

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/371248558>

Stress monitoring using wearable sensors: IoT techniques in medical field

Article in *Neural Computing and Applications* · June 2023

DOI: 10.1007/s00521-023-08681-z

CITATIONS

40

READS

628

2 authors:



Fatma M. Talaat

Kafrelsheikh University

100 PUBLICATIONS 2,074 CITATIONS

[SEE PROFILE](#)



Rana Mohamed Elbalka

Mansoura University

6 PUBLICATIONS 45 CITATIONS

[SEE PROFILE](#)



Stress monitoring using wearable sensors: IoT techniques in medical field

Fatma M. Talaat¹ · Rana Mohamed El-Balka²

Received: 7 February 2023 / Accepted: 10 May 2023
© The Author(s) 2023

Abstract

The concept “Internet of Things” (IoT), which facilitates communication between linked devices, is relatively new. It refers to the next generation of the Internet. IoT supports healthcare and is essential to numerous applications for tracking medical services. By examining the pattern of observed parameters, the type of the disease can be anticipated. For people with a range of diseases, health professionals and technicians have developed an excellent system that employs commonly utilized techniques like wearable technology, wireless channels, and other remote equipment to give low-cost healthcare monitoring. Whether put in living areas or worn on the body, network-related sensors gather detailed data to evaluate the patient’s physical and mental health. The main objective of this study is to examine the current e-health monitoring system using integrated systems. Automatically providing patients with a prescription based on their status is the main goal of the e-health monitoring system. The doctor can keep an eye on the patient’s health without having to communicate with them. The purpose of the study is to examine how IoT technologies are applied in the medical industry and how they help to raise the bar of healthcare delivered by healthcare institutions. The study will also include the uses of IoT in the medical area, the degree to which it is used to enhance conventional practices in various health fields, and the degree to which IoT may raise the standard of healthcare services. The main contributions in this paper are as follows: (1) importing signals from wearable devices, extracting signals from non-signals, performing peak enhancement; (2) processing and analyzing the incoming signals; (3) proposing a new stress monitoring algorithm (SMA) using wearable sensors; (4) comparing between various ML algorithms; (5) the proposed stress monitoring algorithm (SMA) is composed of four main phases: (a) data acquisition phase, (b) data and signal processing phase, (c) prediction phase, and (d) model performance evaluation phase; and (6) grid search is used to find the optimal values for hyperparameters of SVM (C and gamma). From the findings, it is shown that random forest is best suited for this classification, with decision tree and XGBoost following closely behind.

Keywords Stress monitoring · Wearable sensors · Medical field · IoT

1 Introduction

The Internet of things (IoT) has transformed the healthcare industry by enabling remote monitoring of patients’ health status [1]. The integration of wearable sensors with IoT technologies has enabled continuous monitoring of physiological signals, leading to early detection of health problems and the ability to intervene before the condition worsens. Stress is a common health problem that can lead to various physical and mental disorders. IoT has the potential to be utilized in numerous industries, including efficient energy, transportation, agriculture, university campus connectivity, healthcare, logistics, and others,

✉ Fatma M. Talaat
fatma.nada@ai.kfs.edu.eg

Rana Mohamed El-Balka
rana.m.elbalka@gmail.com

¹ Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh, Egypt

² Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt

allowing physical objects to communicate and share information with one another through the internet [1].

IoT technology and machine learning (ML) have significantly impacted healthcare by shifting routine medical tests and healthcare services from hospitals to homes, making it easier for patients and healthcare professionals to utilize medical equipment [2]. With IoT integration, portable sensors can provide more precise data, and medical devices' usability can be enhanced by implementing an android program in conjunction with IoT. The implementation of various technologies, particularly IoT, is likely to have a significant impact on all sectors, particularly in the medical field, as it can improve people's quality of life [2].

IoT and ML have various applications in healthcare and daily life, with the internet's rapid growth decreasing the use of traditional patient service methods and increasing the use of electronic healthcare systems, providing patients and healthcare professionals with access to advanced medical equipment through IoT technology. ML and IoT devices are beneficial in various categories, such as remote monitoring and mechanical automation, offering convenience, cost savings, and increased patient satisfaction in medical care applications [3].

In the IoT healthcare system, a sensor can be identified as a "thing" based on three key characteristics. Firstly, it should be able to recognize and gather environmental data such as temperature, light, and precipitation, as well as monitor pulse-related functions like electrocardiogram, blood glucose levels, and blood oxygen levels. Secondly, it should be capable of transmitting data autonomously to a centralized controller, either dynamically or through another system. Lastly, it should be able to be inactive after the operation is completed, alerting medical professionals to take immediate action if necessary [1, 4]. Additionally, DNA origami has emerged as versatile nanomachines for transportation, sensing, and computing in two-dimensional patterns, and three-dimensional assembly [5–7].

Researchers have conducted innovative studies to improve healthcare systems by offering various analytical applications for managing data sources, including electronic health records (EHRs) and medical images [8]. While the development of apps and services in healthcare is tailored to user needs, it is clear that services are designed based on what the developer has to offer. Recently, various ML techniques have been employed such as convolutional neural networks [9, 10] in multiple applications across various fields, such as efficiently grading alcohol dependence, estimating accident severity in severe injuries, and identifying emotions in functional technologies [11, 12].

In conclusion, IoT and ML have brought about significant advancements in healthcare and daily life. With the integration of IoT, wearable sensors, and ML, healthcare

professionals can obtain real-time data on patients' health status, enabling early detection and intervention before the condition worsens. The convenience, cost savings, and increased patient satisfaction of IoT devices have attracted attention in medical care applications, and the use of integrated technologies has the potential to improve electronic information management services, controlled communications, and system processing in various ways [2].

Stress is a common health problem that can lead to various physical and mental disorders. However, the integration of wearable sensors with IoT technologies has allowed for continuous monitoring of physiological signals, leading to early detection of stress-related problems and the ability to intervene before the condition worsens [13].

The use of IoT and ML in stress monitoring has gained attention in recent years. For example, a study proposed a wearable sensor system that combines physiological measurements, such as heart rate variability and skin conductance, with contextual data, such as location and activity level, to detect stress and track its progression [14]. This system utilized ML algorithms to analyze the data collected by the wearable sensors and provide personalized feedback and intervention to the user.

Other studies have explored the use of various sensors in stress monitoring, such as electroencephalography (EEG) and electromyography (EMG) sensors [15, 16]. These sensors measure brain activity and muscle tension, respectively, and can provide valuable information for stress monitoring and intervention.

The integration of IoT and ML in stress monitoring has the potential to improve the accuracy and efficiency of stress detection and intervention. By utilizing ML algorithms, the collected data can be analyzed in real time, allowing for early detection and intervention before the stress condition worsens. This can have a significant impact on the individual's quality of life and overall health.

In conclusion, stress monitoring by IoT and ML has the potential to revolutionize the healthcare industry by allowing for continuous monitoring of physiological signals and providing personalized feedback and intervention to the user. With advancements in wearable sensor technology and ML algorithms, the accuracy and efficiency of stress detection and intervention are likely to improve in the future.

