Optimizing LSTM-Based Model with Ant-Lion Algorithm for Improving Thyroid Prognosis

Maria Yousef
Department of Computer Science, University of Petra, Amman, Jordan

***Abstract*—In the healthcare sector, early and accurate disease detection is essential for providing appropriate care on time. This is especially crucial in thyroid problems, which can be difficult to diagnose because of their many symptoms. This study aims to propose a new thyroid disease prediction model by utilizing the Ant Lion Optimization (ALO) approach to enhance the hyperparameters of the Long Short-Term Memory (LSTM) deep learning algorithm. To achieve this, after the preprocessing step, we utilize the entropy technique for feature selection, which selects the most important features as an optimal subset of features. The ALO is then employed to optimize the LSTM, identifying the optimal hyperparameters that can influence the model and enhance its efficiency. To assess the suggested methodology, we chose the widely used thyroid disease data. This dataset contains 9,172 samples and 31 features. A set of criteria was used to evaluate the model's performance, including accuracy, precision, recall, and F1 score. The experimental results showed that: 1) the entropy technique in the feature selection step can reduce the total number of features from 31 to 10; 2) the recommended strategy, which selected the optimal hyperparameter for the LSTM using the Alo algorithm, improved the classifier overall by 7.2% and produced the highest accuracy of 98.6%.**

Keywords*—Thyroid disease; LSTM; ALO; prediction model; optimization algorithm

I. Introduction

Incidences of thyroid illness have increased recently. The thyroid gland has one of the most significant functions in controlling metabolism. The thyroid gland secretes levothyroxine (T4) and triiodothyronine (T3), and irregularities in the gland can result in a variety of disorders [1]. Inadequate thyroid hormones can cause hyperthyroidism and hypothyroidism, the most prevalent disorders. If individuals with thyroid disease do not manage their metabolism and body weight appropriately, they may experience severe problems such as reduced taste, diminished smell, sadness, and memory loss. According to the World Health Organization, 20 million people have had some form of thyroid disease in the past 2 years [2]. Moreover, thyroid disorders affect 12% of the population at least once in their lives. Statistics indicate that we should not disregard thyroid-related illnesses lightly. As a result, the use of prediction systems for thyroid disease detection has grown in significance and is currently a field of research.

Health care professionals are always concerned with finding a cure for illnesses, so making an accurate diagnosis when a patient needs it is crucial. In the traditional way, only an experienced doctor is able to thoroughly assess the situation when predicting a thyroid problem; otherwise, it can be a challenging procedure that results in an incorrect diagnosis. A proactive prediction of thyroid illness is critical to appropriately treating the patient at the appropriate time, saving lives and medical [3].

Current data processing and computer technologies have made it possible to identify different forms of thyroid illness, such as hyperthyroidism and hypothyroidism, and forecast thyroid disease early on. These advances have also made machine learning and deep learning approaches more accessible. Most existing research efforts focus on binary classification issues, categorizing individuals as either thyroid sufferers or healthy. In contrast, there are very few multiclass-based detection methods. Prior research in this area also concentrated on directly implementing machine learning algorithms and contrasting their outcomes, regardless of the attributes employed, their significance, and their quantity, which impacts the models' accuracy and gives rise to the high-dimensionality problem. Despite the high accuracy of reporting techniques, their evaluation on less than 1000 samples yields inconsistent results. To address these concerns, this study presents the following contributions:

- Proposed a new thyroid prediction model that employs a long-short-term memory (LSTM) deep learning algorithm, bolstered by the Ant Lion Optimization (ALO) metaheuristic algorithm. This model serves as an enhanced machine learning-based method for predicting thyroid illness, specifically targeting a multi-class problem.
- Handling the high dimensionality problem and selecting the optimal subset of features that can enhance the model's performance through machine learning-based feature selection using an additional tree classifier.
- Demonstrate the impact of the ALO optimization algorithm on the performance of the LSTM classifier through the suggested experiments.

The rest of the article is structured as follows: Section II summarises the previous attempts to identify and classify thyroid-related illnesses. Section III provides a background on all of the algorithms and techniques used in this study. Section IV presents the suggested approach to solving the thyroid illness prediction issue. Section V describes the experimental results we achieved in our investigation, while Section VI presents the study's conclusion along with our inputs.

II. Related Work

This section covers summaries of many machines learning based thyroid disease prediction systems.

A study by [4] suggested a way to predict thyroid disease that uses five different classification algorithms: Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Multilayer Perceptron, and Naive Bayes. The authors of this paper gathered the dataset from the Iraqi population, which included 1250 male and female individuals ranging in age from 1 year to 90 years. It includes three types of classes: hypothyroidism, normal, and hyperthyroidism—as well as 17 attributes that represent patient information and relevant medical analyses. A comparison and implementation of several algorithms performances was conducted. The result shows that RF outperformed other algorithms in the ability to predict; it achieved 98% accuracy.

In an effort to improve the effectiveness of the KNN algorithm in thyroid prediction, Abbad Ur Rehman et al. [5] examined two well-known feature selection methods: chi-square feature selection and L1-norm feature selection. This study utilized two datasets for training one from KEEL dataset repository that contains 7200 cases, with 13 features and another from a registered hospital in Pakistan was used for testing. It contains 3300 instances and 11 features. After assessing the performance of unique distance functions of KNN, Euclidean and Cosine distance functions obtained 97% accuracy at the k value of 1 with the use of chi-square-based feature selection approach for new suggested dataset.

A comparative study on thyroid illness detection was conducted by [6]. In this study, the authors utilizing a multilayer neural network, a learning vector quantization-based neural network, and a probability-based neural network. This study uses the UCI machine learning database's thyroid illness dataset. There are five features and 215 samples in total. The experiment's findings demonstrated that neural network structures can effectively assist in thyroid illness diagnosis; the probability-based neural network outperformed other neural networks, achieving an accuracy rate of 94.8%.

