Journal of Reliable Intelligent Environments (2021) 7:341–353
https://doi.org/10.1007/s40860-020-00128-9

ORIGINAL ARTICLE

A cost-effective alertness-rating tool to enable situational awareness among on-duty static security guards in Covid-19 pandemic

Debabrata Bej[1] · Shubhranil Chakraborty[2] · Sekh Arif Mahammad[2] · Ashis Kumar Mal[1]

Received: 8 August 2020 / Accepted: 22 December 2020 / Published online: 9 January 2021

Abstract
The workload of the static security guards has doubled due to the Covid-19 outbreak. In addition to their regular duties, they undertake some additional tasks to evaluate each individual's body temperature and welcome them with a hand sanitizer. In this scenario, their situational awareness is hugely desirable to perform these activities for the entire campus's safety. This situational awareness of guards means their ability to observe, inspect, and make the right decisions. However, due to their fatigue and other secondary activities, such as cell phone use, they cannot perform their duties correctly. In this context, this paper presents a method for sending random alarms in real-time to the on-duty guards, who are executing their work at the campus gates, remotely monitoring the alertness throughout the day from the head security office. For alertness detection, the system uses a simple client–server model. The system is designed using NodeMCU Wi-Fi modules. The algorithm of the Client, server, and repeater has been developed. The prototype has been tested by placing it on the working individuals' desk inside the departmental lab inside the campus. The system records the response time of the working individuals. These data are further used to calculate their percentage of alertness. In addition, an alertness-rating/scoring method has been developed to improve their work performance. This system can be an economical solution to enable the awareness of on-duty guards.

Keywords Situational awareness · Surveillance system · Distraction · Alert fatigue · Alertness · Cost efficiency

1 Introduction

Security guards have a primary responsibility to watch after the premises, allow entries at the gates, and protect the locations. Since the Covid-19 outbreak, they carry out several additional activities, such as greeting the visitors with a hand sanitizer and a temperature measurement gun [1]. They are usually two groups depending on the recruiting backgrounds, i.e., proprietary security guards and agency security guards. Agency security guards provide better security to the entire premises because the security agency usually hires qualified individuals and trains them for their particular jobs. Security guards are divided into three categories according to their duties, i.e., static, patrol, and undercover. Static guards are the ones who are typically posted at the gates (entrances and exits). Patrol guards travel in groups and are generally equipped with bicycles or motorcycles for easy transport. The undercover guard's job is to stay secret in the casual dress to make it seem like he is only a general person [2]. In Covid-19 situations, static security guards play a crucial role in inspecting visitors' body temperature using a temperature measurement gun, forcing them to sanitize their hands, and wearing a mask. Based on their observations, they may or may not allow visitors to enter the premises. Suppose they found any visitor with high body temperature. In that case, they have to report immediately to their central security office, local police station, local hospital, etc., to take immediate action to process his/her Covid-19 test. In this way, static security guards help the premises and society ensure safety against these Covid situations. To perform these tasks, their situational awareness (SA) is hugely desired. SA means knowing what is going on around us [3]. From the perspective of pervasive computing, this situation identification approaches involve three key issues: the representation of the situation, specification of the situation, and reasoning

✉ Debabrata Bej
debabrata.bej@gmail.com

[1] Department of Electronics and Communication Engineering, National Institute of Technology Durgapur, Mahatma Gandhi Road, Durgapur, West Bengal 713209, India

[2] Department of Electronics and Communication Engineering, Jalpaiguri Government Engineering College, Jalpaiguri, West Bengal 735102, India

about the situation [4]. This SA improves the detection process of motion disorders of patients with Autism Spectrum Disorders using a formal intelligent reasoner [5]. Further, this concept can be related to the risk assessment [6]. From the perspective of a static security guard, SA means his/her ability to observe, quickly analyze the situation, and make the right decisions in an emergency [7]. However, due to their fatigue and other secondary tasks, like mobile phone use, static guards may not perform their duties correctly.

Many anti-drowsy alarm devices are commercially available in the market, which can be applied to alert the on-duty static security guards. In [8], A tool, Nap Zapper, can identify nod off of static guards by detecting their head's tilt using a position sensor and alerting them by a beep sound at their ear. However, it creates irritation and is responsible for their health issues, such as Otalgia, headaches, and eye irritation. In addition, remote tracking of their work performance is not possible using this tool. In [9], the Guard Sleep GSM Alarm (GSGA) system generates beep sounds to the guards every 30 min and waits for their response for 1 min. Suppose the guard fails to make the response/acknowledgment in the system. In that case, it recognizes that the guard is in a sleepy/distracted condition and simultaneously sends a Short Message Service (SMS) or call to his/her supervisor's mobile phone. However, this system fails to track the guard randomly. In case of random alarm, he/she cannot trace the alarm timing, so he/she must always be alerted to do their task. In [10], the QR patrol system is a remote guard tour management system. It monitors security guards based on Global Positioning System (GPS). However, GPS-based tracking device sometimes suffers from signal inaccuracy. In our case, the GPS signal may be weak from the blockage by infrastructures like tall buildings. In addition, GPS-based device suffers from battery drainage problem. In addition to this, users have to pay out a comparatively massive cost every month to use this system. In [11], the 360 guard alertness-monitoring system detects security guard alertness utilizing a smartphone application. However, the product covers many other complex security features, such as visitor tracking, vehicle tracking, incident logs, and maintenance. The authors have searched for the cost of this product. However, it is found that the price is not available on the website. As the product covers many features into a single device, the authors assume the cost will be higher. In [12], Wow security provides a mobile app-based guard alert system. Users have to buy the app at a high price. Above all, most of the commercial products are noneconomic.

In the literature, researchers mostly strived to detect drivers' drowsiness to put aside them from the accidents. Reviewing those techniques can also be useful to identify the drowsiness of on-duty guards and alert them. Mainly three types of detection approaches were proposed in the individual or combined way to detect the drivers' drowsiness. Those were the physiological approach, the behavioral approach, and the vehicular approach. Physiological and behavioral methods for drivers can also be applied for the alertness detection of on-duty static security guards. However, vehicular methods are entirely inappropriate for the guards.

For drowsiness detection of the driver from the physiological signal, the Respiratory Rate Variability (RRV) was monitored using the induction plethysmography belt in [13]. Photoplethysmogram (PPG)-derived respiration was monitored using a PPG sensor in [14]. Electroencephalographic (EEG) signal was analyzed to detect the change in brain wave in [15, 16], and the NicoletOne ambulatory EEG machine logged the brainwave pattern of the drowsy driver in [17]. Change in Heart Rate (HR) was monitored for the drowsy driver in [18]. These techniques also can be used for the application of security guards. However, drowsiness detection through physiological signal sometimes gets annoying to the on-duty person (driver/security guard) because this method directly places the sensors/devices on their body. In the next paragraph, the authors have reviewed the behavioral techniques applied to drivers' fatigue detection.