Research gap:

- Limited studies have explored the integration of IoT and ML in stress monitoring, despite the potential benefits it offers.
- Few studies have focused on the use of multiple sensors for stress monitoring, with most research focusing on a single type of sensor.

- There is a need for further research to develop more accurate and efficient ML algorithms for stress detection and intervention.

Main contributions:

- The integration of wearable sensors with IoT and ML technologies can provide real-time stress monitoring and personalized feedback and intervention to the user.
- The use of multiple sensors, such as EEG and EMG, can provide valuable information for stress monitoring and intervention.
- ML algorithms can analyze the collected data in real time, allowing for early detection and intervention before the stress condition worsens, thus improving the individual's quality of life and overall health.

The structure of the remaining work is outlined as follows: Sect. 2 discusses recent research related to IoT in healthcare systems, Sect. 3 presents the proposed framework, Sect. 4 provides details of the experimental evaluation, and Sect. 5 concludes the effort.

2 Related work

This section presents a discussion of various relevant research areas, including (1) human behavior recognition applications, (2) RL techniques for Internet of things (IoT), (3) wearable technology for stress monitoring and detection, and (4) challenges in physiological stress monitoring.

2.1 Human behavior recognition applications

In the field of Human behavior recognition applications, there have been several studies. Zhao et al. [17] proposed a novel Res-Bidir-LSTM network to tackle HAR problems. This approach has shown high precision in the early stages of training, even though it takes a long time to develop. The Res-Bidir-LSTM method can be applied to challenging, large-scale HAR issues when sensor fusion is required. The HAR's input should include time series through the basic structure of the LSTM. One of the benefits of this technique is that it successfully solves the gradient vanishing problem.

Chang et al. [18] introduced a smart system, known as ST-Med-Box, based on deep learning techniques to identify medications. This system may help patients with chronic diseases to take numerous medications accurately and ensure they take the medication. Patients can download the available Android app and scan the QR codes on their drug packets to store the relevant medication information. Then, they can be reminded to take their medication.

2.2 RL techniques for internet of things (IoT)

To find the optimum decision-making approach for the Internet of Things (IoT), a number of RL techniques have been investigated [19]. These include Q-learning, SARSA, Expected SARSA, and Monte Carlo. By utilizing RL techniques, the fog node's idle time may be minimized and its use of resources and services can be increased. In [20–22], a trustworthy cloud management system is proposed using RL.

2.3 Wearable technology for stress monitoring and detection

The proposed study is influenced by a number of ongoing studies in the area of wearable technology for stress monitoring and detection. The RespiBAN and Empatica E4 wearables were used to construct the WESAD (Wearable Stress and Affect Detection) dataset [23]. The SWELL-KW dataset [24] combined computer logging, video (for facial expression), and Kinect (a 3D body posture sensor) to track the stress levels of 25 people while they worked on conventional knowledge tasks (doing a presentation, reading, producing reports, and sending emails) under time constraints.

The Affective-Road dataset [25] used Zephyr BioHarness 3.0 and Empatica E4 to measure stress levels throughout 10 drives while they were under various driving circumstances. Healey et al. [26] created a wearable glove with a photoplethysmogram (PPG) sensor inserted to track the levels of stress in 10 drivers as they traveled various routes. Similarly, Shi et al. [27] created a multi-node stress monitoring system based on ECG, EDA, and PPG data, and Muaremi et al. [28] used a smartphone and a Wahoo wearable breast belt to identify various levels of stress. A multi-sensor dataset of nurses who were employed by the hospital during the COVID-19 outbreak was produced by Hosseini et al. [29].

2.4 Challenges in physiological stress monitoring

The literature study leads to the conclusion that there is still some debate over the best measuring strategy for physiological stress monitoring. The classification accuracy obtained by several research, albeit using the same physiological variables and classifiers, varied greatly. In the research, it is also unclear how sensitive and specific stress-related bio physiological measures like heart rate and respiration rate are [30, 31]. Table 1 presents the models that are commonly utilized in stress prediction, including random forest, XGBoost, and decision tree.

Table 1 Comparative analysis of the prevalent models employed in the field

Model	Algorithm description	Pros	Cons
Random forest	Random forest is a decision tree-based ensemble learning algorithm that builds multiple decision trees and aggregates their predictions to improve accuracy and reduce overfitting. Each tree in the forest is trained on a randomly selected subset of features and data samples. Random forest is widely used in various applications such as finance, marketing, and healthcare	<ul style="list-style-type: none"> Can handle high dimensional datasets with a large number of features Can handle missing values and categorical variables without requiring preprocessing Can provide feature importance rankings that can be useful for feature selection Has low bias and high variance, which can be beneficial for complex problems 	<ul style="list-style-type: none"> May not perform well on small datasets with limited samples Can be slow and computationally expensive compared to other algorithms May overfit on noisy datasets
XGBoost	XGBoost (Extreme Gradient Boosting) is a boosting algorithm that trains multiple weak learners in a sequential manner, where each subsequent learner tries to correct the errors of the previous one. XGBoost is known for its scalability, efficiency, and high performance, and has been used in various fields such as natural language processing, image classification, and financial forecasting	<ul style="list-style-type: none"> Can handle missing values and categorical variables without requiring preprocessing Has a faster training time and better performance than other boosting algorithms Can handle imbalanced datasets and produce accurate results with a small number of samples Provides feature importance rankings that can be useful for feature selection 	<ul style="list-style-type: none"> Can be sensitive to the choice of hyperparameters May overfit on small datasets with limited samples Can require more memory compared to other algorithms
Decision tree	A decision tree algorithm creates a treelike model of decisions and their possible consequences. It is a simple and interpretable algorithm that can handle both categorical and continuous data. Each node in the tree represents a feature, and each branch represents a decision rule based on the feature's value. Decision trees are used in various applications such as fraud detection, credit risk assessment, and medical diagnosis	<ul style="list-style-type: none"> Easy to understand and interpret Can handle both categorical and continuous data Can provide feature importance rankings that can be useful for feature selection Can handle missing values without requiring preprocessing 	<ul style="list-style-type: none"> Easy to understand and interpret Can handle both categorical and continuous data Can provide feature importance rankings that can be useful for feature selection Can handle missing values without requiring preprocessing

Table 1 provides a comparative analysis of the commonly used models for stress prediction, namely random forest, XGBoost, and decision tree. Each algorithm is described in detail, highlighting its advantages and limitations. This table serves as a helpful reference for researchers and practitioners who are looking for the most suitable algorithm for their stress prediction task. By comparing the pros and cons of each algorithm, users can make an informed decision about which algorithm to use based on the specific requirements of their application.

Research gaps can be summarized in the following points:

- Limited amount of data points for deep LSTM neural network in human behavior recognition applications.
- Lack of comprehensive statistical analysis of the dataset in stress monitoring studies.
- Variability in classification accuracy for stress-related physiological measures across different studies.
- Small sample sizes in existing stress monitoring datasets.
- Limited comparison of different machine learning models for stress prediction.