Ahmad et al. in [7] described a mixed decision support system that uses k nearest-neighbour (kNN) weighted preprocessing, an adaptive neurofuzzy inference system (ANFIS), and linear discriminant analysis (LDA). The LDA-kNN-ANFIS approach starts with LDA, which lowers the illness dataset dimensionality and removes pointless features. The second step uses a kNN-based weighted preprocessor to preprocess a subset of the attributes. In the final step, the adaptive neurofuzzy inference system uses preprocesses, chosen characteristics as an input for diagnosis. The suggested technique tested the system's overall performance using a thyroid illness dataset from the University of California, Irvine's machine learning repository. The calculated classification analysis using this approach's accuracy, sensitivity, and specificity values came out to be 98.5, 94.7, and 99.7%, in that order.

III. BACKGROUND

A. Ant Lion Optimization (ALO)

It is a metaheuristic optimizer that draws inspiration from nature, the ALO algorithm emulates the natural predatory behaviour of ant lions [8]. The algorithm aims to resolve complex optimization issues by emulating the communication between antlions, representing superior solutions, and ants, representing potential answers. By building traps to catch other ants, antlions direct the hunt for the best answers [9]. The ALO algorithm successfully explores the search space by using a random walk mechanism for ants that is impacted by the placement of antlions. The ALO algorithm guarantees fast convergence to high-quality solutions and thorough coverage of the solution space by striking a balance between exploration and exploitation. To direct the search for the best answers, it makes use of heuristic data and pheromone trails [10], where the probability of the pheromone update is defined as follows:

$$P_i = \left[\frac{F_i^n}{\sum_i F_i^n}\right] \quad (1)$$

Where:

P_i: is a possibility that any path will be selected.

F_i^n: is the strength of the pheromone on every path.

The ALO algorithm is divided into two populations: an ant population and an antlion population. The general steps that ALO employs to alter these two sets and get the global optimum for a specific optimization issue are as follows:

- The primary search agents of the ALO are the Ant set, which is first initialized with random values.
- In every iteration, the fitness value of every ant is assessed using an objective function.
- Ants use random walks to get around the antlions in the search area.
- There is never an assessment of the antlion population. In actuality, antlions moved to the new placements of ants in subsequent iterations if the ants improved from their initial position, which they presumed to be on.
- Each ant has a single antlion, which updates its location whenever the ant becomes fitter.
- Additionally, there is an elite antlion that affects ant movement regardless of distance.
- Any antlion that surpasses the elite will be replaced by the elite.
- Once an end condition is satisfied, steps b through g are repeated.
- The elite antlion's location and fitness value are given back as the most accurate estimate of the global optimum.

By repeating this procedure, the ALO algorithm efficiently converges to optimum or nearly optimal solutions for a range of optimization issues. Fig. 1 displays the ALO algorithm flowchart:

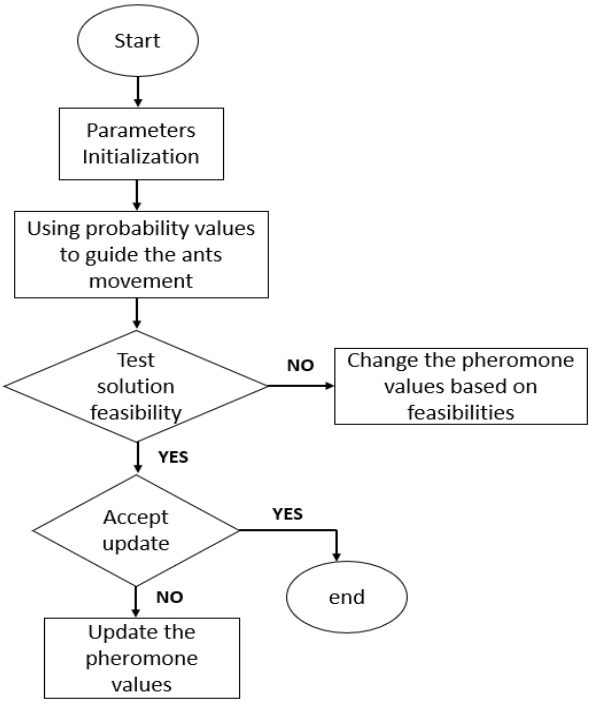

Fig. 1. ALO flowchart.

B. Long Short Term Memory (LSTM)

A unique kind of recurrent neural network (RNN) architecture was created to handle sequential input and overcome the shortcomings of conventional RNNs in identifying long-term relationships [11]. Long-term state memory (Long-term Memory) is achieved by LSTM networks through the use of memory cells that hold onto information over extended periods of time [12]. Each cell in LSTM consists of three gates: the input gate, the forget gate, and the output gate. These gates control the information flow by selecting which input elements to keep, which to reject, and which to go on to the subsequent time step [13]. For applications requiring time series, prediction processing, and other sequential data, LSTM networks are especially useful because they can learn to remember pertinent information and discard irrelevant data by dynamically modifying these gates.

The operation of an LSTM can be explained through the repetitive process that follows by each element in an ordered series. For each time step, the input gate determines how much a new input should impact the cell's current state. Conversely, the forget gate decides whether to remember parts of the previous cell state [14]. The output gate then generates the current output by merging the updated cell state with the new input. This output is not only forwarded to the following layer or ultimate prediction but also returned to the cell in order to analyze the subsequent element in the sequence. This approach enables LSTMs to learn relationships spanning several time steps, which is essential for modelling sequential data well. It also lets LSTMs keep an updated version of short-term memory at each step [15].

In traditional RNN, this repeating unit has a straightforward structure, such as the Tanh layer. While the LSTM and repeating modules have the same design, they differ in their architecture [16]. Rather than having a single neural network layer, Fig. 2 shows four layers that interact in a very special way.