For fatigue detection of drivers from behavioral approaches, facial expressions, such as blinking, yawning, and head nodding, can be considered. The driver's face was tracked from the camera's video stream using the Multiple Convolutional Neural Networks Kernelized Correlation Filter (MC-KCF) algorithms. The facial key region, eye, and mouth were identified by the feature extraction, and finally, eye closure, blinking, and yawning were evaluated [19]. A mobile application was developed and utilized to estimate the Percentage of Eyelid Closure (PERCLOS) [20]. In [21], a dome camera was used to record the behavioral changes of the bus driver at the head, shoulder, face, and eyes, head, and shoulder were identified and located using the Histogram of Oriented Gradients (HOG) descriptor and Support Vector Machine (SVM), Open Source Computer Vision (OpenCV) was used to detect the position of the face, Open/close state of the eyes was recognized using the OpenCV eye detector (CV-ED) and I2R eye detector (I2R-ED); finally, PERCLOS was estimated to detect the fatigue. The driver's body's orientation and position and the head were tracked using Northern Digital Inc. (NDI) Hybrid Polaris Spectra machine, and driver fatigue was determined from the head-nodding angle [17]. A fusion of two or three different technologies, such as control systems, networks and distributed systems, and learning and reasoning systems, may shape Intelligent Sensor Actuator Systems (ISAS) and make driving safer by sensing and responding to driver exhaustion [22]. To detect a person's fatigue from behavioral features, mostly camera, Image Processing, and intelligence computation techniques are used. In that process, a vast amount of data are to be stored/accessed.

It has been seen that most of the researchers estimated the drowsiness of on-duty persons using the sensor/camera, and it requires additional high computation techniques to process the signal and extract the useful information for alertness detection. In addition, the system cost is very high. Thus, drowsiness detection and alert systems are not readily available to users such as drivers and security guards, in developing countries. To enable alertness among the guards, researchers should think about the cost-effective solution for randomly monitoring their vigilance. To enhance professional efficiency and retain the highly skilled, fully aware performance of security guards, their distraction mode control is significant. Based on the evaluation of their distraction, a perfect rating system can also be introduced by the higher authority at the head security office (HSO). Salary, promotion, demotion, even status of the employment of guards depends on that rating system. Then, it can be assuredly avoided the laziness and unawareness of them. Therefore, awareness monitoring of a guard is beneficial because it prepares him/her to execute efficient work through lifestyle changes. In [23], the authors presented a cost-effective, systematic, sensor-less solution to trigger random alarms to the distracted drivers/nurses and detect their responses. By following the concept of [23], this paper presents a very cost-effective solution for remotely identifying the static security guards' presence at the campus gates and for remotely monitoring their alertness from HSO by triggering random alarms to them and detecting their responses.

The rest of the paper is organized as follows. The proposed alarm system is depicted in Sect. 2. Section 3 illustrates the used components and techniques in prototype development. Section 4 describes the results and discussions of the proposed system. Finally, we conclude this work in Sect. 5.

2 Proposed system to enable the awareness of on-duty static security guards

This paper considers the specific situation where several static security guards are assigned to watch over and secure the campus gates (G) throughout a day. However, sometimes, they fail to perform their duty due to their fatigue or other secondary tasks. This fatigue or other task leads to insecure the gates, and they fail to trace the unexpected issues at campus gates in this Covid-19 pandemic situation. In this case, the present system (S) suggests a systematic, straightforward, and cost-effective solution, which triggers random alarms remotely to these gates in an unspecified manner from the HSO. The guards of those gates have to respond to the alarms. The system understands the mode of their awareness as alerted or distracted from their response recording. If they do not respond, the system identifies that guard as distracted. HSO records the real-time data of individual static security guards in a database (DB). Then, the method includes calculating their percentage of alertness and assessing their work performance through the implementation of a rating system.

2.1 Proposed system model

In this model, it has been considered that four static security guards have been assigned to watch over the four gates (G1–G4). The proposed system's main objective is to monitor the guards' awareness remotely from the HSO for the identification of those security guards at their locations and recoding their alertness level during their duty. HSO sends signals randomly to the selected nodes (Node 1 to Node 4) situated at the four gates through the Wi-Fi network. S1, S2, S3, and S4 are four systems placed particularly at four nodes to receive the signals from the HSO. The security guards must respond to the received signals to acknowledge the HSO that he/she is on alert mode, and HSO also identifies their alertness level from their response time. Suppose the acknowledgment signal has been not received from that particular node. In that case, HSO can recognize that the security guard on that node may be distracted by fatigue or any other secondary task. In this work, authors have used Wi-Fi over some other wireless technologies as (a) even though RFID and Bluetooth consume less power, their range of communication is short [24], (b) 4G or other LTE services would lead to high costs [24]. Hence, because of making a low-cost solution, these technologies will not fit best, and (c) ZigBee seems to be one of the most promising technologies. It has the advantage of low-power consumption, but when it comes to data transfer speed, it contributes 100 KBps, whereas Wi-Fi has a data transfer speed of 100 MBps [24]. In embedded systems, a Wi-Fi network can be established easily using the NodeMCU module, which simultaneously acts both as server and client, minimizing the load of using separate modules for the receiver and sender. The proposed system model is depicted in Fig. 1.

2.2 Block diagram of the proposed system

The proposed system has been used to create a private Wi-Fi network for monitoring the alertness of the static security guards based on the simple relationship between the server and the client. The block diagram of the proposed system is depicted in Fig. 2. The components of the system are client, repeaters, and servers. The client has been placed at the HSO, servers have been placed at the end of the guards, and to collect the data from the various servers, Client has to ping the servers because Client can receive the data only after requesting it from the servers. As the proposed private Wi-Fi network has to cover a long distance, repeaters have been placed between the client and the respective servers.

