Elevint: A Cloud-based Internet of Elevators

Sarah Mohammed Aljadani, Shahd Mohammed Almutairi, Saja Saeed Ghaleb, Lama Al Khuzayem
Department of Computer Science
King Abdulaziz University
Jeddah, Saudi Arabia

***Abstract*—With the significant growth of the number of high-rise buildings nowadays, the dependence on elevators has also increased. The issue that faces elevator passengers in case of breakdowns is the long waiting time for the arrival of the maintenance engineers to perform the repair, as the process of reporting is done manually. The safety concern increases when people are trapped. Most state-of-the-art approaches detect faults without providing means to facilitate the communication between elevator owners, maintenance companies, and engineers or notify them in case of breakdowns. Moreover, none of the proposed fault detection solutions rely on rules specified by experts in the field. This paper aims at addressing these issues by proposing a system that manages, monitors, detects faults and informs users of any faults instantly by sending notifications. Specifically, the paper proposes a mobile application, Elevint, that is, cloud-based and exploits the Internet of Things (IoT) technology. Elevint provides real-time monitoring of elevator operating conditions collected from sensors. The data is then transferred to the cloud, where faults are detected by applying rules that compare the current conditions with severity levels determined by experts. In the case that a fault is detected, elevator owners and maintenance companies are automatically notified. Moreover, through Elevint, maintenance companies can assign engineers to repair the fault and elevator owners can view and re-schedule the engineer's visit if needed. Testing our system on an elevator model shows 98% accuracy. In future, we intend to test it on real elevators to verify its applicability in practice.**

***Keywords*—Internet of things; elevator; fault detection; monitoring; notification; real-time**

I. Introduction

The use of elevators has increased dramatically throughout the world, due to the congestion of population in main cities. Thus, cities have tended to increase the number of skyscrapers to accommodate the high number of populations. Successively, the reliance on elevators around the world has increased. Safety remains a critical issue in the design of elevators, since its failure may endanger people, disrupt their schedule, and threaten their lives.

Presently, when an elevator breaks down or people are trapped, there is an extensive waiting time before the relevant people are notified. This is because the process of informing the maintenance technicians is done manually. If this process is somehow automated, the maintenance team will arrive faster, and the breakdown time will be reduced significantly.

Therefore, to ensure the safety of elevator passengers, it is necessary to speed up the process of notifying the elevator owners and maintenance companies about failures. Developing notice of the breakdown in a timely manner will not only reduce the downtime and save money, but will also allow preventive maintenance of the elevator [1].

With the advancement of technology in the 21st century, researchers in academia have considered investigating the safety control system of elevators through the Internet of Things (IoT) with the aim of moving the industry from preventative maintenance to predictive maintenance [2]. Most state-of-the-art research in this area proposes a monitoring elevator system that exploits IoT techniques to achieve the supervisory functions of early warning to reduce elevator accidents [3] while using smart sensors and cloud services [4]. Monitoring will help do analysis research to identify the main factors that cause accidents; thus, it will eventually help reduce accidents and injury rates [5].

On the other hand, elevator companies have also started to make smart elevators and are investing billions of dollars in this new technology [6]. However, it is not possible for all elevator owners to replace their old elevators with new smart ones, since there would be significant costs. This research aims at filling the gap by elevating current conventional elevators to smart elevators, through installing safety control systems that reduce downtime by analyzing the elevators' data and notifying the right people in a timely manner. Specifically, this paper proposes an IoT cloud-based elevator monitoring and fault detection mobile application which includes the following features:

1) Enable elevator owners and maintenance companies to monitor their elevators' operating conditions from anywhere in real-time.

2) Facilitate the communication between elevator owners, maintenance companies, and technicians through instant messaging services.

3) Detect major underlying factors for elevator faults based on some expert-defined heuristic rules.

4) Notify elevator owners and maintenance companies instantly about detected faults.

The remainder of this paper is organized as follows: in Section II, we review related work in this area and in Section III, we explain our proposed solution. We present our results in Section IV, followed by Section V which concludes the paper.

II. Related Work

Wang et al. [7] developed a system called EleSense, which is a framework for high-rise building structure monitoring. They used two kinds of sensing devices, the sensors are two sets of vibrating wire strain gauge sensors attached at the inner tube and outer tube in each floor, and temperature sensors in most of the floors. A cluster head installed on each floor sends the data gathered from sensors to the base station. The base station is attached in the top of the elevator and while the elevator is moving, the base station moves with it and collects data from sensors. The results show that EleSense can significantly reduce communication costs, while providing reliable data.