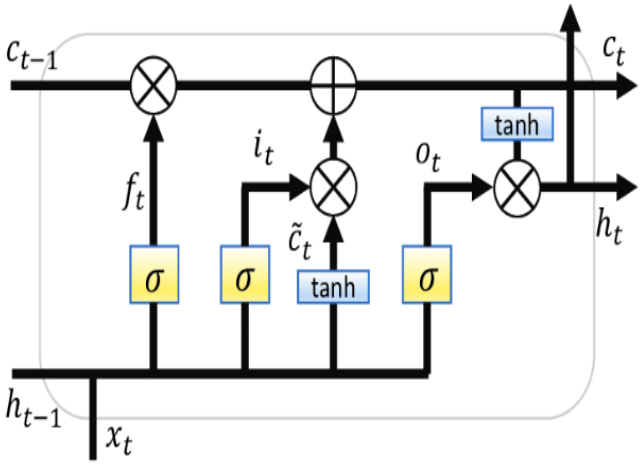

Fig. 2. LSTM architecture.

Where, h_t stands for the current layer's output, x_t for the current input, and h_{t-1} for the output of the previous layer. C_t indicates the current layer's cell status. While the symbols C_{t-1}, ft, and $\tilde{C}_t$ represent the cell state, forget gate layer, and candidate vector, respectively.

We can establish the system by using the forget gate layer given by Eq. (2):

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f) \quad (2)$$

Where, σ: is the sigmoid function, W_f :denotes the forget layer's weights, and b_f: denotes the forget layer's bias. Next, we employ the following Eq. (3), and (4) to update the input layer:

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = tanh(W_c . [h_{t-1}, x_t] + b_c) \quad (4)$$

Where, i_t : denotes the output of the input gate layer, W_i: denotes the weights of the input gate layer, b_i : indicates its bias, the output of the input candidate layer is represented by $\tilde{C}_t$. Tanh is the output obtained from the input candidate layer, W_c stands for the hyperbolic tangent function, and b_c is the candidate layer's bias. Next, we employ the following Eq. (5) to defined the output layer:

$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o) \quad (5)$$

Where, the output of the output gate layer is represented by the symbol o_t. W_o, and $\boldsymbol{b_o}$, which stand for output gate weights and bias, respectively. Eq. (6) provides the output of the output gate layer. Eq. (6) yields the output of the output gate layer, and Eq. (7) provides the current cell state.

$$h_t = o_t * tanh(C_t) \quad (6)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t) \quad (7)$$

IV. Research Methodology

Fig. 3 illustrates the five primary steps of the suggested early diagnostic model. The first stage explains the Thyroid disease dataset. Preprocessing is the second stage, where the main goals are analysis, resizing, and enhancing the quality of the dataset. Using the decision tree entropy technique, features are selected in the third step to create the best possible subset of data. The LSTM-based model uses these as training vectors and then employs a meta-heuristic technique to maximize convergence and improve performance. Ultimately, a range of assessments will be conducted to classify the illness classes and evaluate the suggested system utilizing a variety of criteria.

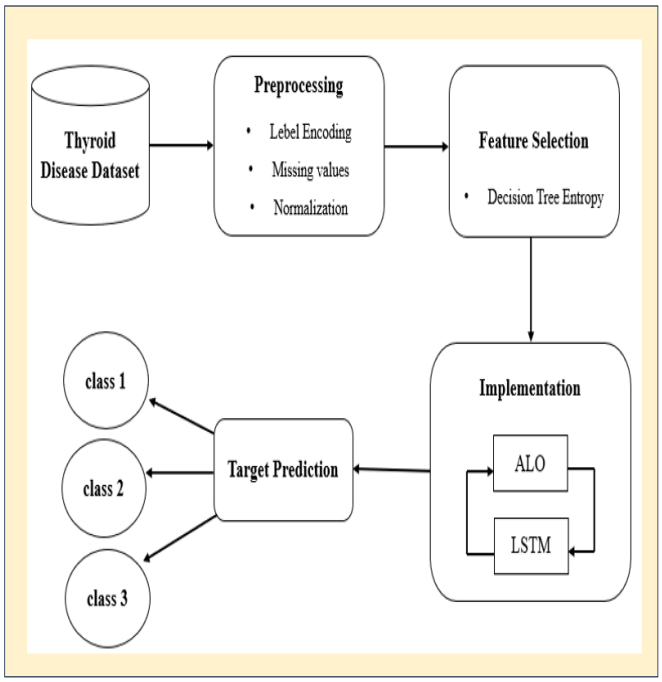

Fig. 3. The proposed system schema.

A. Thyroid Disease Dataset

This investigation used the thyroid dataset from the University of California, Irvine (UCI) repository [17]. We chose this particular data set because it is well-established in the field of thyroid illness classification from earlier studies. The 9,172 samples and 31 features in the dataset record a variety of medical characteristics and thyroid-related test findings. The main features contain patient demographic traits like sex and age, and multiple medical necessities and treatments such as sick, on_thyroxine, thyroid_surgery, TSH measurement, and pregnancy. The dataset also includes various thyroid hormone levels (TSH, T3, TT4, T4U, FTI, and TBG), with some missing values. Additionally, it records the referral source and a target variable, indicating the outcome or diagnosis. Additionally, the dataset contains many thyroid hormone levels with some missing data (TT4, T3, TSH, TT4, FTI, TBG, and T4U). It also logs the referral source and a target variable that indicates the diagnosis or outcome. This dataset is useful for analyzing thyroid diseases because it provides a comprehensive view of patient health and thyroid-related medical history. Table I displays a description of each feature.

