Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.038589
Article

Priority Detector and Classifier Techniques Based on ML for the IoMT

Rayan A. Alsemmeari[1,*], Mohamed Yehia Dahab[2], Badraddin Alturki[1] and Abdulaziz A. Alsulami[3]

[1]Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
[2]Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
[3]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia

*Corresponding Author: Rayan A. Alsemmeari. Email: ralsemmeari@kau.edu.sa

Received: 20 December 2022; Accepted: 06 June 2023; Published: 30 August 2023

Abstract: Emerging telemedicine trends, such as the Internet of Medical Things (IoMT), facilitate regular and efficient interactions between medical devices and computing devices. The importance of IoMT comes from the need to continuously monitor patients' health conditions in real-time during normal daily activities, which is realized with the help of various wearable devices and sensors. One major health problem is workplace stress, which can lead to cardiovascular disease or psychiatric disorders. Therefore, real-time monitoring of employees' stress in the workplace is essential. Stress levels and the source of stress could be detected early in the fog layer so that the negative consequences can be mitigated sooner. However, overwhelming the fog layer with extensive data will increase the load on fog nodes, leading to computational challenges. This study aims to reduce fog computation by proposing machine learning (ML) models with two phases. The first phase of the ML model assesses the priority of the situation based on the stress level. In the second phase, a classifier determines the cause of stress, which was either interruptions or time pressure while completing a task. This approach reduced the computation cost for the fog node, as only high-priority records were transferred to the fog. Low-priority records were forwarded to the cloud. Four ML approaches were compared in terms of accuracy and prediction speed: K-nearest neighbors (KNN), a support vector machine (SVM), a bagged tree (BT), and an artificial neural network (ANN). In our experiments, ANN performed best in both phases because it scored an F_1 score of 99.97% and had the highest prediction speed compared with KNN, SVM, and BT.

Keywords: Machine learning; priority detector; Internet of Medical Things; IoMT; fog computing; cloud computing

1 Introduction

When confronted with a challenging and stressful scenario, people may react to the stress by becoming incredibly anxious or feeling under pressure. In medicine, stress is a psycho-physiological condition that can be brought on by various stressors, including discomfort, annoyance, sadness, and anxiety [1]. Workplace stress can put pressure on an individual and cause a change in their mental or physical behavior that significantly impacts the organization and its other employees [2]. Workplace stress is a physical and emotional reaction that occurs when an employee does not have the skills, ability, or resource to complete a task [3]. Stress can lead to health problems, like heart disease and psychological disorders [4]. Additionally, stress can cause financial losses because it can reduce staff productivity [5].

The Internet of Things (IoT) is considered a critical component in healthcare systems, and it is called the Internet of Medical Things (IoMT) [6]. The IoMT comprises sensors, a fog layer, and a cloud layer, as shown in Fig. 1 [7]. Sensors are crucial in the IoMT because they collect different types of data relating to health, and the data are used to diagnose various disorders and their severity [1]. Sensor readings are sent to the fog layer via IoT devices. The data can be processed in the fog layer and used to monitor the patient. Although fog nodes offer various advantages, such as decentralization, a fast response time, and mobility [7], they have limited storage and processing capacity. Thus, if there are large amounts of data, they are sent to the cloud. However, because of communication difficulties like latency and high bandwidth costs, cloud computing can be inappropriate for IoMT applications that depend on real-time processing [8,9].

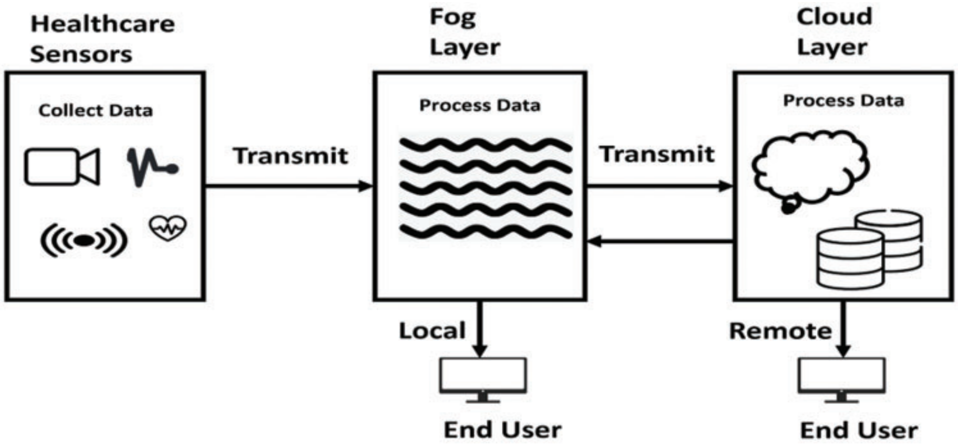

Figure 1: Architecture of the IoMT

Machine learning (ML) can autonomously detect a person's stress level by using physiological data collected during stressful conditions. This can aid in monitoring their stress and preventing harmful stress-related disorders. Different ML and deep-learning approaches have been used to detect stress and determine whether a person is stressed [10,11]. An IoMT system can harness the power of ML to improve the detection of health issues by analyzing data collected by sensors [12]. An early diagnosis can save lives; however, due to the limitations of fog nodes, to avoid overwhelming the fog layer with massive amounts of data [13], only essential data (i.e., priority records) should be handled by the fog layer. Thus, an ML model could be implemented in the IoMT to assess the priority of the data [14]. The ML model can use the features collected from sensors as its input, and then assess the status of the healthcare record and set its priority. Moreover, ML can be used to improve the scheduling of IoT traffic [15]. For example, an ML system could decide which data are to be processed in the fog layer and which are be processed in the cloud layer based on their priority. High-priority records could be handled in the fog layer, while the remaining data could be processed in the cloud.

To assess workplace stress, this paper proposes an ML system with two phases, both of which run locally in the fog layer. The first phase determines the priority of employees' records, based on whether they are stressed or not. Records from employees who have been identified as stressed are sent for further analysis. The second phase attempts to classify the cause of the stress.

The main contributions of this research are as follows:

- We propose a priority-based ML detector to reduce the amount of computation in the fog layer. Only high-priority records are processed in this layer.
- We studied the prediction accuracy of K-nearest neighbors (KNN), a support vector machine (SVM), a bagged tree (BT), and an artificial neural network (ANN). We evaluated each ML model using the SWELL knowledge-work dataset [16,17] using precision, recall, F_1 score, specificity, and accuracy.
- We evaluated the efficiency of each ML model by recording the training time and prediction speed.
- According to our experiments, ANN is the most suitable model for phases 1 and 2 because its F_1 score was 99.97% in phase 1% and 100% in phase 2. In addition, ANN needed the minimum amount of training time and had the highest prediction speed compared with the other models.

