# MODELLING AN EFFICIENT REGULARIZED OPTIMIZATION APPROACH FOR HEALTH CARE APPLICATION

**K.GUNASEKARAN[1,*], V.D.AMBETH KUMAR[2], P. SHERUBHA[3], S.P. SASIREKHA[4]**

[a]Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India.
[b]Department of Computer Engineering, Mizoram University, Aizawl, India.
[c]Department of Information Technology, Karpagam College of Engineering, Coimbatore, India
[d]Department of Computer Science and Engineering, Karpagam Academy of Higher Education, Coimbatore, India.
Corresponding mail id: k.gunasekaran411@gmail.com, sherubha0106@gmail.com, sugi.sasi29@yahoo.com

## ABSTRACT

Chronic conditions like diabetes, heart disease, cancer, and chronic respiratory disorders pose a threat to people everywhere. Among these, the diagnosis of heart disease is made more difficult by its variable symptoms or traits. Internets of Things (IoT) solutions are crucial for healthcare detection. The suggested approach combines fog, edge and cloud computing to deliver quick and accurate results. The hardware elements gather information from various patients. To obtain important features, signals are subjected to cardiac feature extraction. Additionally, data on the feature extraction of other properties are acquired. An optimized cascaded convolution neural network collects all these features and subjects them to the detection system. Squirrel Optimizer is adopted over the auto-encoder (SOAE) technique to optimize the AE hyper-parameters. The suggested SOAE is 4%, 4%, 4%, 8%, 68%, 49%, 34%, 11% and 8% more accurate than PSO, GWO, WOA, DHO, DNN, RNN, LSTM, CNN, and RCNN, respectively, according to performance studies. The comparison analysis shows that the proposed system performs better than conventional models.

**Keywords:** *Regularized Model, Encoding, Optimization, Health Care, Prediction*

## 1. INTRODUCTION

The contemporary economy depends on Internet services to give clients access to resources for on-demand service. It is constructed using computing concepts which gained much attention [1]. These fields are now crucial components of both academia and business. However, the substantial time delay of cloud computing makes it unsuitable for real-time applications requiring replies [2]. Recent technologies have seen substantial growth due to their capacity to provide a variety of response characteristics based on target applications including fog, edge, IoT, and big data [3]. Because these technologies may provide processing, improving and supporting network capacity, security, privacy, and mobility with minimal delay constraints through storage and communication to edge devices, Applications that require real-time or high latency performance benefit from fog computing [4]. Recently, architecture for cloud computing that provides reliable and strong infrastructure and services has supported innovative applications [5]. Using the least amount of power, the latency on the network, and response time, fog computing also uses gateways, nodes, and routers. Recent research studies fog computing's difficulties in medical applications, the hardest and most important quality of service constraints to improve in real-world fog settings are reaction time and latency [6]. Healthcare is the most crucial field for accuracy, "fog computing," and real-time developments. By bringing resources closer to consumers to achieve the lowest latency, fog computing can boost security in the healthcare industry [7]. Achieving earlier results makes it possible to take the necessary and speedier actions to treat critically ill heart patients. Although it delivers results faster, it still needs to work on complicated data and produce extremely precise outcomes [8]. Deep learning and the many deep learning techniques can achieve high accuracy variations employed in recent studies and trained on enormous datasets. Using diverse devices, such as IoT sensors and file input data, modern methodologies have recognized that two methods are used to collect healthcare data, particularly for cardiac patients [9]. Data about healthcare patients are reportedly downloaded from higher speeds—up to 250 MB per minute—over the network [10]. Traditional methods are insufficient for capturing and delivering video and data results, so edge and cloud resources must be used to support programs that use large amounts of data. After being gathered and aggregated from "smart devices of IoT networks," edge servers or cloud nodes, data is processed and stored [11].To provide heart patients and other customers with quality computer services who seek accurate results for healthcare and other purposes with quick responses, minimal energy use, and high accuracy, a comprehensive "Edge-Fog-Cloud-derived computation model," is advised.

Several applications, such as speech recognition, computer vision, etc., have witnessed considerable growth in deep learning. This freshly established discipline has demonstrated noteworthy results by processing natural language, predicting sequences, and issues requiring mixed-modality data sets [12]. Additionally, a lot of machine learning techniques benefit from the usage of ensemble learning to improve performance. The estimator for the "bag classifier," one of the efficient ensemble techniques, is trained by fitting several random data subgroups. The final projected outcomes are then obtained by voting or averaging the individual identifications of these groupings [13]. These estimators work better than a single estimator at reducing variance by randomizing the data. High accuracy predicting and classification rates for healthcare data have been reached by advanced deep learning systems [14]. On the other hand, deep learning is widely applied in healthcare applications, which necessitates a sizable computer resource commitment for training and recognition and additional time for data analysis and constructing such intricate neural networks. The previous methodologies may need help in IoT applications connected to healthcare because it can be challenging to attain the real-time application accuracy rate. The traditional healthcare system relies on experts to diagnose and predict medical issues. However, these experts can sometimes miss new data patterns and important details. Machine and deep learning can help by automatically identifying these patterns and details. With the increase in digital healthcare data, research using these techniques has grown. These methods are very effective with large amounts of data and excel at recognizing patterns and representing features. Integration of IoT-based computing and deep learning are used to make high-precision discoveries in practical applications. Edge computing offers a fresh research method due to its huge benefit of reducing response time. Using IoT and fog computing technologies, automatic cardiac disease detection models, and enhanced deep learning applications, are required for critically important healthcare applications [15]. The proposed model is focused on doing so to quickly implement this method for supporting patients with heart disease, which also streamlines processes for making decisions. The fundamental contribution of the following highlights the created smart healthcare model:

(i) A novel meta-heuristic SO-based deep learning model is executed in IoT environment for clinical heart disease identification. To obtain important data from common devices to process data and learn more about related-diseases and medical history

(ii) To suggest an autonomous diagnostic system for cardiac disorders that uses an optimized AE and optimizes a few parameters using the SO algorithm and to verify the heart disease detection efficiency using various performance metrics and compares the outcomes with other approaches.

The following is a list of the remaining parts. Section 2 examines related works. The adoption of IoT over healthcare applications and deep learning to predict heart illness is examined in Section 3. Section 4 examines the numerical outcomes of the SOAE model. Section 5 provides a conclusion to this article.

## 2. RELATED WORKS

Recently, various methods for anticipating heart disease have been presented. Using several ensemble classifiers, 86% of heart disease risk predictions are accurate [16]. Combining a rough sets-based attribute reduction technique using a type-2 interval fuzzy logic system and the chaotic firefly algorithm results in an 86% accurate model for the detection of heart disease [17]. The performance accuracy of a hybrid machine learning model that uses both the linear method (LM) and random forest (RM) methodologies to predict heart disease [18] is 88.7%. A comprehensive decision-support tool for forecasting the likelihood of heart failure achieves 91.0% accuracy by using an ANN for data classification and fuzzy analytic hierarchy technique for feature weighting [19]. It is suggested to use DNN and a statistical model for feature learning and classification tasks in a clever system for heart disease diagnosis. The model achieves 92% accuracy, 94% specificity, and 90% sensitivity, respectively [20].