3 Stress monitoring algorithm (SMA) using wearable sensors

The wearable sensors that are part of the suggested stress monitoring concept are simple for everyone to use. The proposed stress monitoring algorithm (SMA) is composed of four main phases as depicted in Fig. 1: (1) data acquisition phase, (2) data and signal processing phase, (3) prediction phase, and (4) model performance evaluation phase.

The proposed stress monitoring algorithm (SMA) using wearable sensors is a four-phase algorithm that aims to

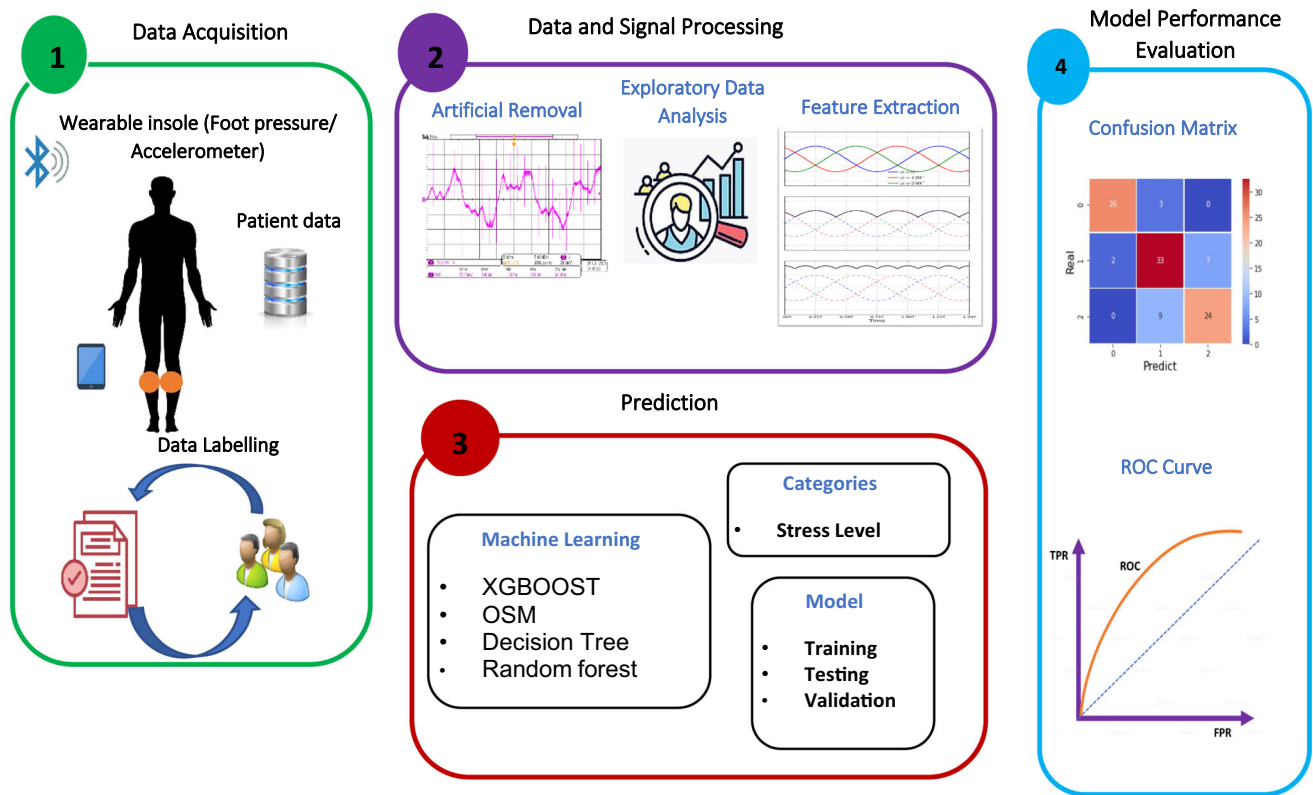


Fig. 1 Architecture of the proposed stress monitoring using wearable sensors

monitor and predict stress levels in individuals. Here is a brief overview of each phase:

1. **Data Acquisition Phase** In this phase, data are collected from the wearable sensors, which include physiological signals such as heart rate variability (HRV), skin conductance, and temperature. The sensors are worn by the individual throughout the day, and the data are collected continuously.
2. **Data and Signal Processing Phase** The data collected from the sensors undergo several processing steps to extract the relevant features that are indicative of stress. This phase includes preprocessing of the data, feature extraction, feature selection, and normalization.
3. **Prediction Phase** Once the relevant features have been extracted, the next step is to use machine learning algorithms to predict the stress levels of the individual. The machine learning algorithms used for this phase include decision trees, support vector machines, and artificial neural networks.
4. **Model Performance Evaluation Phase** In the final phase, the performance of the model is evaluated. The model is tested on a separate dataset to assess its accuracy in predicting stress levels. The performance of the model is evaluated based on various metrics.

3.1 Data acquisition

The proposed architecture aims to benefit from wearable technologies created for real-time monitoring of physiological parameters related to stress. (1) Electrocardiogram (ECG), (2) infrared sensors, (3) hypertension monitoring, and (4) smoking behavior analysis are a few examples of wearable-based technologies.

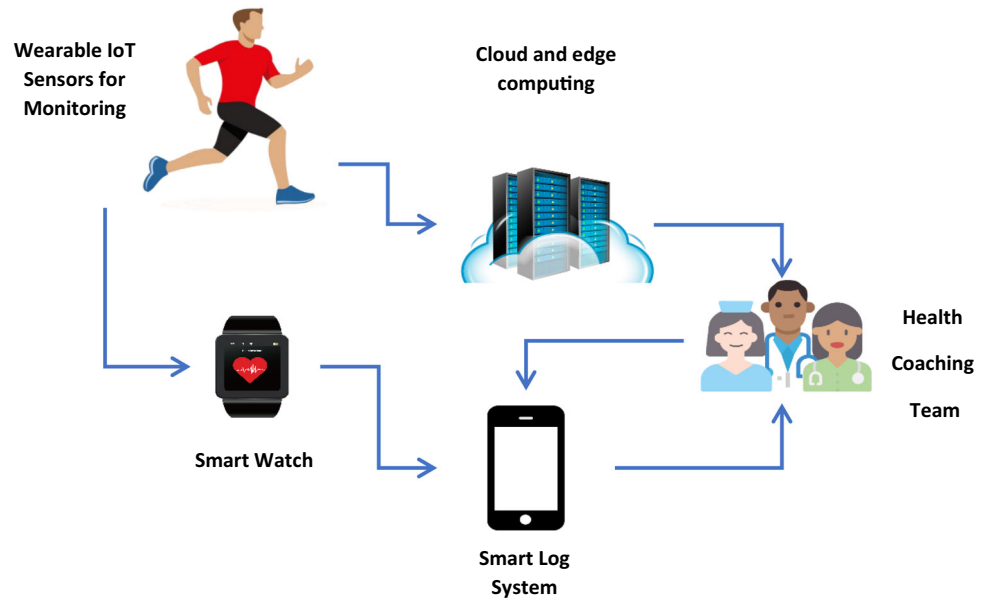
3.1.1 Electrocardiogram (ECG) for heart rate monitoring

These sensors will provide real-time heart rate monitoring using an ECG sensor to support the suggested model [32, 33]. Additionally, an IoT infrastructure with a prognosis algorithm is used to more efficiently notify doctors and patients' loved ones. A strategy for predicting the highs and lows of glucose levels was suggested [34, 35]. Figure 2 depicts this architecture.