The remainder of this paper is organized as follows: Section 2 reviews the literature relevant to this research. Section 3 discusses the dataset used in our experiments to evaluate our model. Section 4 presents the system architecture proposed in this research. Section 5 discusses the experiment and research findings. Finally, Section 6 presents our conclusions and recommendations.

2 Literature Review

The IoT produces an enormous amount of data that are often processed based on priority [18]. However, the devices at the edge of the IoT network have limited and heterogeneous resources. Jobs are distributed to these devices in a diverse environment [19]. Jobs should be assigned to relevant nodes with sufficient resources. Efficiently allocating jobs on heterogeneous devices can improve the accuracy and efficiency of the process and also increase central processing unit (CPU), random access memory (RAM), storage, and bandwidth usage while maintaining the delay, latency, and energy. Scheduling in fog computing includes task scheduling and resource allocation [20]. Task scheduling is important in the fog when there are a large number of diverse jobs waiting to be executed by devices with resource constraints. Jobs are queued based on priority. An efficient task scheduling method can improve performance while reducing the processing time [21]. Resource allocation focuses on efficiently distributing jobs with various quality of service (QoS) criteria to heterogenous and resource-constrained devices to speed up responses and reduce energy, resource usage, and costs [22]. Several surveys [18,23–26] have discussed state-of-the-art scheduling in fog computing. However, those papers did not focus on recent ML scheduling approaches. Scheduling approaches using ML have been reviewed [27]. The authors of that paper illustrated the applicability of ML for resource management in fog computing.

2.1 Traditional Job Scheduling

A prioritized and efficient resource algorithm that aims to reduce costs and response times has been proposed for a fog–cloud architecture [28]. Once a job is assigned, its priority is computed based on its deadline. Prioritized jobs are distributed to the fog layer for processing. In the fog layer, several small data centers and constrained devices communicate with each other. If no fog device can handle the job because of a lack of resources, the job is sent to the cloud for processing. This algorithm was implemented in the simulator *CloudAnalyst* [29]. The results were promising, as this prioritized job scheduling reduced costs and response times. To reduce network and energy consumption, as well as latency, four algorithms—namely first come, first served (FCFS), shortest job first (SJF), round robin, and genetic programming (GP) [30]—were implemented in the simulator *iFogSim* [31]. The FCFS algorithm had lower energy consumption and latency than the other three algorithms.

A job scheduling algorithm based on the weighted round-robin (WRR) algorithm has been developed [32]. The algorithm chooses the ideal fog device for processing preventative jobs based on their available computational power and remaining energy. If the fog device is unable to process the job, then the job is distributed either to other fog devices or to the cloud. The spanning-tree protocol was used in the fog network to collect and route data. The authors used commonly used simulators like *Network Simulator* (*NS3*) [33] and *iFogSim*. The latency of the proposed method was better compared to other job scheduling techniques.

Traditional algorithms are usually static. Therefore, they perform best when the workload does not change. Thus, these algorithms are unsuitable for processing variable workloads if task priorities are dynamic or the fog layer is unpredictable.

2.2 Using ML for Scheduling in Fog Computing

A scheduling method called Quantum Computing Inspired-Neural Network (QCi-NN) has been developed, especially for real-time scheduling [34]. It has three layers: monitoring, learning, and prediction. The monitoring layer collects load processing requirements from a temporal database. During the learning phase, QCi-NN records its observations of the temporal behavior (time of arrival, capacity needed, and processing time) of the jobs given to a particular device in the temporal database. Finally, the computing node is chosen by the prediction layer using probabilistic quantum parameters, temporal behavior, and database information. Notably, once the system has reached a suitable level of stability and a job is issued by an IoT device, the fog node is automatically allocated using probabilistic predictors. The proposed strategy had good results regarding task execution time and energy consumption. However, job dependencies were not considered in that work. Moreover, it can be crucial to reschedule jobs if high waiting times are too high. Therefore, an approach called Clipped Double Deep Q-learning has been proposed based on deep reinforcement learning [35]. This job scheduling method reduces energy, costs, and latency (service delay). Each fog node has a parallel scheduler, which aims to reduce the waiting time of jobs while optimizing the allocation of resources through parallel queueing. Jobs are assigned to servers according to their latency and length. If a job has a high waiting time, then a dual-queue strategy is used to reschedule jobs. An algorithm called Brain-Inspired Rescheduling Decision-making uses actor–critic reinforcement learning to meet job deadlines through rescheduling [36]. To mimic voluntary motor movements, the system mirrors the human brain's decision-making paradigm. When rescheduling preempted jobs to fog nodes, it ensures the deadlines are met by maintaining the optimal performance of the fog nodes through load balancing.

A job scheduling method that considers energy consumption in a fog computing context with software-defined networking has been proposed [37]. A method based on deep reinforcement learning was developed for dynamic job scheduling and assignment when energy is limited. The proposed method guarantees lower latency and energy usage. A Q-learning strategy based on reinforcement learning was proposed for latency and privacy-aware job scheduling that considers QoS latency requirements [38]. Jobs are divided into low, medium, and high categories based on the capacity of fog devices and their computation requirements. The proposed method performed well while ensuring task security with fewer delays and conflicts when allocating jobs.

2.3 Handling Big Data in the Cloud and in the Fog Layer

In the healthcare industry, patient wearables, medical gadgets, and the different sensors used in medical centers all produce vast amounts of data. Healthcare applications that can quickly process the data can enhance patient care and the transport system by optimizing care procedures and minimizing the overall cost for each patient. Most smart city applications, including smart healthcare, have started using the powerful capability of cloud computing for data analytics. However, applications that require a large amount of data or are sensitive to time might not be suitable for cloud computing because the servers in the cloud may be too distant from the devices at the edge of the network. The data sent may clog up the network [39]. Good examples are processing electrocardiogram (ECG) signals, which must be done in real time, and processing the huge amounts of data from magnetic resonance imaging. The healthcare sector needs data to be analyzed in real time near the edge of the network.

Rather than sending all the data from sensors to cloud servers for storage and computation, the data can be transferred to fog gateways, where the data are fused, analyzed, or processed. This can decrease the amount of data that needs to be sent to and stored on the cloud, thus lowering expenses, reducing latency, and better utilizing network bandwidth [40]. However, this does not mean that the data processing and storage power of the cloud are not exploited. Instead, data can be distributed to either the fog or the cloud based on the data analytics needs of applications. For example, real-time analysis applications might process data in a fog node. In contrast, applications that do not require real-time analysis can run in the cloud. Such a system can process data effectively by using the power of both the fog and the cloud.