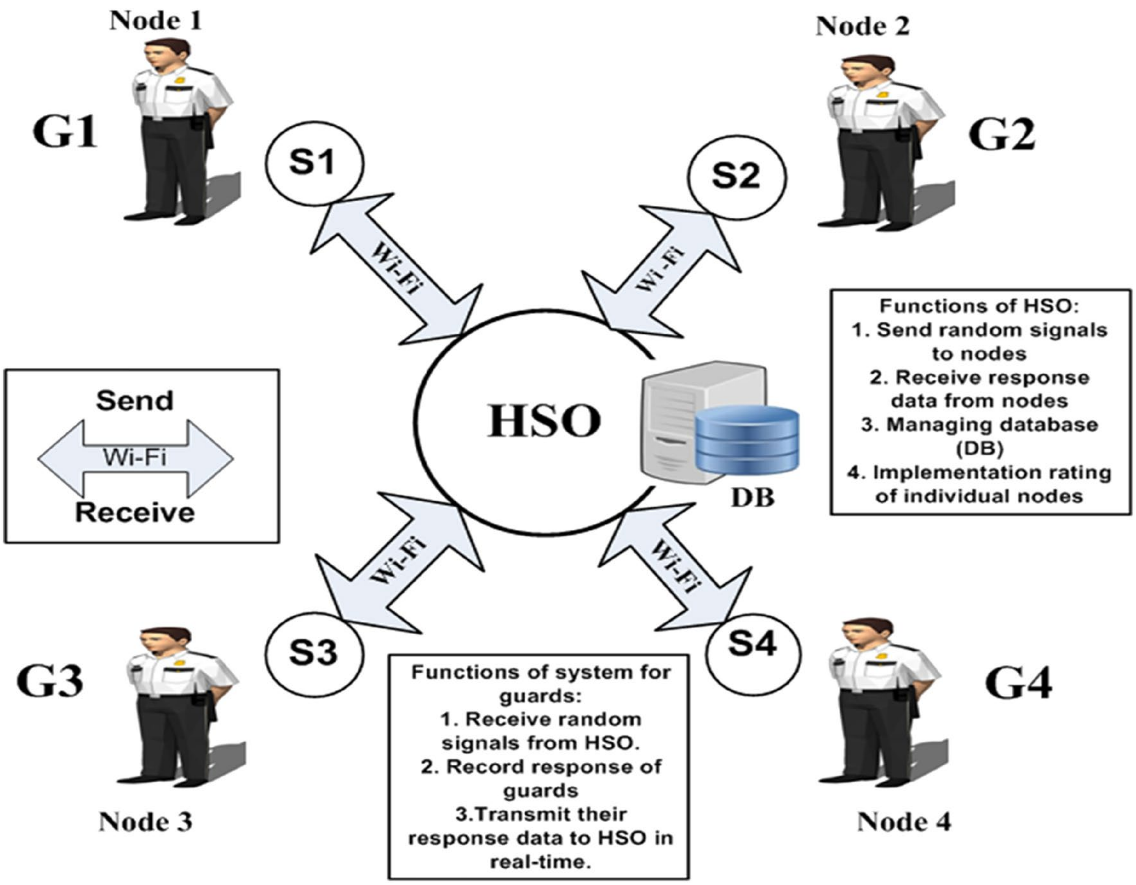

Fig. 1 Proposed system model

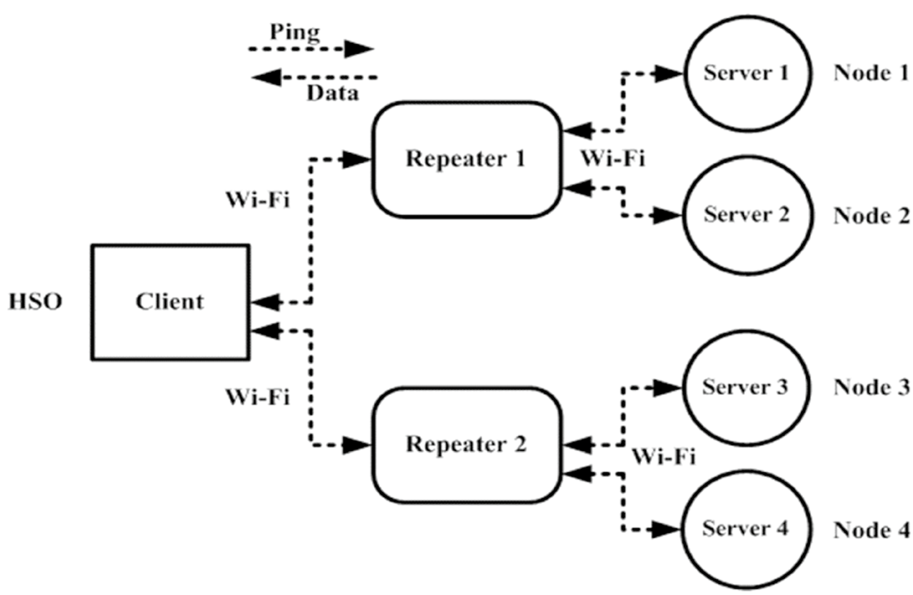

Fig. 2 Block diagram of the proposed system

In this study, the authors consider the client has two repeaters (repeater 1 and 2). Repeater 1 has two servers (server 1 and 2), and repeater 2 has two servers (server 3 and 4). The client randomly pings to the respective repeaters. These repeaters replicate the incoming ping signal and guide it to the respective servers randomly. Security guards at the server sides have to respond to those signals to acknowledge the HSO. The client records their responses and calculates their percentage of alertness level. Further, these data are converted to the rating of individual guards.

In Fig. 3a, detailed block diagram has been presented for the communication network of server 1 with the client. At HSO, one Wi-Fi module and computer act as the client of the system. The Wi-Fi module pings to the repeater, and the computer is connected with the Wi-Fi module to record the received day-wise alertness data of the static security guards. A DB has been maintained in the computer for further analysis of the collected data. The repeater has been used between the client and the server. The repeater's primary purpose is to extend the working distance between the two Wi-Fi modules, i.e., sender and receiver. Repeater replicates the received ping signal and guides it to the respective server. At the guards' end, one Wi-Fi module has been used. The purpose of this Wi-Fi module is to receive the signal from the repeater and create an alarm to the guards by activating a Light Emitting Diode (LED) and sounding a buzzer for a fixed time duration. Guards have to press the push-button switch within that time. From this response, the client can identify the alertness level of the individual static security guards.

2.3 Working principle of the proposed system

The proposed system works on the simple principle of communication between two or more Wi-Fi modules. For this purpose, the authors create a private Wi-Fi network between the modules. Our sole objective is to track the on-duty static security guards, identify the distracted guards, and record their day-wise performance level. The proposed system has a client, which is a Wi-Fi module. The client is the brain of the system. It is responsible for essential functions such as randomly pinging the respective repeaters, storing the guards' time of response. As the distance covered by the private network must be large, the authors have placed repeaters between them. These repeaters amplify the incoming ping and guide it to the respective servers. These are the same module as the client, i.e., a Wi-Fi module. The repeaters act as both server and client. The repeater acts as a server or an Access Point (AP) to receive an incoming ping request. As soon as the AP receives a request, it switches to a client and randomly calls the connected server. The servers are coupled with a LED, buzzer, and a push-button switch. When the respective server receives a ping request, it turns on the LED and sounds the buzzer for a fixed time. Within this time, the guard has to respond by pressing the push-button switch. If the guard presses the switch, a protected message is sent to the client. If the guard fails to do so, a compromised message is sent to the client, and the conclusion is that the guard is in distracted mode, and thus another alarm in a shorter time is necessary. In Fig. 4, the flow chart of the system is presented.