3.1.2 Infrared sensors for blood glucose monitoring

Using near-infrared sensors, a portable noninvasive blood glucose monitoring system was created. This sensor, which is implanted at the fingertip to opportunistically measure blood sugar levels, computes the blood glucose concentrations. The signal is then increased and filtered before

Fig. 2 Using IoT architecture in health monitoring systems



going to the microcontroller. The patient's glucose level was predicted using the voltages that were measured and the glucose information that would be shown on the LCD and transferred through Wi-Fi to a mobile device.

3.1.3 Hypertension monitoring

One of the factors that affects patients' chances of having a stroke is hypertension. Thus, one of the variables that must be carefully tracked in order to follow patients' conditions and make the best decision in order to avoid stroke is hypertension. For monitoring hypertension, it has previously been recommended to look at the blood pressure, the diagnosis, and the results in a hypertension patient setting [36].

3.1.4 Smoking behavior analysis

Wearable sensors have excelled at accurately identifying smoking events and individual puffs. The precise analysis of a person's smoking behavior requires the employment of these sensing techniques over the course of several weeks or months [37]. These sensors can help with the suggested model for tracking the smokers.

3.2 Data and signal processing

The concept "digital signal processing" (DSP) refers to the application of digital processing to a range of signal processing tasks, such as those carried out by computer systems or more specialized digital signal processors. The digital signals examined in this manner are a set of values that represent samples of a continuous variable in a domain

such as time, space, or frequency. In digital electronics, a pulse train—which is often generated by transistor switching—represents a digital signal. Algorithm 1 illustrates the DSP algorithm's steps. DSP consists of four main steps: (1) *Step 1* preprocessing of data; the first step is to preprocess the data collected by the wearable sensors. This includes filtering the data to remove any noise or artifacts, normalizing the data to account for any variations in sensor readings, and segmenting the data into windows of a fixed length for analysis.

(2) *Step 2* Feature Extraction: the second step involves extracting features from the preprocessed data. The features are statistical measures that describe the characteristics of the data, such as mean, variance, skewness, and kurtosis. These features are used as input to the prediction model in the next phase. (3) *Step 3* Feature Selection: in this step, a subset of the extracted features is selected for use in the prediction model. This is done to reduce the dimensionality of the data and improve the accuracy of the prediction model. Various feature selection techniques can be used for this purpose, such as principal component analysis (PCA) or correlation-based feature selection. (4) *Step 4* Signal Processing: this step involves applying signal processing techniques to the preprocessed data. These techniques include spectral analysis, time–frequency analysis, and wavelet analysis. The goal of this step is to extract more information from the data that can be used in the prediction model.

Algorithm 1: DSP Algorithm

```

• Input:
  ○ Raw data
• Output:
  ○ Selected features for prediction model
• Steps:
  def data_signal_processing(raw_data):
    # Step 2: Filter the signal
    filtered_data = apply_low_pass_filter(raw_data)
    filtered_data = apply_high_pass_filter(filtered_data)

    # Step 3: Feature extraction
    time_domain_features = calculate_time_domain_features(filtered_data)
    frequency_domain_features = calculate_frequency_domain_features(filtered_data)

    # Step 4: Feature selection
    important_features = select_important_features(time_domain_features,
    frequency_domain_features)

    # Step 5: Feature normalization
    normalized_features = normalize_features(important_features)

    # Step 6: Data transformation
    transformed_features = transform_features(normalized_features)

    return transformed_features

```

3.3 Prediction

In this phase, a set of selected models are trained and tested and then the model with the highest performance is selected according to the steps illustrated in Algorithm 2. The prediction phase combines five main steps: (1) *Step 1* Preprocessing: Preprocess the data obtained from the previous phase, data and signal processing, by removing any noise or artifacts and normalizing the data. (2) *Step 2* Feature Extraction: Extract relevant features from the preprocessed data, such as heart rate variability, skin temperature, and respiration rate. (3) *Step 3* Model Selection: Select a set of models to train and test on the extracted

features. Examples of models that can be used include support vector machines (SVM), random forests, and neural networks. (4) *Step 4* Training and Testing: (a) Train each selected model on a portion of the extracted feature data and test the trained model on the remaining portion of the data. (b) Use metrics such as accuracy, sensitivity, and specificity to evaluate the performance of each model.

(5) *Step 5* Model Performance Evaluation: (a) select the model with the highest performance based on the evaluation metrics obtained in the previous step. (b) Evaluate the selected model using cross-validation techniques to ensure that it is not overfitting the data.

Algorithm 2: Prediction Algorithm

- **Input:**
 - Preprocessed and extracted feature data
 - Set of selected models
 - **Output:**
 - Selected model with the highest performance
 - **Steps:**
 - Step 1: Preprocessing**
 - Remove noise and artifacts from the feature data
 - Normalize the data
 - Step 2: Feature Extraction**
 - Extract relevant features from the preprocessed data
 - Step 3: Model Selection**
 - Select a set of models to train and test on the extracted features
 - Step 4: Training and Testing**
 - For each selected model:
 - Train the model on a portion of the extracted feature data
 - Test the trained model on the remaining portion of the data
 - Calculate evaluation metrics such as accuracy, sensitivity, and specificity
 - Step 5: Model Performance Evaluation**
 - Select the model with the highest performance based on the evaluation metrics obtained in the previous step
 - Evaluate the selected model using cross-validation techniques to ensure that it is not overfitting the data
- Return the selected model with the highest performance*

The prediction phase compares four ML algorithms: (1) XGBoost, (2) OSM, (3) decision tree, and (4) random forest.

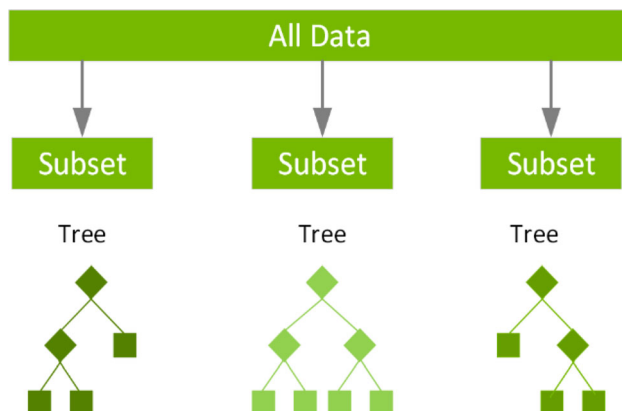


Fig. 3 General architecture for XGBoost algorithm

1. XGBoost

Extreme gradient boosting model XGBoost is a sequential ensemble model. This is a gradient boost extension. The iterative XGBoost decision tree method employs a variety of decision trees. Each tree eventually