A platform based on the fog and the cloud has been developed to analyze big data from the IoT [41]. In this architecture, the main component is called the IoT management broker. It can handle a number of requests from various smart houses. Data preparation, mining, categorization, prediction, and visualization are methods of managing data in fog nodes. Fog computing has been used to solve data management problems in healthcare [42]. Data combination, integration, and aggregation are all components of data fusion from large, diverse datasets. An architecture for big data has been developed that spreads the data analytics over three layers, namely the cloud, the fog, and the edge, to exploit the strengths of each layer [43]. The edge layer controls the actuators and devices, the fog layer provides data aggregation and ingestion, and the cloud applies other analytics that needs high-performance computing resources.

2.4 QoS Requirements of Healthcare Applications

QoS requirements must be met if smart applications are to be deployed in a fog environment. For instance, a connected automobile application needs to communicate in real time and also needs mobility assistance. Augmented reality applications require high processing power and real-time communications. Furthermore, healthcare services need continuous connectivity and low energy costs [44]. In healthcare, the various services have a diverse set of QoS requirements, including low latency, real-time processing, high reliability, low energy consumption, low costs, high availability, and awareness of location and mobility [45]. However, the benefits of cloud computing, such as increased data storage, affordable resources, frequent maintenance, low latency, location awareness, and extensive content distribution, have not yet been fully exploited [46]. For real-time healthcare applications, latency may be too high for systems that deliver data to the cloud, process the data there, and then send the results back to the application. Additionally, any failure of a cloud server or a connectivity loss might affect the availability of the application and could be fatal. To satisfy the QoS requirements, it is advisable to reduce the amount of data distributed across the network. Due to its location, awareness of location, geographical distribution, and online processing, the fog can be a good fit for healthcare applications that require low power wastage, tight bandwidth, and low processing power and that rely on resource-constrained devices.

The IoT and fog computing can be used by the medical industry for remote monitoring, allowing healthcare practitioners to provide seamless consultations and support to their patients. There has been a rapid increase in providing remote healthcare services. The associated IoT products are collectively called the IoMT. In particular, intelligent stress monitoring has gained prominence due to rising levels of stress. Therefore, many researchers have started monitoring stress using the IoMT. There are several datasets relevant to stress, including the multimodal SWELL knowledge-work dataset [43].

2.5 Machine Learning

This paper compares four ML algorithms: KNN, ANN, SVM, and BT. KNN is a supervised ML algorithm that uses proximity to make a prediction or classification about the grouping of an individual data point. It predicts a class by calculating the distance between a new data point and the training points. The KNN algorithm presumes that similar data exists nearby [47]. ANNs are algorithms that imitate the sophisticated human brain. They mimic how neurons in the human nervous system can learn from experience. An ANN has three layers. The first layer is the input layer. Each node in this layer represents a feature in the sample dataset. The second layer is a hidden layer. This is where all the computations and processing take place. This layer correlates the input features with possible outcomes by applying weights to the inputs through an activation function [48]. Finally, the third layer is the output layer, which delivers the outcome. This layer has the same number of nodes as there are categories. SVM is also a supervised ML algorithm that can be utilized for classification and regression problems, although it performs best in classification tasks. SVM is a robust algorithm that can classify data by finding the optimal hyperplane (line) that best separates data points [49]. BT is an algorithm that can be combined with any other ML algorithm. However, using it with a decision tree is useful as the BT can reduce the high variance and thus, the test errors. Furthermore, this algorithm aims to enhance decision-making by using many decision trees instead of a single decision tree [50].

Features extracted from the benchmark SWELL knowledge-work dataset have been classified with KNN and SVM [51]. The SWELL knowledge-work dataset is an important dataset for stress. The best features considerably affect the detection of stress. An attempt was made to identify the optimum set of features that yields the highest classification accuracy. The proposed methodology produced state-of-the-art results, with accuracies for KNN and SVM of, respectively, 76.9% and 86.4%. Scalable deep-learning algorithms have been explored for rapidly and precisely detecting stress and its impact in real time [52]. Common problems with cloud-based computation, such as excessive latency and a lack of privacy, were addressed by performing inference locally on low-cost wearables and other devices with limited resources. A customized algorithm called FastGRNN was developed to detect stress and emotion. For three datasets widely used in the field, namely WESAD, the SWELL knowledge-work dataset, and DREAMER, the algorithm had very short inference durations yet excellent accuracy. The accuracy with the SWELL knowledge-work dataset was 87.78%.

A self-supervised learning method to identify emotions using ECG signals has been developed [53]. The proposed architecture comprises two primary networks. In the self-supervised learning stage, unlabeled data are used to train the first network to identify particular predetermined signal alterations. The weights of the convolutional layers in this network are then passed to the emotion detection model, where two dense layers are learned to categorize arousal and valence scores. This self-supervised strategy assists the model in learning the ECG feature manifold necessary for emotion identification, with the model performing as well as or better than the fully supervised version. The accuracies for the fully supervised convolutional neural network (CNN) and the self-supervised CNN were, respectively, 89.4% and 93.3%.

A method for representing model uncertainty in Bayesian neural networks via approximations utilizing Monte Carlo dropout has been proposed [54]. For active learning, this was paired with appropriate acquisition functions. In experiments with the SWELL knowledge-work dataset, this framework delivered a significant efficiency improvement during inference, with a significantly low number of obtained pool values during active learning over multiple acquisition functions. An accuracy of 91.92% was achieved using active Bayesian learning. In addition, a multimodal artificial intelligence (AI) framework has been developed to track a person's work habits and stress levels [55]. It effectively identified workload-related stress by concatenating readings from a diverse range of sensors. This study made two contributions. First, it applied a multimodal AI-based fusion technique to identify a stress pattern across time. Second, the method detected stress and its intensity. With the SWELL knowledge-work dataset, an accuracy of 96.09% was realized for stress detection and categorization.

A fog computing approach has been developed to reduce latency in healthcare applications [56]. Fog computing and a random forest ML technique were used to segregate patient data and improve latency automatically. The primary goal of each planned job was to provide personal care to a patient on time. Random forest classifiers were trained and distributed using data gathered by IoT devices. This technique reduces the number of needless patient trips to various physicians, saving both time and money. This model achieved an overall reduction in latency of between 92% and 95%. Notably, how ML and IoT research can substantially contribute to the fight against illness has been reviewed [57]. Emerging applications of ML for combating diseases, such as COVID-19, include detection, diagnosis, prediction, segmentation, and medication development. This means that ML models could be important in classifying and detecting disease and stress.

