IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 September 2023, accepted 17 October 2023, date of publication 19 October 2023, date of current version 26 October 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3326129

RESEARCH ARTICLE

Early Mental Stress Detection Using Q-Learning Embedded Starling Murmuration Optimiser-Based Deep Learning Model

SYED KUMAYL RAZA MOOSAVI[1], MUHAMMAD HAMZA ZAFAR[2], (Graduate Student Member, IEEE), FILIPPO SANFILIPPO[2,3], (Senior Member, IEEE), MALIK NAVEED AKHTER[4], AND SHAHZAIB FAROOQ HADI[5]

[1]School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan
[2]Department of Engineering Sciences, University of Agder, 4879 Grimstad, Norway
[3]Department of Software Engineering, Kaunas University of Technology, 51368 Kaunas, Lithuania
[4]School of Mechanical and Manufacturing Engineering, National University of Sciences and Technology, Islamabad 44000, Pakistan
[5]College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad 44000, Pakistan

Corresponding author: Filippo Sanfilippo (filippo.sanfilippo@uia.no)

This work was supported by the Biomechatronics and Collaborative Robotics Research Group at the Top Research Center Mechatronics (TRCM), University of Agder (UiA), Norway.

ABSTRACT Stress affects individual of all ages as a regular part of life, but excessive and chronic stress can lead to physical and mental health problems, decreased productivity, and reduced quality of life. By identifying stress at an early stage, individuals can take steps to manage it effectively and improve their overall well-being. Feature selection is a critical aspect of early stress detection because it helps identify the most relevant and informative features that can differentiate between stressed and non-stressed individuals. This paper firstly proposes a variance based feature selection technique that uses q-learning embedded Starling Murmuration Optimiser (QLESMO) to choose relevant features from a publicly available dataset in which stresses experienced by nurses working during the Covid'19 Pandemic is recorded using bio-signals and user surveys. Furthermore, a comparative study with other metaheuristic based feature selection techniques have been demonstrated. Next, to evaluate the efficacy of the proposed algorithm, 10 benchmark test functions have been used. The reduced feature subset is then classified through a 1D convolutional neural network (CNN) model (QLESMO-CNN) and is seen to perform well in terms of the evaluation metrics in comparison to other competitive algorithms. Finally, the proposed technique is compared with the State-of-the-Art methodologies present in literature. The experiments provide a strong basis to determine features that are most relevant for early mental stress classification using a hybrid model combining CNN, Reinforcement Learning and metaheuristic algorithms.

INDEX TERMS Machine learning, stress detection, reinforcement learning, starling murmuration optimiser, feature selection, 1DCNN, metaheuristic algorithms.

I. INTRODUCTION

Technological advancements have led to the generation of large amounts of data in various fields, including healthcare, finance, and social media. Healthcare, in particular, has seen a massive increase in data generation with electronic health records, medical imaging, genomics, and sensor data [1]. Storing, managing, analysing, and interpreting this large and complex data has become a challenge. Traditional database tools are inadequate to handle such vast amounts of data with varying structures. However, the development of advanced analytics and machine learning techniques has enabled the processing and analysis of this data to generate valuable insights for improving patient care, clinical outcomes and

The associate editor coordinating the review of this manuscript and approving it for publication was Antonio J. R. Neves.

healthcare in smart cities. For example the authors in [2] developed a concept of smart cities model integrating IoT and AI technologies to enhance citizens' daily lives for stress management.

The implementation of big data analytics in healthcare has immense potential to revolutionise the way healthcare is delivered and has already shown promising results in improving patient outcomes [3]. For example, the analysis of electronic health records can help identify disease patterns and predict outcomes, leading to more targeted and effective treatments. The use of genomics data can also help develop personalised treatments and therapies for patients based on their genetic makeup.There is an increasing body of research indicating that mobile health (mHealth) systems' technological advancements can be leveraged to create diverse intervention approaches to aid individuals with autism spectrum disorder (ASD) in coping with their problem behaviours and adapting to their surroundings [4], [5], [6]. These advancements in healthcare data analytics are transforming the way healthcare is delivered and hold the promise to reshape the field.

In recent years, there has been an increasing interest in detecting and managing stress, which has become a common problem among individuals due to various factors, such as work pressure, lifestyle, and social environment. Early detection of stress is crucial in preventing its adverse effects on an individual's physical and mental health. Stress detection refers to the process of identifying and monitoring signs of stress in individuals using data from wearable sensors, social media, and other sources [7].

Wearable sensors are being used to monitor physiological and behavioural indicators of stress, such as heart rate variability, skin conductance, and activity levels [8]. For example, a study used a wearable device to collect data on physiological indicators of stress in individuals during a public speaking task. The data was then analysed using machine learning techniques to predict stress levels with high accuracy [9]. Social media data, such as posts, likes, and comments, are being used to analyse an individual's emotional state and mood changes. For instance, a study used Twitter data to analyse the relationship between stress and sleep deprivation [10].

The data generated from wearable sensors and social media can be analysed using advanced analytics and machine learning techniques to identify patterns and predict stress levels. Machine learning algorithms such as Random Forest and Support Vector Machines are being used to develop predictive models for stress detection [11]. Natural language processing (NLP) techniques are being used to analyse text data from social media and emails to identify linguistic indicators of stress. Computer vision techniques are being used to analyse facial expressions and body language to identify signs of stress.

The integration of stress detection systems with digital health platforms is enabling real-time stress monitoring and management for individuals. For example, a study developed a smartphone application that used machine learning to predict stress levels based on physiological data collected from wearable sensors. The application provided personalised stress management strategies to users based on their stress levels and preferences [12]. Similarly, a wearable health sensor monitoring system based on a multi-sensor fusion approach was presented in [13] and [14]. The combination of fog computing, IoT sensors, and a Random Forest machine learning technique was proposed by the authors in [15] as a model to effectively mitigate healthcare delay and improve patient care.

While the collection of large amounts of data is crucial for stress detection, it also presents significant challenges in terms of storage, management, and processing. With the increasing use of wearable devices, sensors, and mobile apps for collecting physiological and behavioural data, the amount of data generated has significantly increased. The storage and management of such large amounts of data require advanced computing infrastructure and storage systems that can handle the volume, velocity, and variety of data [16].

Furthermore, processing such large amounts of data requires sophisticated data analytics and machine learning algorithms to identify patterns and relationships within the data. This is particularly important in stress detection, where subtle changes in physiological and behavioural data can be indicative of stress levels. The processing of such large amounts of data also presents significant challenges in terms of computational resources and time. Another challenge in storing large amounts of data for stress detection is ensuring data privacy and security. With the increasing use of wearable devices and mobile apps for data collection, there is a growing concern about the security and privacy of such data [17].

To solve the challenges accompanied with data storage and analysis feature selection can be used as a means of reducing the amount of unnecessary data that needs to be stored and processed, thus addressing some of the challenges posed by the storage of large amounts of data. By selecting only the most relevant features for stress detection, feature selection can reduce the dimensionality of the dataset and improve the efficiency of data storage and processing. This can also lead to improved accuracy and performance of stress detection algorithms, as irrelevant features can introduce noise and reduce the discriminatory power of the data.

Various methods for feature selection exist in the literature, such as filter methods, wrapper methods, and embedded methods [18]. Filter methods rank features based on their correlation with the target variable and select the top-ranked features for analysis, while wrapper methods evaluate the importance of features using a machine learning algorithm and select the subset that yields the best performance. Embedded methods involve incorporating feature selection into the model training process, such as through the use of regularisation techniques. Research has shown that feature selection can significantly improve the efficiency and performance of stress detection algorithms. For example, a study used feature selection to reduce the dimensionality of physiological and behavioural data for stress detection and found that the reduced dataset yielded similar performance to the full dataset [19]. Another study used a combination of feature selection and deep learning techniques for stress detection and found that the selected features improved the accuracy and efficiency of the algorithm [20].

Feature selection is a crucial step in building machine learning models that can perform well on a given task. One popular approach to feature selection is metaheuristic-based methods, which leverage optimisation algorithms inspired by natural processes like evolution, swarm behavior, and immune systems. These methods can effectively search through large and complex feature spaces to identify a subset of relevant features that can improve model performance. Unlike traditional optimisation techniques, metaheuristic algorithms are able to handle difficult optimisation problems that may be impractical or infeasible for other methods. In the context of stress detection, metaheuristic algorithms can be used to optimise the feature selection process, thereby improving the efficiency and performance of stress detection algorithms.

This study aims at analysing the feasibility of using the physiological and behavioural data of nurses working during the Covid-19 pandemic in hospitals to predict stress levels in real-world scenarios. Within this framework, a reinforcement learning combined with a nature inspired algorithm is proposed for early mental stress detection. The dataset utilised is first run through a feature selection process to generate a subset dataset of relevant features. Consequently, the subset is used as input for a novel convolutional neural network (CNN) model for classification and analysis. The contributions of this paper, illustrated in Figure 1, are highlighted as follows:

- A mathematical model of a Reinforcement Learning based Starling Murmuration Optimiser algorithm (QLESMO) is proposed. The combination offers the ability to efficiently locate global minima while also mitigating the issue of model overfitting.
- The effectiveness of the proposed QLESMO is shown by its evaluation on a set of 10 benchmark test functions.
- QLESMO method is used to conduct feature selection on a dataset related to the detection of mental stress. The objective is to identify a subset of relevant features that indicate early indicators of stress levels among nurses working in a hospital environment during the Covid-19 epidemic.
- A Comparative Analysis is conducted using competitive metaheuristic algorithms in order to demonstrate the algorithm's superiority in the context of feature selection.
- The hybrid QLESMO model is combined with a 1DCNN architecture for multiclass classification of mental stress detection.
- The proposed QLESMO-based CNN model is evaluated and compared with other models, namely MFA-CNN, AOA-CNN, and GWO-CNN, particularly in the domain of classification tasks.

II. RELATED WORK

The use of metaheuristic algorithms in feature selection has been a popular research topic in recent years. These algorithms do not rely on explicit problem-specific knowledge and can be applied to a wide range of problems in various domains. Metaheuristic algorithms are often used in optimisation problems where the search space is large and complex, and traditional methods may not be effective. Metaheuristic algorithms differ from traditional optimisation algorithms in that they do not guarantee optimal solutions but aim to find good solutions within a reasonable amount of time. These algorithms often use stochastic processes and randomisation to explore the search space and avoid getting trapped in local optima. Metaheuristic algorithms also allow for parallelisation and distributed computing, which can speed up the optimisation process [21].

Various metaheuristic algorithms, such as genetic algorithm (GA) [22], particle swarm optimisation (PSO) [23], ant colony optimisation (ACO) [24], and simulated annealing (SA) [25], have been applied to feature selection in various domains, including healthcare, finance, and engineering. From an inspirational standpoint, metaheuristic algorithms can be subdivided into biology-based, evolutionary-based, mathematics-based, physics-based, and Human Social Interaction-based [26], [27], [28]. Figure 2 shows the weighted radar chart of research that has been conducted in the last two decades on different applications for metaheuristic algorithms.