TABLE I. ASD Dataset Features

Features	Description	Type
age	The patient age	integer
sex	Identifies a sex of patient	String
on_thyroxine	Whether the patient is taking thyroxine medication	Bool
on_antithyroid_meds	If the patient takes antithyroid medication	
query on thyroxine	Did the patient performed a thyroxine test?	
sick	Does the patient suffer from other diseases?	
pregnant	Whether patient is pregnant	
thyroid_surgery	Whether patient has undergone thyroid surgery	
I131_treatment	Whether the patient is receiving therapy with I131	
query_hypothyroid	Whether the person thinks they are hypothyroid	
lithium	If the patient takes lithium medication	
goitre	Whether the patient suffers from goitre	
tumor	If the patient has a tumour	
hypopituitary	Whether the patient has an overactive pituitary gland	Float
psych	If the patient takes psychiatric or psychological medication	Bool
TSH_measured	Whether TSH was measured in the blood	
TSH	TSH level in blood from lab work	Float
T3_measured	Whether a blood test for T3 was performed	Bool
T3	T3 level in blood from lab work	Float
TT4_measured	Whether TT4 was measured in the blood	Bool
TT4	T4 level in blood from lab work	Float
T4U_measured	Whether a blood test for T4U was performed	Bool
T4U	T4U level in blood from lab work	Float
FTI_measured	Whether a blood test for FTI was performed	Bool
FTI	FTI level in blood from lab work	Float
TBG_measured	If blood TBG levels were tested	Bool
TBG	TBG level in blood from lab work	float
referral_source	Refers to the healthcare facility or physician from whom the patient was referred for additional assessment, diagnosis, or treatment.	String
target	Final diagnosis	String
patient_id	Unique patient ID	

B. Preprocessing

The information prehandling process is regarded as a critical step in data mining tasks. Crude information frequently contains missing or erroneous data. This leads to a state of confusion in machine learning forecasting. This step includes cleaning, extracting, and modifying data into a suitable format for machine processing [18].

1) Label encoding: The categorical variables in the thyroid illness dataset used in this study consist of categorical values that must be converted into a quantitative form. The data is transformed into certain numerical forms using the one-hot encoding technique [19]. This labelling procedure assigns a value of 0 or 1 to each data point based on its category, enabling machine learning algorithms to differentiate between the two groups. During this phase, we convert data for features such as "sex", "sick", etc. from Boolean type values (yes/no) or (t/f) to binary values (1/0), as shown in Fig. 4.

1: age Numeric	2: sex	3: on_thyroxine	4: query_on_thyroxine	5: on_antithyroid_meds	6: sick	7: pregnant	8: thyroid_surgery	9: I131_treatment	10: query_hypothyroid
29.0	0	1	0	0	0	0	0	1	
29.0	0	1	0	0	0	0	0	0	
41.0	1	1	0	0	0	0	0	0	
36.0	1	1	0	0	0	0	0	0	
32.0	1	1	1	0	0	1	0	0	
60.0	1	0	1	0	0	0	1	0	
77.0	1	0	1	0	0	0	1	0	
28.0	0	0	1	0	0	0	1	0	
28.0	0	1	0	0	0	1	1	0	
28.0	0	0	1	0	0	1	0	0	
54.0	1	0	0	0	0	1	0	0	
42.0	1	0	1	0	0	1	0	0	
51.0	1	1	0	0	0	1	0	0	
51.0	1	0	0	0	0	0	0	0	
37.0	0	1	0	0	0	0	0	0	
16.0	1	0	0	0	0	1	0	0	

Fig. 4. Part of dataset after levelling.

2) Missing values: Missing data is a common issue, especially in the health industry. Almost every feature and patient record includes some missing data. Thyroid disease dataset has missing values for a number of features, including "TSH," "goitre," "TT4", and "T3." 3. Removing samples or features that have missing values is ineffective because it may lead to the loss of important information during the diagnosis process [20]. At this stage, we apply a filter approach that uses the mean of the training data to fill in the gaps and eliminate missing values from the dataset.

3) Normalization: The process of converting data into a suitable format that may be used for mining is known as data transformation. One method of data scaling is normalization, which is the act of bringing attribute values into a specified range [21]. The normalization strategy has been applied prior to the feature selection and modelling stages since varying attribute values make it more difficult to compare attributes and reduce the learning capacity of algorithms. Therefore, this study applied the Min-Max normalization approach to numeric data types. Each feature's greatest value is changed to 1, the lowest value to 0 [22], and all other values are transformed to a decimal interval. Fig. 5 shown the part of dataset after normalization.

22: TT4 Numeric	23: T4U_measured String	24: T4U Numeric	25: FTI_measured String	26: FTI Numeric	27: TBG_measured String	28: TBG Numeric
0.2006688963210...	0	0.47685185185185...	0	0.07457935425...	0	0.0645322661...
0.2107023411371...	0	0.63888888888888...	0	0.05979990904...	0	0.0595297648...
0.1989966555183...	0	0.7546296296296...	0	0.06093678944...	1	0.0545272636...
0.1989966555183...	0	0.24537037037037...	0	0.04956798544...	1	0.1295647823...
0.0903010033444...	0	0.37499999999999...	0	0.08367439745...	1	0.1795897948...
0.1086956521739...	0	0.18055555555555...	0	0.05297862664...	1	0.1295647823...
0.0535117056856...	0	0.28240740740740...	0	0.07457935425...	1	0.2696348174...
0.1906354515050...	0	0.66203703703703...	0	0.08140063665...	0	0.1145572786...
0.1237458193979...	0	0.64351851851851...	0	0.08367439745...	0	0.2146073036...
0.1354515050167...	0	0.28240740740740...	0	0.08481127785...	0	0.1545772886...
0.2190635451505...	0	0.14351851851851...	0	0.08253751705...	0	0.1345672836...
0.1722408026755...	0	0.27777777777777...	0	0.09163256025...	0	0.1445722861...
0.1638795986622...	0	0.28240740740740...	0	0.07457935425...	0	0.1295647823...
0.1906354515050...	0	0.09259259259259...	0	0.10982264665...	0	0.1595797898...
0.2023411371237...	0	0.66203703703703...	0	0.04956798544...	0	0.1545772886...