The results of this literature review suggest that most technical solutions focus on using non-ML algorithms to distribute data between the fog and the cloud in the IoMT. In contrast, we use ML as a detector and classifier when distributing data in an IoMT architecture. The novelty of this research is the replacement of such non-ML models with ML models because ML models perform better at detection and classification than traditional approaches. Therefore, we use ML to detect the priority of employee records. High-priority data are sent to the fog and low-priority data to the cloud. Finally, the high-priority data are processed by an ML classifier in the fog. Table 1 lists a comparison of technical solutions to distribute data between fog and to cloud that were discussed in the literature.

Table 1: Comparison of literature about methods used to distribute data between fog and cloud

Reference	Method	Type	Cost	Latency	Execution time	Energy	Platform
[28]	Prioritized efficient resource	Non-ML	√	√	x	x	CloudAnalyst
[30]	FCFS, SJF, Round-Robin, GP		x	√	√	√	iFogSim
[31]	FCFS		√	√	x	x	iFogSim
[32]	WRR		x	√	x	x	NS3, iFogSim
[34]	Quantum computing	ML	x	x	√	√	iFogSim
[35]	Clipped double deep Q-learning		√	√	√	√	Python
[36]	Brain-inspired rescheduling decision-making		x	√	x	x	N/A
[37]	Deep reinforcement learning		x	√	√	√	Mininet
[38]	Q-learning		x	√	x	x	Python

3 Dataset

The SWELL knowledge-work dataset [16,17] was used to evaluate the effectiveness and efficiency of the ML model proposed in this research. The dataset was gathered by a group of researchers at Radboud University from 25 participants. The participants were doing routine tasks at work, such as preparing presentations, writing reports, and reading an email. To manipulate the experienced job load, the researchers subjected the participants to typical work stressors, such as interrupting them by sending them unexpected emails or asking them to complete an assigned task before the specified time. Therefore, in this experiment, the participants were in one of three scenarios:

- No stress: A participant was allowed to work on a task and complete it within 45 min without interruption and without being told that they needed to finish it by a specified time.
- Time pressure: A participant was assigned a task and asked to finish it in 2/3 of the time usually needed.
- Interruptions: A participant received eight emails while working on a task. Some of the emails were relevant to the task whereas the others were not.

The following data were collected by sensors:

- Computer interactions, such as keystrokes, mouse movements, and applications used
- Facial recognition, including facial movements, head orientation, action units, and emotion
- Body posture, including joint angles, bone orientation, and distance
- Physiology, such as skin conductance and heart rate

The dataset consists of 34 features and 451,355 records in total. A detailed description of the dataset is available from [16,17]. Table 2 lists the number of records for the no stress, interruption, and time pressure scenarios. Notice that the numbers of no stress and stress records are roughly similar. There are 128,714 stress records caused by interruption and 78,243 stress records caused by time pressure. Fig. 2 summarizes this information. Fig. 2a shows that 54% of the data points contain no stress records, whereas 46% do. Fig. 2b shows that for 62% of the records, the source of stress was interruptions and for 38%, the time pressure.

Table 2: Summary of condition labels in the SWELL knowledge-work dataset

Scenario	Number of records
No stress	244,398
Stress: Interruption	128,714
Stress: Time pressure	78,243
Stress (Total)	206,957

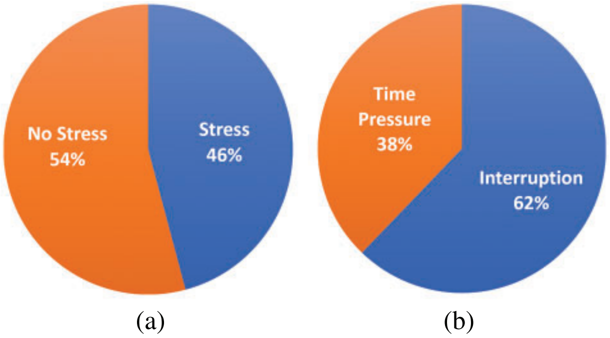

Figure 2: Pie charts for the SWELL knowledge-work dataset: (a) distribution of stress and no stress records and (b) distribution of time pressure and interruption records

4 Methodology

This section describes the research methodology and the system architecture and also explains the flow in the proposed model. The topology of the system architecture is illustrated in Fig. 3, and a flowchart of the proposed system is depicted in Fig. 4. Data are collected using IoMT sensors. The sensor layer has devices capable of running ML algorithms. This architecture can be considered as standard, as it has been used by many researchers [9]. As the first phase, the data collected are transmitted to the ML priority detector, which assesses the priority of the employee record. Immediate action is required if the record is high priority, as it means the employee is under stress, so the date are sent to the fog layer for processing. A low-priority record means the employee is not under stress, and it is transmitted to the cloud because it does not require a fast response. In the second phase, the fog layer processes each high-priority record to find the cause of stress using the ML classifier. In our case, there are two reasons for stress: stress caused by interruption and stress caused by time pressure. The end user can monitor the decisions made by the classifier in the fog layer. After the record is processed in the fog layer, it is sent to the cloud for long-term storage. All records in the cloud can be accessed by the end user.