To select relevant features, binary vector representations are commonly used. A solution vector in a feature selection algorithm is usually represented as (10101100 . . . ..), where each 1 indicates a selected feature and each 0 indicates a non-selected feature in the subset. Literature survey reveals many different techniques using metaheuristic techniques, variants, and hybridised versions for creating subset features of many multifaceted datasets. Using discrete cosine transformation Tiwari [29] used an algorithm for face recognition problem. Nakamura et al. [30] developed a binary Bat Algorithm (BA) using a sigmoid function to restrict the bat agent's position to binary variables, which was evaluated on five datasets using the optimum path forest classifier. In [31], the authors improved the Krill Herd Algorithm (KHA) named as IG-MBKH, using a hyperbolic tangent function and adaptive transfer function along with information gain feature ranking. Allam and Nandhini [32] developed a binary Teaching Learning Based Optimiser (TLBO) with a threshold value for breast cancer datasets, and Agrawal et al. [33] proposed a novel binary version of the Gaining Sharing Knowledge (GSK) algorithm with KNN classifier for 23 benchmark datasets from the University of California

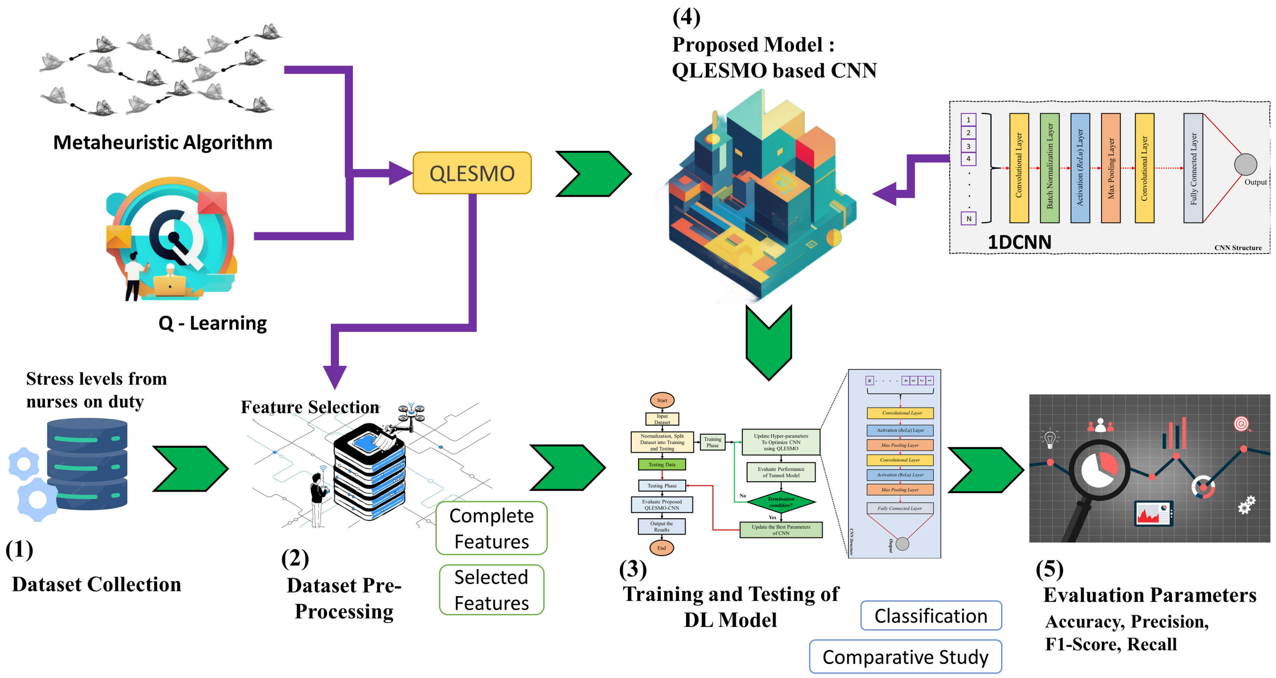


FIGURE 1. System flow diagram for the QLESMO-based CNN model that is used for the selection and classification of mental stress dataset features.

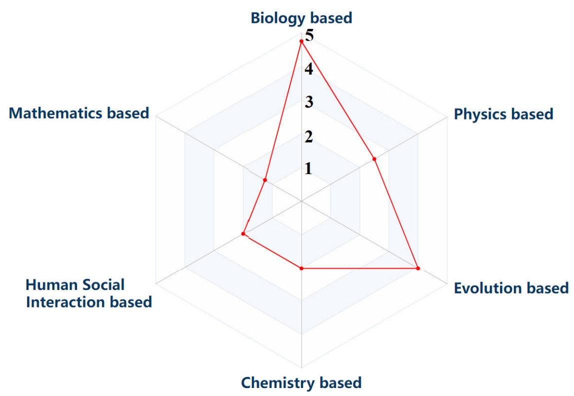


FIGURE 2. Weighted radar chart for amount of publications in literature for different types of metaheuristic algorithms.

Irvine (UCI) repository. Additionally, several physics-based algorithms, such as multi-verse optimiser [34], sine-cosine algorithm [35], and gravitational search algorithm [36], have also been proposed for feature selection.

Hybrid versions of metaheuristic algorithms were developed to improve the performance of feature selection methods in terms of accuracy and efficiency. Hybridisation involves combining two or more metaheuristic algorithms to create a new algorithm that is better suited for the specific problem at hand. In addition, they can provide better diversity in the search process and prevent the algorithm from getting stuck in local optima. This is achieved by combining different types of search strategies that explore the search space in different ways. In [38], the authors developed a hybrid version of Gorilla Troops Optimiser (GTO) and Bird Swarms Algorithm (BSA) and tested it on four IOT based datasets and found increase in the final output and improved convergence curves compared to other state-of-the-art techniques. A synchronisation neural network structure named Double Layer Tree Parity Machine (DLTPM) is suggested in [39]. The paper proposes whale optimisation-based DLTPM. This model employs a whale algorithm optimised weight vector for quicker synchronisation.

A nature-inspired feature selection algorithm was developed by Das et al. [40], by hybridising Binary Bat Algorithm (BBA) with Late Acceptance Hill-Climbing (LAHC) to select the optimal subset of well-known features and deep learning-based features for identifying different Indian languages based on audio signals, with the aim of reducing model complexity and training time, and to achieve high accuracy rates. In [41], the authors proposed a less expensive computational model for emotion classification through speech analysis using a meta-heuristic feature selection method called Golden Ratio based Equilibrium Optimisation (GREO) algorithm. The optimally selected features by the model are fed to the XGBoost classifier, and the proposed model achieved impressive recognition accuracy of 97.31% and 98.46% on two standard datasets, SAVEE and EmoDB, respectively.

A subsclass of deep learning models namely; Reinforcement learning is another popular technique that has been used for solution optimisation in various domains, including healthcare, finance, and engineering. Reinforcement learning is a machine learning technique that involves an agent learning to take actions in an environment to maximise a reward signal. These algorithms combine the strengths

FIGURE 3. A synchronised murmuration behaviour of starlings (Photo taken from the atlantic shown from a distance near, Israel, on February 10, 2016 [37].).

of reinforcement learning and metaheuristic algorithms to improve the efficiency and performance of the model.

Many reinforcement learning based metaheuristic algorithms have been proposed in recent years [42]. For instance, Zhao et al. [43] proposed an inverse reinforcement learning framework with Q-learning mechanism named IRLMFO, to improve the performance of the moth-flame optimisation (MFO) algorithm in a large-scale real-parameter optimisation problem. The framework aimed to choose the right strategy using historical data from the strategy pool, and strengthen the exploitation capability of the IRLMFO algorithm with a competition mechanism. The performance of the algorithm in [44] namely; a hybrid metaheuristic algorithm called QBSO-FS that integrates a reinforcement learning algorithm with Bee Swarm Optimisation metaheuristic (BSO), was evaluated on 20 well-known datasets for feature selection. The vanilla Sine Cosine Algorithm was embedded with Q-learning technique, in [45], and evaluated on several benchmark problems and engineering case studies demonstrating that the results outperform other optimisation algorithms in terms of fitness value and convergence speed. In [46], the exploration and exploitation phase of the Sand Cat Swarm Optimisation (SCSO) metaheuristic algorithm was improved by automatic switching of these two phases using reinforcement learning techniques.

The preceding papers have demonstrated the importance of feature selection based on metaheuristic approaches. However, choosing the optimal feature selection technique and classification algorithm is crucial. The selection must take into account factors such as accuracy and precision improvement, training time reduction, and minimal error. In a dataset, certain features may exhibit high correlation, leading to data redundancy. Other features may introduce noise and cause the classification algorithm to overfit, adversely affecting the system's performance. This study aims to compare various metaheuristic algorithms for feature selection against the proposed method and determine which algorithm can significantly reduce and select the features that have a greater impact on the system, improving the accuracy and performance of early stress detection. Following that, the architecture proposed in this study utilises the hybridisation of metaheuristic algorithms and reinforcement learning for the feature selection and accurate classification of mental stress.

III. FUNDAMENTAL

This section introduces the concepts and mathematical models of the vanilla Starling Murmuration Optimiser (SMO) [47], Q-Learning Algorithm and the proposed hybrid algorithm.

A. STARLING MURMURATION OPTIMISER

One of nature's most spectacular displays is the murmuration of starlings. During this awe-inspiring event, multiple flocks of starlings come together to form intricate and coordinated flight patterns in the sky above their roost for approximately thirty minutes, as depicted in Figure 3. This remarkable behaviour is typically triggered by the presence of a predator, such as a peregrine falcon or hawk, at dusk. The starlings expertly split and recombine their flocks in a highly synchronised manner during the murmuration. To avoid being preyed upon, they also frequently change their direction of motion, but the slightest uncertainty in their movements can disrupt the cohesion between the flocks and lead to wayward starlings being hunted down by predators.

The optimised decision-making process of the starlings enables them to achieve this impressive synchronisation. Each bird aligns its direction of motion with its neighbours while avoiding collisions and flying as closely as possible. This behaviour leads to frequent changes in the neighbours of each starling, producing a dynamic neighbourhood and various typologies within the murmuration. This section provides a detailed description the mathematical model that has been derived, inspired by the unique movement of these birds.

A dynamic multi-flock construction is employed to model the behaviour of the starlings in which each iteration moves the starlings to another flock in the population. To explore and exploit promising solution areas within the search space the movement strategy of the starlings are divided into three distinct strategies namely; diving, whirling and separating. To begin with, a solution candidate (X) matrix is determined, set initially as random values within a search space, that would represent the initial positional vectors of the starlings, as shown in Equation (1). Each position vector also accompanies an initial fitness function value.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^d \\ x_2^1 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \dots & \vdots \\ x_N^1 & x_N^1 & \dots & x_N^d \end{bmatrix}, \tag{1}$$

where X represents the candidate solution of the preys in the search space, d represents the number of dimensions of the current problem and N is defined as the number of candidates utilised for the metaheuristic algorithm. When the solution X moves using exploration and exploitation strategy, the movement can sometimes go beyond the scope of the search space region. Therefore boundary conditions are required to keep the search space within an enclosed area. Equation (2) represents the search space boundary conditions where X_{min} is the minimum value that the current solution of X at iteration t can achieve while X_{max} is the maximum possible attainable value that can be achieved. These values are set according to the problem at the start of the execution of the algorithm:

$$\begin{aligned} X_j^i(t) = Range \text{ --- } [X_{min} \text{ - } X_{max}] \ ; \\ where \begin{cases} i = 1, 2, \dots, d \\ j = 1, 2, \dots, N \end{cases} \end{aligned} \tag{2}$$

Starlings form the murmuration M in which a portion of the starling is separated to enhance the population diversity using a separating search strategy as shown in Equation (3). Then, each constructed flock flies using either the diving or whirling search strategy:

$$P_{sep} = \frac{log(t+d)}{2log(It_{max})}, \tag{3}$$

where P_{sep} represents the construction of the new population of the starlings based on the iterative level t. The iterative update method of the separating strategy of starlings between different flocks is shown in Equation (4):

$$X_i^{t+1} = X_i^t + \mathfrak{Q} \times (X_{r'}^t - X_r^t), \tag{4}$$

where X_i^t represents the current positional vector of the starling i and X_i^{t+1} is the next position of the starling. $X_{r'}$ is the random position of starling from the selected flock of the population while X_r is the random starling in the entire population of the starlings. $\mathfrak{Q}$ is defined as the separation operation that uses a quantum harmonic oscillator to maintain the population diversity. The details of the quantum harmonic oscillator mathematical model is provided in the original SMO paper [47].