Fig. 5. Part of dataset after normalization.

C. Feature Selection

The information prehandling process is regarded as a critical step in data mining tasks. Crude information frequently contains missing or erroneous data. This leads to a state of confusion in machine learning forecasting. This step includes cleaning, extracting, and modifying data into a suitable format for machine processing [18].

The extra tree classifier [25], using the training dataset, generates numerous decision trees at random. After dividing the decision tree's nodes, we apply either the entropy criteria or the Gini index. In our feature selection procedure, we used entropy measures [26]. The entropy is calculated using Eq. (8). The proportion of rows in the dataset with the label i is represented by pi, while the number c denotes the unique class labels.

$$Entropy(E) = \sum_{i=1}^{c} p_i log_2(p_i) \quad (8)$$

Entropy evaluates the information about the condition of the features with respect to the target. We calculate the entropy of the features derived from each choice, and identify the significant features by computing the cumulative entropy values for each feature. Referred to as the shortlisted features, this group of traits has a high entropy. Fig. 6 uses Multilevel Feature Selection (MLFS) [27] to illustrate the significance of the features. With n_estimators = 300 and max_depth = 21, we employed an ETC classifier for MLFS, which rated the features according to their value. In this step, entropy was able to reduce the number of features from 30 to 10. The subset of features with scores greater than 0.02 is selected to train the learning models, namely 'Thyroid_surgey', 'pregnant', 'tumor', 'psych','TSH','T3', 'TT4', 'T4U', 'FTI' and 'TBG'.

D. Implementation

This stage involves using the Ant Lion Optimizer (ALO) as an optimizer for an LSTM model. During the optimization process, we may modify the algorithm movement equations to update the LSTM model weights and biases. The goal is to find the ideal values for the model's parameters to minimize the loss function and enhance performance.

There are numerous essential steps in the suggested approach when utilizing the ALO algorithm to optimize the hyperparameters of LSTM networks. The first step is establishing the hyperparameter search space for the LSTM model, which includes variables like learning rate, batch size, number of LSTM layers, and number of neurons. An "ant" represents every potential solution in this area, while "antlions"

symbolize the ideal or nearly ideal solutions. ALO creates an initial population of ants, each representing a collection of hyperparameters. A roulette wheel selection procedure guides ants towards antlions at the start of each iteration to replicate their exploration and exploitation of the solution space. The LSTM model assesses each ant's fitness based on its performance. Ants update their locations in the search space based on their interactions with antlions, and either keep or modify the top-performing solutions—antlions—as needed. This iterative procedure continues until convergence to ensure that the chosen hyperparameters effectively maximize LSTM performance. The following steps explain how to achieve this:

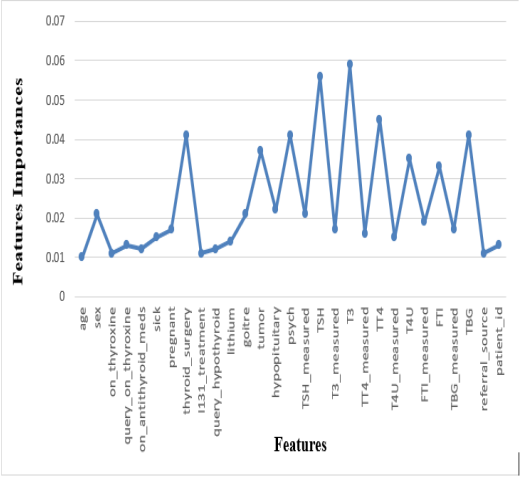

Fig. 6. Entropy value of the thyroid dataset features.

1) Initialization: Set random biases and weights at the beginning of the LSTM model. Determine how many antlions (M) and ants (N) are present. In the search area, place the ants at random starting points [28]. Set the antlions' starting positions at random throughout the search area. Decide on the maximum number of iterations (MaxIter).

2) Fitness evaluation: Use the objective function, which is usually the LSTM model's loss function, to determine each ant's and antlion's level of fitness [29].

3) Movement of ants: Every ant travels at random around the closest antlion inside a hypersphere. Ant movement is determined by the following formulas:

$$c_i(t) = Antlion_j(t) + l_c(t) \quad (9)$$

$$q_i(t) = Antlion_j(t) + h_q(t) \quad (10)$$

Where:

$c_i(t)$ and $q_i(t)$: represent the ith ant's lowest and maximum values at time t $Antlion_j(t)$: represents the location of the jth antlion at t. $l_c(t)$: represents the variable with the lowest value at time t among all variables is represented by $h_q(t)$: represents the variable that has the highest value at time t among all variables.

4) Movement of antlions: Within its capture radius, each antlion moves towards the most fit ant. The following equation determines how antlions move:

$$Antlion_j(t+1) = Antlion_j(t) + rand(). (Antlion_j(t) - X_f) \quad (11)$$

where, $rand()$ is a random value between 0 and 1, and $-X_f$ is the location of the fittest ant inside the capture radius of the jth antlion

5) Building trap: Utilizing a roulette wheel selection process, the fittest antlions are chosen according to their fitness scores [30]. To update the LSTM model's weights and biases inside their capture radius, the chosen antlions are employed.

6) Update: The movement equations are used to update the positions of every ant and antlion [31]. The revised antlion locations are used to update the weights and biases of the LSTM model. The modified LSTM model is used to reassess the fitness of each ant and antlion.

7) Termination: The algorithm ends when it reaches the maximum number of iterations or satisfies a stopping condition. where N represents the number of ants, M stands for the number of antlions, MaxIter signifies the maximum number of iterations, and lc(t) represents the lowest value of all the variables at time t [32].