Jiang et al. [8] proposed a system that provides fault detection based on real-time data from sensors that collect various parameters of running elevators, such as the weight of the lift car, the signal of portal crane, the signal of the layer precision of the elevator and the signal of safety gear. The system also enabled trapped people to communicate with maintenance staff. Moreover, the system has video and audio monitoring with Wimax (Worldwide Interoperability for Microwave Access) technology, which provides video and audio transmission based on the Internet Protocol version 4.

Zarikas and Tursynbek [9] described a system for an intelligent elevator, integrated with a smart building. The system’s aim is to build a decision engine that can control the elevator's actions through AI techniques.

Salim and Akin [10] proposed an IoT-based elevator system that predicts and diagnoses errors from elevators. In this system, after troubleshooting, the data is collected from the elevators via sensors. This data is then transferred from the elevator to the control system, over the Internet to be sent to the data maintenance company to solve the problems. The system used fire sensors, low electricity and door status to detect faults. The advantage of this system is that it has a website to monitor the status of elevators, in addition to saving all elevator data and extracting reports.

Zhang et al. [11] measured the level of comfort during the elevator ride by measuring the acceleration on three directions axes x, y and z using sensors embedded in a smartphone. The elevator ride comfort level is evaluated based on the collected data, and the assessment results are uploaded to a structural health monitoring site. The results of the experiment show that their method has met the engineering requirements. Using the sensors inside the mobile device to monitor the elevator vibration may be easier and saves the time and effort of installing these sensors into a device on the elevator.

Suárez et al. [12] proposed an application that informs the maintenance company of the elevators’ failures and trapped people. The application, which is developed using Microsoft Azure, implements on a cloud that is responsible for receiving data from elevators and processing it, then sending notifications accordingly. The disadvantage of this system is that it only deals with elevator faults as opposed to other systems which can also cater for management and monitoring in real-time.

Olalere et al. [13] developed a system to reduce elevators breakdowns, by early reporting of faults and diagnosis from historical data gathered from monitoring the vibration and noise using sensors. The system is divided into three layers. The first layer is the sensors layer that includes two kinds of sensors for vibration and audio. The second layer contains the Yun microcontroller, which is an Arduino development board that analyses the signals from the sensors and helps to achieve the IoT communication. The third layer includes the web application server. They measured the results over a month, and the results showed that the elevator stops were noticeably reduced, due to notification of breakdowns. This system is very close to our proposed system but lacks some extra sensors and a monitoring camera.

Ming et al. [14] proposed an elevator safety monitoring system based on IoT technology and included audio and video monitoring. The research used sensors for vibration, acceleration, speed, diving noise, direction, floor station, elevator car door switch, voltage, temperature, and all these sensors were used to analyze data and increase safety.

Li [15] proposed a framework for an elevator security monitoring system based on cloud computing to monitor the running status of the elevator in real time. The system can provide reminders of faults, show alarm information and can classify the alarm information according to the severity.

Zhou et al. [16] introduced a remote elevator monitoring system based on IoT through real-time monitoring, fault diagnosis, alarm, and maintenance, without mentioning any details of the proposed system and results.

Huang et al. [17] proposed a warning system, which can monitor elevators failures through sensors and send the data to the remote monitoring platform. The system can send the information and remind the maintenance personnel to deal with it as soon as possible.

An et al. (a) [18] proposed an elevator monitoring system with the aim of helping managers in elevator maintenance companies to detect elevator faults early and to ensure the safety of passengers. Several sensors such as, temperature, vibration, speed, and load were exploited to capture elevators’ data in real-time. This data is then sent through an industrial gateway to an InterProcess Communication (IPC) server before being sent to a SQL server through a General Packet Radio Service (GPRS) network. The web application also includes real-time video-monitoring, and a fault prediction model based on a Prognostics and Health Management (PHM) technology. The disadvantage of the system, however, is that it is a web-based application rather than a mobile application and does not provide notifications in case a fault is being detected.

An et al. (b) [19] proposed an intelligent elevator management system based on Building Information Modeling (BIM) and IoT technologies. Specifically, sensors such as temperature, humidity, speed, vibration, and load are employed to collect input data, then this data is sent through a RS485/CAN communication to the cloud for processing. The desktop application also includes real-time video monitoring as well as some functions such as, voice assistance and emergency assessment. The disadvantage of the system, however, is that it is a desktop application rather than a mobile application and does not provide means for detecting faults.