## 3. METHODOLOGY

The "smart heart disease prediction system" gathers patient information from IoT or smart devices. Hardware components also refer to items like activity, medical, and environmental sensors attached to patients. The body transmits information like activity level, heart rate, blood pressure, EEG, oxygenation, EMG, and ECG. To anticipate heart disease, gateway devices analyze the data collected and send it to worker or broker nodes. The signals are isolated, and independent calculations are made of them. Some metrics are harmonic distortion, peak amplitude, zero-crossing rate, heart rate, entropy, energy and SD. Skewness, kurtosis, and standard deviation, as well as the smallest and greatest means, are also calculated to identify the characteristics of additional attributes. The cloud data centre, worker nodes, and broker nodes are all components of the proposed approach for predicting smart heart disease, which mainly relies on the bus. After obtaining the extracted features, the diagnosis system uses an upgraded encoding model to determine whether the patient has a heart illness. SOAE technique is used to optimize the hidden neurons, cascaded network layers, and activation functions. Reducing prediction loss based on the mean square error is the major goal of the proposed diagnosing heart disease (MSE) strategy. It receives normal and pathological output classes to assure system safety and vigilance.

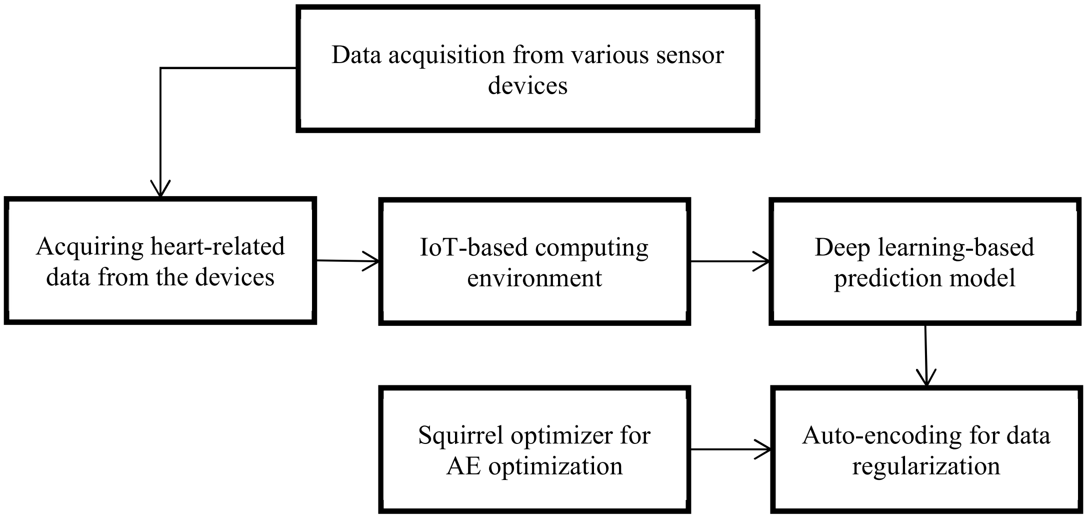
Data acquisition from various sensor devices
Acquiring heart-related data from the devices
IoT-based computing environment
Deep learning-based prediction model
Squirrel optimizer for AE optimization
Auto-encoding for data regularization

Fig 1 Block diagram

### 3.1. Data acquisition

On a set of data that was manually gathered, the proposed model was assessed. The following data is provided by sensors, such as glucose, temperature, oxygen, EMG, EEG, and ECG sensors, for patients who utilize medical sensors. 1) Sensor for respiratory rate, 2) Temperature gauge, 3) Pulse oximeter, 4) EMG, 5) EEG signals and 6) ECG sensors.

### 3.2. Encoding healthcare data

We review our recommended method to train and validate our clinical risk prediction model. An auto-encoder (AE) model is widely used to learn unsupervised dataset properties since it is a symmetrical neural network. AE is trained using an unsupervised, greedy layer-by-layer methodology to generate $K$ hidden layer deep learning architecture. Auto-encoder is trained using the first hidden representation layer $h$ is produced using the distorted, noisy, and beginning input data $x$. Higher-level symbols, like $h_1^e$, are created utilizing the data from the identified hidden layer $h$ by training the following auto-encoder. Auto-reconstruction encoder's layer is moved to the network design's final second layer, h. The reconstruction procedure is followed backwards, concluding with the higher-level representation $h_2^e, \dots, h_k^e$. By including a softmax regression layer, the rebuilt feature layer $x'$ that results from clinical risk prediction can be carried out using greedy layer-wise pre-training. The final result is a clinical risk prediction problem-specific deep neural network (DNN). The length and quantity of output nodes for the patient sample vector are fixed to be the same. The nonlinear activation function can encode the vector $h \in R^{|h|}$ given a patient vector $x$ and noisy input vector $\tilde{x}$ (where $|h|$ denotes the size).

$$h = f(W^e \tilde{x} \ + b^e) \tag{1}$$

The parameters of a single AE are $W$ and $b$, and the sigmoid function is $f(x) \ = \left(\frac{1}{1+expexp\,(-x)}\right)$. $W$ stands for the encoder weight matrix. The bias vector is $b$. The input vector $x$ must be recreated while the hidden feature vector $h$ is present is modified as follows during the decoding stage:

$$x' \ = \ g(W^d h + b^d) \tag{2}$$

The decoder parameters are the decoder function $g(.)$, the decoder weight matrix ($W^d R^{M*|h|}$), and the bias of the decoder $b^d \in R^M$. To reduce the parameters used, it is recommended that $W^d = W^{e^T}$, which connects the learnt weights for the coding layer and decoding layer. A hidden vector, $h$, is created for each patient's sample, $x$ and then restored to its original state, $x$. The reconstruction error is minimized to improve the parameters:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(x_i, x_i') = \frac{1}{N}\sum_{i=1}^{N} L\{x_i, g\theta^d \ \left[f_{\theta^e(x_i')}\right]\} \tag{3}$$

Where $L(x_i, x_i')$ is loss function and the average inaccuracy in reconstruction across $N$ patient samples. Cross-entropy is used to calculate the loss function as follows:

$$L(x_i, x_i') = -\sum_{j=1}^{M} (x_{ij}\ log\ log\ x_{ij}' + \left(1 - x_{ij}\right) log\ log\ (1 - x_{ij}')) \tag{4}$$

The cost function is optimized after parameter vector is shrunk to extremely less values close to zero, and the quasi-Newton method, a second-order optimization strategy based on the updating mechanism, is employed. The anticipated model is trained using gradient descent to create an architecture with $K$ hidden layers, starting with lowest layer of the encoder. Hidden layer representation $h_1 = f_{\theta^o}(x)$, is created by optimizing the noise in patient sample input $(x)$. The subsequent AE can then be trained to create higher-level representations using the input data from this layer, and so on.