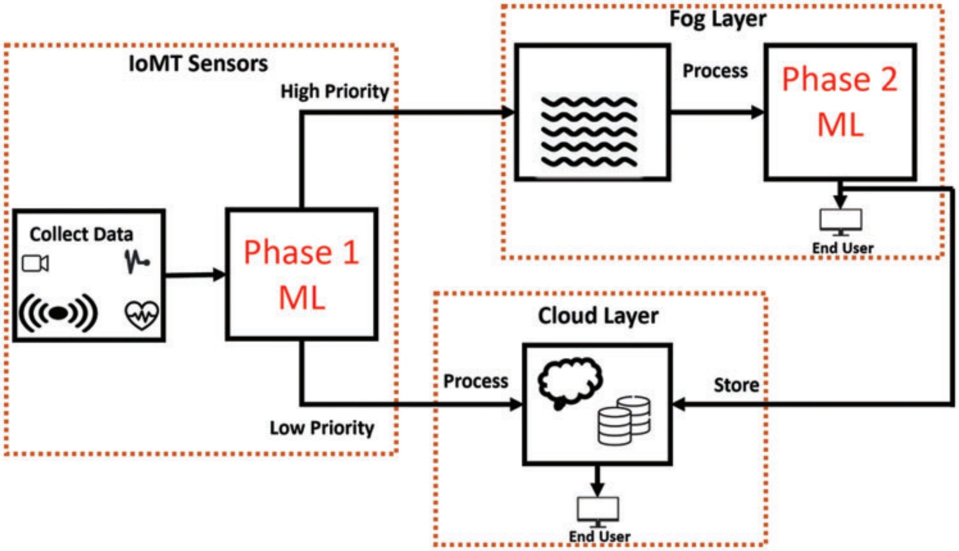

Figure 3: System architecture

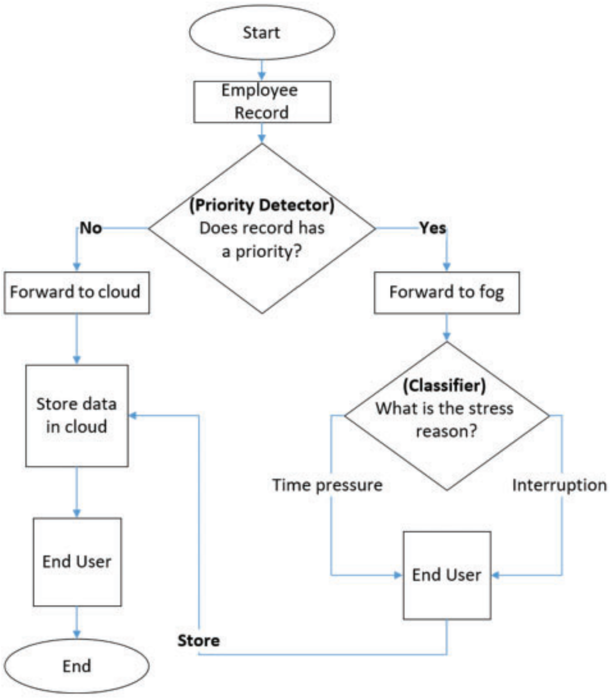

Figure 4: System flowchart

Algorithms 1 and 2 are pseudocode for the proposed models. In Algorithm 1, the number of employee records is assigned to the variable *total*, which, in our case, is 451,355. The variable *length* is the number of features in each record, which is 34. The while loop runs from the first record to the last record. In line 5, the record is processed by ML1 to detect the priority. The record is sent to the fog layer if its priority is high, otherwise it is sent to the cloud. Algorithm 2 shows the process of identifying the source of stress by ML2. The output of ML2 is either “interrupt” or “timePressure”.

Algorithm 1: Priority detector

```
total ← numberOfEmployeeRecords
length ← lengthOfRecord
counter ← 1
while counter <= total do
  x ← ML1(employeeRecords(counter (1:length))
  if x is True then
    fogLayer()
  else
    CloudLayer()
  end if
  counter + +
end while
```

Algorithm 2: fogLayer()

```
y ← ML2(employeeRecords(counter (1:length))
if y = interrupt then
  return employeeRecords(counter, length + 1) = interrupt
else
  return employeeRecords(counter, length + 1) = timePressure
end if
```

5 Results and Discussion

This section explains the experiments and the results obtained. First, the training procedure is described. Then the metrics used to validate the accuracy are explained and discussed. In addition to studying the effectiveness of each ML model, the prediction speeds are calculated and compared.

The four ML algorithms (KNN, ANN, SVM, and BT) were trained separately for each of the two phases. In phase 1, the four ML models were trained and tested with the SWELL knowledge-work dataset. The output of a model was either high or low priority. In phase 2, the same ML algorithms were trained and tested, but the output was now the cause of stress: interruption or time pressure. Section 3 contains detailed information about the dataset, which was partitioned as 80% for training and 20% for testing. The trained models were validated using k-fold cross-validation, a technique used to compute the accuracy of ML models by partitioning the dataset into k sets of equal size. Each set was used as a testing set after training with the other sets [58]. In our case, the k value was equal to 5.

The results of each model were evaluated using precision, recall, F_1 score, specificity, and classification accuracy. Precision is the number of true positives as a percentage of the number of predicted positives [59]. Recall is the number of true positives as a percentage of the number of all positives [59]. The F_1 score is the harmonic mean of recall and precision [59]. Specificity is the percentage of true negatives as a percentage of all negatives [60].

Precision is calculated as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100 \tag{1}$$

where TP is the number of true positives, i.e., where a participant was under stress and was correctly classified. FP is the number of false positives, indicating that a participant was not under stress and was incorrectly classified.

Recall is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN} \times 100 \tag{2}$$

where FN is the number of false negatives, where a participant was under stress and was incorrectly classified.

The F_1 score and specificity are computed as follows:

$$F_1 \text{ score } = \frac{TP}{TP + (FP + FN)/2} \times 100 \tag{3}$$

$$\text{Specificity } = \frac{TN}{TN + FP} \times 100 \tag{4}$$

where TN is the number of true negatives, which indicates that a participant was not under stress and was correctly classified.

Finally, the classification accuracy is calculated using [60,61]

$$\text{Accuracy } = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \tag{5}$$

The experimental results for the ML models plus a non-ML model are listed in Table 3 for each phase. As mentioned above, the ML algorithms are run in two phases. Phase 1 determines the priority of the employee records, and phase 2 determines the cause of the stress. Overall, three ML models (KNN, ANN, and BT) had an accuracy higher than 99%. The reason is that supervised ML models perform better in binary classification than in multi-classification [62]. The low performance of SVM occurred because an SVM generally does not perform well with an imbalanced dataset compared with other methods [63]. Therefore, using two ML models improved the classification accuracy compared to a single ML model. In addition, it is clear from Table 3 that the KNN, ANN, or BT models could be selected for either phase 1 or 2 because they had better accuracy compared with other models. A rule-based system (RBS) was also evaluated in this study. This non-ML model computes the distances between each feature vector and the mean of each class. The performance of RBS was insufficient because the class was predicted using a static threshold.

Table 4 shows the training time and the prediction speed of each ML model. In addition, it shows the setup parameters used in our experiments. According to our experiments in phase 1, ANN had the lowest training time of 2,527.4 s compared with KNN, SVM, and BT. The prediction speed is the number of observations (data points) processed per second. ANN was the fastest model because it processed 130,000 data points per second. In phase 2, ANN was also the fastest model when training or processing data. Fig. 5 compares the training times in seconds for the four ML models. Overall, SVM was the slowest. Fig. 6 shows the prediction speeds. ANN had the highest speeds in comparison with the other models. Therefore, based on the obtained accuracies and prediction speeds, we suggest that ANN is the most suitable model for both phase 1 and phase 2.

To further assess the accuracy of our selected model (ANN), we compared it with other algorithms, as listed in Table 5. For a fair and reasonable comparison, we consider only studies that used the SWELL knowledge-work dataset. Several groups worked on this dataset between 2017 and 2022. Table 4 shows that our ANN performed better than the existing methods.