Next each starling position in each flock in the population is updated with the diving strategy or the whirling strategy. The selection of strategies by starlings is determined based on the following equation:

$$X_i^{t+1} = \begin{cases} Diving \ \ Strategy & Q_q < \mu_q \\ Whirling \ \ Strategy & Q_q \geq \mu_q \end{cases}, \tag{5}$$

where Q_q is the quality of flock q (shown in Equation (6)) and μ_q is the average of the quality of all flocks in the population.

$$Q_q(t) = \frac{\sum_{i=1}^{k} \frac{1}{N} \sum_{j=1}^{N} fv_{ij}(t)}{\frac{1}{n} \sum_{i=1}^{n} fv_{qi}(t)}, \tag{6}$$

where fv_{ij} is the fitness value of the i_{th} starling in the proportion of the flock population and k is the number of flocks in a murmuration M.

The diving technique involves the exploration phase and the whirling technique involves the exploitation phase of the metaheuristic algorithm. The diving search approach is intended to effectively explore the search space when the flock quality is considered bad ($Q_q < \mu_q$). Given that each flock q, including starlings k, is at an unfavourable location. This area is then avoided by the diving search method, which includes uphill and downward quantum dives as well as a quantum random dive (QRD) operator for picking quantum dives. On the other hand, When flock has a good quality ($Q_q > \mu_q$), then the next position of each starling k is determined using the whirling search strategy. Inspired by the nature of murmuration, the whirling search strategy exploits the search zones using Equation 7. The details of the two strategies for the updation of the position of the starlings are further elaborated in the original SMO paper [47].

$$X_i(t+1) = X_i(t) + \cos(r_a) \times (X_R(t) - X_N(t)), \tag{7}$$

where r_a is a random value between [0, 1], X_R is the random member from the flocks in the population and X_N is a distinct starling which has not been picked from previous iterations in the flocks. Algorithm 1 provides the pseudo code of the SMO Algorithm.

B. Q-LEARNING TECHNIQUE

By ways of interacting with the environment, a reinforcement technique is developed in literature. By providing the concept

Algorithm 1 Pseudo Code of the SMO

Set the number of max_iter, population size, dimension d
Initialise the starling population according to equation (1).

Evaluate initial candidate solution of the population
Determine Global Best (GB) from the entire population
while iter $\leq$ max_iter **do**
  Determine new population distribution at each iter
  Use separation strategy to select the proportion of starlings using equation (3)
  Run multi-flock construction algorithm of the SMO
  Compute quality of flock Q_q and the average quality of the population flock μ_q
  if $Q_q < \mu_q$ **then**
    Determine new starling position using diving strategy
  else
    Determine new starling position using whirling strategy
  end if
  Evaluate boundary conditions using equation (2)
  Calculate Starling population fitness
  Determine the GB position from population
  iter = iter + 1
end while
return GB position

of reward or punishment the learning agent is taught to perform at optimum capacity without having the need to specify exactly the task that the agent needs to achieve. In this context, the intelligent agent (learner) takes action based on its current state and is rewarded or penalised by the environment. In order to optimise the cumulative reward, the agent takes into account the actions it has taken in the past. Formally, we define the set of all states in the complete environment as a set $S = s_1, s_2, \ldots .s_n$ and all possible actions that can be taken by the agent at each state S is represented as a set $A = a_1, a_2, \ldots .a_n$. Whenever the agent performs an action from a state, it is provided with a reward r_t where t represents the reward at that instead from the specific action state protocol. The reward can be negative indicating punishment receival. Using this information the agent is assigned to task to maximise its value by learning the policy π for $S \rightarrow A$ combination:

$$V^*(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} \ldots, \tag{8}$$

where the value of *gamma*, the discount parameter, is determined within the range [0,1]. Interestingly, the specific value of this hyperparameters largely determines the behaviour of the learner. If the value is close to 0, the learner will tend to forget the past reward points achieved by the learner and will only focus on the current reward. Conversely, if its closer to 1, the learner will tend to use the rewards received in the past as a determining factor for the value maximisation.

The most common type of reinforcement learning methodology is the Q learning technique which consists of learning the value of Q which is the reward received immediately upon executing action a from state s, plus discount parameter γ computed using equation (8). Q is estimated recursively based on this equation:

$$Q(s, a) = r(s, a) + \gamma \times Q_{max}(\delta(s, a), a'), \tag{9}$$

where $\delta(s, a)$ denotes the resultant state that is applied on applying the action a state s pair. In the context of metaheuristic algorithms, when learning, the agent faces an important choice of movement in the exploitation phase or exploration the exploration phase. The challenge for the agent is to determine the appropriate balance between exploring unknown states and actions and exploiting early visited states and actions in order to maximise reward accumulation. The Q-learning algorithm 2 begins by initialising a table with zero values for each possible state in the search space. As the agent moves through different states and takes actions, the algorithm observes the resulting rewards and updates the Q-table accordingly.

Algorithm 2 Pseudo Code of the Q-Learning Algorithm

Set the Q-table entries to zero
Observe current state
Select an action from the list of possible actions in the current state
Observe reward obtained from selecting that action
Update Q-table entry according to equation (9)
Repeat Q-table updation until stopping criteria achieved

C. METAHEURISTIC FEATURE SELECTION TECHNIQUE

In feature selection, the goal is to identify the most suitable features from the complete dataset. This involves generating a binary output of 1s and 0s to indicate the selected features. To represent a solution, a one-dimensional vector is used with each cell assigned a value of either 1 or 0. A value of 1 indicates that the associated attribute is chosen while 0 indicates that it is not.

The fitness function used in the algorithm determines the values assigned to each attribute in the vector. This approach allows for an efficient selection of the most appropriate features from the original dataset. The traditional fitness function involves the maximisation of the Accuracy of the subset feature set and the number of unselected features according to equation (10):

$$F.F = \alpha \cdot Accu._{KNN} + (1 - \alpha) \cdot \frac{Tot\_feat. - sel.\_feat.}{Tot\_feat.} \tag{10}$$

While the use of metaheuristic algorithm certainly improve the performance of the feature selection technique, a classification algorithm i.e. KNN inhibits the computational time of the feature selection process as the metaheuristic algorithm has to determine the fitness value of the population for each iteration. This equates to the fact that each iteration of the

TABLE 1. The structure of the Q-table in the proposed model with initial conditions.

	$Action_1$ (r_1=H & r_2=H)	$Action_2$ (r_1=H & r_2=M)	$Action_3$ (r_1=M & r_2=H)	$Action_4$ (r_1=M & r_2=M)	$Action_5$ (r_1=H & r_2=L)	$Action_6$ (r_1=L & r_2=H)	$Action_7$ (r_1=M & r_2=L)	$Action_8$ (r_1=L & r_2=M)	$Action_9$ (r_1=L & r_2=L)
$State_1$ (Den=H & Dis=H)	0	0	0	0	0	0	0	0	0
$State_2$ (Den=H & Dis=M)	0	0	0	0	0	0	0	0	0
$State_3$ (Den=M & Dis=H)	0	0	0	0	0	0	0	0	0
$State_4$ (Den=M & Dis=M)	0	0	0	0	0	0	0	0	0
$State_5$ (Den=H & Dis=L)	0	0	0	0	0	0	0	0	0
$State_6$ (Den=L & Dis=H)	0	0	0	0	0	0	0	0	0
$State_7$ (Den=M & Dis=L)	0	0	0	0	0	0	0	0	0
$State_8$ (Den=L & Dis=M)	0	0	0	0	0	0	0	0	0
$State_9$ (Den=L & Dis=L)	0	0	0	0	0	0	0	0	0

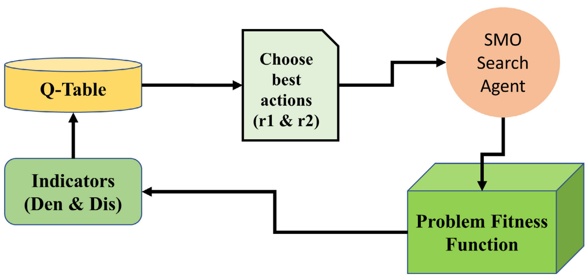


FIGURE 4. Overview of the QLESMO algorithm.

population will have to go through the KNN classification algorithm. To circumvent such issues in the process of feature selection, we propose a variance based feature selection technique as shown in equation (11) that uses the data variance as a tool to determine the fitness value. The new approach is a minimisation problem. This results in a significant reduction in the algorithm's processing time to produce optimal features:

$$F.F = \alpha \cdot Var._{Dataset} + (1-\alpha) \cdot \frac{Tot\_feat. - unsel.\_feat.}{Tot\_feat.} \tag{11}$$

IV. PROPOSED TECHNIQUE

In this section, first, the novel Q-learning embedded Starling Murmuration Optimiser (QLESMO) is presented highlighting the use of this approach for optimisation problems. Successively, the architecture of the proposed 1D-Convolutional Neural Network (CNN) is explained. An important part of 1DCNN is the hyperparameters, which are explained subsequentially. In the last part, CNN training using QLESMO is outlined.

A. Q-LEARNING EMBEDDED STARLING MURMURATION OPTIMISER

The QLESMO algorithm incorporates the Q-Learning approach to guide SMO search agents towards discovering the entire search space in a more efficient manner, avoiding local optima during runtime. This differs from the traditional SMO technique, which employs flock quality to switch between diving and whirling equations. Instead, our proposed algorithm uses Q-learning to control two critical parameters, namely r_1 and r_2, replacing the flock quality parameters Q_q and μ_q. The movement of the search agent within the search space is heavily influenced by these parameters, which are critical in directing its exploration. The Q-table is used to modify the values of these parameters in accordance with the values stored in the Q-table. (See Table 1)

So, when r_1 is lower than r_2, the SMO algorithm will be in the exploitation mode and conduct the diving strategy. On the other hand, the algorithm performs the whirling strategy when r_1 is higher than r_2 and traverses the search space as an explorative agent. Now the architecture of the Q-table consists of 9 actions. Initially all the values in the Q-table are set to zero so that the initial stage will have an equally likely reward from the Q-table. Once the action is performed on the SMO algorithm, if the agent finds a better fitness solution, the reward is set to 1 and conversely if the agent does not find the better solution than the reward is set to a penalty of -1. Using the equation (9), the Q-table is updated.

The parameter α is utilised to signify the importance of both the variance quality and subset length, and its value is predetermined prior to the commencement of the algorithm. In this study, α has been set to 0.5.

Next, the positions of the agents are measured and various states are identified based on their locations relative to the population location. To determine the population status and the position of individual agents in relation to the global best position, two indicators are employed: population density and population distance [48], shown in equations (12) and (13), respectively:

$$Density = \frac{1}{N|L|} \times \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{D} (X_i^j - \overline{X^j})^2} \tag{12}$$

$$Distance = \frac{\sum \sqrt{|P - X(i,:)|^2}}{\sum (Bound_{Upper} - Bound_{Lower})^2} \tag{13}$$

Algorithm 3 Steps for QLESMO

Set the number of max_iter, population size, dimension d and α for the fitness function
Initialise the SMO population and Q-table
Evaluate initial candidate solution of the population and indicators(Den, Dis)
Determine Global Best (GB) position from the entire population
while iter $\leq$ max_iter **do**
  Determine new Starling population distribution at each iter
  Compute Indicator values
  Evaluate action based on Q-table from state values
  Determine values for r_1 and r_2
  if $r_1 \leq r_2$ **then**
    Diving Action Execute
  else
    Whirling Action Execute
  end if
  Calculate Starling fitness population using eq. (11)
  Inject boundary conditions using eq. (2)
  Determine reward value based on state-action update
  Update Q-table
  Determine the GB position from population
  iter = iter + 1
end while
return GB position

In this equation, N represents the total number of agents, L stands for the longest diagonal length in the search space, D denotes the dimension of the search space, X_i^h is the value of agent i at dimension j, and X_j refers to the mean value of all QLESMO agents at dimension j. It is important to note that the indicators' values fall within the range of [0,1] and are further divided into three categories: L (0 to 0.333), M (0.334 to 0.666), and H (0.667 to 1). By categorising states and actions in this way, QLESMO can efficiently explore the search space and handle a wide range of real-world problems with robustness. The pseudocode of QLESMO algorithm combined with the proposed feature selection methodology as the fitness function evaluator, equation (11), is given in Algorithm 3.