### 3.3. Risk analysis

AE replicated the features unsupervised even though it was successfully used as an extractor to address classification issues. The learning technique might include risk information from the training dataset to enhance feature representations' reconstructed clinical risk prediction capability. To successfully implement intra- and inter-risk-level repulsion restrictions into AE using clinical risk data from training patient samples. Be aware that patients with the same risk rating may be forced to comply with these dual restrictions to have their rebuilt feature representations patients with varying risk levels should have their representations as widely apart as is practical. We assume that for two patient samples, $x_m$ and $x_n$ belong to the same risk category, and linked encoding quality $P(h_{1,m}|x_m)\ and\ P(h_{1,m}|x_n)$ is comparable. Intra-risk-level limitation for entire encoding is suggested to lower cost function:

$$L_{intra} = \frac{1}{2}\sum_{k=1}^{K}\sum_{m,n,\mu_{m,n}}^{K} \left|\left|P\left(h_{k-1,m}\right) - P\left(h_{k-1,n}\right)\right|\right|^{2} \quad (5)$$

When two patient samples come from the same risk group, the symbol "$\mu$" denotes "1," otherwise it $\mu$ denotes "0." $h_{i-1,m}$ *and* $h_{l-1,n}$ equivalent counterparts are $x_m$ and $x_n$. It is expected that the relevant encoding properties with samples $x_m$ and $x_n$ are risk groups, $P(h|x_m)$ and $P(h|x_n)$, will change. Inter-risk-level restriction for complete encoding is suggested to lower cost function:

$$L_{intra} = \frac{1}{2}\sum_{k=1}^{K}\sum_{m,n,}^{K}\left(1 - \mu_{m,n}\right).\, exp- \left|\left|P\left(h_{k-1,m}\right) - P\left(h_{k-1,n}\right)\right|\right|^{2} \quad (6)$$

$x_m$ and $x_n$ are patient samples that came from the same risk group, in that case, then "$\mu$" = "1," otherwise it $\mu$ equals 0. $h_{k-1,m}$ and $h_{k-1,n}$ have respective counterparts of $x_m$ and $x_n$. We create the following learning objective function that considers both $L_{intra}$ and $L_{inter}$ constraints:

$$J'(\theta) = \frac{1}{N}\sum_{n=1}^{N}(x_n, x'_n) + \lambda_{intra}L_{intra} + \lambda_{inter}L_{inter} \quad (7)$$

Where $\lambda_{intra}$ and $\lambda_{inter}$ are two regularization variables that govern the significance of intra- and inter-level limitations on risk, it is possible to compute the gradients between two regularization terms and the learning parameters.

$$\frac{\partial L_{intra}}{\partial w} = \frac{\partial \sum_{k=1}^{K}\sum_{m,n}^{K}\mu_{m,n}\left|\left|P\left(h_{k-1,m}\right) - P\left(h_{k-1,n}\right)\right|\right|^{2}}{\partial w} \quad (8)$$

The proposed regularized encoding can ensure the peculiarities of patient risk information. Feature representations among various risk levels are kept as far away as possible. In contrast, the risk level is kept closer in practical. Using this regularised encoding, we pre-train the clinical risk prediction with EHRs. A softmax regression model with a layered auto-encoder with regularization is created when pre-training is complete by adding the rebuilt feature representation layer on top of a softmax regression layer. Then, we carry out clinical risk prediction tasks. With Eq. (9) for the softmax layer and lowering the cross-entropy loss, the model is particularly optimized by back-propagation.

$$L_{tune}\left(\theta_{encoding}\right) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{r=1}^{R} I(y_i = r)\log\left(\frac{\exp\left(h_{encoding\,(x_i)}\right)}{\sum_{i=1}^{R}\exp\left(h_{encoding}\,(x_i)\right)}\right) \quad (9)$$

The indication function $I(.)$ gives 1 if the given statement is true and 0 if it is false. The output for the input $x$ is $h(x)$, where $R$ represents the number of risk levels. The pre-training phase's encoding/decoding weights are then used for the $K$ hidden layers' weights to be initialized. However, the softmax layer scales begin with extremely low random values. The initial step in this method involves estimating the vector of $\theta$ parameters. After that, a min-batch gradient descent technique is used to minimize the cost:

$$L_{tune}\left(\theta_{encoding}\right) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{r=1}^{R} I(y_i = r)\log\left(\frac{\exp\left(h_{encoding\,(x_i)}\right)}{\sum_{i=1}^{R}\exp\left(h_{encoding}\,(x_i)\right)}\right) \quad (10)$$

In this part, we offer a method for selecting instructive risk factors for patients while considering various concerns. Patient features are more re-constructible if there are fewer reconstruction errors. Patient attributes with higher re-constructibility more accurately reflect the fundamental qualities. In this regard, we forecast that a patient sample inside a certain risk category of the trained regularized model will offer a negligible reconstruction error from the raw and rebuilt data. On the other hand, we argue that every patient sample, regardless of risk level, exhibits unique behaviour in response to its risk variables, which results in a significant reconstructive mistake. The discrepancy between the test model and the input patient sample during the discriminative learning phase will likely result in a sizable reconstruction error. Fig 2 illustrates our method for choosing prospective risk variables that are easier to reconstruct as discriminative features in this situation. Formally, the following Eq. (11) can be used to determine the corrected error of one feature, $i(i = 1, M)$, concerning certain risk level, $r \in R$ where $R$ stands for "low, medium and high risk level".

$$e_{ri} = \sqrt{\frac{\sum_{n=1}^{N_r}(x'_{ni} - x_{ni})^2}{N_r}}$$

Where $x_{nl}$ represents the $n^{th}$ patient's $i^{th}$ feature value, the reconstruction error among the patient feature $x'_{ni}$ and the original feature value $x$ is calculated for total of $N$ patient samples at $r^{th}$ risk level. Positive large $e_{ri}$ patients are more likely than negative big $e_{ri}$ patients to fall into $r^{th}$ risk category.

