.9470
Epoch 4/100
40/40 [=====] - 0s 3ms/step - loss: 0.1573 - accuracy: 0.9497 - val_loss:
0.1183 - val_accuracy: 0.9576
Epoch 5/100
40/40 [=====] - 0s 3ms/step - loss: 0.1429 - accuracy: 0.9529 - val_loss:
0.1017 - val_accuracy: 0.9576
    
```

(c)
Fig. 6 (a) Loss analysis of SR-LSTM, (b) Accuracy analysis of SR-LSTM, (c) Model training

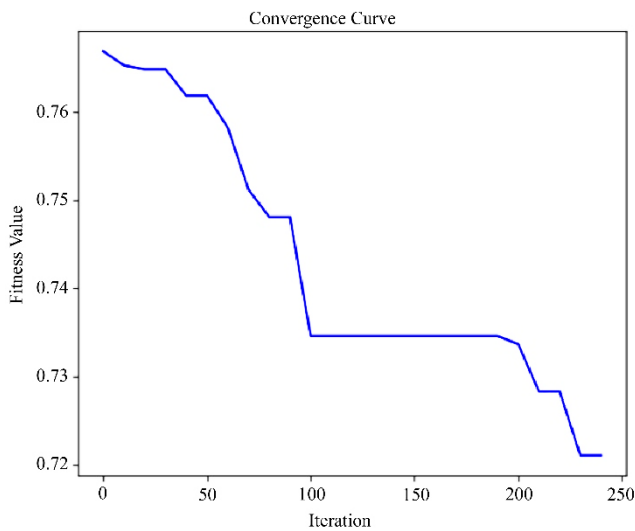


Fig. 7 Convergence curve of GTO

The performance of the optimization process can be evaluated based on two key aspects: the quality of the solution obtained and the convergence rate analysis, which provides insights into the randomness and speed of the optimizer.

Figure 7 illustrates the convergence curve of the Giant Trevally Optimizer (GTO) used for hyperparameter tuning. This curve depicts how the optimization process progresses over iterations. By analyzing the convergence curve, we can assess the effectiveness of GTO in finding the optimal solution and observe the rate at which it converges towards the best solution.

Table 3 provides a comparison of the performance of the SR-LSTM algorithm with other prediction models. The results indicate that the SR-LSTM model outperformed the

other models in terms of precision (Pre), recall (Rec), accuracy (Acc), and F1-score rates. The SR-LSTM model achieved a maximum accuracy of 98.7%, highlighting its effectiveness for risk analysis applications.

Figure 8 visually represents the suitability of the SR-LSTM model for risk analysis applications.

Table. 3 Performance analysis of an optimized SR-LSTM model

	Pre	Rec	Acc	F-Score
SVM	89.97	90.38	94.8	91.67
LSTM	93.11	94.9	97.2	96.2
SR-LSTM	95.8	97.89	99.2	98.1

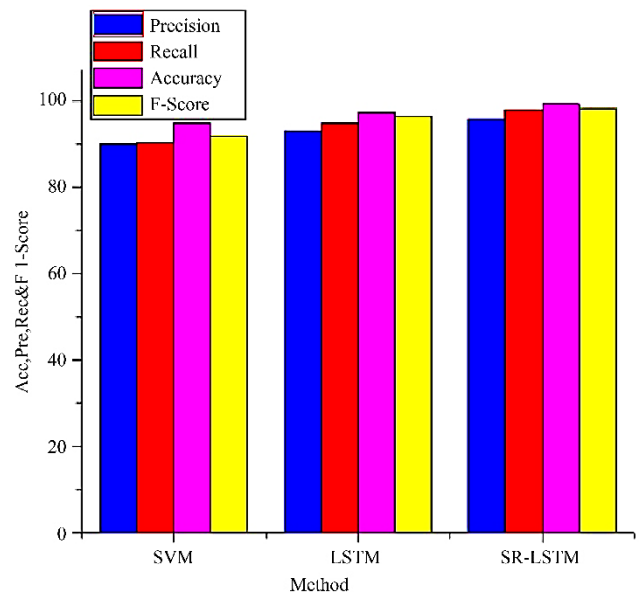


Fig. 8 Performance of the model in comparison with other models

5. Conclusion

The medical field has witnessed an increasing adoption of machine learning (ML) algorithms due to their high accuracy and adaptability. However, one of the main limitations of these techniques is the manual selection of features, which can significantly impact the quality of results. A novel approach leveraging deep learning (DL) based prediction models has been proposed to overcome these challenges. In this particular study, a new model called SR-LSTM was developed specifically for predicting thyroid disease. This model effectively addresses temporal dependencies by incorporating cascaded LSTM layers and attention mechanisms. By leveraging the power of DL, the SR-LSTM model can automatically learn and capture complex patterns and dependencies in the input data, leading to improved prediction accuracy.