B. CNN MODEL

1D Convolutional Neural Networks (1DCNNs) are a type of neural network commonly used in machine learning applications for processing one-dimensional data, such as time series or audio signals.

1DCNNs work by applying a set of filters, also known as kernels, to the input data. The filters are typically small in size and slide along the input data, computing the dot product between the filter weights and a small segment of the input data at each position. The result of each convolution operation is then passed through an activation function, such as the rectified linear unit (ReLU), to introduce nonlinearity into the model.

The output of each convolution operation is often referred to as a feature map, and multiple feature maps can be produced by using multiple filters. These feature maps are then typically subsampled, often using a technique called max pooling, which involves taking the maximum value within a small window of the feature map. This downsamples the feature maps and can help to reduce the dimensionality of the input data.

Finally, the resulting feature maps are typically flattened and passed through one or more fully connected layers, which can be used to produce a final output, such as a classification or regression prediction.

In a Convolutional Neural Network (CNN), the core operation is the convolution, which can be expressed mathematically as:

$$Y(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i+m, j+n) \cdot K(m,n) \tag{14}$$

In this equation, $Y(i,j)$ represents the output feature map value at position (i,j), $X(i+m, j+n)$ is the input feature map value at position $(i+m, j+n)$, $K(m,n)$ is the convolution kernel (or filter) with its associated parameters, and b is the bias term. The double summation computes the weighted sum of the input values within a local receptive field, where M and N denote the dimensions of the kernel.

The convolution operation involves sliding the kernel over the input feature map and computing the weighted sum at each position to produce the output feature map. The bias term b is added to introduce a bias shift in the output values.

The output of the convolution operation is a sequence of values, one for each position in the input signal. The resulting sequence can then be passed through an activation function, such as the ReLU function to introduce nonlinearity

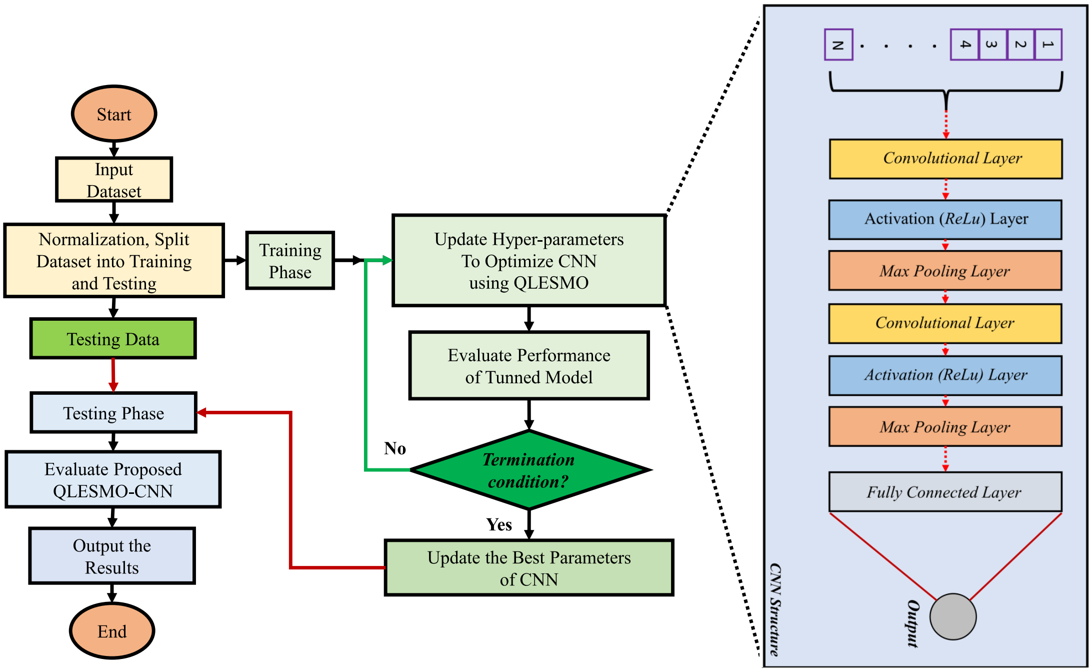


FIGURE 5. QLESMO-based CNN model for classification.

into the model.

$$ReLU(x) = \max(0, x) \tag{15}$$

The mathematical equation for max pooling can be expressed as follows:

$$p_i = \max_{j=1}^{S} y_{i+(j-1)S}, \tag{16}$$

where S is the stride of the pooling operation. 1D Convolutional Neural Networks (1DCNNs) have several benefits when it comes to classification tasks. Firstly, 1DCNNs are able to learn features automatically from the input data. This can be particularly useful when working with time series data, where it can be difficult to identify relevant features by hand. By using 1DCNNs, the network can learn to identify relevant features directly from the input signal, without the need for feature engineering. Secondly, 1DCNNs are able to capture temporal dependencies within the input data. This is important in classification tasks where the order of the input data matters. For example, when classifying speech signals, the order of the audio samples is critical in identifying the spoken word. Thirdly, 1DCNNs are capable of producing highly accurate results in classification tasks, often outperforming other machine learning algorithms. This is due in part to their ability to learn complex features from the input data, as well as their ability to capture temporal dependencies. The input to the 1D Convolutional Neural Network (CNN) comprises mental stress data features, preprocessed and selected for features by the QLESMO. The 1D CNN employs convolutional layers with rectified linear unit (ReLU) activation to autonomously extract temporal features from the sensor data. Each convolutional layer uses a filter size of 3, a hyperparameter meticulously tuned by QLESMO to enhance the CNN's efficiency. Subsequent to convolution, max pooling layers are applied to condense the feature maps and diminish their dimensions. These feature maps are then flattened and input into fully connected layers for the purpose of classification. Given the binary nature of the classification problem (stressed versus not stressed), binary cross-entropy loss is adopted as the appropriate loss function. The Adam optimiser is employed to iteratively adjust the CNN weights, aiming to minimise the loss. Additionally, batch normalisation is strategically placed after each convolutional layer, aiding in the normalisation of activations and promoting convergence. Notably, no dropout is integrated into the architecture, a decision informed by the fine-tuning of regularisation hyperparameters through QLESMO. The quantity of filters and units in each layer stands as critical hyperparameters, meticulously optimised by QLESMO to derive the optimal CNN architecture.

However, there are also some potential demerits when it comes to hyperparameter tuning in 1DCNNs. One potential challenge is tuning the number and size of the filters used in the network. Choosing the wrong number or size of filters can lead to overfitting or underfitting the data, resulting in poor performance on new, unseen data. Another potential challenge is selecting the appropriate activation function for the network. While ReLU is a popular choice for many applications, other activation functions, such as sigmoid or tanh, may be more appropriate in certain situations. Finally, tuning the stride length and padding used in the network can also be important for achieving optimal performance.

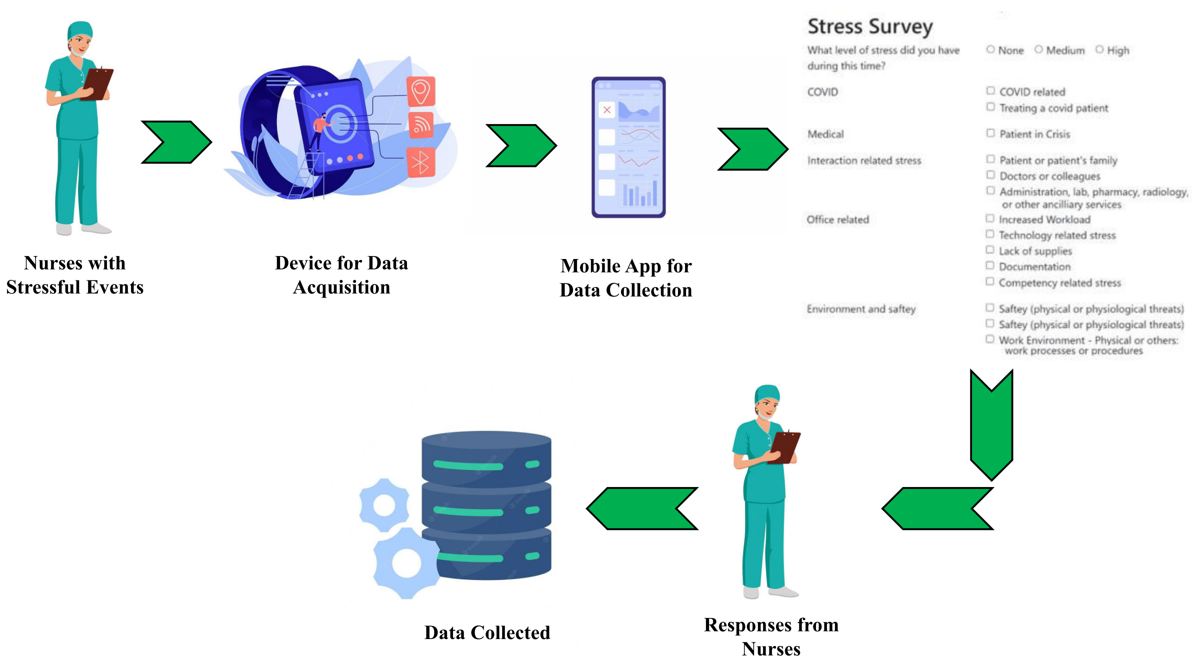


FIGURE 6. Procedure of data collection for mental stress in nurses.

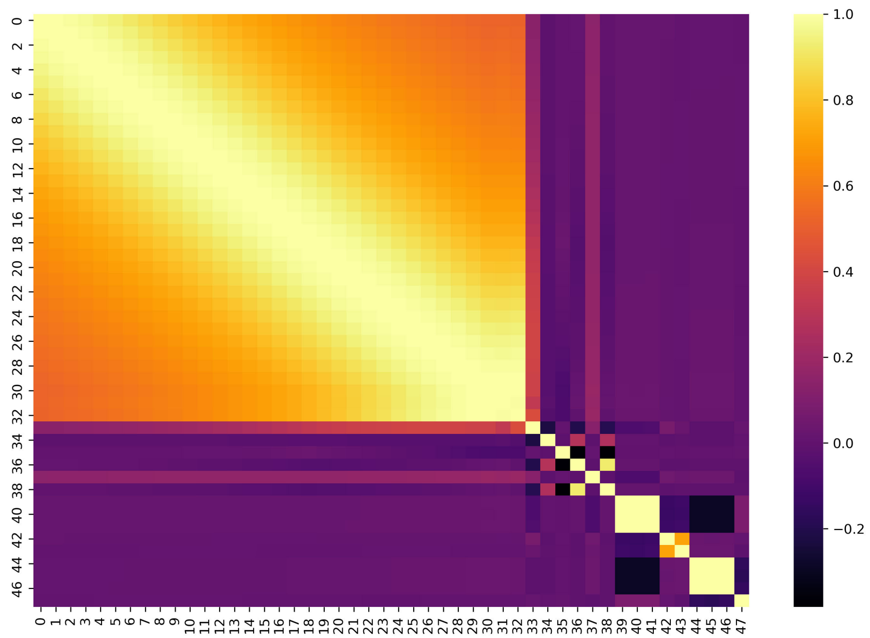


FIGURE 7. Mental stress dataset (47 features) correlation heatmap.