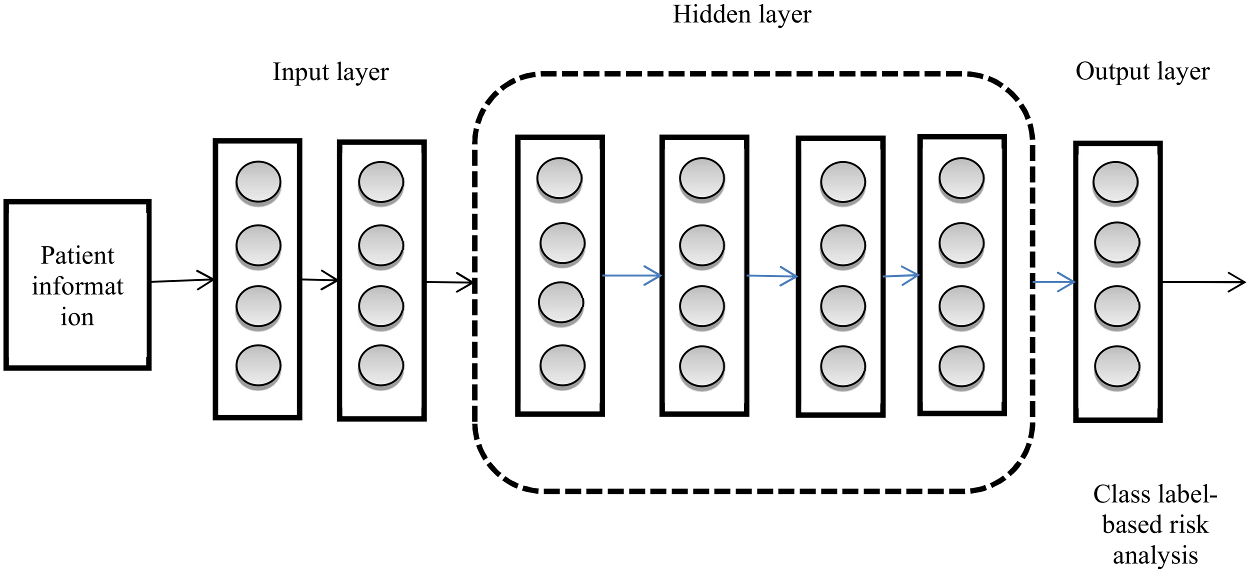

*Fig 2 Internal Architecture Of AE*

### 3.4. Squirrel Optimization (SO)

Here, SO is an effective method microscopic mammals use for long-distance travel and is a dynamic FLS foraging movement seen in Asian and European deciduous forests. The squirrels move around during the summer by gliding from one woodland tree to the next in search of food. To get their daily energy needs met, they need only look for acorn nuts. The search for winter-saved hickory nuts begins at that point. They depend on storing hickory nuts to give them the energy they need as they become less active in the winter. The warmer it gets, the busier the flying squirrels are. The procedure mentioned above is carried out repeatedly throughout the squirrels' lives and forms the basis of the SO.

Each squirrel uses a population-based approach to scour a multidimensional search space for food. The locations of the squirrels are viewed as separate design variables, and the separation between the particular squirrel and the food corresponds to the goal function's fitness value. Particular squirrels in the SO relocate to new, more suitable locations. In a deciduous forest, it is assumed that there are n squirrels and that there is only one squirrel per tree. It is believed that the woodland contains all three kinds of trees. The three types of trees are hickory, acorn, and regular. There should be $N$ trees in the forest; a hickory tree is one of them. The remaining trees are merely standard trees without any food, and there are no acorn trees. A hickory tree is a convenient place for squirrels to browse. SO imitates a squirrel's foraging behaviour. Population dispersal, varying dietary habits, seasonal adaptive intelligence, and ad hoc transfers of people after the winter are all factors that are the four strategies each squirrel employs to change its location. The seasonal monitoring condition of the Squirrel Search Algorithm has the benefit of enabling more efficient and improved search space exploration. Furthermore, SO helps achieve results that manage exploitation and exploration difficulties and eliminate over-fitting problems. But like other intelligent evolutionary algorithms, SO have several limitations, including a slow convergence speed and poor convergence accuracy. The hypothetical optimization based on FLS nutrition foraging behaviour can be described using the subsequent phases.

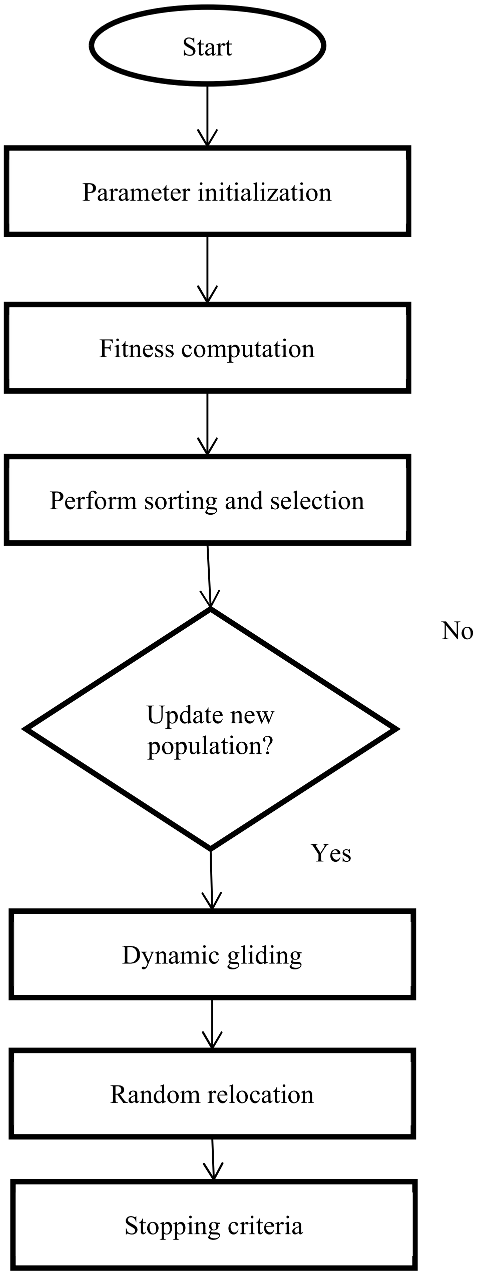

*Fig 3 SO Flow Diagram*

Step 1 (Parameter initialization): The number of decision variables overall ($N_d$), population size overall ($N_P$), population size overall ($I_m$), likelihood that a predator would be present ($P_{pp}$), scaling factor overall ($F_s$), gliding constant overall ($C_g$), and upper and lower boundaries for the choice variables are among the many factors that are taken into consideration ($U_d \ and \ L_d$). These options are chosen at the beginning of the SO procedure (See Fig 3).

Step 2 (Initializing Flying Squirrels randomly): SO starts with a randomly chosen position similar to earlier population-based algorithms, flying squirrels. Flying squirrels (FLS) are a particular species only found in woods. A consistent distribution establishes each FLS's initial location inside the forest. The FLS coordinates are initialized at random in the search process as follows:

$$FLS_{i,j} = L_d + rand\;(\,) * (U_d - L_d); \quad i = 1,2,\dots,N_p, \quad j = 1,2,\dots,N_d \tag{12}$$

Where the random number returned by $rand()$ has a uniform distribution and ranges from $[0,1]$.

Step 3 (Fitness computation): The choice variable's values are input into a fitness function provided by the user, and the relevant values are then returned and computed to determine each FLS. How food is supplied an FLS is seeking—whether an ideal, typical, or nonexistent one and, consequently, its chances of survival—are indicated by the FLS's position's fitness value. The position of a flying squirrel's fitness ($f = (f_1 f_2, \dots, f_{NP})$ value can be determined by using the following procedure to input the values of a fitness function's choice variables:

$$f_i = f_i\left(FLS_{i,1}, FLS_{i,2}, \dots, FLS_{i,n}\right); \quad i = 1,2,\dots,NP \tag{13}$$

Step 4 (Declaration, Selection, and Sorting): Each FLS position's fitness values are noted before ascending order is used to sort the array. Within the hickory nut tree sits an FLS with a low fitness level. It is anticipated that the three best FLS that followed them are currently on acorn but will soon transition to hickory nut trees. On common trees should be the remainder of FLS. On the assumption that they have satisfied their daily calorie requirements, it is thought that some squirrels relocate to the hickory nut tree at random. The squirrels that are still alive will head for the acorn nut trees. Predators always have an impact on the FLL's foraging habits. Indexes should be sorted in ascending order. The quality of food sources is then ranked in order of increasing fitness value based on FLS locations:

$$\left[sorted_f, sorted_{index}\right] = sort\;(f) \tag{14}$$

Step 5 (Aero-dynamic gliding): In each case, it is hypothesized that there is no predator around; the FLS flies around the forest, effectively looking for its target prey, but is compelled to take a hasty, arbitrary flight to find cover when a predator shows up. Gliding to a new site entails the following steps:

$$FLS_{at}^{new} = \{FLS_{at}^{old} + d_g C_g (FLS_{ht}^{old} - FLS_{at}^{old}\; if\; R_1 \geq P_{pp}\; random\; location\;\; otherwise \tag{15}$$

$R_1$ generates a number on the range $[0,1]$ from the uniform distribution, where the random gliding distance is denoted by $d_g$ and $C_g$ is the gliding constant. Using either random or velocity values, Aerodynamic gliding is utilized to estimate the new position. While moving them, restrict the new sites' lower and higher boundaries.