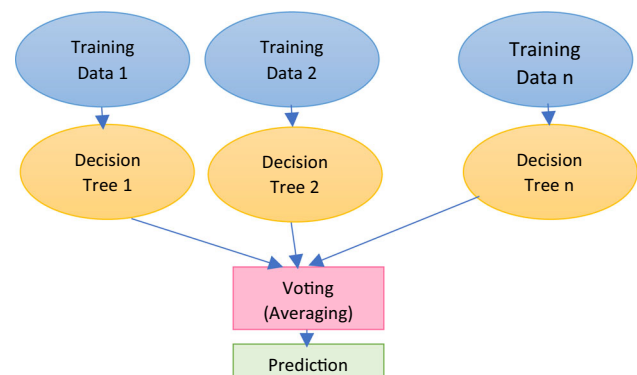


Fig. 4 General architecture of random forest algorithm

picks up knowledge from the leftovers of every other tree. As opposed to accepting the final output of random forest that received the majority of votes, as illustrated in Eq. (1), the predicted output of XGBoost is the sum of all the results.

$$y_i = \sum_{k=1}^n f_k(x_i), f_k \in F \quad (1)$$

where F is the set of regression trees, f_k denotes a tree, and y_i denotes the projected value for the i th occurrence of x_i . Figure 3 shows a general architecture for XGBoost algorithm. Figure 4 depicts the general architecture of random forest.

2. Optimized support vector machine (OSM)

This stage outlines the optimization procedure used for the support vector machine hyperparameters (SVM). The best values for the SVM's hyperparameters (C and gamma) are found via grid search [38]. SVMs may effectively carry out nonlinear classification in addition to linear classification utilizing the kernel method, as shown in Fig. 5. SVM uses a number of kernel functions to get the best solution. The type of the kernel function and the trade-off constant C must be specified in order to implement SVMs. The soft margin's parameter C, which regulates how each support vector's influence is felt, is used. Algorithm 3 illustrates the total steps of the hyperparameters optimization algorithm.

Algorithm 3: SVM Hyper parameters Optimization Steps

Input: Initial values for the hyper parameters of SVM (C and gamma) and training dataset

Output: Optimal values for C and gamma, highest accuracy

Steps:

1. Train the model with AffectNet dataset
2. Test the model
3. Calculate the accuracy (Acc)
4. Divide dataset into subsets
5. For each subset
 - a. Use Grid search to redefine new values for C and Gamma (C_new, Gamma_new)
 - b. Train the model with the new values C_new and Gamma_new
 - c. Test the model
 - d. Calculate the accuracy (Acc_New)
6. Next
7. Retrieve C_new, Gamma_new, and Acc_New

3. Decision tree

The decision tree algorithm is a type of supervised learning approach. The decision tree technique can handle both classification and regression issues, unlike other

supervised learning methods. Using simple decision rules discovered from prior data, a training model is created with the use of a decision tree that may be used to forecast the class or value of the target variable (training data).

4. Random forest repressors

Random forest [39] uses the ensemble learning method for classification and regression as a supervised learning algorithm. It is a set of classification or regression trees that have not been trimmed that were produced using a random sample of training data. The qualities picked at random during the induction process. The prediction is created by combining the ensemble's guesses. Figure 4 displays the overall random forest architecture.

4 Implementation and evaluation

This section introduces the used dataset and the performance evaluation.

4.1 Human stress detection dataset

Titles for the Stress-Lysis.csv file [40] include "Humidity—Temperature—Step count—Stress levels." The stress levels of the human being are here identified and assessed based on physical activity. Human body humidity, body

temperature, and number of steps done by the user are all included in a collection of 2001 samples. Low stress, typical stress, and high stress are three different categories of stress that are evaluated.

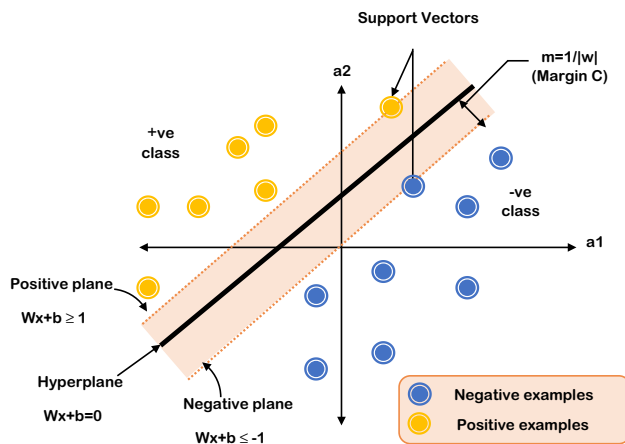


Fig. 5 Separated data by a hyperplane defined by several support vectors

4.2 Stress level detection

A sample of dataset is depicted in Table 2.

4.3 Data categorization

A sample of categorized dataset is depicted in Table 3.

4.4 Data visualization

The results of the data visualization phase are illustrated in Fig. 6. The heatmap is shown in Fig. 7. The results for each feature are shown in Table 4. The results of the used ML algorithms are shown in Table 5 and in Fig. 8.

The score convergence curve for each model, as depicted in Fig. 9.

From the findings above, it is shown that random forest and OSM are best suited for this classification, with decision tree and XGBoost following closely behind.

The performance of four popular machine learning algorithms (random forest, OSM classifier, decision tree, and XGBoost) was evaluated for stress level detection using data from wearable sensors. The results showed that random forest and OSM classifier performed the best, with perfect accuracy and ROC-AUC scores of 1.0. Decision

tree and XGBoost also performed well, but with slightly lower accuracy and ROC-AUC scores.

The success of random forest and OSM classifier in stress level detection can be attributed to their ability to handle large datasets and noisy features, as well as their high accuracy in handling both classification and regression tasks. On the other hand, decision tree and XGBoost are also well-known for their effectiveness in classification tasks but may not perform as well with noisy data.

Overall, the results suggest that machine learning algorithms can effectively detect stress levels using wearable sensor data, with random forest and OSM classifier being the most suitable for this task. These findings can have significant implications in healthcare, as wearable sensor technology continues to become more accessible and may be used to monitor stress levels and other health indicators in real time.

4.5 Results discussion

The results of this study suggest that the proposed stress monitoring algorithm (SMA) using wearable sensors is effective in detecting stress levels in humans. The dataset used in this study was the Stress-Lysis.csv file, which contains 2001 samples of human stress levels based on physical activity, humidity, temperature, and step count.

In the data categorization phase, the data were categorized into three different stress levels: low, normal, and high. This categorization was used to train and test various machine learning classifiers to determine which classifier was most effective in detecting stress levels. The results of the machine learning classifiers were evaluated using the ROC-AUC score, which measures the overall performance of the classifier.

The results of the machine learning classifiers showed that the random forest classifier achieved the highest score, with a perfect score of 1.000 for both the Score and ROC-AUC Score. The OSM classifier also achieved a perfect score of 1.000 for both measures. The XGBoost and decision tree classifiers had lower scores, but still performed well with scores of 0.963 and 0.920 for the Score measure, respectively.