Algorithm 1, written in Python, describes a classifying model for this study. The configuration of the LSTM model includes 50 units and a 972 softmax activation function to facilitate multi-class classification. The code utilizes the ALO algorithm to optimize the LSTM model. Through the model's training step, the ALO algorithm uses specific hyperparameters, such as 150 iterations and 15 ant lions.

Algorithm 1: Pseudocode of LSTM

```
# Define the LSTM-based model
model = Sequential ()
model.add (LSTM (units=50, input_shape= (time_steps,
num_features)))
model.add (Dense (units=num_classes,
activation='softmax'))
# Define the Ant Lion Optimizer
optimizer=AntLionOptimizer (num_iter=150,
num_antlion=15,
num_dimensions=model. count_params ())
# Compile the model
model. compile (optimizer=optimizer,
loss='categorical_crossentropy', metrics=['accuracy'])
# Train the model
model.fit (x_train, y_train, epochs=100, batch_size=32)
# Make predictions
predictions = model. predict(x_test)
```

V. Result and Discussion

To perform our experiments, we used the following software and hardware requirements: an Intel Core i7-6500U CPU with 16 GB of RAM and a clock speed of 4 GHz.

In the testing stage, we assessed how successfully the ALO_ LSTM-based model classified various kinds of thyroid illness.

After training the model on 80% of the dataset, we tested 30% to evaluate its performance. A confusion matrix [33], which included metrics like true positive, true negative, false positive, and false negative, was used to determine the results. Metrics including accuracy, recall, precision, and F1-Score were employed to assess the performance of the classification model. The following four parameters can be used to calculate the metrics and measurements that are employed.

1) True positive (TP): denotes correctly predicted positive values; in other words, it indicates that both the predicted and actual class values are true.

2) True negative (TN): these are the accurately predicted negative values, which indicate that neither the expected nor the actual class value is false.

3) False positive (FP): in situations where the predicted class is true but the actual class is false.

4) False negative (PN): This occurs when the predicted class is false, but the actual class is true.

The accuracy, recall, precision, and F1 score may be determined based on the previously mentioned parameters. These can be expressed as stated using the following equations:

$$Accuracy = \frac{TP+TN}{TP+FP+TP+TN} \quad (12)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (13)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (14)$$

$$F1 - score = \frac{TP}{TP+FN} \quad (15)$$

A. First Experiment

This study conducted two experiments to examine how the ALO metaheuristic algorithm may enhance the LSTM-based model's performance. In the first experiment, we employed the LSTM algorithms directly, regardless of the AlO, and used 'Adam' as an optimizer parameter. In contrast, in the second experiment, we used the ALO optimizer to improve the accuracy of the classification model. Table II shows the parameters used for LSTM in the first experiment. Where he Fig. 7 present the confusion matrix values used in determined the model performance.

TABLE II. THE LSTM PARAMETER FOR FIRST EXPERIMENTAL OUTCOMES

Parameters	Values
Model	Sequential
Hidden layer	4
Hidden unites	50
Neurons per hidden layer	30,30,30,1
Type of layer	LSTM
Activation hidden layer	Softmax
Epoch	100
Optimizer	ALO
Time step	1

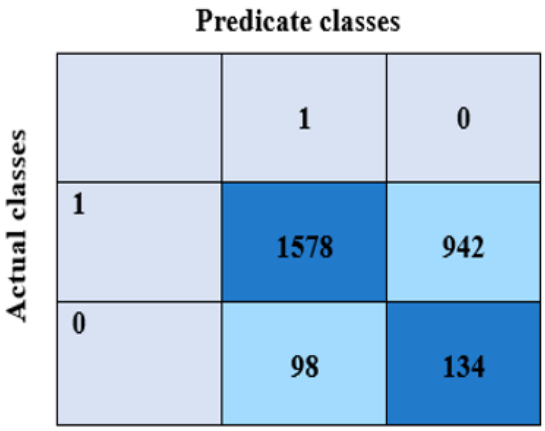

Fig. 7. Confusion matric of LSTM with Adam optimizer.

B. Second Experiment

The second experiment uses the ALO method as an optimizer for LSTM model parameters. Its purpose is to identify the optimum combination of weights and biases for the LSTM model, reducing the loss function while simultaneously speeding convergence through effective search space travel. The ALO algorithm employs specific hyperparameters, such as 150 iterations and 15 ant lions, during the model's training phase. The LSTM parameters for the second experiment are shown in Table III, and Fig. 8 details the confusion matrix from the second experiment. Additionally, Table IV displays the outcomes of our studies.

TABLE III. THE LSTM PARAMETER FOR SECOND EXPERIMENTAL OUTCOMES

Parameters	Values
Model	Sequential
Hidden layer	4
Hidden unites	50
Neurons per hidden layer	30,30,30,1
Type of layer	LSTM
Activation hidden layer	Softmax
Epoch	100
Optimizer	ALO
Time step	1

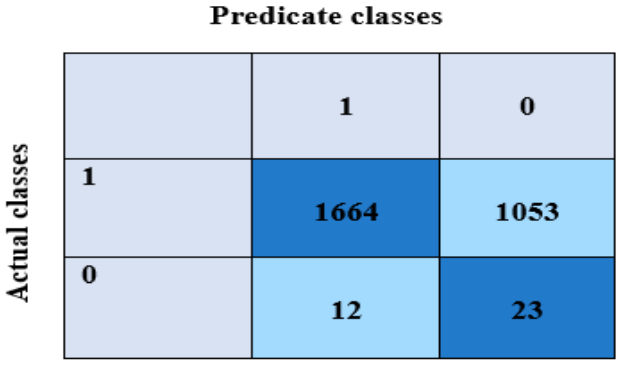

Fig. 8. Confusion matric of LSTM with ALO optimizer.