Step 6 (Stopping criteria): A common convergence criterion called the function tolerance criterion allows for a negligible but allowable discrepancy among finally the two outputs. There are times when the maximal time is employed as pause condition. The majority of iterations are employed in this experiment as a halting condition.

```
Algorithm 1: Squirrel Optimization

Input: Initialize population with random values based on upper and lower bound limit
Output: convergence solution

1. Select location randomly based on total flying squirrels ($n$);  //Eq. (12)
2. Evaluate the fitness function based on input features for $n$ samples with an error rate and $k$ neighbours;  //Eq. (13)
3. Sort the flying squirrel location in ascending order based on fitness value computation; //Eq.(14)
4. Generate new location via gliding;  //Eq. (15)
5. Repeat step 1 to 4 based on the maximal number of iterations.
```

## 4. NUMERICAL RESULTS

MATLAB 2020a was used to run the suggested model for diagnosing heart disease. Standard performance measures were used to compare the created system's efficacy to those of traditional models. The created model was examined using several optimization strategies comprising long short-term memory (LSTM), deep neural networks (DNN), CNN, RCNN, and recurrent neural networks (RNN). Optimizers like Whale Optimization Algorithm (WOA), Deer Hunting Optimization (DHO), Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO) are used. The experimentation is done on 4 GB RAM, Intel Core i3, 64-bit OS, windows 8. There are several estimated performance indicators like true positives (TP), true negative (TN), false positives (FP) and false negatives (FN).

1) F1-score is a harmonic average among precision and recall which is a statistical metric adopted to evaluate performance.

$$F1-score = \frac{2TP}{2TP + FP + FN} \qquad (16)$$

2) MCC: coefficient of correlation calculated with four values.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN}} \qquad (17)$$

3) NPV: The probability that individuals who receive a negative screening test don't have the disease.

$$NPV = \frac{TN}{FN + TN} \qquad (18)$$

4) FDR: all of the rejected hypotheses' total number of false positives.

$$FDR = \frac{FP}{FP + TP} \qquad (19)$$

5) FPR: ratio among total number of negative forecasts to the number of incorrect positive predictions.

$$FPR = \frac{FP}{FP + TN} \qquad (20)$$

6) FNR: percentage of positives that result in failed test results with the test.

$$FNR = \frac{FN}{TN + TP} \qquad (21)$$

7) Sensitivity: number of precisely recognized true positives.

$$Sen = \frac{TP}{TP + FN} \qquad (22)$$

8) Specificity: number of clearly determined true negatives.

$$Spe = \frac{TN}{TN + FP} \tag{23}$$

9) Precision: The proportion of precisely expected positive observations to all precisely predicted positive observations."

$$Pr = \frac{TP}{TP + FP} \tag{24}$$

10) Accuracy: It represents the proportion of properly predicted observations to all observations.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{25}$$

11) $K$ −fold validation: it is utilized to compute the skills based on the provided data which is also used to analyze performance.

### 4.1. Performance analysis

Current meta-heuristic-based algorithms are used to assess the effectiveness of the suggested system for predicting smart heart disease, as shown in Fig 4 to Fig 5. By changing the learning percentages from 35% to 85%, a newly built GSO is tested with a range of frequently used performance indicators to demonstrate how well it can diagnose heart disease. Due to its effectiveness, the suggested GSO is more accurate than current algorithms. Nevertheless, accuracy is preserved at the early stages of schooling, just as it is for other persons. However, GSO finds that as learning percentages increase, performance improves as measured by the performance metrics. GSO outperforms PSO, GWO, and DHO in terms of accuracy by 2%, 4%, and 8%, respectively, and WOA by 85%. The developed GSO exhibits larger error rates at starting percentages when measured using the FNR metric and lesser error is shown by 86%. SOAE performs better when compared to other performance metrics, utilizing IoT technologies, demonstrating a higher prediction rate.

*Table 1 Comparison based on the healthcare model using existing vs. proposed optimizer*

| Measures | PSO | GWO | WOA | DHO | GSO | SOAE |
|---|---|---|---|---|---|---|
| Accuracy | 94 | 93 | 93 | 94 | 94 | 95 |
| Sensitivity | 93 | 95 | 92 | 99 | 93 | 99.5 |
| Specificity | 94 | 94 | 94 | 90 | 97 | 98 |
| Precision | 94 | 94 | 94 | 91 | 98 | 99 |
| FPR | 0.05 | 0.05 | 0.05 | 0.09 | 0.02 | 0.01 |
| FNR | 0.06 | 0.07 | 0.07 | 0.007 | 0.065 | 0.05 |
| NPV | 0.94 | 0.94 | 0.94 | 0.90 | 0.97 | 0.98 |
| FDR | 0.05 | 0.05 | 0.05 | 0.088 | 0.019 | 0.015 |
| F1-score | 0.94 | 0.93 | 0.93 | 0.95 | 0.95 | 0.97 |
| MCC | 0.88 | 0.87 | 0.86 | 0.89 | 0.89 | 0.92 |

*Table 2 Comparison based CV on existing vs. proposed optimizer ($k = 10$)*

| Measures | PSO | GWO | WOA | DHO | GSO | SOAE |
|---|---|---|---|---|---|---|
| Accuracy | 95 | 94 | 94 | 94 | 97 | 98 |
| Sensitivity | 95 | 94 | 94 | 93 | 97 | 98 |
| Specificity | 95 | 94 | 94 | 94 | 96 | 97 |
| Precision | 95 | 94 | 94 | 94 | 97 | 98 |
| FPR | 0.04 | 0.05 | 0.05 | 0.05 | 0.03 | 0.02 |
| FNR | 0.04 | 0.06 | 0.05 | 0.06 | 0.023 | 0.020 |
| NPV | 0.95 | 0.94 | 0.94 | 0.94 | 0.96 | 0.97 |
| FDR | 0.04 | 0.05 | 0.05 | 0.05 | 0.023 | 0.010 |
| F1-score | 0.95 | 0.94 | 0.94 | 0.93 | 0.97 | 0.98 |
| MCC | 0.90 | 0.88 | 0.89 | 0.88 | 0.94 | 0.95 |