2.4 Proposed algorithms of the system

Algorithm 1: Algorithm for Client

```
Input: SSID, password, rn, r
Output: data, resp, state, line
Initialization:
1.      Setup ip, gateway, subnet
        LOOP PROCESS
2.      while(true)
3.      rn: = randomly generate unique repeater number to connect
4.      if rn = previously generated repeater number (r) then
         goto 1
         else
connect to repeater using specified SSID and password
5.      end if
6.      if rn = CONNECTED then
          data: = server connected
          resp[data]: = successful ping (1) / unsuccessful ping (0)
          stat[data]: = response
          line[data]: = response time
7.      end if
8.      serial print resp, state and line
9.      end while
```

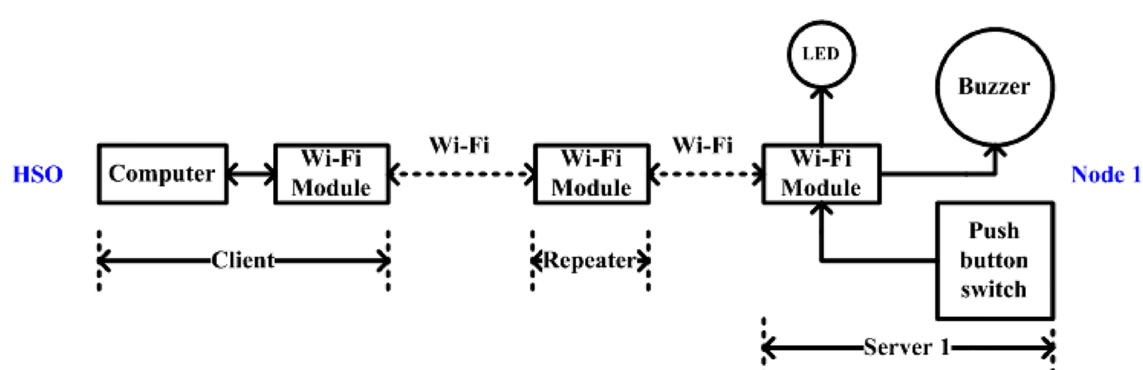

Fig. 3 Detailed block diagram of the communication between one individual server with client

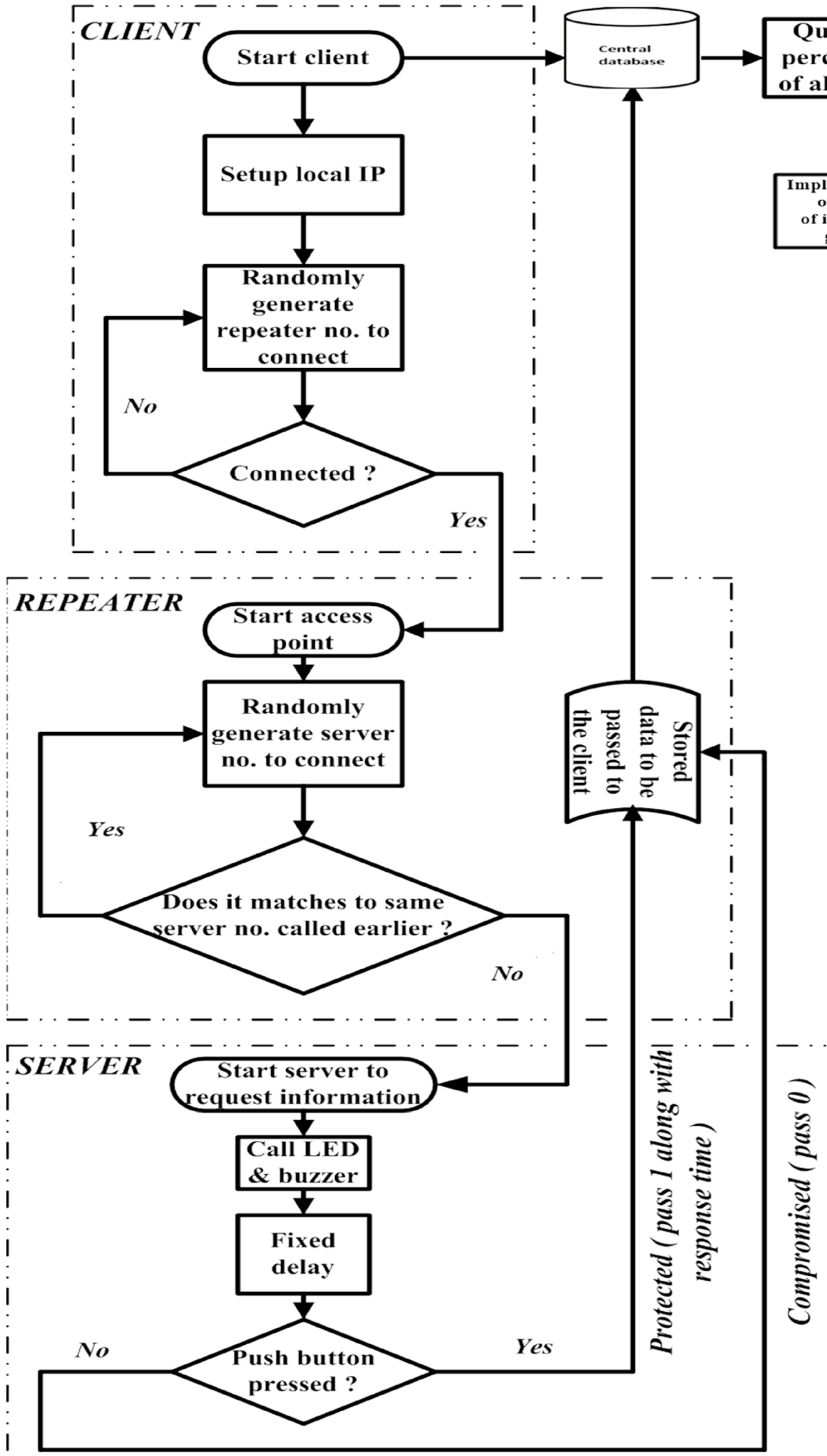

Fig. 4 Flow chart of the proposed system

The authors have prepared three algorithms (Algorithm 1–3) for the Client, repeater, and server. These algorithms are required to program the respective modules at the client, repeater, and server end. The client's algorithm is designed so that the Wi-Fi module has complete autonomy to request information from all the servers in any order and presented in Algorithm 1. It means the process is entirely random. There is generally no pattern unless set by the programmer. The first step of the algorithm requires it to generate the repeater number with which it wants to get connected. On generating, it checks if it has connected to the repeater on its previous iteration. If it has, it again generates another repeater number to get connected. The next step involves getting connected and fetching the data. On getting the unique repeater number, it connects and requests data from the server. It means the repeater gets connected to a unique server, fetches the data, and returns it to the client. The client then calculates the percentage of awareness of the guards and records it in a Comma-Separated Values (CSV) file in the database. Further, these data convert to individual guards' rating, which is useful to evaluate their work performance.