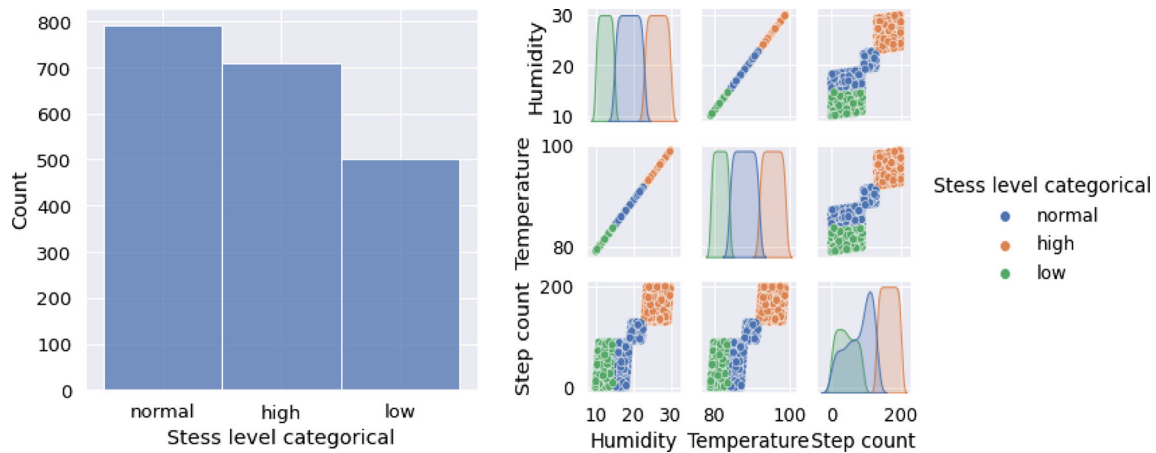
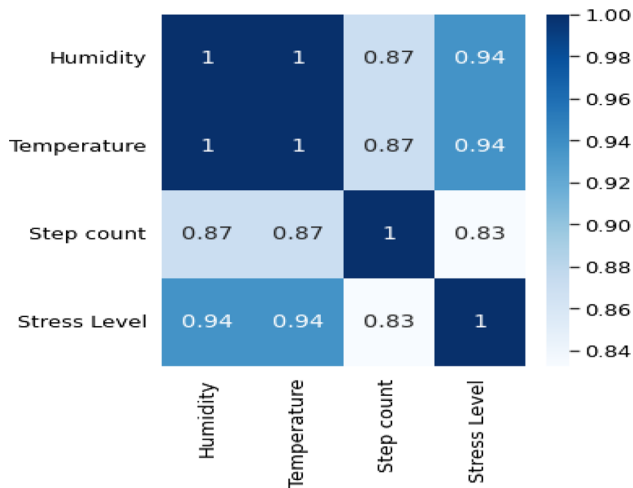
Overall, these findings suggest that the proposed SMA using wearable sensors and machine learning classifiers can be effective in detecting stress levels in humans. The random forest classifier in particular showed excellent performance in this study. However, further research is needed to validate these results on a larger dataset and to determine the applicability of this algorithm in real-world settings. Additionally, the use of other types of sensors, such as heart rate monitors and electroencephalography (EEG) sensors, could be explored to improve the accuracy of stress level detection.

Table 2 A sample of the used dataset

Humidity	Temperature	Step count	Stress level
21.33	90.33	123	1
21.41	90.41	93	1
27.12	96.12	196	2
27.64	96.64	177	2
10.87	79.87	87	0

Table 3 Categorized data

Humidity	Temperature	Step count	Stress level	Stress level categorial
21.33	90.33	123	1	Normal
21.41	90.41	93	1	Normal
27.12	96.12	196	2	High
27.64	96.64	177	2	High
10.87	79.87	87	0	Low

**Fig. 6** Data visualization**Fig. 7** Data heatmap**Table 4** Results for each feature

	Count	Mean	std	Min	25%	50%	75%
Humidity	2001.0	20.000000	5.777833	10.0	15.0	20.0	25.0
Temperature	2001.0	89.000000	5.777833	79.0	84.0	89.0	94.0
Step count	2001.0	100.141429	58.182948	0.0	50.0	101.0	150.0
Stress level	2001.0	1.104448	0.771094	0.0	0.0	1.0	2.0

Table 5 Summarized achieved results of the used machine learning classifiers

Model_after_resampling	Score	ROC_AUC_Score
Random forest	1.000000	1.000000
OSM classifier	1.000000	1.000000
XGBoost	0.963195	0.986748
Decision tree	0.920125	0.957978

5 Conclusion

The study included the uses of IoT in the medical area, the degree to which it is used to enhance conventional practices in various health fields, and the degree to which IoT may raise the standard of healthcare services. The main contributions in this paper are as follows: (1) importing

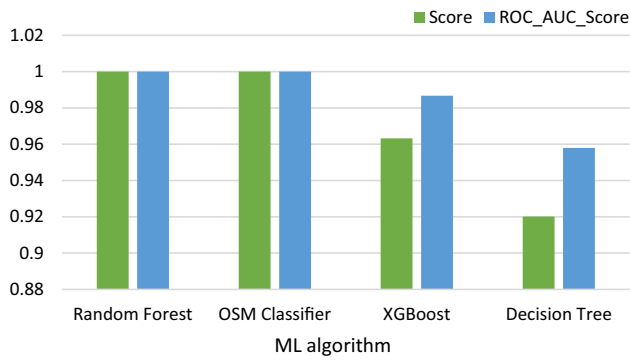


Fig. 8 Results of the used ML algorithms

signals from wearable devices, extracting signals from non-signals, performing peak enhancement; (2) processing and analyzing the incoming signals; (3) proposing a new stress monitoring algorithm (SMA) using wearable sensors; (4)

comparing between various ML algorithms; (5) the proposed stress monitoring algorithm (SMA) is composed of four main phases: (a) data acquisition phase, (b) data and signal processing phase, (c) prediction phase, and (d) model performance evaluation phase; and (6) grid search is used to find the optimal values for hyperparameters of SVM (C and gamma). From the findings, it is shown that random forest is best suited for this classification, with decision tree and XGBoost following closely behind. The findings of this study have implications for the future of healthcare services, as the use of IoT in medical monitoring can improve the accuracy and efficiency of diagnosis and treatment. Future work can explore other ML algorithms and correlation methods to further improve the performance of the proposed SMA. Additionally, the use of other techniques such as OCNN may also be explored. In the future work, we can use OCNN [41] to achieve better