*Table 3 Comparison based on the healthcare model using existing vs. proposed method*

| Measures | DNN | RNN | LSTM | CNN | RCNN | SOAE |
|---|---|---|---|---|---|---|
| Accuracy | 67 | 75 | 80 | 90 | 91 | 94 |
| Sensitivity | 97 | 88 | 98 | 94 | 90 | 93 |
| Specificity | 42 | 68 | 61 | 86 | 92 | 97 |
| Precision | 58 | 66 | 73 | 88 | 91 | 98 |
| FPR | 0.57 | 0.34 | 0.38 | 0.13 | 0.07 | 0.02 |
| FNR | 0.02 | 0.11 | 0.01 | 0.053 | 0.09 | 0.065 |
| NPV | 0.42 | 0.64 | 0.61 | 0.86 | 0.92 | 0.97 |
| FDR | 0.41 | 0.33 | 0.26 | 0.11 | 0.08 | 0.019 |
| F1-score | 0.73 | 0.75 | 0.84 | 0.91 | 0.90 | 0.95 |
| MCC | 0.47 | 0.54 | 0.65 | 0.81 | 0.83 | 0.89 |

*Table 4 Comparison based CV on existing vs. proposed method ($k = 10$)*

| Measures | DNN | RNN | LSTM | CNN | RCNN | SOAE |
|---|---|---|---|---|---|---|
| Accuracy | 83 | 86 | 89 | 89 | 92 | 97 |
| Sensitivity | 91 | 99 | 89 | 87 | 93 | 97 |
| Specificity | 77 | 76 | 89 | 91 | 91 | 96 |
| Precision | 76 | 76 | 90 | 91 | 89 | 97 |
| FPR | 0.22 | 0.23 | 0.10 | 0.08 | 0.08 | 0.03 |
| FNR | 0.08 | 0.001 | 0.10 | 0.12 | 0.06 | 0.023 |
| NPV | 0.77 | 0.76 | 0.89 | 0.91 | 0.91 | 0.96 |
| FDR | 0.23 | 0.23 | 0.09 | 0.088 | 0.10 | 0.02 |
| F1-score | 0.83 | 0.86 | 0.90 | 0.89 | 0.91 | 0.97 |
| MCC | 0.68 | 0.76 | 0.79 | 0.79 | 0.84 | 0.94 |

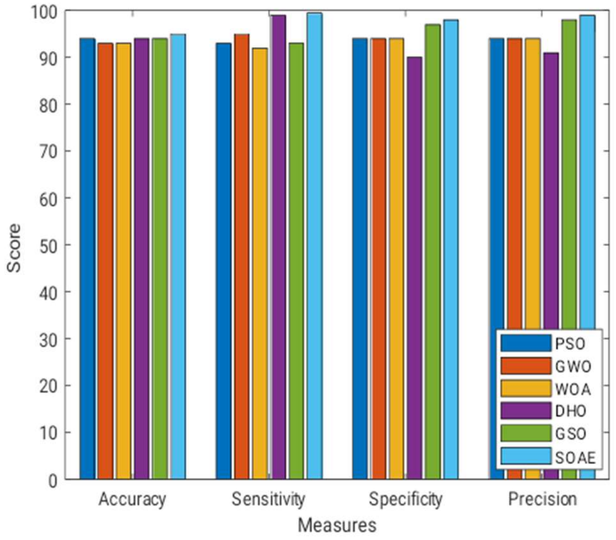

*Fig 4a Comparison based on the healthcare model using existing vs. proposed optimizer*

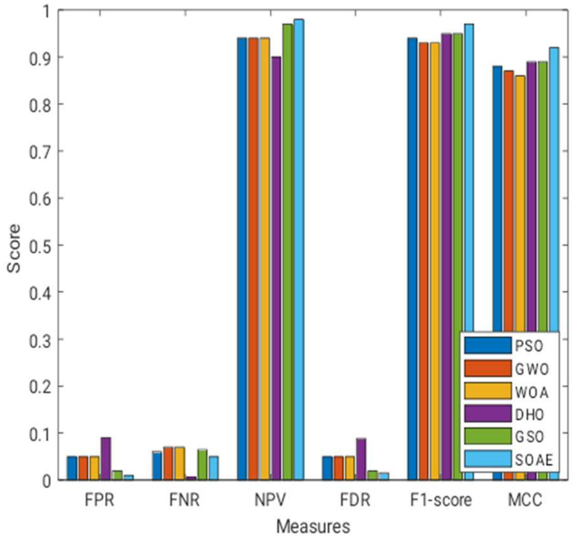

*Fig 4b Comparison based on the healthcare model using existing vs. proposed optimizer*

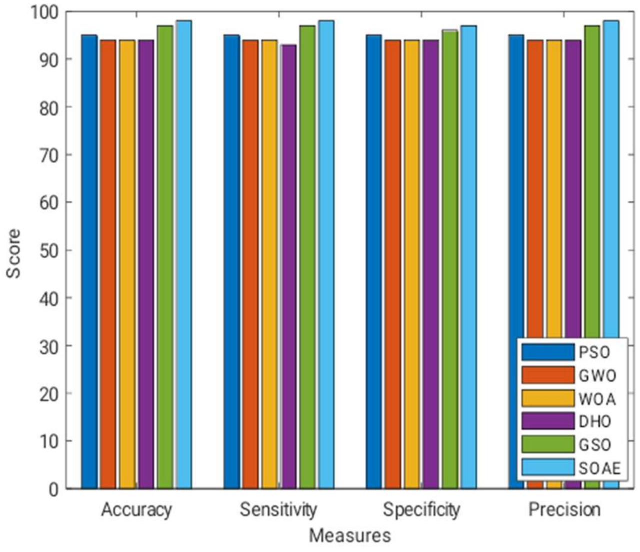

*Fig 4c Comparison based CV on existing vs. proposed optimizer* $(k = 10)$

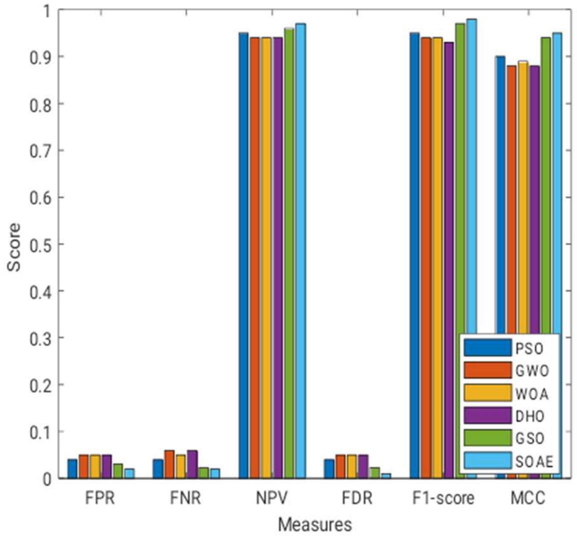

*Fig 4d Comparison based CV on existing vs. proposed optimizer* $(k = 10)$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

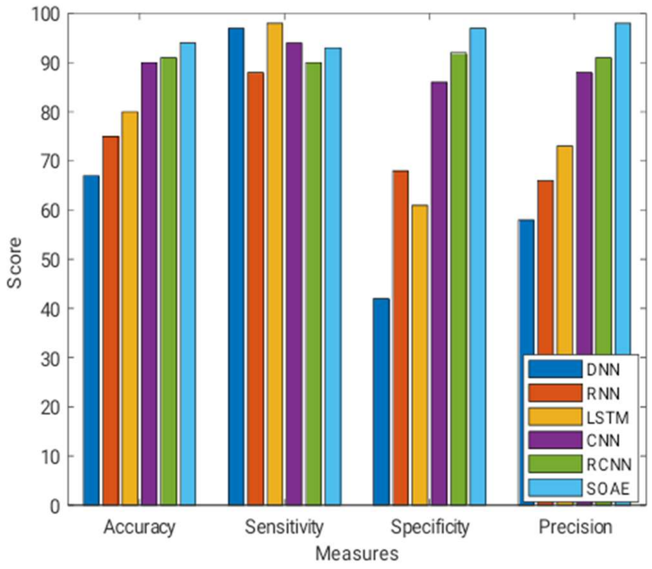

*Fig 5a Comparison based on the healthcare model using existing vs. proposed method*