Shen et al. [20] proposed a predictive maintenance system using IoT and Machine Learning (ML) for a Permanent

Magnet Synchronous Motor (PMSM) traction elevator. First, raw data from the temperature and encoder sensors are captured then processed by Arduino. Data analysis using MATLAB is then performed on the digital data to compile it into a 30-minute data file. The data file then gets analyzed to determine which data requires predictive maintenance. The data is then classified into four categories: long, medium, short, and urgent according to predefined thresholds. Finally, this data is used for training the K-nearest Neighbor (KNN) ML model, so new data can be predicted accordingly. The accuracy of the system is 95.5%. The disadvantage of the system, however, is that it relies on two sensors' data only, and it is not a mobile application and does not provide notifications.

Guo et al. [21] described a system to monitor elevators in real-time based on multi-sensor fusion to ensure detecting common elevators faults then present the results on a website. The sensors consisted of magnetic switches, network camera, barometer, and accelerometer. The system was installed on a real elevator to test it.

Bai et al. [22] proposed a fault-prediction model based on improved PSO-BP which is optimized by an improved particle swarm optimization algorithm to enhance elevators safety using real-time data collected from the SCADA of the elevator. The convergence rate of the improved PSO-BP model is increased by 35.47%, and the prediction accuracy is improved by 49.12%

Gupta et al. [23] proposed a solution to use the elevators during COVID-19, without touching any surface to maintain sterilization and safety by deploying facial recognition software that uses pattern recognition, voice command using speech analysis method, and the body temperature to notify the people around them if someone has more than the suggested temperature. All these sensors are embedded in one device.

Table I shows a comparison between the related work according to the aim and criteria that we have developed. The first criterion (A) indicates whether the system is cloud-based or not. Secondly, (B) specifies if the system includes a fault detection model or not. Thirdly, (C) specifies if the system have video monitoring. Subsequently, (D) indicates whether the system has a mobile application or not. Finally, we determine if the system is implemented (E), available (F) or provide notifications (G). Out of the 18 methods, seven are cloud-based, fourteen systems includ fault detection (some of them were predictions), nine systems include video monitoring, only three systems are mobile applications, twelve are implemented, three systems are available, and only five systems provide notifications. As can be seen from the table, all approaches include at most four features, except for Guo et al. which includes five.

As far as we know, there isn't a system that provides all features. Therefore, we aim to build an IoT cloud-based fault detection mobile application, called Elevint, where data is collected from several sensors, combining related elevators worldwide. Data will then be analyzed in the cloud to be monitored and faults can be detected using rules determined by experts. Once a fault is detected, the system will alert the maintenance company, which in return will repair it. Thus, Elevint will include all features except for video monitoring, which we leave as future work.

TABLE I. COMPARISON BETWEEN RELATED WORK

Research by	Aim	A	B	C	D	E	F	G
Wang et al. (2011)	Reduce Data Collection Delay	×	×	×	×	✓	×	×
Jiang et al. (2015)	Fault Detection System	×	✓	✓	×	×	×	×
Zarikas and Tursynbek (2017)	Control the Elevators Actions	×	✓	✓	×	×	×	×
Salim and Akin (2017)	Fault Prediction	×	✓	×	×	✓	×	×
Zhang et al. (2017)	Elevator Monitoring	×	✓	×	✓	✓	✓	×
Suárez et al. (2018)	Inform About Failure	×	✓	✓	×	✓	×	✓
Olalere et al. (2018)	Remote Fault Indication	✓	✓	×	×	✓	×	✓
Ming et al. (2018)	Elevator Monitoring	×	×	✓	✓	×	×	×
Li (2018)	Elevator Monitoring	✓	✓	×	×	✓	×	×
Zhou et al. (2019)	Elevator Monitoring	×	✓	×	×	×	×	✓
Huang et al. (2020)	Fault Detection System	×	✓	×	✓	✓	×	✓
An et al. (a) (2021)	Fault Prediction System	✓	✓	✓	×	✓	×	×
An et al. (b) (2021)	Elevator Monitoring	✓	×	✓	×	✓	×	×
Shen et al. (2021)	Fault Prediction System	×	✓	×	×	✓	×	×
Guo et al. (2021)	Elevator Monitoring & Fault Detection	✓	✓	✓	×	✓	✓	×
Bai et al. (2021)	Fault Prediction	×	×	×	×	×	×	×
Gupta et al. (2022)	Elevator Monitoring & Fault Detection	×	✓	✓	×	×	×	✓
Our Proposed Solution: Elevint	Detection, Monitoring, Managing, Notification System	✓	✓	×	✓	✓	✓	✓