Table 3: Accuracy of the two-phase models

Model	Precision		Recall		F_1 score		Specificity		Classification accuracy	
	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2
KNN	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
ANN	99.94%	100%	100%	100%	99.97%	100%	99.93%	100%	99.97%	100%
SVM	75.19%	86.00%	67.61%	81.56%	71.20%	83.75%	66.22%	74.82%	67.06%	79.24%
BT	99.93%	99.95%	99.86%	99.98%	99.89%	99.98%	99.92%	99.92%	99.89%	99.98%
RBS	58.32%	54.22%	46.96%	46.28%	52.03%	49.94%	71.61%	76.23%	60.03%	64.90%

Table 4: Training times and prediction speeds

ML	Training time (s)		Prediction speed (Observations per second)		Model hyperparameters
	Phase 1	Phase 2	Phase 1	Phase 2	
KNN	4,218.7	1,399.4	350	580	Number of neighbors: 1; distance metric: Euclidean; distance weight: Equal;
ANN	2,527.4	1,365.2	130,000	110,000	Fully connected layers: 3; 1st layer size: 10; 2nd layer size: 10; 3rd layer size: 10; activation: ReLU; Iteration limit: 1000;
SVM	34,244	3,745.3	410	3,800	Kernel function: Linear; Box constraint level: 1; kernel scale: Automatic; multiclass method: One-*vs.*-One;
BT	4,581.1	1,485.5	37,000	53,000	Ensemble method: Bag; learner type: decision tree; number of learners: 30;

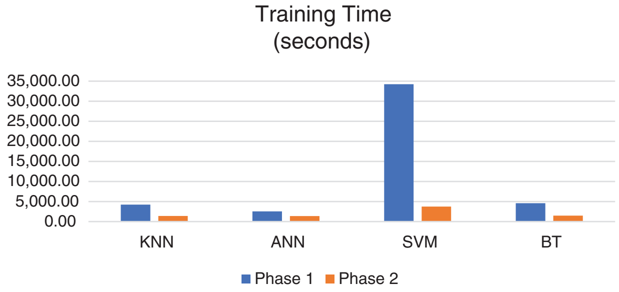

Figure 5: Training times for the four ML algorithms for each phase

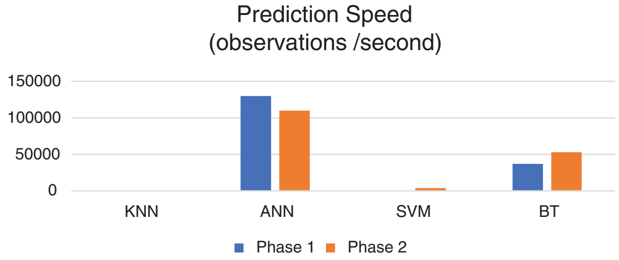

Figure 6: Prediction speeds for the four ML algorithms for each phase

Table 5: Comparison of available methods

Study	Year	Algorithm	Accuracy
Sriramprakash et al. [51]	2017	KNN	76.9%
		SVM	86.4%
Ragav et al. [52]	2019	FastGRNN	87.87%
Sarkar et al. [53]	2020	Fully supervised CNN	89.4%
		Self-supervised CNN	93.3%
Ragav et al. [54]	2020	Active Bayesian learning	91.92%
Walambe et al. [55]	2021	Nasa TLX	96.67%
Our ANN model	2023	ANN	99.97%

6 Conclusions and Feature Work

This paper proposes a priority detection and classification approach using two phases of ML in the IoMT domain. The first phase uses a priority detector to identify the priority of employee records collected from the sensor layer. A record marked as high priority is sent immediately to the fog layer for processing. A low-priority record is instead transmitted to the cloud layer because it does not need instant attention. As a result, this reduces the computation overhead on the fog layer because it processes only high-priority records. In the second ML phase, a stress classifier in the fog layer determines the source of the stress and informs the end user whether the employee faces stress caused by interruption or time pressure. This approach enhances decision-making by autonomously identifying the cause of stress. The proposed architecture could be used in healthcare sectors that have an IoMT with fog and cloud layers. Moreover, it could be applied to similar architectures where the fog layer processes data from IoT devices. The KNN, SVM, ANN, and BT models were compared using the SWELL knowledge-work dataset by calculating standard metrics. The prediction speed was calculated to assess the efficiency of each ML model. In the experiments, our selected ANN scored between 99.97% and 100% for precision, recall, F_1 score, specificity, and classification accuracy, and it outperformed the other models. In addition, the prediction speed of ANN exceeded that of the other models. For future work, we recommend improving the privacy of the data transmitted to the fog and the cloud layers by applying available technology to avoid exposing personal information about an employee within an enterprise.

Acknowledgement: The authors are thankful to the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah 21589, Saudi Arabia, for funding this work under Grant No. IFPIP: 1181-611-1443. The authors, therefore, acknowledge with thanks DSR for technical and financial support.

Funding Statement: This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under Grant No. IFPIP: 1181-611-1443.

Author Contributions: The authors confirm their contribution to the paper as follows: Methodology, R.A.A., B.A., and A.A.A.; conceptualization, R.A.A., and A.A.A.; software, R.A.A. validation, M.Y.D., and B.A.; formal analysis, A.A.A.; investigation, R.A.A.; resources, B.A.; data curation, R.A.A.; writing—original draft preparation, R.A.A. and A.A.A.; writing review and editing, R.A.A, A.A.A., B.A., and M.Y.D.; visualization, B.A.; supervision, A.A.A., and M.Y.D.; project administration, R.A.A.; funding acquisition, R.A.A. and B.A.; All authors reviewed the results and approved the final version of the manuscript.

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

References

[1] 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.

[2] E. D. Valle, S. Palermi, I. Aloe, R. Marcantonio, R. Spera *et al.,* "Effectiveness of workplace yoga interventions to reduce perceived stress in employees: A systematic review and meta-analysis," *Journal of Functional Morphology and Kinesiology*, vol. 5, no. 2, pp. 33, 2020.

[3] W. Salama, A. H. Abdou, S. A. K. Mohamed and H. S. Shehata, "Impact of work stress and job burnout on turnover intentions among hotel employees," *International Journal of Environmental Research and Public Health*, vol. 19, no. 15, pp. 9724, 2022.

[4] L. Pepa, A. Sabatelli, L. Ciabattoni, A. Monteriù, F. Lamberti *et al.,* "Stress detection in computer users from keyboard and mouse dynamics," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 1, pp. 12–19, 2020.