TABLE IV. COMPARATIVE CLASSIFICATION PERFORMANCES OF LSTM BASED MODEL

Models	Accuracy	Precision	Recall	F1 - score
LSTM based model	91.4%	94.1%	92.1%	93.1%
LSTM with ALO algorithm	98.6%	99.2%	89.6%	98.6%

Based on the findings, we conclude that the model performs well in every experiment, with high precision values ranging from 94.1% to 99.2%. This means that a high rate of the predicted positive instances is correct. The values of recall also show promising performance, ranging from 99.2% to 89.6%, meaning that a substantial proportion of the actual positive samples are predicted correctly. We establish the F1-scores at 98.9%, indicating a good overall performance and maintaining a balance between recall and accuracy. In the second experiment, the ALO algorithm produced an average accuracy, recall, and F1-score of 98.6%, indicating consistently strong classification performance across all classes. This indicates that the classification model is successful in correctly classifying the various thyroid disease classes.

From Table III, the recommended strategy, which selected the optimal hyperparameter for the LSTM using the AlO algorithm, improved the classifier overall by 7.2% compared to the achieved accuracy. With excellent accuracy and consistency across all criteria, the classification performance results show that the model performs robustly and consistently when it comes to differentiating between the different thyroid classes.

As mentioned in references [21], and [22], pre-processing and transforming the study's database into a regular form enhanced the performance of the proposed model. The use of the feature selection algorithm also helped to reduce the number of features used in training the model by focusing on the features with high impact and importance in the prediction process.

Additionally, after comparing the outcomes of the proposed strategy with those of alternative approaches found in the literature. It validates that, even with the same database but a different technique, the efficiency attained by the suggested system for prediction when using the ALO algorithm to enhance the performance of the LSTM algorithm surpasses other research with greater classification accuracy. It was also observed that although some previous research not employed any approaches in feature selection and these factors might have a big impact on how well the classification methods work.

VI. CONCLUSION

The primary goal of this research is to create a new model that predicts thyroid illness by combining the ALO metaheuristic algorithm with the LSTM algorithm. The proposed methodology is based on three steps: preprocessing, feature selection, and classification. By using the commonly used data on thyroid disorders, we trained and evaluated the proposed technique. We employed several metrics, such as accuracy, precision, recall, and F1 score, to assess the model's performance. The results of the experiments showed that 1) using entropy in the feature selection step can cut the total number of features from 31 to 10; and 2) using the ALO algorithm in the suggested strategy made the LSTM classifier 7.2% better overall and gave the best accuracy of 98.6%. The model's high level of accuracy and usefulness illustrate its practical relevance and significance in enhancing patient care. Future studies in this area should concentrate on investigating more intricate model architectures that are capable of capturing even more minute patterns in the thyroid data. Additionally, the model's performance might be improved, and more thorough insights could be obtained by employing different feature selection approaches other than the entropy technique used in this work. We can bridge these gaps and strive for innovative advancements. We can continue to refine the classification models for thyroid illnesses, enabling more accurate and reliable identification of thyroid diseases. This could revolutionize healthcare delivery and improve outcomes for individuals suffering from thyroid conditions.

ACKNOWLEDGMENT

We gratefully acknowledge University of Petra for their invaluable support and resources throughout this research. The realization of our project was significantly aided by our research group and all its members. Their unwavering support and the conducive academic environment provided invaluable assistance throughout this endeavour. Our heartfelt thanks to our colleagues and students for their insightful contributions and steadfast support.

REFERENCES

[1] V. Leso, I. Vetrani, L. De Cicco, A. Cardelia, L. Fontana, G. Buonocore, & I. Iavicoli, " The impact of thyroid diseases on the working life of patients: a systematic review," *International Journal of Environmental Research and Public Health*, vol. 17, no.12, pp. 4295, 2020. https://doi.org/10.3390/ijerph17124295

[2] L. Hegedüs, A. C. Bianco, J. Jonklaas, S. H. Pearce, A. P. Weetman, & P. Perros, " Primary hypothyroidism and quality of life," *Nature Reviews Endocrinology*, vol. 18, no. 4, pp. 230-242, 2022. https://doi.org/10.1038%2Fs41574-021-00625-8

[3] D. Dahiwade, G. Patle, & E. Meshram, "Designing disease prediction model using machine learning approach," *In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1211-1215, 2019. https://doi.org/10.1109/ICCMC.2019.8819782

[4] E. Sonuç, " Thyroid disease classification using machine learning algorithms," *In Journal of Physics: Conference Series*, vol. 1963, no. 1, p. 012140, 2021. https://doi.org/ 10.1088/1742-6596/1963/1/012140.

[5] H. Abbad Ur Rehman, C. Y. Lin, & Z. Mushtaq, "Effective K-nearest neighbor algorithms performance analysis of thyroid disease," *Journal of the Chinese Institute of Engineers*, vol. 44, no.1, pp. 77-87, 2021. https://doi.org/10.1080/02533839.2020.1831967.

[6] F. Temurtas, "A comparative study on thyroid disease diagnosis using neural networks." *Expert Systems with Applications*," vol. 36, no. 1, pp. 944-949, 2009. https://doi.org/10.1016/j.eswa.2007.10.010

[7] W. Ahmad, A. Ahmad, C. Lu, B. A. Khoso, & L. Huang, "A novel hybrid decision support system for thyroid disease forecasting." *Soft Computing*, vol. *22*, pp. 5377-5383, 2018. https://doi.org/10.1007/s00500-018-3045-9

[8] S. Mirjalili, " The ant lion optimizer, "*Advances in engineering software*, vol. *83*, pp. 80-98, 2015. https://doi.org/10.1016/j.advengsoft.2015.01.010

[9] L. Abualigah, M. Shehab, M. Alshinwan, S. Mirjalili, & M. A. Elaziz, "Ant lion optimizer: a comprehensive survey of its variants and applications," *Archives of Computational Methods in Engineering*, vol. *28*, pp. 1397-1416, 2021. https://doi.org/10.1007/s11831-020-09420-6