Choosing the wrong stride length or padding can result in loss of important features or information, leading to poor performance on classification tasks.

Metaheuristic optimisation algorithms have been found to be effective for hyperparameter tuning in 1DCNNs, as they are able to efficiently search through the vast space of potential hyperparameter values and find optimal solutions. Unlike traditional grid search or random search methods, metaheuristic algorithms are able to intelligently explore the space of hyperparameters and find optimal values that may not have been considered otherwise.

C. QLESMO BASED CNN FOR CLASSIFICATION

One of the main demerits of the CNN architecture is that it involves a large number of hyperparameters, including the size of the filters, the number of filters, and the learning

TABLE 2. Hyperparameters for the proposed QLESMO tuned CNN model.

Model Name	Hyper Parameters	Value
Convolutional Layers	No. of Units in CNN Layer	150
	Filter size in each Layer	3
QLESMO	No. of agents	30
	Algorithm Hyper Parameter	100
Learning Configuration	Learning Rate	10^{-2}
	Dropout Rate	NA

rate for the optimiser. Tuning these hyperparameters can be a time-consuming and challenging process, requiring extensive trial and error experimentation.

To address this challenge, metaheuristic optimisation algorithms can be used to effectively tune the hyperparameters of the CNN architecture. These algorithms are designed to search for optimal hyperparameters efficiently by exploring the hyperparameter space using a combination of heuristic techniques and mathematical optimisation methods. In this work, we employed the QLESMO algorithm to tune the hyperparameters of the CNN architecture. Figure 5 shows the model flow diagram for the proposed QLESMO baed CNN technique. Additionally, Table 2 shows the model parameters for the QLESMO tuned CNN network.

The training process begins by utilising the training set to iteratively train the weights of the Convolutional Neural Network (CNN). To enhance the efficiency and effectiveness of this training, it's conducted in batches, involving smaller subsets of the training samples. This batch-wise approach significantly contributes to the model's ability to generalise well to unseen data. For each batch during the training phase, a series of steps are executed. Firstly, the batch is propagated forward through the layers of the CNN, producing predictions. Subsequently, a loss is computed by comparing these predictions to the true labels of the training data. Gradients are then calculated using backpropagation, which provides information about the rate at which the loss changes with respect to the weights and biases. The weights of the network are updated to minimise the loss using the Adam optimiser and a specific learning rate. This process is vital for refining the model's predictive capabilities. Multiple epochs, which involve passing the entire training set through the network, are employed to iteratively train the CNN, allowing the model to learn from the data over multiple rounds.

Following each epoch, the model's performance is evaluated using a validation set. The QLESMO (Quantum Leap Enhanced Sequential Model Optimisation) algorithm is utilised to optimise hyperparameters such as the learning rate, number of units, activation functions, and more. This fine-tuning helps enhance the performance of the model on the validation set. The model exhibiting the best performance on the validation set is selected and further assessed on a separate test set. Diverse performance metrics, including accuracy, precision, recall, and F1-score, are meticulously calculated on this test set. These metrics provide a comprehensive understanding of the model's performance and efficacy in making predictions and classifications.

V. RESULTS

A. DATASET: MENTAL STRESS DETECTION

Data were obtained for about seven days from fifteen female nurses who were on regular shifts at a medical center. The age of the nurses varied from 30 to 55 years. Throughout two stages, data was obtained from 04/14/2020 to 12/11/2020, amounting to 1,250 hours of data. Phase-I ran from 4/15/2020 to 8/6/2020, and Phase II ran from 10/08/2020 to 12/11/2020. Nurses who were following their usual work schedules were the participants recruited for the research. Wearable devices were utilised to continuously track the physiological signals with minimal interference during regular shifts. The procedure for the data collection process is illustrated in Figure 6. The objective of this study was to determine the nurses' stress levels during their everyday routine using wearables. To better highlight the distribution of the characteristics, Figure 7 shows a heatmap that depicts the correlation between all of the features in the dataset.

To ensure minimal data loss, a Empatica E4 device was placed on the dominant arm's wrist, and participants were instructed to keep their phone nearby and wear the device in close contact with their skin. The study only included signals obtained continuously during the nurse's eight-hour work shift. The nurse could stop data collection by turning off the device, and any missing data during the shift was discarded, which could result in the absence of Inter-Beat Interval (IBI) and Heart Rate data due to PPG sensor noise artifacts. No missing data from the EDA sensor was observed [49]. Figure 6 shows the framework for the dataset collection. Furthermore, once the nurses registered as experiences stress, a survey with various questions pertaining to present conditions was presented and recorded. The dataset itself contains a total of 12445 entries from data recorded from 15 nurses over the span of their 8-hour work shifts. The complete set of survey questions and bio-markers make up 47 features for the division of stress and non-stressed instances. The complete dataset is available on Dryad [50].

B. FEATURE SELECTION RESULTS

The experimental results in this section showed that our proposed strategy that focuses on a larger set of parameters was able to effectively and efficiently solve the multi-objective optimisation problem related to mental stress detection dataset. The results demonstrate that our strategy outperformed the baseline approaches and was able to find solutions that were closer to the optimal solution, while also taking into account the practicality and realism of the solutions. In particular Variance based FS proves to be a far more effective strategy for feature selection than the traditional KNN based approach.

The Feature selection algorithm is run 30 times using each metaheuristic algorithm in comparison. To draw a fair comparison of the technique each algorithm is run for 10 iterations and a population size of 30 agents is used.

TABLE 3. Average results of 30 runs with 10 iterations and 30 population size of agents.

S.No.	FS Type	Metric	Algorithms						
			QLESMO	SMO	AOA	GWO	MFA	PSO	WOA
1	Variance	Accuracy	**0.949549**	0.915284	0.915457	0.923033	0.874247	0.936119	0.913964
2		S.D	**0.00671**	0.034731	0.023859	0.035438	0.021251	0.023288	0.039367
3		Precision	**0.949827**	0.915459	0.915879	0.923026	0.874481	0.936327	0.913905
4		F1-Score	**0.94965**	0.915291	0.91562	0.922982	0.874296	0.936138	0.913776
5		Time	**21.78802**	38.76465	25.05564	24.97033	52.69665	23.41998	24.67906
6		No. of features	**8.857143**	10.42857	14.14286	14	34.85714	9.714286	17.42857
7	KNN	Accuracy	**0.942375**	0.807783	0.802215	0.831889	0.869311	0.8506	0.763301
8		S.D	**0.011926**	0.136611	0.123064	0.097142	0.012921	0.109067	0.118564
9		Precision	**0.942595**	0.804161	0.798416	0.831048	0.869503	0.849152	0.758794
10		F1-Score	**0.94246**	0.804966	0.799277	0.83137	0.869266	0.849445	0.759798
11		Time	**234.5026**	430.255	251.5114	253.8618	2295.545	247.2163	251.2514
12		No. of features	**10.71429**	11.14286	14.57143	21.71429	35.28571	14.14286	18.42857

TABLE 4. Best results of 30 runs with 10 iterations and 30 population size of agents.

S.No.	FS Type	Metric	Algorithms						
			QLESMO	SMO	AOA	GWO	MFA	PSO	WOA
1	Variance	Accuracy	**0.959823**	0.953797	0.955002	0.956609	0.900763	0.956207	0.955002
2		Precision	**0.959801**	0.954027	0.955081	0.956828	0.900928	0.956241	0.954879
3		F1-Score	**0.959809**	0.953894	0.955009	0.9567	0.900834	0.956185	0.95493
4		Time	**20.37101**	37.21625	23.86199	24.0896	50.20213	22.50135	22.12482
5		No. of features	**6**	9	14	14	30	8	**6**
6	KNN	Accuracy	**0.95902**	0.946967	0.948172	0.9546	0.887103	0.955404	0.952591
7		Precision	**0.959115**	0.947114	0.94796	0.954774	0.887748	0.95527	0.952686
8		F1-Score	**0.95906**	0.947003	0.948044	0.95465	0.8874	0.95533	0.952615
9		Time	**211.1683**	407.2095	245.3324	240.9927	1675.444	234.6942	241.4366
10		No. of features	**7**	9	14	14	32	11	**7**

The average results of the 30 runs of the metrics used are shown in Table 3 while Table 4 depict the best result obtained from the 30 runs. As the data suggests it can be devised that the proposed QLESMO algorithm was able to find global minima solution far better than the conventional algorithms in comparison.

In this stage, feature vectors from various windows are organised sequentially to create a 2D input sample for the 1D Convolutional Neural Network (CNN), where each row corresponds to a distinct time window. This stacking process encapsulates the temporal information within the data. Prior to feeding this input into the CNN, standardisation is carried out by subtracting the mean and dividing by the standard deviation, effectively normalising the data distribution. This standardised input essentially functions as a pseudo-timeseries, preserving temporal structure while enhancing the informativeness of the features. During the training phase, the 1D CNN model is trained using batches of these standardised input samples. When it comes to testing, a similar set of preprocessing and feature extraction steps is applied to previously unseen data, converting them into standardised input samples compatible with the CNN model for evaluation and prediction.

C. BENCHMARK TEST FUNCTION RESULTS

In this section the set of simulation results is performed on 10 benchmark test functions (taken from CEC'20 test

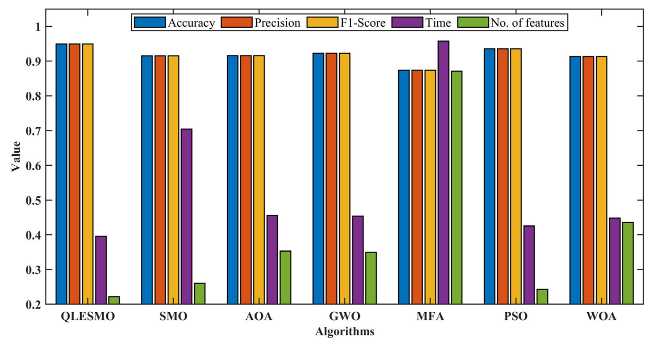


FIGURE 8. Variance average.

suite [51]) with QLESMO using 30 agents of population size, and 100 iterations run 30 times. The convergence curve comparison for the algorithms can be seen in Figure 10. Detailed analysis of each function against different test functions is prepared. Referring to Table 5, QLESMO has gained the far most performance by finding the global minima solution better than other compared optimisers. The best results are written in bold for better visualisation of each test function. Results indicates that QLESMO algorithm using Q-learning is far better at balancing between exploitation and exploration phases compared to other meta-heuristic algorithms.

TABLE 5. Best/Worst/Average/STD achieved by algorithms for standard functions in 100 iterations with 30 population size in 30 runs.