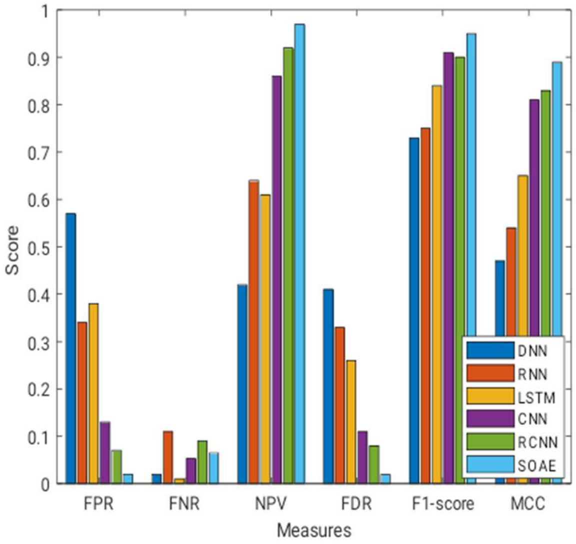

*Fig 5b Comparison based on the healthcare model using existing vs. proposed method*

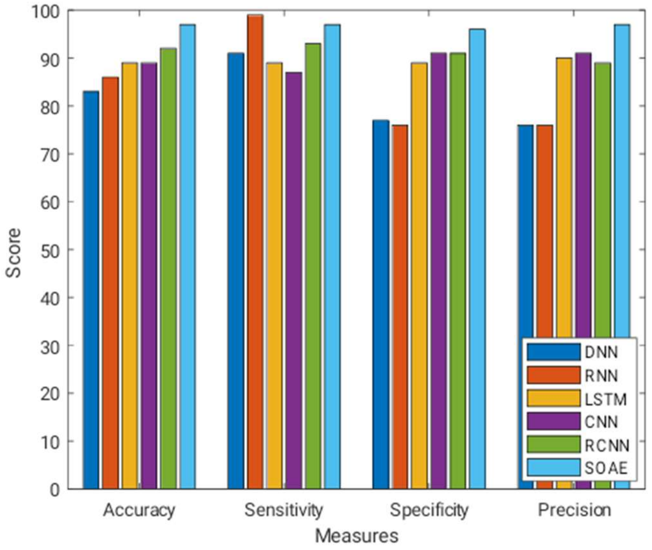

*Fig 5c Comparison based CV on existing vs. proposed method* $(k = 10)$

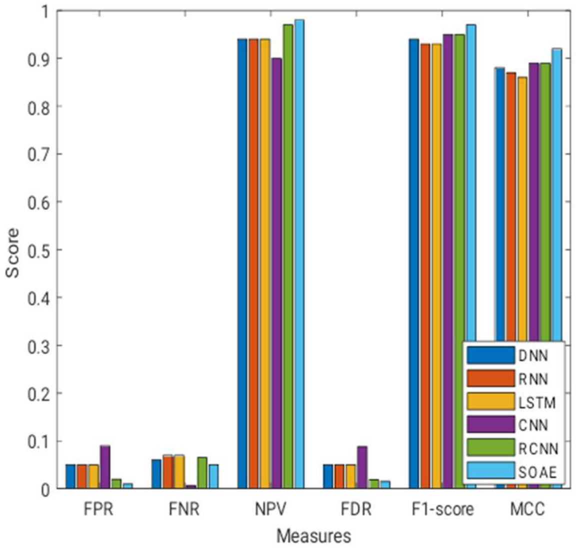

*Fig 5d Comparison based CV on existing vs. proposed method* $(k = 10)$

Fig 4 to Fig 5 looks at the accuracy, sensitivity, specificity, and precision of a complicated "prediction model" for heart disease and FNR, FPR, FDR, NPV, F1-score, and MCC. To show that changing the "learning percentages" is effective Compared to how well the current classifiers performed, the suggested intelligent model for predicting heart disease performs better. The proposed SOAE has a higher prediction and lower error rates for each performance indicator. Compared to RNN, DNN, LSTM, CNN, and RCNN, each 35%, SOAE is boosted by 67%, 53%, 33%, 11%, and 7%. LSTM, CNN, and DNN are outperformed by SOAE with improvements of 92%, 88%, 87%, 73%, and 45%, respectively, when the error metrics are considered. This results in lower error rates for the FPR. The recommended smart heart disease prediction model performs admirably because SOAE outperforms other classifiers in every performance metric when compared to them. The model's performance is assessed by changing the various $k$ −folds when comparing the smart healthcare prediction model to Fig 4a, 4b and Fig 5a, 5b depicts meta-heuristic classifiers and algorithms, respectively. The enhanced performance of the SOAE-based smart healthcare prediction model is measured using various performance measures. The $k$ −fold is two because the anticipated SOAE is 10%, 11%, 12%, and 8% superior to PSO, WOA, GWO and DHO, respectively. The SOAE also receives 83%, 83%, 64%, 51%, and 53% less FDR than RNN, DNN, LSTM, CNN, and RCNN at 5-fold CV. Because of this, the constructed smart heart disease detection model performs better when tested using the k-fold validation method. Based on the attained results, it is proven that the model acts as a smart healthcare model using IoT and provides earlier alert to help the needed people.

### 4.2. Comparative analysis

The proposed model adopts three major technologies like deep learning, optimization with heuristic approach and IoT for smart prediction. Some existing approaches fail to work on cross-domain to give promising outcomes. Tables 3 and 4 use a range of meta-heuristic-based classifiers and algorithms to utilize fog computing with IoT support and assess the proposed smart healthcare paradigm's general efficacy. Comparing DNN, RNN, LSTM, CNN, and RCNN with PSO, GWO, WOA and DHO, respectively, the suggested SOAE is 41%, 26%, 18%, 5%, and 4% more accurate. Similarly, the smart healthcare model's higher performance is apparent and has shown encouraging outcomes compared to traditional approaches. Tables 3 and 4 compare the proposed model for smart healthcare with computing supported by IoT, where the $k$ −fold is 10. The tables that compare different classifiers and meta-heuristic-based methods employ the resampling technique of cross-validation to assess models using machine learning on a small sample of data. The suggested 3 and 4 accuracies are 3%, 4%, 3%, 4%, 17%, 13%, 9%, 9%, and 6% more advanced than the acronyms for CNN, RCNN, RNN, LSTM, and DNN. It follows that the established smart healthcare model is superior to conventional methods. However, the model fails to analyze the computational complexity while executing the proposed model. Also, real time analysis with the real-time samples is not performed in the proposed work.