[10] E. Emary, H. M. Zawbaa, & A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54-65, 2016. https://doi.org/10.1016/j.neucom.2015.06.083

[11] A. Graves, & A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37-45, 2012. https://doi.org/10.1007/978-3-642-24797-2_4

[12] S. Hochreiter, & J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735

[13] G. Van Houdt, C. Mosquera, & G. Nápoles, " A review on the long short-term memory model, " *Artificial Intelligence Review*, vol. *53 no.*8, pp. 5929-5955, 2020. https://doi.org/10.1007/s10462-020-09838-1

[14] Y. Eren, & İ. Küçükdemiral, "A comprehensive review on deep learning approaches for short-term load forecasting," *Renewable and Sustainable Energy Reviews*, vol. *189*, pp. 114031, 2024. https://doi.org/10.1016/j.rser.2023.114031

[15] S. M. Al-Selwi, M. F. Hassan, S. J. Abdulkadir, A. Muneer, E.H. Sumiea, A. Alqushaibi, & M. G. Ragab, "RNN-LSTM: From applications to modeling techniques and beyond—Systematic review," *Journal of King Saud University-Computer and Information Sciences*, PP. 102068, 2024. https://doi.org/10.1016/j.jksuci.2024.102068

[16] C. Avci, B. Tekinerdogan, & C. Catal, " Analyzing the performance of long short-term memory architectures for malware detection models," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 6, pp. 1-1, 2023. https://doi.org/10.1002/cpe.7581

[17] https://www.kaggle.com/datasets/emmanuelfwerr/thyroid-disease-data.

[18] P. Dhawas, m. a. Ramteke, A. Thakur, P. V. Polshetwar, R. V. Salunkhe, & D. Bhagat, "Big Data Analysis Techniques: Data Preprocessing Techniques, Data Mining Techniques, Machine Learning Algorithm, Visualization," *In Big Data Analytics Techniques for Market Intelligence*, pp. 183-208, 2024. https://doi.org/10.4018/979-8-3693-0413-6.ch006

[19] P. Rodríguez, M. A. Bautista, J. Gonzalez, S. & Escalera, " Beyond one-hot encoding: Lower dimensional target embeddin, " *Image and Vision Computing*, vol. 75, pp. 21-31, 2018. https://doi.org/10.48550/arXiv.1806.10805

[20] A. E. Karrar, "The effect of using data pre-processing by imputations in handling missing values," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 10, no. 2, pp. 375-384, 2022. http://dx.doi.org/10.52549/ijeei.v10i2.3730

[21] S. G. O. P. A. L Patro., & K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*. https://doi.org/10.48550/arXiv.1503.06462

[22] L. Al Shalabi, Z. Shaaban, & B. Kasasbeh, " Data mining: A preprocessing engine, " *Journal of Computer Science*, vol. *2, no.* 9, pp. 735-739, 2006. https://doi.org/10.3844/JCSSP.2006.735.739

[23] X. Song, Y. Zhang, W. Zhang, C. He, Y. Hu, J. Wang, & D. Gong, " Evolutionary computation for feature selection in classification: A comprehensive survey of solutions, applications and challenges," *Swarm and Evolutionary Computation*, vol. 90, pp. 101661, 2024. https://doi.org/10.1016/j.swevo.2024.101661

[24] D. Theng, & K. K. Bhoyar, "Feature selection techniques for machine learning: a survey of more than two decades of research, " *Knowledge and Information Systems*, vol. 66, no. 3, pp. 1575-1637, 2024. https://doi.org/10.1007/s10115-023-02010-5

[25] Priyanka, & D. Kumar, "Decision tree classifier: a detailed survey," *International Journal of Information and Decision Sciences*, vol. 12, no. 3,pp. 246-269, 2020. https://doi.org/10.1504/ijids.2020.10029122

[26] Y. Zhu, D. Tian & F. Yan, F, "Effectiveness of entropy weight method in decision-making, "*Mathematical Problems in Engineering*, vol. 1, pp. 3564835, 2020. https://doi.org/10.1155/2020/3564835

[27] H. Li, F. He, Y. Chen, & Y. Pan," MLFS-CCDE: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution," *Memetic Computing*, vol. 13, pp. 1-18, 2021. https://doi.org/10.1007/s12293-021-00328-7

[28] V. K. Pathak, S. Gangwar, R. Singh, A. K. Srivastava, & M. Dikshit, M. " A comprehensive survey on the ant lion optimiser, variants and applications," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 36, no.4, pp. 511-562, 2024. https://doi.org/10.1080/0952813X.2022.2093409

[29] A. C. Johnvictor, V. Durgamahanthi, R. M. Pariti Venkata, & N. Jethi, " Critical review of bio-inspired optimization techniques. Wiley Interdisciplinary Reviews," *Computational Statistics*, vol. 14, no. 1, pp. e1528, 2022. https://doi.org/10.1002/wics.1528

[30] S. Mirjalili, S. " The ant lion optimizer," *Advances in engineering software*, vol. 83, pp. 80-98, 2015. https://doi.org/10.1007/978-3-030-12127-3_3

[31] S. Kumar, & A. Kumar, "A brief review on antlion optimization algorithm," *In 2018 international conference on advances in computing, communication control and networking (ICACCCN)*, pp. 236-240, 2018. https://doi.org/10.1109/ICACCCN.2018.8748862

[32] A. A. Heidari, H. Faris, S. Mirjalili, I. Aljarah, & M. Mafarja, "Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks," *Nature-inspired optimizers: theories, literature reviews and applications*, PP. 23-46, 2020. https://doi.org/10.1007/978-3-030-12127-3_3

[33] M. Heydarian, T. E. Doyle, & R. Samavi, " MLCM: Multi-label confusion matrix," *IEEE Access*, vol. 10, pp. 19083-19095, 2022. http://dx.doi.org/10.1109/ACCESS.2022.3151048