Fun.	Params	QLESMO	MFA	AOA	WOA	GWO	PSO
F1	Best	1.56e-104	3.60e-80	8.06e-41	9.98e-01	3.86e+03	1.63e+05
	Worst	1.76e-103	3.14e-78	2.71e-26	7.18e+00	5.46e+03	2.71e+05
	Avg	4.43e-98	6.12e-64	3.22e-04	4.99e+02	2.83e+03	1.12e+05
	Std	1.23e-98	1.16e-64	1.05e-04	2.70e+02	8.43e+03	3.98e+05
	Rank	1	2	3	4	5	6
F2	Best	2.05e-121	3.19e-41	6.98e-54	1.69e-01	2.47e+01	6.91e+00
	Worst	3.15e-108	1.51e-38	8.90e-52	6.23e-01	3.35e+01	9.74e+00
	Avg	1.98e-72	6.56e-35	6.94e-47	5.08e+00	9.36e+00	1.93e+00
	Std	3.73e-73	2.76e-35	1.43e-47	4.54e+00	4.17e+01	1.13e+01
	Rank	1	3	2	4	6	5
F3	Best	4.80e-81	5.08e-78	5.41e-19	2.82e+01	7.28e+03	2.22e+03
	Worst	3.53e-74	1.18e-66	1.06e-03	2.02e+02	9.62e+03	3.41e+03
	Avg	2.99e-66	1.85e+00	3.17e-02	5.69e+03	7.40e+03	1.67e+03
	Std	6.04e-67	3.38e-01	2.21e-02	4.62e+03	1.68e+04	5.04e+03
	Rank	1	2	3	4	6	5
F4	Best	2.97e-55	1.26e-40	6.00e-17	1.38e+00	2.01e+01	1.08e+01
	Worst	1.63e-53	4.29e-38	1.21e-03	2.33e+00	2.72e+01	1.23e+01
	Avg	1.09e-49	4.16e-34	2.06e-02	8.48e+00	5.90e+00	3.57e+00
	Std	2.34e-50	1.04e-34	3.57e-02	8.58e+00	3.34e+01	1.62e+01
	Rank	1	2	3	4	5	6
F5	Best	0.93e-06	1.11e-05	1.79e-06	1.26e-02	3.40e-01	3.86e-02
	Worst	2.85e-05	6.78e-05	2.26e-05	2.62e-02	1.25e+00	1.02e-01
	Avg	1.63e-04	2.85e-04	3.25e-04	2.20e-01	2.37e+00	7.12e-02
	Std	1.84e-04	3.30e-04	3.10e-04	1.67e-01	3.06e+00	1.64e-01
	Rank	1	3	2	4	6	5
F6	Best	2.17e-36	1.57e-32	6.73e-01	1.12e+00	8.52e+01	4.84e+00
	Worst	3.01e-31	1.55e-32	7.28e-01	1.62e+00	1.87e+04	7.08e+00
	Avg	5.56e-47	6.65e-10	4.67e-02	2.64e+05	9.85e+05	1.09e+01
	Std	1.60e-32	1.22e-10	7.77e-01	5.37e+04	5.65e+05	1.41e+01
	Rank	1	2	3	4	6	5
F7	Best	3.08e-04	3.11e-04	4.37e-04	7.89e-04	3.39e-04	5.25e-04
	Worst	3.08e-04	3.43e-04	5.02e-04	1.55e-03	1.00e-03	7.39e-04
	Avg	1.07e-05	2.78e-04	2.74e-02	1.41e-02	2.48e-02	1.31e-03
	Std	3.16e-04	5.22e-04	1.77e-02	1.49e-02	1.45e-02	1.34e-03
	Rank	1	2	4	5	3	6
F8	Best	1.08e-32	1.34e-32	2.74e+00	3.36e+00	2.68e+05	2.24e+01
	Worst	1.14e-32	1.38e-32	2.82e+00	4.15e+00	8.89e+05	4.61e+01
	Avg	7.04e-48	5.56e-08	7.13e-02	5.49e+05	6.13e+06	3.51e+03
	Std	1.84e-38	1.34e-22	2.87e+00	1.08e+05	6.68e+06	1.40e+03
	Rank	1	2	3	4	6	5
F9	Best	-3.98e+00	-3.88e+00	-3.86e+00	-3.83e+00	-3.88e+00	-3.86e+00
	Worst	-3.96e+00	-3.82e+00	-3.85e+00	-3.89e+00	-3.88e+00	-3.88e+00
	Avg	1.91e-02	8.15e-02	1.32e-02	9.66e-02	5.6060e-02	1.15e-04
	Std	-3.84e+00	-3.78e+00	-3.83e+00	-3.70e+00	-3.84e+00	-3.86e+00
	Rank	1	2	3	4	5	6
F10	Best	-1.03e+01	-1.03e+01	-7.24e+00	-9.26e+00	-1.01e+01	-1.01e+01
	Worst	-1.01e+01	-1.01e+01	-4.73e+00	-5.28e+00	-1.00e+01	-1.02e+01
	Avg	1.42e-05	2.69e-01	1.37e+00	2.27e+00	3.33e+00	3.56e+00
	Std	-1.01e+01	-1.01e+01	-3.67e+00	-2.76e+00	-5.07e+00	-5.87e+00
	Rank	1	2	3	4	5	6

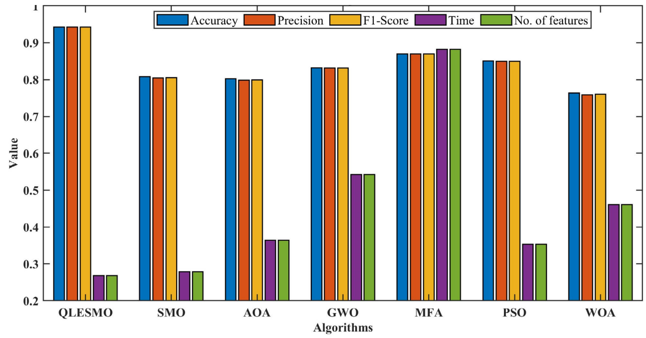


FIGURE 9. KNN average.

D. CLASSIFICATION RESULTS

Based on selected features, as shown in Table 6, the four deep learning models also achieved promising classification results for mental stress detection. QLESMO-CNN, MFA-CNN, AOA-CNN, and GWO-CNN achieved accuracy of 98.67 %, 97.59 %, 95.50 %, and 94.74 %, respectively. These results suggest that the selected features contain sufficient information for the models to accurately detect mental stress. Notably, the models achieved slightly lower accuracy compared to using the full feature set, but the results are still considered effective for mental stress detection. Precision, recall, accuracy, specificity and F1 score comparison between competing techniques is presented in Figure 11.

Based on the full features of the dataset, as shown in Table 6, all four models also showed promising results for the classification of mental stress detection. QLESMO-CNN achieved an accuracy of 96.95 %, MFA-CNN achieved an accuracy of 95.26 %, AOA-CNN achieved an accuracy of 94.17 %, and GWO-CNN achieved an accuracy of 91.52 % on the mental stress dataset. Precision, recall, accuracy, specificity and F1 score comparison between competing techniques is presented in Figure. 12.

Using selected features can also help to reduce the computational complexity of the models and improve their efficiency without significantly affecting their performance. Therefore, selecting important features is a viable strategy to consider when working with large datasets. The classification results demonstrate the potential of deep learning models, including QLESMO-CNN, MFA-CNN, AOA-CNN, and GWO-CNN, in accurately detecting mental stress, both with the full feature set and with selected features.

VI. DISCUSSION

The optimisation strategy in this work is demonstrated on three levels. First, the proposed QLESMO algorithm is utilised to address the challenge of feature selection for wearable-based mental stress detection. Second, the QLESMO algorithm is tested on and compared with competitive metaheuristic algorithms for 10 CEC benchmark test function. Finally, a QLESMO based CNN architecture is devised which performs classification for the dataset employed in this study i.e. the stresses observed in nurses working during the Covid-19 pandemic. In each use case, it is observed that the integration of reinforcement learning, metaheuristic algorithms and CNN architecture to develop the QLESMO-CNN technique outperforms other comparative techniques in terms of the evaluation metrics used in this study.

For the domain of feature selection a new approach is devised in which the fitness function is calculated using the dataset variance minimisation as compared to the traditional KNN accuracy maximisation method. This strategy is seen to provide optimal features 10 times faster and with comparably higher accuracy as seen in Table 3 and Table 4. The analysis with the benchmark test functions further incorporates the idea that QLESMO algorithm excels in striking a balance between exploitation and exploration phases, a crucial aspect in optimisation problems, and its superiority over other meta-heuristic algorithms in terms of convergence.

Utilising the QLESMO-CNN architecture, the task of early mental stress detection is classified. QLESMO-CNN appeared to outperform other models across various metrics. Additionally, using selected features, as shown in Table 6, can enhance computational efficiency, making it a practical choice for large datasets. Classifying mental stress using a deep learning model is instrumental in achieving early detection, which is pivotal for timely intervention and support. These models provide objective assessments, can screen large populations, offer continuous monitoring, optimise resource allocation, and generate valuable data-driven insights into stress factors. By leveraging technology in this way, we can proactively address mental stress, mitigate its impact, and promote better mental well-being for individuals.

Insights into the State-of-the-Art landscape of mental stress detection approaches are provided by the comparison analysis, which is shown in Table 7. It highlights the complexity of the issue by showcasing a variety of techniques, from wearables and surveys to EEG signals. When various approaches are compared, our suggested model stands out with an accuracy of 0.9867, outperforming the other techniques. This shows that the QLESMO-CNN technique performs effectively, making it a potential development in the recognition of mental stress. It considers the most recent advancements in the area and can potentially provide a more reliable method for spotting early signs of mental stress.

The model selection process opted in this study does not adopt the cross validation technique due to the dataset's temporal dependencies and reliability. When performing cross-validation, these structures in datasets are regularly ignored, resulting in serious underestimation of predictive error [58]. Furthermore, at each run of the feature selection process, the model instances are randomised so as to produce a generalised reduced feature subset that represents the dataset in a more robust and computationally cost-effective manner.

The model selection process opted in this study does not adopt the cross validation technique due to the dataset's temporal dependencies and reliability. When performing

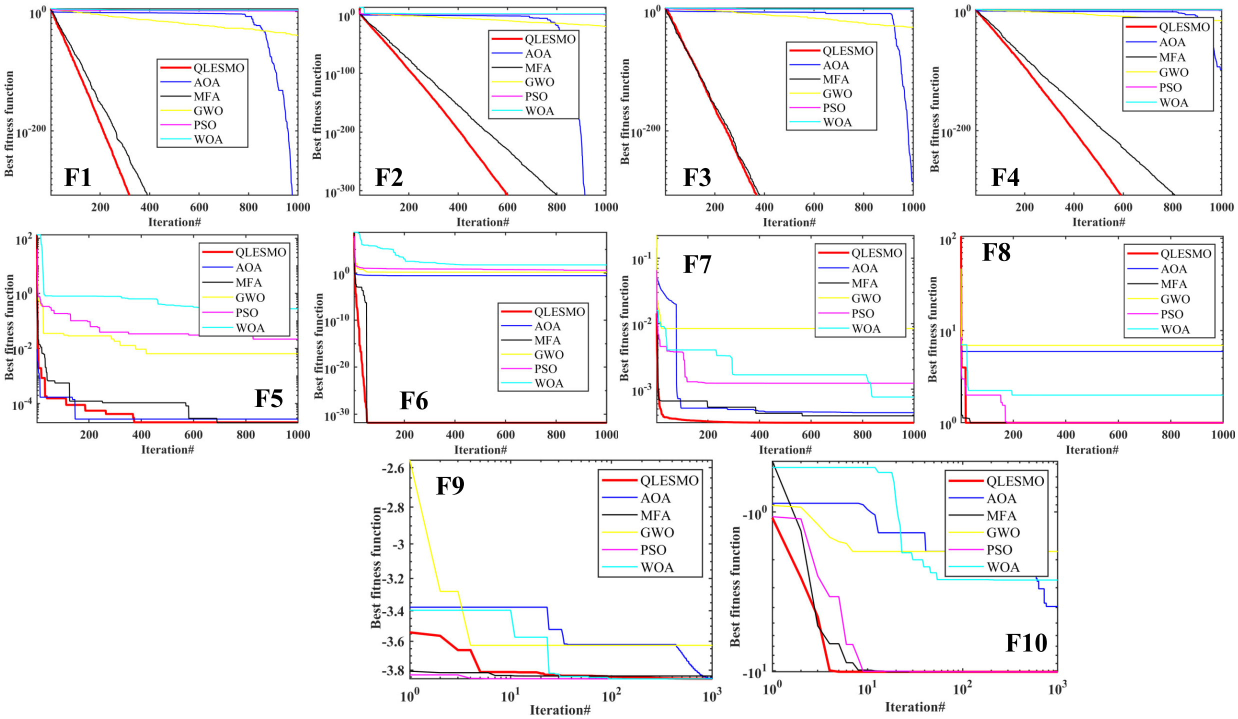


FIGURE 10. Cost vs iterations comparison on standard functions.