Algorithm 2: Algorithm for Server

```
Input: ping, case
Output: time, response
Initialization:
1.      Setup Access Point
 LOOP PROCESS
2.      while (ping = True)
          call func_LED
3.      end while
4.      function func_LED ()
           if case = 1
               intensity: = LOW
           else if case > 1
                intensity: = HIGH
5.      end if
          LED_intensity: = intensity
          BUZZER_intensity: = intensity
LOOP PROCESS
6.      while (time <= 120 Sec)
            if BUTTON_press = HIGH
            response: = 1
              break
7.      end if
8.      time: = time + 1
9.      end while
10.     end func_LED()
11.     return time, response
```

The algorithm for the server is presented in Algorithm 2. This algorithm is designed in such a way that the Wi-Fi module act as an AP. The module running as an AP is set up with a unique Service Set Identifier (SSID) and password. If it receives a client's request through the SSID and password, the server call func_LED () function. The func_LED () function is designed in such a way that on getting called, it activates the LED and buzzer, which are connected with the module. The LED and buzzer are set on a high or low intensity depending on the guards' previous response. If the guard has failed to respond to an earlier ping, the buzzer and LED will glow up with high intensity; else, it has a low intensity. The devices are activated for a window of fixed delay if the guard presses the key present in the system and responds to the ping, the Wi-Fi module record the time taken by him/her to respond and return the same to the client where the percentage of alertness of the guard is calculated.

Algorithm 3: Algorithm for Repeater

```
Input: SSID, password, rn, r, ping, case
Output: data, resp, state, line, response
Initialization:
1.      Setup AccessPoint
         LOOP PROCESS
2.      while (ping = True)
          call switch_client()
3.      end while
4.      function switch_client()
           Disconnect as Access Point
           Set up gateway, ip, subnet
5.      rn: = randomly generate unique server number to connect
6.      if rn = previously generated server number (r) then
          goto 5
         else
           connect to server using specified ssid and password
7.      end if
8.      if rn = CONNECTED then
          data: = server connected
          resp[data] := successful ping(1) / unsuccessful ping(0)
          stat[data] :=  response
          line[data]: = response time
9.      end if
10.     end switch_client()
11.     return resp, state and line to client
```

The algorithm designed for the repeater is presented in Algorithm 3. It is a mix of both the client and the server algorithms. On receiving a ping/request from the client, the repeater, which was in AP mode until now, switches to a client mode itself. It generates a random server number from the servers available to it and checks if it has not been produced on a previous iteration. If it is unique, it connects to the server and requests it to set the LED and the buzzer high with a fixed delay window. If the guard responds, the repeater fetches the response time and returns a "1" (high value) to the Client. If the guard fails to do so, it returns a "0" (low value) to the Client.

3 Components and techniques used

3.1 Prototype development of the system

For prototype development of the proposed system, NodeMCU (ESP8266) development board [25], LED, buzzer, and push-button switch are used. All the components used in the system have been presented in Fig. 5.

3.1.1 NodeMCU development board

NodeMCU [25] is an open-source software and hardware development environment for the Internet of Things (IoT). It has built-in support for Wi-Fi connectivity. ESP8266 is an inexpensive System On Chip (SOC) that is used for the Wi-Fi network. In the present work, the NodeMCU module is used to create only the Wi-Fi network among the Client, servers, and repeaters. The maximum communication range of this module is approximately 300 m (line-of-sight outdoors). To achieve more range, it is required to orient in the 'best' direction, i.e., usually parallel to each other.

3.1.2 LED, buzzer, and a push-button switch

LED is used to generate a physical output of the alarm system at the server-side. Piezo buzzer is used at the server-side to create the sound of the alarm system to the guard. A push-button switch is used to record the on-duty guard's response at the server-side in the alarm system.

3.1.3 Programming tools/software

3.1.3.1 Arduino IDE The open-source Arduino Software/Integrated Development Environment (IDE) [26] is used to write code and upload it to the NodeMCU module. For this purpose, the ESP8266 library has been installed in the Arduino software. The Arduino IDE supports C and C++ language. The authors have used the Arduino IDE version 1.8.9 to write and upload the code to the NodeMCU module.

3.1.3.2 CoolTerm CoolTerm [27] is a serial port terminal application. It is used to capture the serial data and save it in a CSV file. In the present system, it is used with a computer at the Client. It collects all the data into a database regarding the guards' response from the NodeMCU module of the Client, which is connected with the computer through the serial port. These data are further used to calculate the percentage of alertness level of individual guards and analyze their work performance.

3.1.4 Picture of the final prototype

The prototype of the proposed system has been presented in Fig. 6. In part A, B, and C, Client, repeater, and server are shown, respectively.

4 Result and discussion

The prototype was tested inside the laboratory by placing it in front of the four working individuals studying for the whole night at different regions of the respective department. This similar system can be utilized for the alertness detection of guards. A private Wi-Fi network was created for communication purposes between the client, servers, and repeaters. Servers were placed in front of those working individuals to alert them and record their responses. The tested working range of the Wi-Fi network of this prototype is around 50 m. The servers are represented by the four nodes, i.e., Node 1–Node 4. Alarms were sent to all four servers from the client, which was connected with a computer to record the responses of the working individuals. These responses were shown in terms of response time, i.e., the time taken by the working individuals to give the response at those alarms. Those response data represent their alertness level. Repeaters were used to extend the working range of the Wi-Fi network. Ping signals were sent to the randomly selected repeaters by the Client, and repeaters sent the alarm signal to the randomly chosen servers to alert the four working individuals. Alarms were generated by the system using a random time from 7 to 15 min at every cycle. The alarm signal could stay for 2 min (120 s) and be represented by glowing a LED and beeping a buzzer at the respective server with the four working individuals. The prototype was tested at night at different laboratories inside the department for two different durations of time, i.e., 12–2 AM and 3–5 AM.

From Figs. 7 and 8, it can assure that the system generates an alarm in a different interval of time randomly to check the level of alertness at the different nodes at a time. The working person presents in individual nodes generally give the response according to their vigilance, which has been calculated using Eq. 1. The level of alertness percentage has been seen to be higher at a node where the response time is less, and it has been seen to be lower at a node where the response time is high. It has been seen 0% when no response has occurred on the particular node. In Fig. 7 and 8, red,

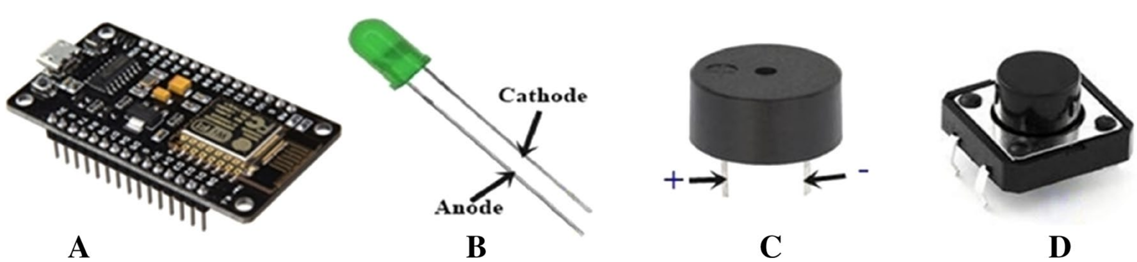

Fig. 5 Components used for prototype development. **a** NodeMCU (ESP8266), **b** LED, **c** Buzzer, and **d** Push button switch

blue, and green color means distracted, moderately alert, and highly alert, respectively.

their level of alertness. In Fig. 10, an overview of the proposed system has been presented.

$$\text{Level of alertness at individual alarm} = \frac{(\text{Maximum alarm time} - \text{Time taken to response})}{\text{Maximun alarm time}} \times 100\% \quad (1)$$

The present experiment's main objective is to establish a system to enable awareness among the static security guards. For this purpose, the authors measured the average alertness level of the individual guards from the different nodes at a time by generating the alarm to them. The system selected the nodes randomly to generate arm in a random interval of time.