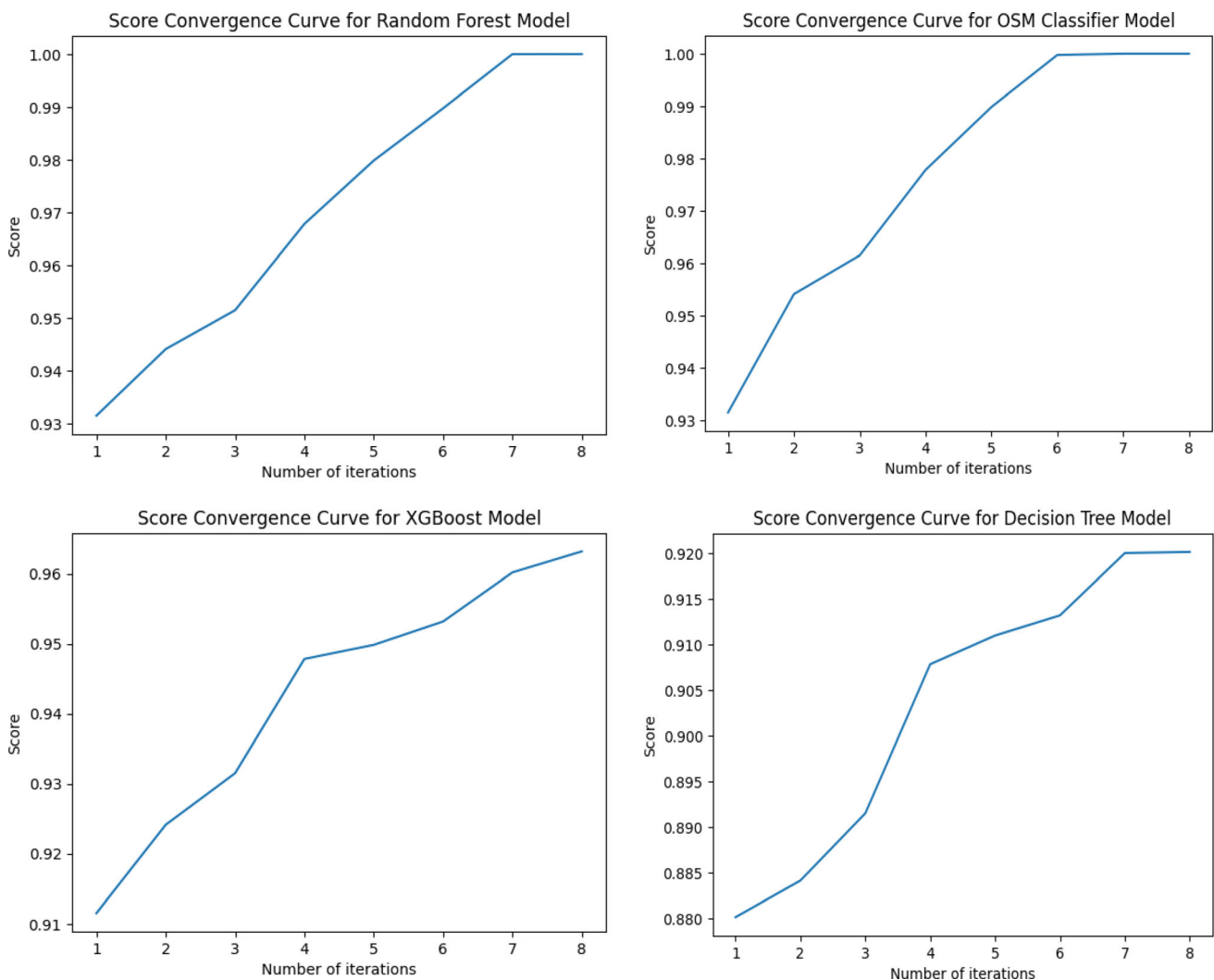


Fig. 9 Score convergence curve for each model

results as it achieved a good performance in [42–48]. We can also use correlation methods like [49].

Author contributions It is a collaborative effort where Fatma and Rana worked together. Fatma came up with the idea and wrote the abstract and the proposal and made the experiments, while Rana contributed by making comparisons.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). The authors received no specific funding for this study.

Data availability <https://www.kaggle.com/datasets/laavanya/stress-level-detection>.

Declarations

Ethical approval There is no any ethical conflicts.

Conflict of interest The authors declare that they have no conflicts of interest to report regarding the present study.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Kadhim KT, Alsahlany AM, Wadi SM, Kadhum HT (2020) An overview of patient's health status monitoring system based on Internet of Things (IoT). *Wirel Pers Commun* 114(3):2235–2262
- Mohammed CM, Askar S (2021) Machine learning for IoT healthcare applications: a review. *Int J Sci Bus* 5(3):42–51
- Bolhasani H, Mohseni M, Rahmani AM (2021) Deep learning applications for IoT in health care: a systematic review. *Inform Med Unlocked* 23:100550
- Chen Y, Shi P, Bai Y, Li J, Feng J, Sareh P (2023) Engineered origami crease perforations for optimal mechanical performance and fatigue life. *Thin-Walled Struct* 185:110572
- Jalali E, Soltanizadeh H, Chen Y, Xie YM, Sareh P (2022) Selective hinge removal strategy for architecting hierarchical auxetic metamaterials. *Commun Mater* 3(1):97
- Chen Y, Xu R, Lu C, Liu K, Feng J, Sareh P (2023) Multi-stability of the hexagonal origami hyper based on group theory and symmetry breaking. *Int J Mech Sci* 247:108196
- He Z, Shi K, Li J, Chao J (2023) Self-assembly of DNA origami for nanofabrication, biosensing, drug delivery and computational storage. *iScience* 26:106638
- Palanisamy V, Thirunavukarasu R (2019) Implications of big data analytics in developing healthcare frameworks—a review. *J King Saud Univ-Comput Inf Sci* 31(4):415–425
- Zhang P, Fan W, Chen Y, Feng J, Sareh P (2022) Structural symmetry recognition in planar structures using convolutional neural networks. *Eng Struct* 260:114227
- Eidini M, Paulino GH (2015) Unraveling metamaterial properties in zigzag-base folded sheets. *Sci Adv* 1(8):e1500224
- Jayadev PG, Bellary S (2022) A hybrid approach for classification and identification of iris damaged levels of alcohol drinkers. *J King Saud Univ-Comput Inf Sci* 34(8):5273–5285
- Rao Z, Tung PY, Xie R, Wei Y, Zhang H, Ferrari A, Klaver TP, Körmann F, Sukumar PT, Kwiatkowski da Silva A, Chen Y (2022) Machine learning-enabled high-entropy alloy discovery. *Science* 378(6615):78–85
- Sun W, Guo Z, Yang Z, Wu Y, Lan W, Liao Y, Wu X, Liu Y (2022) A review of recent advances in vital signals monitoring of sports and health via flexible wearable sensors. *Sensors* 22(20):7784. <https://doi.org/10.3390/s22207784>
- Mozos OM, Sandulescu V, Andrews S, Ellis D, Bellotto N, Dobrescu R, Ferrandez JM (2017) Stress detection using wearable physiological and sociometric sensors. *Int J Neural Syst* 27:1650041
- Ahn JW, Ku Y, Kim HC (2019) A novel wearable EEG and ECG recording system for stress assessment. *Sensors* 19:9. <https://doi.org/10.3390/s19091991>
- Pourmohammadi S, Maleki A (2020) Stress detection using ECG and EMG signals: a comprehensive study. *Comput Methods Programs Biomed* 193:105482. <https://doi.org/10.1016/j.cmpb.2020.105482>. (Epub 2020 May 5 PMID: 32408236)
- Zhao Y, Yang R, Chevalier G, Xu X, Zhang Z (2018) Deep residual bidir-LSTM for human activity recognition using wearable sensors. *Math Probl Eng*
- Chang WJ, Chen LB, Hsu CH, Lin CP, Yang TC (2019) A deep learning-based intelligent medicine recognition system for chronic patients. *IEEE Access* 7:44441–44458
- Talaat FM, Ali HA, Saraya MS et al (2022) Effective scheduling algorithm for load balancing in fog environment using CNN and MPO. *Knowl Inf Syst* 64:773–797. <https://doi.org/10.1007/s10115-021-01649-2>
- Duggan M, Flesk K, Duggan J, Howley E, Barrett E (2016) A reinforcement learning approach for dynamic selection of virtual machines in cloud data centres. In: 2016 sixth international conference on innovative computing technology (INTECH), pp 92–97. <https://doi.org/10.1109/INTECH.2016.7845053>
- Talaat FM, Alshathri S, Nasr AA (2022) A new reliable system for managing virtual cloud network. *Comput Mater Continua* 73(3):5863–5885. <https://doi.org/10.32604/cmc.2022.026547>
- Talaat FM, MSabry, AI Saleh, SH Ali, (2020) A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment. *J Ambient Intell Humaniz Comput*. <https://doi.org/10.1007/s12652-020-01768-8>
- Schmidt P, Duerichen R, van Laerhoven K, Marberger C, Reiss A (2018) Introducing WESAD, a multimodal dataset for wearable stress and affect detection. In: Proceedings of the 20th ACM international conference on multimodal interaction, Boulder, CO, USA, 16–20 October 2018; pp 400–408
- Koldijk S, Sappelli M, Verberne S, Neerinx MA, Kraaij W (2014) The swell knowledge work dataset for stress and user modeling research. In: Proceedings of the 16th international conference on multimodal interaction, Istanbul, Turkey, 12–16 November 2014, pp 291–298
- el Haouij N, Poggi J-M, Sevestre-Ghalila S, Ghozi R, Jaïdane M (2018) AffectiveROAD system and database to assess driver's attention. In: Proceedings of the 33rd annual ACM symposium on applied computing, Pau, France, 9–13 April 2018, pp 800–803