TABLE 6. Evaluation metrics comparison with selected and full set features.

Technique	With Selected Features					With Full Features				
	Precision	Recall	Accuracy	Specificity	F1score	Precision	Recall	Accuracy	Specificity	F1score
QLESMO-CNN	0.9848	0.9821	0.9867	0.9932	0.9834	0.962	0.9618	0.9695	0.9849	0.9619
MFA-CNN	0.9705	0.9662	0.9759	0.988	0.9683	0.9409	0.9484	0.9526	0.9766	0.9443
AOA-CNN	0.9427	0.9463	0.955	0.9779	0.9444	0.9261	0.9363	0.9417	0.9718	0.9308
GWO-CNN	0.9396	0.9317	0.9474	0.9731	0.9354	0.9068	0.8903	0.9152	0.9559	0.8975

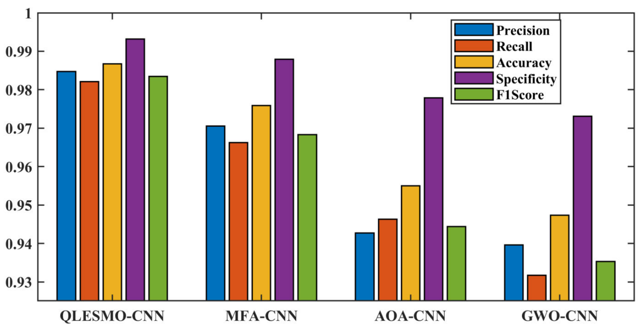


FIGURE 11. Bar chart comparison for evaluation metrics with selected features.

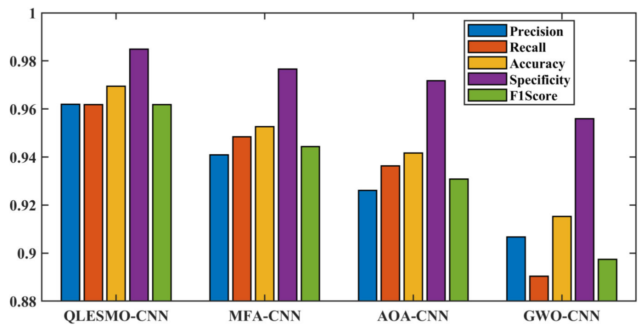


FIGURE 12. Evaluation metrics comparison with full features.

cross-validation, these structures in datasets are regularly ignored, resulting in serious underestimation of predictive error [58]. On the context of this study, at each run of the feature selection process, the model instances are randomised so as to produce a generalised reduced feature subset that represents the dataset in a more robust and computationally cost-effective manner. Furthermore, by using a separate validation set, we aimed to ensure that our model's performance was rigorously tested on unseen data, which we believe is crucial for the reliability of our findings.

The perception of stress levels might vary across individuals. Determining the absence of other contributing elements, whether personal or professional, in the nurse's assessment of a stress-related event is a challenging task. Moreover, considering that the dataset was obtained from nurses under an abnormally elevated workload, it may be necessary to adapt the model's conclusions to typical user case

TABLE 7. Comparative analysis of proposed technique with the state-of-the-art.

Reference	Year	Dataset Conditions	Dataset Collection	Underlying Methodology	Evaluation Metrics
[52]	2019	Students of Jaypee Institute of Info. Tech. before exams and due to usage of the internet	Survey	KNN, Naïve Bayes, RF, SVM	Accuracy : 0.8571 Recall : 0.7500 Specificity: 1.0
[53]	2022	Mental arithmetic tasks	EEG Signals	WOA-SVM	Accuracy : 0.9726 Precision : 0.9627 F1-Score : 0.9723 Recall : 0.9826 Specificity : 0.9932
[54]	2015	Trier Social Stress Test (TSST) with public speaking and cognitive tasks	Wearables	SVM	Accuracy : 0.8308 Precision : 0.8387
[55]	2016	The SWELL Knowledge Work (SWELL-KW) Dataset for Stress and User Modeling Research	Camera facial expressions, body postures, wearables	SVM, DT, ANN	Accuracy : 0.9003
[56]	2020	drivedb - Physionet Dataset (COVID-19 pandemic related stress)	ECG signals	Cubic SVM using Gaussian Kernel	Accuracy : 0.983
[57]	2022	Mental calculations, and sound interference	Wearables	CNN	Accuracy: 0.827 Sensitivity: 0.970 Specificity: 0.690
Our Model	2023	Nurses working 8-hour shifts during Covid-19 Pandemic	Wearables, Survey	QLESMO-CNN	Accuracy : 0.9867 Precision : 0.9848 F1-Score : 0.9834 Recall : 0.9821 Specificity : 0.9932

settings. One potential area of attention for academics is the development of transfer learning methodologies that might enhance the adaptability of models to particular datasets. The preservation of data privacy and security is a significant problem when dealing with the acquisition of information on a massive scale. In addition, it is essential to address ethical issues pertaining to factors such as maintaining anonymity, processing data, and ensuring secure storage.

VII. CONCLUSION AND FUTURE WORK

Stress is a common experience that affects individuals of all ages, and managing it effectively can significantly improve overall well-being. Early detection of stress is essential to prevent chronic stress-related health problems, reduced productivity, and a decreased quality of life. The proposed feature selection technique using the Q-Learning Embedded Starling Murmuration Optimiser (QLESMO) algorithm has shown promising results in identifying the most relevant and informative features that can differentiate between stressed and non-stressed individuals, specifically in the context of stresses experienced by nurses during the Covid-19 pandemic.

The variance reduction-based feature selection technique used in this study outperformed the traditional KNN-based feature selection approach. The QLESMO algorithm embedded with q-learning showed superior performance compared to other feature selection methods, as demonstrated through the benchmark test functions. The reduced feature subset obtained through the proposed algorithm was then classified using the QLESMO-CNN model, which further validates the effectiveness of the proposed approach.

The results of this study demonstrate that the QLESMO algorithm is a robust and efficient feature selection method that can be applied to a variety of stress detection tasks. Furthermore, the proposed methodology has the potential to aid in the development of automated stress detection systems that can improve the accuracy and speed of detecting stress in real-time. Future work could encompass longitudinal studies for prolonged algorithm validation, integration of multiple data modalities for enhanced accuracy and real-world deployment testing. Additionally, the algorithm's applicability in other domains, its interpretability, and ethical considerations should be explored to ensure responsible and impactful stress management solutions. Furthermore, considering the potential of transfer learning for cross-validation studies across different stress contexts could provide valuable insights into the algorithm's generalisability.

REFERENCES

[1] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: Promise and potential," *Health Inf. Sci. Syst.*, vol. 2, no. 1, pp. 1–10, Dec. 2014.

[2] A. Kishor, C. Chakraborty, and W. Jeberson, "Intelligent healthcare data segregation using fog computing with Internet of Things and machine learning," *Int. J. Eng. Syst. Model. Simul.*, vol. 12, nos. 2–3, pp. 188–194, 2021.

[3] E. J. Topol, "High-performance medicine: The convergence of human and artificial intelligence," *Nature Med.*, vol. 25, no. 1, pp. 44–56, Jan. 2019.

[4] S. Balaji, F. Sanfilippo, M. W. Gerdes, and D. Prattichizzo, "A perspective on intervention approaches for children with autism spectrum disorder," in *Intelligent Technologies and Applications*. Cham, Switzerland: Springer, 2022, pp. 132–143.

[5] L. J. Wali and F. Sanfilippo, "A review of the state-of-the-art of assistive technology for people with ASD in the workplace and in everyday life," in *Digital Transformation for a Sustainable Society in the 21st Century*. Cham, Switzerland: Springer, 2019, pp. 520–532.

[6] F. Sanfilippo and K. Raja, "A multi-sensor system for enhancing situational awareness and stress management for people with ASD in the workplace and in everyday life," in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, Maui, HI, USA, 2019, pp. 4079–4086.

[7] T. Zhang, A. El Ali, C. Wang, A. Hanjalic, and P. Cesar, "CorrNet: Fine-grained emotion recognition for video watching using wearable physiological sensors," *Sensors*, vol. 21, no. 1, p. 52, Dec. 2020.

[8] S. Gedam and S. Paul, "A review on mental stress detection using wearable sensors and machine learning techniques," *IEEE Access*, vol. 9, pp. 84045–84066, 2021.

[9] H. Yu, E. B. Klerman, R. W. Picard, and A. Sano, "Personalized wellbeing prediction using behavioral, physiological and weather data," in *Proc. IEEE EMBS Int. Conf. Biomed. Health Informat. (BHI)*, May 2019, pp. 1–4.

[10] R. Alonzo, J. Hussain, S. Stranges, and K. K. Anderson, "Interplay between social media use, sleep quality, and mental health in youth: A systematic review," *Sleep Med. Rev.*, vol. 56, Apr. 2021, Art. no. 101414.

[11] Z. Shahbazi and Y.-C. Byun, "Early life stress detection using physiological signals and machine learning pipelines," *Biology*, vol. 12, no. 1, p. 91, Jan. 2023.

[12] G. Vos, K. Trinh, Z. Sarnyai, and M. R. Azghadi, "Generalizable machine learning for stress monitoring from wearable devices: A systematic literature review," 2022, *arXiv:2209.15137*.

[13] F. Sanfilippo and K. Y. Pettersen, "A sensor fusion wearable health-monitoring system with haptic feedback," in *Proc. 11th Int. Conf. Innov. Inf. Technol. (IIT)*, Nov. 2015, pp. 262–266.

[14] F. Sanfilippo and C. Pacchierotti, "A wearable haptic system for the health monitoring of elderly people in smart cities," *Int. J. Online Eng.*, vol. 14, no. 8, pp. 1–15, 2018.

[15] K. N. Mishra and C. Chakraborty, "A novel approach toward enhancing the quality of life in smart cities using clouds and iot-based technologies," in *Digital Twin Technologies and Smart Cities* (Internet of Things). Cham, Switzerland: Springer, 2020, pp. 19–35, doi: 10.1007/978-3-030-18732-3_2.

[16] E. Mehmood and T. Anees, "Challenges and solutions for processing real-time big data stream: A systematic literature review," *IEEE Access*, vol. 8, pp. 119123–119143, 2020.

[17] G. Shin, M. H. Jarrahi, Y. Fei, A. Karami, N. Gafinowitz, A. Byun, and X. Lu, "Wearable activity trackers, accuracy, adoption, acceptance and health impact: A systematic literature review," *J. Biomed. Informat.*, vol. 93, May 2019, Art. no. 103153.

[18] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, May 2003.

[19] Y. Deng, Z. Wu, C.-H. Chu, and T. Yang, "Evaluating feature selection for stress identification," in *Proc. IEEE 13th Int. Conf. Inf. Reuse Integr. (IRI)*, Aug. 2012, pp. 584–591.

[20] F. Albertetti, A. Simalastar, and A. Rizzotti-Kaddouri, "Stress detection with deep learning approaches using physiological signals," in *IoT Technologies for HealthCare*. Cham, Switzerland: Springer, 2021, pp. 95–111.

[21] G. Schryen, "Parallel computational optimization in operations research: A new integrative framework, literature review and research directions," *Eur. J. Oper. Res.*, vol. 287, no. 1, pp. 1–18, Nov. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221719309440

[22] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks: Theory and Applications* (Studies in Computational Intelligence), vol. 780. Cham, Switzerland: Springer, 2019, pp. 43–55, doi: 10.1007/978-3-319-93025-1_4.