## 5. CONCLUSION

This work intends to provide a breakthrough in smart healthcare models using IoT. Information for the suggested model was acquired from various hardware equipment where some qualities' features were also retrieved. The proposed model focuses on modelling an efficient approach to provide an intelligent automated prediction model using the deep learning approaches. The diagnostic system was given all these qualities using the SOAE technique to optimize important encoder parameters using SOAE. Experimental analysis revealed that the anticipated SOAE outperformed DNN, RNN, LSTM, CNN, and RCNN by 3.7%, 3.7%, 3.6%, 7.6%, 67.9%, 48.4%, and 33%, respectively. So far, the performance of the IoT-enabled computing model for smart healthcare has been encouraging. Future iterations of the suggested model include more sophisticated feature representation, optimization and classification to enhance the prediction system's capacity to identify cardiac disease. Additionally, the anticipated SOAE is utilized in real-time applications in future.

**CONFLICT OF INTEREST:** There is no conflict of interest

**STATEMENT OF APPROVAL:** There are no reporting experiments on live vertebrates and/or higher invertebrates

**DATA AVAILABILITY:** The data produced are confidential

**COMPETING INTEREST:** There are no competing interests

## REFERENCES

[1] H. Malik, M. Alam, Y. Moullec, A. Kuusik, Narrowband IoT performance analysis for healthcare applications. Healthcare Appl., 1077–1083 (2018)

[2] S. Moosavi, E. Nigussie, M. Levorato, S. Virtanen, J. Isoaba, Performance analysis of end-to-end security schemes in healthcare IoT. Netw. Secure. 432, 439 (2018)

[3] C. Fonseca, D. Mendes, M. Lopes, A. Romao, P. Parreira, Deep learning and IoT to assist multimorbidity home-based healthcare. J. Health Med. Inf. 8, 1–4 (2017)

[4] D. Dziak, B. Jachimcrzyk, W. Kulesza, IoT based information system for healthcare

application: Design methodology approach. Appl. Sci. 7, 2–26 (2017)

[5] M. Kang, A study on the continuing usage of IoT-based healthcare wearable devices. Blue Eyes Intell. Eng. Sci. 8, 74–76 (2019)

[6] Z. Alansari, S. Soomro, M. Belgaum, S. Shamshriband, The rise of the internet of things in big healthcare data. Internet of things in healthcare 564, 675–685 (2018)

[7] S. Pinto, J. Cabral, T. Gomes, We care An IoT-based healthcare system for elderly people. Healthcare Syst., 1378–1383 (2017)

[8] H. Zakaria, N. Bakar, N. Hassan, S. Yaccob, IoT security risk management model for secured practice in the healthcare environment. Procedia Comput. Sci. 161, 1241–1248 (2019)

[9] P. Shakeel, S. Baskar, V.R. Dhulipala, S. Mishra, M. Jaber, Maintaining security and privacy in healthcare system using learning based deep Q networks. J. Med. Syst. 42, 1–10 (2018)

[10] Kumar, S. Lokesh, R. Varatharian, G. Babu, P. Parthasarathy, Cloud and IoT based disease prediction and diagnosis system for healthcare using the fuzzy neural classifier. Futur. Gener. Syst. 86, 527–534 (2018)

[11] Qayyum, J. Qadir, M. Bilal, A. Fuqaha, Secure and robust machine learning for healthcare: A survey. Biomed. Eng., 1–22 (2020)

[12] Verma, S. Sood, Cloud-centric IoT based disease diagnosis healthcare framework. J. Parallel Distrib. Comput. 116, 27–38 (2018)

[13] Greco, G. Percannella, P. Ritrovato, F. Tortorella, M. Vento, Trends in IoT solutions for healthcare: Moving AI to the edge. Pattern Recogn. Lett. 135, 346–353 (2020)

[14] Bholwal, Extensive study Of IoT in healthcare based on machine learning and cloud. Innov. Eng. Technol. 12, 014–018 (2019)

[15] Kumar, K. Sood, S. Kaul, R. Vasuja, Big data analytics and its benefits in healthcare, in Big Data Analytics in Healthcare, Studies in Big Data, vol. 66, (Springer, Cham, 2020), pp. 2013–2235

[16] Sherubha, "Graph-Based Event Measurement for Analyzing Distributed Anomalies in Sensor Networks", Sådhanå(Springer), 45:212, https://doi.org/10.1007/s12046-020-01451-w

[17] Sherubha, "An Efficient Network Threat Detection and Classification Method using ANP-MVPS Algorithm in Wireless Sensor Networks", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8 Issue-11, September 2019

[18] Sherubha, "An Efficient Intrusion Detection and Authentication Mechanism for Detecting Clone Attack in Wireless Sensor Networks", Journal of Advanced Research in Dynamical and Control Systems (JARDCS), Volume 11, issue 5, Pg No. 55-68

[19] Sun, Y.; Song, H.; Jara, A.; Bie, R. Internet of Things and Big Data Analytics for Smart and Connected Communities. IEEE Access 2016, 4, 766–773

[20] Qin, Y.; Sheng, Q.Z.; Falkner, N.J.G.; Dustdar, S.; Wang, H.; Vasilakos, A.V. When Things Matter: A Survey on Data-Centric Internet of Things. J. Netw. Comput. Appl. 2016, 64, 137–153.

[21] Zhang, L.; Jeong, D.; Lee, S. Data Quality Management in the Internet of Things. Sensors 2021, 21, 5834.

[22] Siow, E.; Tiropanis, T.; Hall, W. Analytics for the Internet of Things: A Survey. ACM Comput. Surv. 2018, 51, 1–36.

[23] Mahdavinejad, M.S.; Rezvan, M.; Barekatain, M.; Adibi, P.; Barnaghi, P.; Sheth, A.P. Machine Learning for Internet of Things Data Analysis: A Survey. Digit. Commun. Netw. 2018, 4, 161–175.

[24] Mohammadi, M.; Al-Fuqaha, A.; Sorour, S.; Guizani, M. Deep Learning for IoT Big Data and Streaming Analytics: A Survey. IEEE Commun. Surv. Tutor. 2018, 20, 2923–2960.

[25] De Morais, C.M.; Sadok, D.; Kelner, J. An IoT Sensor and Scenario Survey for Data Researchers. J. Braz. Comput. Soc. 2019, 25, 4.

[26] Mahmood, M.A.; Seah, W.K.G.; Welch, I. Reliability in Wireless Sensor Networks: A Survey and Challenges Ahead. Comput. Netw. 2015, 79, 166–187.

[27] Garcia-font, V.; Garrigues, C. A Comparative Study of Anomaly Detection Techniques for Smart City Wireless Sensor Networks. Sensors 2016, 16, 868.

[28] Alghanmi, N.; Alotaibi, R.; Buhari, S.M. Machine Learning Approaches for Anomaly Detection in IoT: An Overview and Future

Research Directions. Wirel. Pers. Commun. 2021, 122, 2309–2324.

[29] Al Samara, M.; Bennis, I.; Abouaissa, A.; Lorenz, P. A Survey of Outlier Detection Techniques in IoT: Review and Classification. J. Sens. Actuator Netw. 2022, 11, 4.

[30] Diro, A.; Chilamkurti, N.; Nguyen, V.D.; Heyne, W. A Comprehensive Study of Anomaly Detection Schemes in IoT Networks Using Machine Learning Algorithms. Sensors 2021, 21, 8320.