[5] T. Androutsou, S. Angelopoulos, I. Kouris, E. Hristoforou and D. Koutsouris, "A smart computer mouse with biometric sensors for unobtrusive office work-related stress monitoring," in *Proc. EMBC*, Mexico, pp. 7256–7259, 2021.

[6] D. Koutras, G. Stergiopoulos, T. Dasaklis, P. Kotzanikolaou, D. Glynos *et al.,* "Security in IoMT communications: A survey," *Sensors*, vol. 20, no. 17, pp. 4828, 2020.

[7] S. Razdan and S. Sharma, "Internet of Medical Things (IoMT): Overview, emerging technologies, and case studies," *IETE Technical Review*, vol. 39, no. 4, pp. 775–788, 2021.

[8] Z. Ahmadi, M. Haghi Kashani, M. Nikravan and E. Mahdipour, "Fog-based healthcare systems: A systematic review," *Multimedia Tools and Applications*, vol. 80, no. 2021, pp. 1–40, 2021.

[9] B. Farahani, M. Barzegari, F. S. Aliee and K. A. Shaik, "Towards collaborative intelligent IoT eHealth: From device to fog, and cloud," *Microprocessors and Microsystems*, vol. 72, pp. 102938, 2020.

[10] B. Pramod and M. Vani, "Stress detection with machine learning and deep learning using multimodal physiological data," in *2nd Int. Conf. on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, pp. 51–57, 2020.

[11] V. Ponnusamy, J. Christopher Clement, K. C. Sriharipriya and S. Natarajan, "Smart healthcare technologies for massive Internet of Medical Things," in *Efficient Data Handling for Massive Internet of Medical Things: Healthcare Data Analytics*, Cham, Switzerland: Springer International Publishing, pp. 71–101, 2021.

[12] P. Manickam, S. A. Mariappan, S. M. Murugesan, S. Hansda, A. Kaushik *et al.,* "Artificial intelligence (AI) and Internet of Medical Things (IoMT) assisted biomedical systems for intelligent healthcare," *Biosensors*, vol. 12, no. 8, pp. 562, 2022.

[13] M. M. Bukhari, T. M. Ghazal, S. Abbas, M. Khan, U. Farooq *et al.,* "An intelligent proposed model for task offloading in fog-cloud collaboration using logistics regression," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–25, 2022.

[14] O. Cheikhrouhou, R. Mahmud, R. Zouari, M. Ibrahim, A. Zaguia *et al.,* "One-dimensional CNN approach for ECG arrhythmia analysis in fog-cloud environments," *IEEE Access*, vol. 9, pp. 103513–103523, 2021.

[15] U. K. Lilhore, A. L. Imoize, C. T. Li, S. Simaiya, S. K. Pani *et al.,* "Design and implementation of an ML and IoT based adaptive traffic-management system for smart cities," *Sensors*, vol. 22, no. 8, pp. 2908, 2022.

[16] K. Nkurikiyeyezu, K. Shoji, A. Yokokubo and G. Lopez, "Thermal comfort and stress recognition in office environment," in *Proc. HEALTHINF*, Prague, Czech Republic, pp. 256–263, 2019.

[17] K. Nkurikiyeyezu, A. Yokokubo and G. Lopez, "The effect of person-specific biometrics in improving generic stress predictive models," *Sensors and Materials*, vol. 32, no. 2, pp. 703–722, 2022.

[18] B. Jamil, H. Ijaz, M. Shojafar, K. Munir and R. Buyya, "Resource allocation and task scheduling in fog computing and internet of everything environments: A taxonomy, review, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11, pp. 1–37, 2022.

[19] C. H. Hong and B. Varghese, "Resource management in fog/edge computing: A survey on architectures, infrastructure, and algorithms," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–37, 2019.

[20] B. M. Nguyen, H. Thi Thanh Binh, T. The Anh and D. Bao Son, "Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud–fog computing environment," *Applied Sciences*, vol. 9, no. 9, pp. 1730, 2019.

[21] M. R. Alizadeh, V. Khajehvand, A. M. Rahmani and E. Akbari, "Task scheduling approaches in fog computing: A systematic review," *International Journal of Communication Systems*, vol. 33, no. 16, pp. e4583, 2020.

[22] I. B. Lahmar and K. Boukadi, "Resource allocation in fog computing: A systematic mapping study," in *Proc. FMECI, Electr. Network*, Paris, France, pp. 86–93, 2020.

[23] X. Yang and N. Rahmani, "Task scheduling mechanisms in fog computing: Review, trends, and perspectives," *Kybernetes*, vol. 50, no. 1, pp. 22–38, 2020.

[24] S. Bansal, H. Aggarwal and M. Aggarwal, "A systematic review of task scheduling approaches in fog computing," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 9, pp. e4523, 2022.

[25] M. S. U. Islam, A. Kumar and Y. C. Hu, "Context-aware scheduling in fog computing: A survey, taxonomy, challenges and future directions," *Journal of Network and Computer Applications*, vol. 180, pp. 103008, 2021.

[26] K. Matrouk and K. Alatoun, "Scheduling algorithms in fog computing: A survey," *International Journal of Networked and Distributed Computing*, vol. 9, no. 1, pp. 59–74, 2021.

[27] M. Fahimullah, S. Ahvar and M. Trocan, "A review of resource management in fog computing: Machine learning perspective," arXiv Preprint, arXiv:2209.03066, 2022.

[28] T. Choudhari, M. Moh and T. Moh, "Prioritized task scheduling in fog computing," in *Proc. ACMSE 2018 Conf.*, New York, NY, USA, pp. 1–8, 2018.

[29] B. Wickremasinghe, R. N. Calheiros and R. Buyya, "CloudAnalyst: A CloudSim-based visual modeller for analysing cloud computing environments and applications," in *Proc. 24th IEEE Int. Conf. on Advanced Information Networking and Applications*, Perth, Australia, pp. 446–452, 2010.

[30] M. Mtshali, H. Kobo, S. Dlamini, M. Adigun and P. Mudali, "Multi-objective optimization approach for task scheduling in fog computing," in *Proc. icABCD*, Drakensberg Resort, Winterton, KZN, South Africa, pp. 1–6, 2019.

[31] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh and R. Buyya, "IFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[32] A. M. Alsmadi, R. M. Ali Aloglah, A. A. Smadi, M. Alshabanah, D. Alrajhi *et al.,* "Fog computing scheduling algorithm for smart city," *International Journal of Electrical & Computer Engineering*, vol. 11, no. 3, pp. 2088–2228, 2021.

[33] Nsnam, "N S3," [Online]. Available: https://www.nsnam.org/

[34] M. Bhatia, S. K. Sood and S. Kaur, "Quantum-based predictive fog scheduler for IoT applications," *Computers in Industry*, vol. 111, pp. 51–67, 2019.