A cost comparison with the previous system is presented in Table 2. In [8], the price of the system is very low among all previous methods. However, it is only used for local fatigue detection of users and alarm generation. There is no scope for remote monitoring. From the rest of the product, it is observed that our present system is the most cost-effective solution for remote alertness monitoring of the user and alarm generation.

$$\text{Average level of alertness at individual node} = \frac{\text{Sum of all level of alertness at individual alarm}}{\text{The total no. of system generated alarm at that node}}. \quad (2)$$

Using Eq. 2, the average level of alertness at node 1 to node 4 has been calculated and represented in Fig. 9.

The average level of alertness percentage is compared with a standard rating scale and represented in Table 1.

From the lab-scale experiment, the rating of working individuals from node 1 to node 4 is provided based on

This work can be extended towards the system-oriented detection of the person with high body temperature and the immediate action taking towards his/her isolation and Covid-19 test. When the guard identifies a person with high body temperature, he/she has to press a specified button available in the system to indicate the detection of a

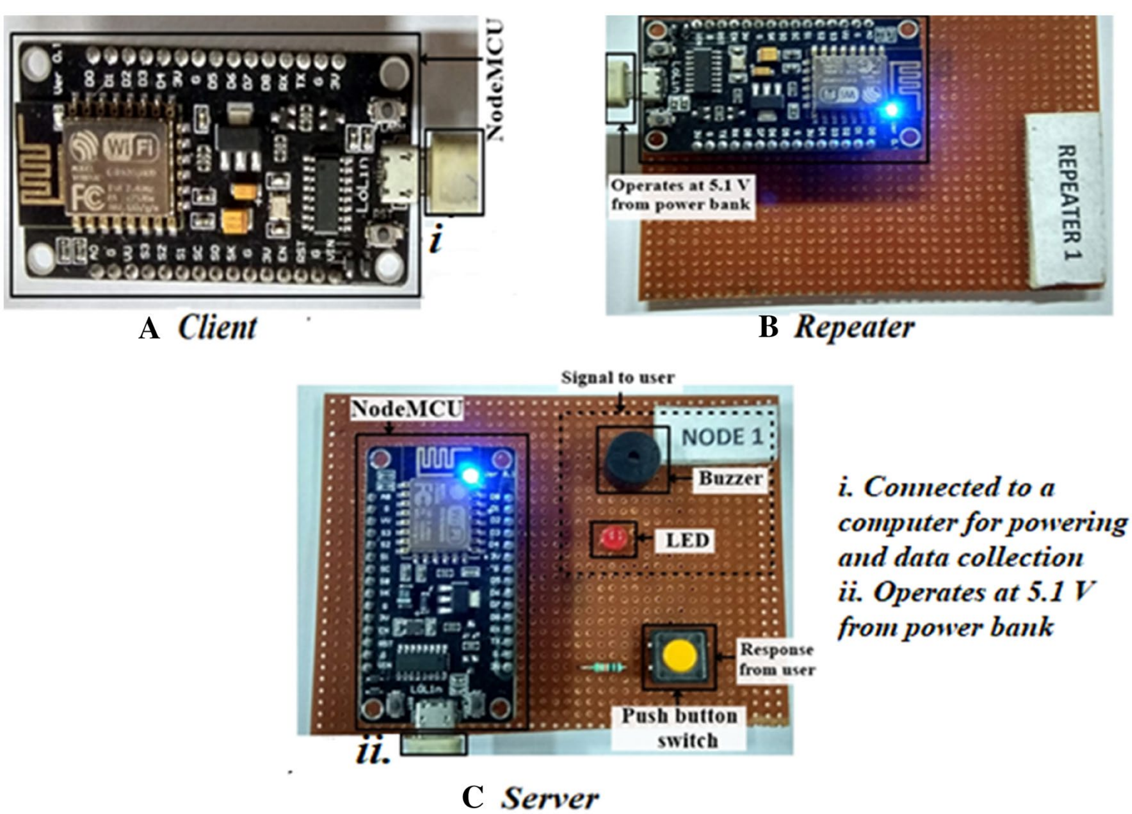

Fig. 6 Picture of the prototype

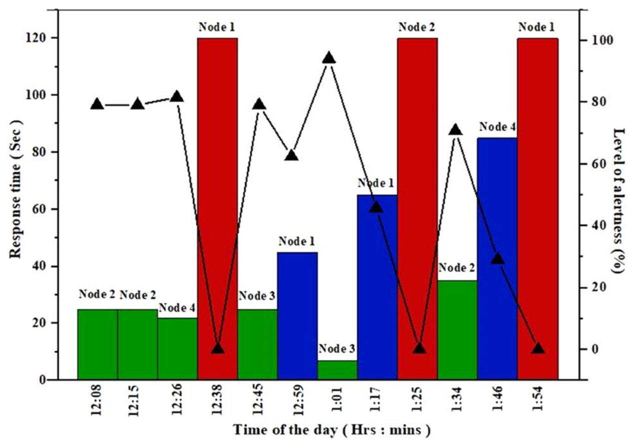

Fig. 7 Alertness detection of working individuals from 12 to 2 AM tested inside the lab

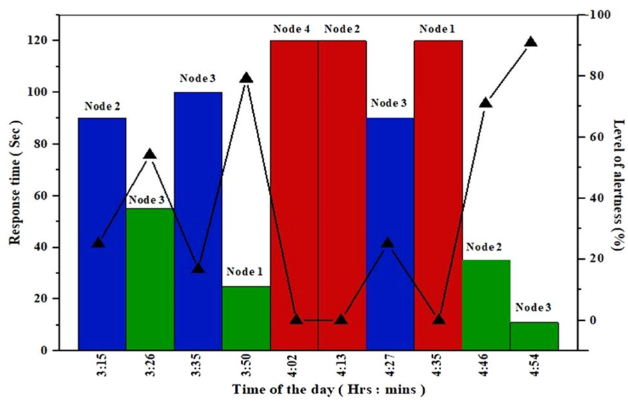

Fig. 8 Alertness detection of working individuals from 3 to 5 AM tested inside the lab

person with high body temperature. This response can be communicated by the system to the local authority, such as local hospital and police station, for taking immediate action of that person's Covid-19 test. Using this method, people can avoid Covid-19 virus contamination and get rid of this worldwide pandemic. The presently engineered system can be useful for the guards' alertness and enable the safety of the campus in the Covid-19 pandemic circumstances. In Fig. 11, the future scope of the proposed system is presented.