26. Healey J, Picard R (2002) SmartCar: detecting driver stress. In: Proceedings of the 15th international conference on pattern recognition, Barcelona, Spain, 6 August 2002, pp 218–221
27. Shi Y, Nguyen MH, Blitz P, French B, Fisk S, de la Torre F, Smailagic A, Siewiorek DP, al'Absi M, Ertin E et al (2010) Personalized stress detection from physiological measurements. In: Int. symp. qual. life technol. 2010, pp 28–29. http://www.humansensing.cs.cmu.edu/sites/default/files/8stress_detect.pdf. Accessed 2 Oct 2022
28. Muaremi A, Arnrich B, Tröster G (2013) Towards measuring stress with smartphones and wearable devices during workday and sleep. *Bionanoscience* 3:172–183
29. Hosseini S, Gottumukkala R, Katragadda S, Bhupatiraju RT, Ashkar Z, Borst CW, Cochran K (2022) A multimodal sensor dataset for continuous stress detection of nurses in a hospital. *Sci Data* 9:255
30. Iqbal T, Redon-Lurbe P, Simpkin AJ, Elahi A, Ganly S, Wijns W, Shahzad A (2021) A sensitivity analysis of biophysiological responses of stress for wearable sensors in connected health. *IEEE Access* 9:93567–93579
31. Iqbal T, Elahi A, Redon P, Vazquez P, Wijns W, Shahzad A (2021) A review of biophysiological and biochemical indicators of stress for connected and preventive healthcare. *Diagnostics* 11:556
32. Patil M, Madankar A, Khandait P (2019) Heart rate monitoring system. In: 2019 5th international conference on advanced computing and communication systems (ICACCS). IEEE
33. Phan D et al (2015) Smartwatch: performance evaluation for long-term heart rate monitoring. In: 2015 International symposium on bioelectronics and bioinformatics (ISBB). IEEE
34. Valenzuela F et al (2020) An IoT-based glucose monitoring algorithm to prevent diabetes complications. *Appl Sci* 10(3):921
35. Nasser AR et al (2021) Iot and cloud computing in health-care: a new wearable device and cloud-based deep learning algorithm for monitoring of diabetes. *Electronics* 10(21):2719
36. AbdulGhaffar A et al (2020) Internet of things based multiple disease monitoring and health improvement system. *J Ambient Intell Humaniz Comput* 11:1021–1029
37. Imtiaz MH et al (2019) Development of a smart IoT charger for wearable cigarette smoking monitor. In: 2019 SoutheastCon. IEEE
38. Saleh AI, Talaat FM, Labib LM (2019) A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. *Artif Intell Rev* 51(3):403–443
39. Fratello M, Tagliaferri R (2018) Decision trees and random forests. *Encycl Bioinform Comput Biol ABC Bioinform* 1
40. <https://www.kaggle.com/datasets/laavanya/stress-level-detection>
41. Talaat FM, Gamel SA (2022) RL based hyper-parameters optimization algorithm (ROA) for convolutional neural network. *J Ambient Intell Human Comput*. <https://doi.org/10.1007/s12652-022-03788-y>
42. Talaat FM, Ali SH, Saleh AI, Ali HA (2020) Effective cache replacement strategy (ECRS) for real-time fog computing environment. *Cluster Comput*. <https://doi.org/10.1007/s10586-020-03089-z>
43. Hassan E, El-Rashidy N, Talaat FM (2022) Review: mask R-CNN models. <https://doi.org/10.21608/njccs.2022.280047>
44. ZainEldin H, Gamel SA, El-Kenawy ES, Alharbi AH, Khafaga DS, Ibrahim A, Talaat FM (2022) Brain tumor detection and classification using deep learning and sine-cosine fitness grey wolf optimization. *Bioengineering* 10(1):18. <https://doi.org/10.3390/bioengineering10010018>
45. El-Rashidy N, ElSayed NE, El-Ghamry A, Talaat FM (2022) Prediction of gestational diabetes based on explainable deep learning and fog computing. *Soft Comput*. <https://doi.org/10.1007/s00500-022-07420-1>
46. El-Rashidy N, ElSayed NE, El-Ghamry A, Talaat FM (2022) Utilizing fog computing and explainable deep learning techniques for gestational diabetes prediction. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-022-08007-5>
47. El-Balka RM et al (2022) Enhancing the performance of smart electrical grids using data mining and fuzzy inference engine. *Multimed Tools Appl* 81(23):33017–33049
48. Talaat FM (2022) Effective deep q-networks (edqn) strategy for resource allocation based on optimized reinforcement learning algorithm. *Multimed Tools Appl* 1–17
49. Talaat FM, Gamel SA (2022) Predicting the impact of no. of authors on no. of citations of research publications based on neural networks. *J Ambient Intell Humaniz Comput*. <https://doi.org/10.1007/s12652-022-03882-1>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.