[35] S. Swarup, E. M. Shakshuki and A. Yasar, "Energy efficient task scheduling in fog environment using deep reinforcement learning approach," *Procedia Computer Science*, vol. 191, pp. 65–75, 2021.

[36] B. Nair and S. M. S. Bhanu, "A reinforcement learning algorithm for rescheduling pre-empted tasks in fog nodes," *Journal of Scheduling*, vol. 25, no. 5, pp. 547–565, 2022.

[37] B. Sellami, A. Hakiri, S. B. Yahia and P. Berthou, "Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network," *Computer Networks*, vol. 210, pp. 108957, 2022.

[38] M. M. Razaq, S. Rahim, B. Tak and L. Peng, "Fragmented task scheduling for load-balanced fog computing based on q-learning," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–9, 2022.

[39] E. Badidi, Z. Mahrez and E. Sabir, "Fog computing for smart cities' big data management and analytics: A review," *Future Internet*, vol. 12, no. 11, pp. 190, 2020.

[40] A. A. Mutlag, M. K. A. Ghani, N. Arunkumar, M. A. Mohammed and O. Mohd, "Enabling technologies for fog computing in healthcare IoT systems," *Future Generation Computer Systems*, vol. 90, pp. 62–78, 2019.

[41] A. Yassine, S. Singh, M. S. Hossain and G. Muhammad, "IoT big data analytics for smart homes with fog and cloud computing," *Future Generation Computer Systems*, vol. 91, pp. 563–573, 2019.

[42] B. Qin, H. Tang, H. Chen, L. Cui, J. Liu *et al.,* "Review on big data application of medical system based on fog computing and IoT technology," *Journal of Physics: Conference Series*, vol. 1423, pp. 012030, 2019.

[43] J. Díaz-de Arcaya, R. Miñon and A. I. Torre-Bastida, "Towards an architecture for big data analytics leveraging edge/fog paradigms," in *Proc. ECSA2019*, Paris, France, pp. 173–176, 2019.

[44] A. Ksentini, M. Jebalia and S. Tabbane, "IoT/cloud-enabled smart services: A review on QOS requirements in fog environment and a proposed approach based on priority classification technique," *International Journal of Communication Systems*, vol. 34, no. 2, pp. e4269, 2021.

[45] A. Mosenia, S. Sur-Kolay, A. Raghunathan and N. K. Jha, "Wearable medical sensor-based system design: A survey," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 3, no. 2, pp. 124–138, 2017.

[46] S. Koldijk, M. Sappelli, S. Verberne, M. A. Neerincx and W. Kraaij, "The SWELL knowledge work dataset for stress and user modeling research," in *Proc. 16th Int. Conf. on Multimodal Interaction*, New York, NY, USA, pp. 291–298, 2014.

[47] W. Xing and Y. Bei, "Medical health big data classification based on KNN classification algorithm," *IEEE Access*, vol. 8, pp. 28808–28819, 2020.

[48] Q. A. Al-Haija and A. A. Alsulami, "High performance classification model to identify ransomware payments for heterogeneous bitcoin networks," *Electronics*, vol. 10, no. 17, pp. 2113, 2021.

[49] M. Mohammadi, T. A. Rashid, S. H. T. Karim, A. H. M. Aldalwie, Q. T. Tho *et al.,* "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems," *Journal of Network and Computer Applications*, vol. 178, pp. 102983, 2021.

[50] E. R. Widasari, K. Tanno and H. Tamura, "Automatic sleep disorders classification using ensemble of bagged tree based on sleep quality features," *Electronics*, vol. 9, no. 3, pp. 512, 2020.

[51] S. Sriramprakash, V. D. Prasanna and O. R. Murthy, "Stress detection in working people," *Procedia Computer Science*, vol. 115, pp. 359–366, 2017.

[52] A. Ragav, N. H. Krishna, N. Narayanan, K. Thelly and V. Vijayaraghavan, "Scalable deep learning for stress and affect detection on resource-constrained devices," in *Proc. ICMLA*, Orlando, FL, USA, pp. 1585–1592, 2019.

[53] P. Sarkar and A. Etemad, "Self-supervised learning for ECG-based emotion recognition," in *Proc. ICASSP*, Barcelona, Spain, pp. 3217–3221, 2020.

[54] A. Ragav and G. K. Gudur, "Bayesian active learning for wearable stress and affect detection," arXiv Preprint, arXiv:2012.02702, 2020.

[55] R. Walambe, P. Nayak, A. Bhardwaj and K. Kotecha, "Employing multimodal machine learning for stress detection," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–12, 2021.

[56] A. Kishor, C. Chakraborty and W. Jeberson, "Intelligent healthcare data segregation using fog computing with Internet of Things and machine learning," *International Journal of Engineering Systems Modelling and Simulation*, vol. 12, no. 2–3, pp. 188–194, 2021.

[57] C. Chakraborty and A. Abougreen, "Intelligent Internet of Things and advanced machine learning techniques for covid-19," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 7, no. 26, pp. 1–14, 2021.

[58] M. Maray, M. Alghamdi and M. B. Alazzam, "Diagnosing cancer using IoT and machine learning methods," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–9, 2022.

[59] H. Azeroual, I. D. Belghiti and N. Berbiche, "A framework for implementing an ML or DL model to improve intrusion detection systems (IDS) in the NTMA context, with an example on the dataset (CSE-CICIDS2018)," in *Proc. ICEAS'22*, Budapest, Hungary, pp. 02005, 2022.

[60] A. A. Nancy, D. Ravindran, P. D. Raj Vincent, K. Srinivasan and D. Gutierrez Reina, "IoT-cloud-based smart healthcare monitoring system for heart disease prediction via deep learning," *Electronics*, vol. 11, no. 15, pp. 2292, 2022.

[61] A. A. Alsulami, Q. Abu Al-Haija, A. Alqahtani and R. Alsini, "Symmetrical simulation scheme for anomaly detection in autonomous vehicles based on LSTM model," *Symmetry*, vol. 14, no. 7, pp. 1450, 2022.

[62] T. Acharya, I. Khatri, A. Annamalai and M. F. Chouikha, "Efficacy of machine learning-based classifiers for binary and multi-class network intrusion detection," in *Proc. I2CACIS*, Shah Alam, Malaysia, pp. 402–407, 2021.

[63] H. Yu, C. Dun, X. Yang, S. Zheng and H. Zou, "Fuzzy support vector machine with relative density information for classifying imbalanced data," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 12, pp. 2353–2367, 2019.