5 Conclusion

The present work represents a simplistic approach for the real-time remote alertness detection of the on-duty static security guards from the HSO through the Wi-Fi network. A simple client–server model is used for this system. In

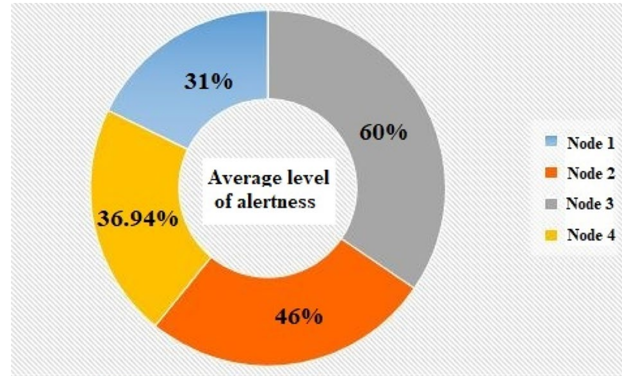

Fig. 9 Average level of alertness of working individual at four nodes

Table 1 Rating scale for the assessment of the professional efficiency of the working individuals

Sl. no	Range of alertness level	Rating decided
1	100–90	5
2	89–80	4.5
3	79–70	4
4	69–60	3.5
5	59–50	3
6	49–40	2.5
7	39–30	2
8	29–20	1.5

between client and server, the repeater is used for the range extension. HSO sends the random alarm signals to the guards and records their responses at a DB. These data are used to calculate the % of guards' alertness, and further, it estimates the rating of the individual guards. For testing purposes, the authors have tested their prototype by placing it in the working individuals' desk in the departmental laboratory, who are studying/working throughout the night. The system records their response time to individual alarms. The average level of individual guards' alertness has been calculated for the four guards as 31%, 46%, 60%, and 36%, respectively. Based on a standard rating scale, the working individuals' rating has been estimated, which can improve their work performance through lifestyle changes. The prototype testing proved that this similar prototype can be applied for the alertness detection of on-duty static guards, who are working at the campus gates. In that case, the distance between sensing nodes will be more to cover the campus gates. Wi-Fi network must cover a long-range. To solve this issue, a required number of repeaters are to be used between the client and the server. This system is a much more cost-effective solution for this purpose compared to the sensors/camera-based guard alert systems. However, the system requires user input for data collection. In addition, as a future scope, this system can be

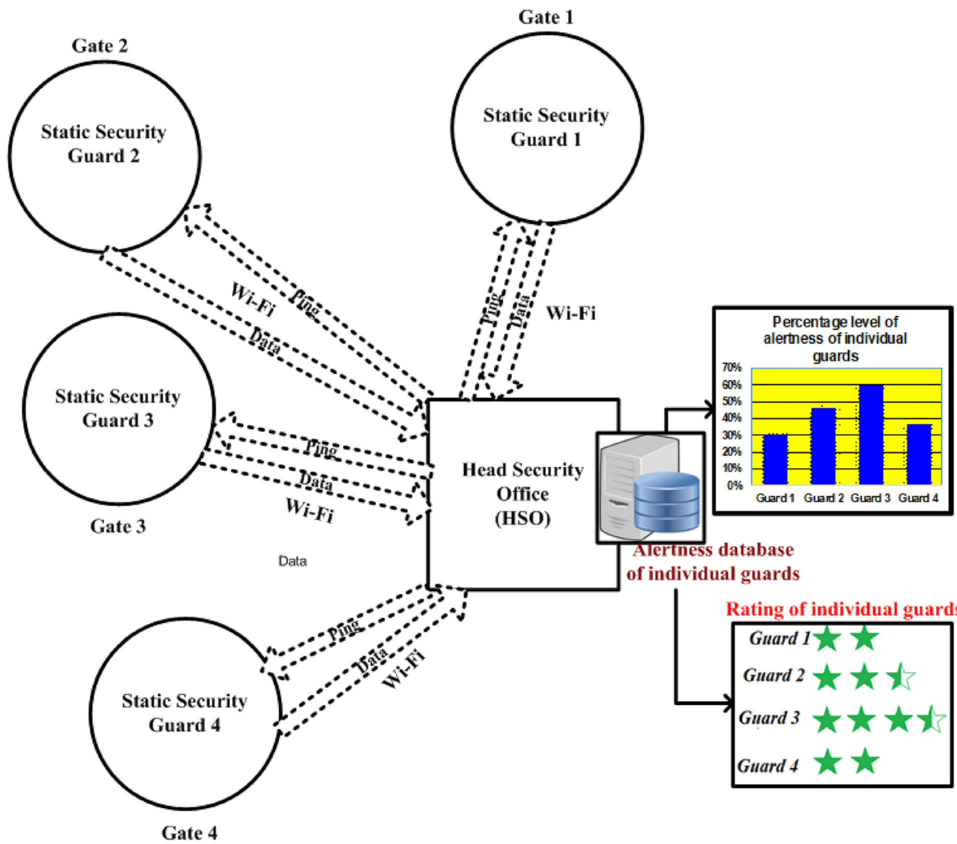

Fig. 10 Overview of proposed system to enable the awareness of static security guards

Table 2 Cost comparison with previous guard alertness-monitoring system

Sl. no	System name	Components used	Cost for one unit in Indian rupee (INR)	Monitoring facility
1	Napzapper [8]	Details are not available	₹ 478/-	Local monitoring; No scope for remote monitoring
2	GSM auto dialer [9]	Microcontroller, GSM module, and SIM card	₹ 4900/-	Remote monitoring
3	QR-Patrol Guard Tour Management system [10]	Details not available	₹ 2,622.55/-	Remote monitoring
4	Wow security–App based guard alert [12]	Mobile App for Android	₹ 6000/year	Remote monitoring
5	Present system-remotely monitoring guard alertness	NodeMCU Wi-Fi module, LED, Buzzer, and Push-button switch	₹ 263/-	Remote monitoring