[23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Aug. 1995, pp. 1942–1948.

[24] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

[25] K. A. Dowsland and J. Thompson, "Simulated annealing," in *Handbook of Natural Computing*. Berlin, Germany: Springer, 2012, pp. 1623–1655, doi: 10.1007/978-3-540-92910-9_49.

[26] M. Gendreau and J.-Y. Potvin, "Metaheuristics in combinatorial optimization," *Ann. Oper. Res.*, vol. 140, no. 1, pp. 189–213, 2005.

[27] I. Fister Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," 2013, *arXiv:1307.4186*.

[28] A. K. Sangaiah, Z. Zhiyong, and M. Sheng, *Computational Intelligence for Multimedia Big Data on the Cloud With Engineering Applications*. New York, NY, USA: Academic Press, 2018.

[29] V. Tiwari, "Face recognition based on cuckoo search algorithm," *Image*, vol. 7, no. 8, p. 9, 2012.

[30] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, and X.-S. Yang, "BBA: A binary bat algorithm for feature selection," in *Proc. 25th SIBGRAPI Conf. Graph., Patterns Images*, Aug. 2012, pp. 291–297.

[31] G. Zhang, J. Hou, J. Wang, C. Yan, and J. Luo, "Feature selection for microarray data classification using hybrid information gain and a modified binary Krill herd algorithm," *Interdiscipl. Sci., Comput. Life Sci.*, vol. 12, no. 3, pp. 288–301, Sep. 2020.

[32] M. Allam and M. Nandhini, "Optimal feature selection using binary teaching learning based optimization algorithm," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 2, pp. 329–341, Feb. 2022.

[33] P. Agrawal, T. Ganesh, and A. W. Mohamed, "A novel binary gaining–sharing knowledge-based optimization algorithm for feature selection," *Neural Comput. Appl.*, vol. 33, no. 11, pp. 5989–6008, 2021.

[34] A. K. Abasi, A. T. Khader, M. A. Al-Betar, S. Naim, S. N. Makhadmeh, and Z. A. A. Alyasseri, "A text feature selection technique based on binary multi-verse optimizer for text clustering," in *Proc. IEEE Jordan Int. Joint Conf. Electr. Eng. Inf. Technol. (JEEIT)*, Apr. 2019, pp. 1–6.

[35] R. Sindhu, R. Ngadiran, Y. M. Yacob, N. A. H. Zahri, and M. Hariharan, "Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism," *Neural Comput. Appl.*, vol. 28, no. 10, pp. 2947–2958, Oct. 2017.

[36] J. P. Papa, A. Pagnin, S. A. Schellini, A. Spadotto, R. C. Guido, M. Ponti, G. Chiachia, and A. X. Falcão, "Feature selection through gravitational search algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 2052–2055.

[37] M. Kahana. *Black Kites Fly as a Murmuration of Starlings Tumbles in the Distance Near Rahat, Israel, on February 10, 2016*. The Atlantic. Accessed: Apr. 16, 2023. [Online]. Available: https://www.theatlantic.com/photo/2019/01/photos-murmurations-starlings/579286/#img04

[38] S. S. Kareem, R. R. Mostafa, F. A. Hashim, and H. M. El-Bakry, "An effective feature selection model using hybrid metaheuristic algorithms for IoT intrusion detection," *Sensors*, vol. 22, no. 4, p. 1396, Feb. 2022.

[39] A. Sarkar, M. Z. Khan, M. M. Singh, A. Noorwali, C. Chakraborty, and S. K. Pani, "Artificial neural synchronization using nature inspired whale optimization," *IEEE Access*, vol. 9, pp. 16435–16447, 2021.

[40] A. Das, S. Guha, P. K. Singh, A. Ahmadian, N. Senu, and R. Sarkar, "A hybrid meta-heuristic feature selection method for identification of Indian spoken languages from audio signals," *IEEE Access*, vol. 8, pp. 181432–181449, 2020.

[41] A. Dey, S. Chattopadhyay, P. K. Singh, A. Ahmadian, M. Ferrara, and R. Sarkar, "A hybrid meta-heuristic feature selection method using golden ratio and equilibrium optimization algorithms for speech emotion recognition," *IEEE Access*, vol. 8, pp. 200953–200970, 2020.

[42] Q. Hamad, H. Samma, and S. A. Suandi, "Q-learning based metaheuristic optimization algorithms: A short review and perspectives," Res. Square, Tech. Rep., 2023, doi: 10.21203/rs.3.rs-1950095/v1.

[43] F. Zhao, Q. Wang, and L. Wang, "An inverse reinforcement learning framework with the Q-learning mechanism for the metaheuristic algorithm," *Knowl.-Based Syst.*, vol. 265, Apr. 2023, Art. no. 110368.

[44] S. Sadeg, L. Hamdad, A. R. Remache, M. N. Karech, K. Benatchba, and Z. Habbas, "QBSO-FS: A reinforcement learning based bee swarm optimization metaheuristic for feature selection," in *Advances in Computational Intelligence*. Cham, Switzerland: Springer, 2019, pp. 785–796.

[45] Q. S. Hamad, H. Samma, S. A. Suandi, and J. Mohamad-Saleh, "Q-learning embedded sine cosine algorithm (QLESCA)," *Expert Syst. Appl.*, vol. 193, May 2022, Art. no. 116417.

[46] A. Seyyedabbasi, "A reinforcement learning-based metaheuristic algorithm for solving global optimization problems," *Adv. Eng. Softw.*, vol. 178, Apr. 2023, Art. no. 103411.

[47] H. Zamani, M. H. Nadimi-Shahraki, and A. H. Gandomi, "Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization," *Comput. Methods Appl. Mech. Eng.*, vol. 392, Mar. 2022, Art. no. 114616.

[48] H. Samma, J. Mohamad-Saleh, S. A. Suandi, and B. Lahasan, "Q-learning-based simulated annealing algorithm for constrained engineering design problems," *Neural Comput. Appl.*, vol. 32, no. 9, pp. 5147–5161, May 2020.

[49] S. Hosseini, R. Gottumukkala, S. Katragadda, R. T. Bhupatiraju, Z. Ashkar, C. W. Borst, and K. Cochran, "A multimodal sensor dataset for continuous stress detection of nurses in a hospital," *Sci. Data*, vol. 9, no. 1, p. 255, Jun. 2022.

[50] S. Hosseini, "A multi-modal sensor dataset for continuous stress detection of nurses in a hospital," *Sci. Data*, vol. 9, no. 1, p. 255, 2021, doi: 10.1038/S41597-022-01361-Y.

[51] J.-J. Liang, B. Qu, D. Gong, and C. Yue, "Problem definitions and evaluation criteria for the CEC 2019 special session on multimodal multiobjective optimization," Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep., 2019, doi: 10.13140/RG.2.2.33423.64164.

[52] R. Ahuja and A. Banga, "Mental stress detection in university students using machine learning algorithms," *Proc. Comput. Sci.*, vol. 152, pp. 349–353, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919306581

[53] L. D. Sharma, V. K. Bohat, M. Habib, A. M. Al-Zoubi, H. Faris, and I. Aljarah, "Evolutionary inspired approach for mental stress detection using EEG signal," *Expert Syst. Appl.*, vol. 197, Jul. 2022, Art. no. 116634. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422001233

[54] V. Sandulescu, S. Andrews, D. Ellis, N. Bellotto, and O. M. Mozos, "Stress detection using wearable physiological sensors," in *Artificial Computation in Biology and Medicine*, J. M. Ferrández Vicente, J. R. Álvarez-Sánchez, F. de la Paz López, F. J. Toledo-Moreo, and H. Adeli, Eds. Cham, Switzerland: Springer, 2015, pp. 526–532.

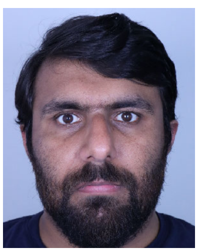
[55] S. Koldijk, M. A. Neerincx, and W. Kraaij, "Detecting work stress in offices by combining unobtrusive sensors," *IEEE Trans. Affect. Comput.*, vol. 9, no. 2, pp. 227–239, Apr. 2018.

[56] M. F. Rizwan, R. Farhad, and M. H. Imam, "Support vector machine based stress detection system to manage COVID-19 pandemic related stress from ECG signal," *AIUB J. Sci. Eng.*, vol. 20, no. 1, pp. 8–16, Apr. 2021.

[57] J. He, K. Li, X. Liao, P. Zhang, and N. Jiang, "Real-time detection of acute cognitive stress using a convolutional neural network from electrocardiographic signal," *IEEE Access*, vol. 7, pp. 42710–42717, 2019.

[58] D. R. Roberts, V. Bahn, S. Ciuti, M. S. Boyce, J. Elith, G. Guillera-Arroita, S. Hauenstein, J. J. Lahoz-Monfort, B. Schröder, W. Thuiller, D. I. Warton, B. A. Wintle, F. Hartig, and C. F. Dormann, "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure," *Ecography*, vol. 40, no. 8, pp. 913–929, Aug. 2017.

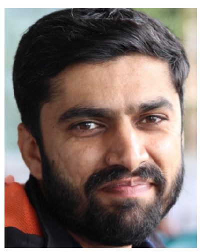
SYED KUMAYL RAZA MOOSAVI received the bachelor's degree in mechatronics engineering and the master's degree in artificial intelligence from the National University of Sciences and Technology, Islamabad, Pakistan, in 2016 and 2023, respectively. He will pursue the Ph.D. degree in robotics and AI. His current research interests include robotics, artificial intelligence, and swarm optimization.



MUHAMMAD HAMZA ZAFAR (Graduate Student Member, IEEE) received the bachelor's degree in electronics engineering and the master's degree in electrical engineering from the Capital University of Science and Technology, Islamabad, Pakistan, in 2015 and 2022, respectively. He is currently pursuing the Ph.D. degree in robotics and AI with the University of Agder, Grimstaad, Norway. His current research interests include robotics, artificial intelligence, and swarm optimization.

FILIPPO SANFILIPPO (Senior Member, IEEE) received the Ph.D. degree in engineering cybernetics from the Norwegian University of Science and Technology (NTNU), Norway, with a focus on alternative and flexible control approaches for robotic manipulators. He is currently a Professor with the Faculty of Engineering and Science, University of Agder (UiA), Grimstad, Norway. He is also an Adjunct Professor with the Faculty of Informatics, Kaunas University of Technology, Kaunas, Lithuania. He has vast experience in participating in European research programs and various national projects from the Research Council of Norway (RCN). He has authored and coauthored several technical papers in various journals and conferences. His current research interests include robotics, wearables, human–robot collaboration, collaborative robotics, artificial intelligence, and control theory. He is currently a member of the IEEE Region 8 Chapter Coordination Committee, the IEEE Public Visibility Committee, and the IEEE R8 Awards and Recognitions (A&R) Committee. He is also the Treasurer of the Norsk Forening for Kunstig Intelligens (NAIS), the Norwegian Association for Artificial Intelligence. He is a reviewer of several international conferences and journals.

MALIK NAVEED AKHTER received the B.E. degree in mechatronics engineering and the M.S. degree in biomedical engineering from the National University of Science and Technology, Islamabad, Pakistan, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree in biomedical sciences with the Auckland University of Technology, Auckland, New Zealand. His current research interests include soft robotics, rehabilitation robotics, upper limb prostheses, and artificial intelligence.



SHAHZAIB FAROOQ HADI received the bachelor's degree in mechatronics engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2016. He is currently pursuing the M.S. degree in artificial intelligence. His current research interests include robotics, artificial intelligence, and optimization algorithms.