Furthermore, to further enhance the performance of the SR-LSTM model, GTO-aided parameter tuning was employed. This optimization technique optimizes the hyperparameters of the model, such as learning rate, batch size, and the number of hidden units, to maximize its performance in terms of accuracy, precision, and F1-score rates.

This integration of GTO enables the model to find the most suitable set of hyperparameters for the specific problem at hand. By combining the strengths of DL-based prediction models, such as SR-LSTM, with the optimization capabilities of GTO, this study offers a promising approach for accurate and reliable predictions in thyroid disease diagnosis.

References

- [1] Ling Chen et al., "Mining Health Examination Records—A Graph-Based Approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2423-2437, 2016. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Feyzullah Temurtas, "A Comparative Study on Thyroid Disease Diagnosis Using Neural Networks," *Expert Systems with Applications*, vol. 36, no. 1, pp. 944-949, 2009. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Fabrizio Monaco, "Classification of Thyroid Diseases: Suggestions for a Revision," *The Journal of Clinical Endocrinology & Metabolism*, vol. 88, no. 4, pp. 1428-142, 2003. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] Amulya.R. Rao, and B.S. Renuka, "A Machine Learning Approach To Predict Thyroid Disease At Early Stages of Diagnosis," *2020 IEEE International Conference for Innovation in Technology (INOCON)*, pp. 1-4, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Ankita Tyagi, Ritika Mehra, and Aditya Saxena, "Interactive Thyroid Disease Prediction System Using Machine Learning Technique," *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 689-693, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] D. Priyadharsini, and S. Sasikala, "Efficient Thyroid Disease Prediction Using Features Selection and Meta-Classifiers," *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1236-1243, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Muntadher Alsaadawi, and Eftal Şehirli, "The Efficiency of Ensemble Techniques in Predicting Thyroid Disorder: A Comparative Study," *2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 834-840, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Sefer Kurnaz et al., "A High Efficiency Thyroid Disorders Prediction System with Non-Dominated Sorting Genetic Algorithm NSGA-II as a Feature Selection Algorithm," *2020 International Conference for Emerging Technology (INCET)*, pp. 1-6, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] Mejdi Karmeni et al., "Towards an Accurate Stacked Ensemble Learning Model for Thyroid Earlier Detection," *2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1-8, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Ying An et al., "High-Risk Prediction of Cardiovascular Diseases Via Attention-Based Deep Neural Networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 1093-1105, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] Y. An, K. Tang, and J. Wang, "Time-Aware Multi-Type Data Fusion Representation Learning Framework for Risk Prediction of Cardiovascular Diseases," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 6, pp. 3725-3734, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Shuai Zheng et al., "Multi-Modal Graph Learning for Disease Prediction," *IEEE Transactions on Medical Imaging*, vol. 41, no. 9, pp. 2207-2216, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] Guoxian Yu et al., "Deepida: Predicting Isoform-Disease Associations by Data Fusion and Deep Neural Networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 4, pp. 2166-2176, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Yang Shen et al., "Ensdeepdp: An Ensemble Deep Learning Approach for Disease Prediction through Metagenomics," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Xingyi Li et al., "A Dual Ranking Algorithm Based on the Multiplex Network for Heterogeneous Complex Disease Analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 4, pp. 1993-2002, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [16] Xianli Zhang et al., "Context-Aware and Time-Aware Attention-Based Model for Disease Risk Prediction with Interpretability," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3551-3562. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Kirubha.M et al., "Analysis of Thyroid Disease Using K Means and Fuzzy C Means Algorithm," *SSRG International Journal of Computer Science and Engineering*, vol. 6, no. 10, pp. 1-6, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] Qiao Ning et al., "SSKM_Succ: A Novel Succinylation Sites Prediction Method Incorporating K-Means Clustering with a New Semi-Supervised Learning Algorithm," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 1, pp. 643-652, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [19] Klaus Greff et al., "LSTM: A Search Space Odyssey," *IEEE Trans Neural Networks Learn Systems*, vol. 28, no. 10, pp. 2222-2232, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [20] Raniah Zaheer, and Humera Shaziya, "A Study of the Optimization Algorithms in Deep Learning," *2019 Third International Conference on Inventive Systems and Control (ICISC)*, pp. 536-539, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Haval Tariq Sadeeq, and Adnan Mohsin Abdulazeez, "Giant Trevally Optimizer (GTO): A Novel Metaheuristic Algorithm for Global Optimization and Challenging Engineering Problems," *IEEE Access*, vol. 10, pp. 121615-121640, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Pei Liu et al., "Optimizing Survival Analysis of Xgboost for Ties to Predict Disease Progression of Breast Cancer," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 1, pp. 148-160, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]