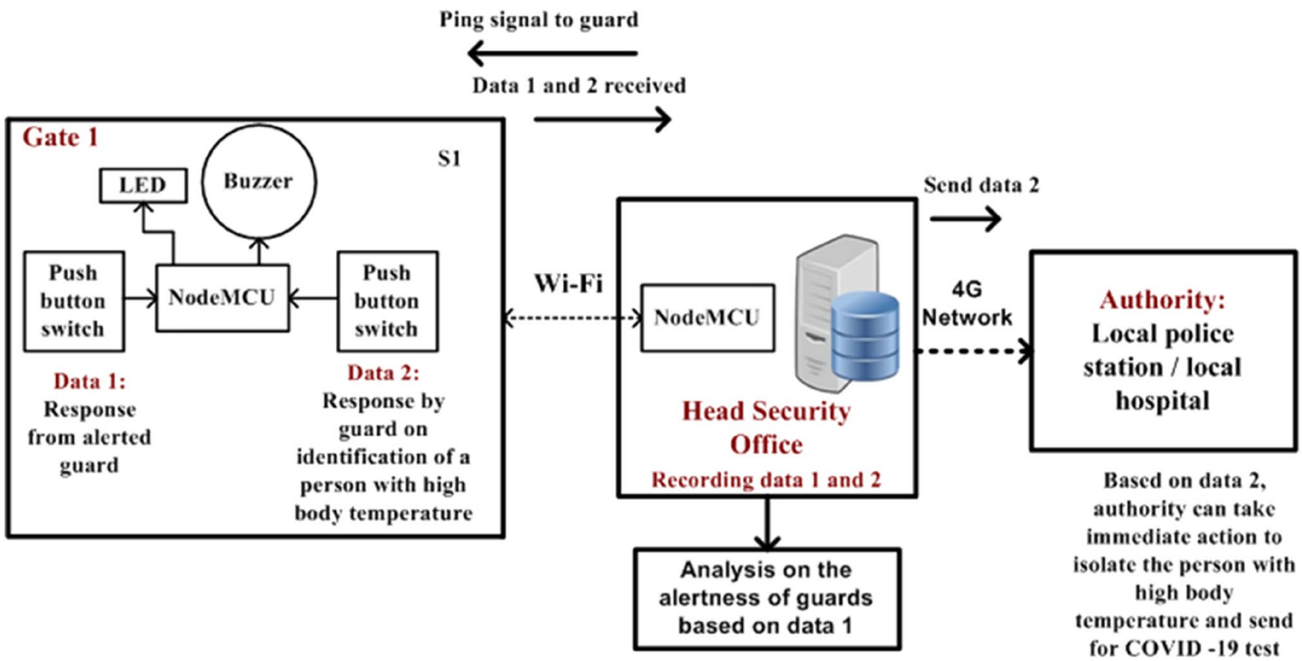

Fig. 11 Future scope of the proposed system

applied to system-oriented detection of the person's high body temperature. As the detected person may be a Covid-19 infected person, the system helps the authority (local police station/local hospital) identify the person quickly to isolate him/her and immediately conduct his/her Covid-19 test. In this way, this system plays a vital role in creating SA among the static guards towards the campus and society's safety.

References

1. Fighting intruders, coronavirus: security guards at IT companies perform double duty (2020) https://www.indiatoday.in/india/story/fighting-intruders-coronavirus-security-guards-at-it-companies-perform-double-duty-1656203-2020-03-16. Accessed 27 July 2020
2. Types and functions of a security guard (2020) http://igssecurity.com/the-types-and-functions-of-a-security-guard/. Accessed 27 July 2020
3. Flin RH, O'Connor P, Crichton M (2008) Safety at the sharp end: a guide to non-technical skills. Ashgate Publishing Ltd., Farnham
4. Ye J, Dobson S, McKeever S (2012) Situation identification techniques in pervasive computing: A review. Pervasive Mobile Comput 8(1):36–66
5. Coronato A, De Pietro G, Paragliola G (2014) A situation-aware system for the detection of motion disorders of patients with autism spectrum disorders. Expert Syst Appl 41(17):7868–7877
6. Coronato A, Cuzzocrea A (2020) An innovative risk assessment methodology for medical information systems. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2020.3023553
7. Cultivating and Improving Situational Awareness Among Your Security Officers (2020) https://www.silvertracsoftware.com/extra/cultivating-and-improving-situational-awareness-among-your-security-officers. Accessed 27 July 2020
8. Nap Zapper anti-sleep alarm system (2020) https://www.thehomesecuritysuperstore.com/products/streetwise-nap-zapper-anti-drowsy-driver-alarm. Accessed 20 Dec 2020
9. GSM Based Guard Sleep Monitoring Alarm System (2020) http://www.safesecuremagazine.com/product_highlight_details.php?id=4. Accessed 27 July 2020
10. QR-Patrol (2020) https://www.qrpatrol.com/. Accessed 1 Aug 2020
11. The 360 (2020) https://the360.in/. Accessed 1 Aug 2020
12. Wow security (2020) https://www.wowsecurity.in/guard-alertness. Accessed 1 Aug 2020
13. Guede-Fernández F, Fernández-Chimeno M, Ramos-Castro J, García-González MA (2019) Driver drowsiness detection based on respiratory signal analysis. IEEE Access 7:81826–81838. https://doi.org/10.1109/ACCESS.2019.2924481
14. Lee B, Lee B, Chung W (2015) Wristband-type driver vigilance monitoring system using smartwatch. IEEE Sens J 15(10):5624–5633. https://doi.org/10.1109/JSEN.2015.2447012
15. Becerra-Sánchez EP, Reyes-Muñoz A, Guerrero-Ibáñez JA (2019) Wearable sensors for evaluating driver drowsiness and high stress. IEEE Latin Am Trans 17(03):418–425. https://doi.org/10.1109/TLA.2019.8863312
16. Budak U, Bajaj V, Akbulut Y, Atila O, Sengur A (2019) An effective hybrid model for EEG-based drowsiness detection. IEEE Sens J 19(17):7624–7631. https://doi.org/10.1109/JSEN.2019.2917850
17. He Q, Li W, Fan X, Fei Z (2015) Driver fatigue evaluation model with integration of multi-indicators based on dynamic Bayesian network. IET Intell Transp Syst 9(5):547–554. https://doi.org/10.1049/iet-its.2014.0103
18. Jo SH, Kim JM, Kim DK (2019) Heart rate change while drowsy driving. J Korean Med Sci 34(8):e56. https://doi.org/10.3346/jkms.2019.34.e56

19. Deng W, Wu R (2019) Real-time driver-drowsiness detection system using facial features. IEEE Access 7:118727–118738. https://doi.org/10.1109/ACCESS.2019.2936663
20. Soares G, de Lima D, Miranda AN (2019) A mobile application for driver's drowsiness monitoring based on PERCLOS estimation. IEEE Latin Am Trans 17(02):193–202. https://doi.org/10.1109/TLA.2019.8863164
21. Mandal B, Li L, Wang GS, Lin J (2017) Towards detection of bus driver fatigue based on robust visual analysis of eye state. IEEE Trans Intell Transp Syst 18(3):545–557. https://doi.org/10.1109/TITS.2016.2582900
22. Schmidtke HR (2018) A survey on verification strategies for intelligent transportation systems. J Reliable Intell Environ 4(4):211–224
23. Bej D, Rakshit S, Mal AK, Mahapatra R (2019) A cost-effective system for triggering alarm to distracted drivers/nurses. Comput Electr Eng 76:24–39
24. Introduction Wireless Communications (2020) http://haitham.ece.illinois.edu/WirelessSU20/Lectures/ZJU-UIUC-Wireless-SU2020_Lec1.pdf. Accessed 20 Dec 2020
25. NodeMCU (2020) https://www.nodemcu.com/index_en.html. Accessed 4 Aug 2020
26. Arduino IDE (2020) https://www.arduino.cc/en/main/software. Accessed 4 Aug 2020
27. CoolTerm (2020) https://freeware.the-meiers.org/. Accessed 4 Aug 2020