III. Proposed Solution

In this section, we explain our system, Elevint, which is a cloud-based remote elevator condition monitoring system that comprises a mobile application. Elevint's system architecture is depicted in Fig. 1 Elevint aims at enabling faster repairs, preventing catastrophic breakdowns, and assisting in fault diagnosis. Specifically, Elevint provides real-time monitoring of elevator operating conditions, such as vibration, temperature, weight, light, and movement including others, through fixed sensors connected to an Arduino microcontroller. The data collected from sensors is then transferred via Wi-Fi to the cloud, where faults are detected by applying rules that compare the current conditions with severity levels determined by an expert. For example, if the temperature exceeds 60 degrees Celsius in the engine room, this indicates a possible failure in the elevator engine. Therefore, notifications are sent to elevator owners and maintenance companies to speed up the process of reporting a breakdown and to reduce the downtime of elevators.

The Android-based mobile application connects elevator owners with maintenance companies and technicians, in which messages can be exchanged between them, as well as other tasks can be performed, such as scheduling periodic maintenance and viewing maintenance history. The main feature of the mobile application is that it enables all three users, to monitor the status of the elevator in real-time i.e., the elevator's current floor, the temperature, the vibration, among others. The data gathered from the sensors is sent to the FireBase real-time cloud-hosted database, through which it gets sent to the mobile application. The mobile application is set up to show updated sensor data at 25-second intervals (changeable). Using a cloud-based architecture provides a quicker maintenance service, as the maintenance team can access the elevator data online and analyze it to find out the likely nature of the fault. For the hardware platform, our system uses a computer with Windows operating system and uses Arduino to create the circuit in which the sensors get connected. As for the software platform, the system uses Basic4Android and Visual Studio to create the application.

A. Hardware Components

Many hardware devices were used to produce an electric circuit connected to an Arduino Mega using the Wi-Fi to collect the real data of the elevator and send it to the application. Following is a description of some sensors that were used.

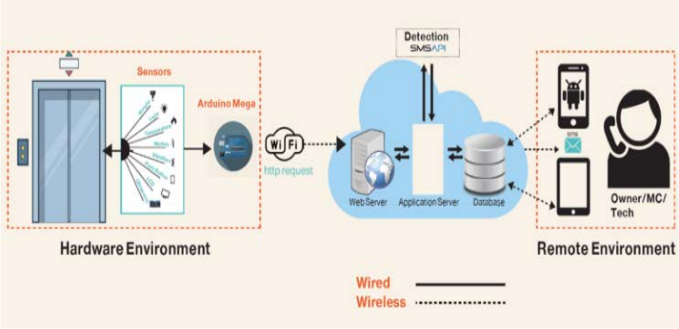


Fig. 1. Elevint System Architecture.

1) Temperature sensor: DS18B20 Temperature Sensor uses MAXIM's 1-wire bus protocol, which requires only 1 wire for receiving and transmitting data. The Temperature Sensor detects the temperature of the object it is attached on.

2) Passive infraRed (PIR) sensor: An electronic sensor that measures infrared (IR) light radiating from objects in its field of view. If the digital pulse is high (3V), it means a motion is detected, while if the digital pulse is low, it means it is idle or no motion is detected.

3) Photo resistor light sensor: When the value read from the photoresistor sensor module goes below a certain threshold value (determined by the developer), it means that it is dark, or the light is OFF. When the analog value from the sensor goes above the threshold value, it means the light is ON.

4) HC-SR04 ultrasonic sensor: The ultrasonic sensor emits an ultrasound that travels through the air. If there is an object or obstacle on its path, it will return to the module. Considering the travel time and the speed of the sound, you can calculate the distance.

5) Vibration sensor: Vibration is a sign of the state of the machine - no matter how accurate and unnoticed by the human senses -. Non-normal vibration of problems in the industrial machine can be detected early and repaired before machine failure occurs. Therefore, vibration analysis is used to determine the state of the equipment, location, and type of problems. Vibration sensors are sensors for measuring, displaying, and analyzing linear velocity, displacement, proximity, or acceleration.

6) Load cell and HX711 weight sensor: The *Load cell* senses the weight and provides an analog value to the *HX711 Load* Amplifier Module, which is an Analog to Digital Converter (ADC) that digitally converts the Load cell output.

7) LCD display: are used in electronic projects, as they are good for displaying information, such as the data that are collected from the sensors.

B. Connecting the System's Circuit

To connect the system's sensors, the circuit was made by connecting the module and sensors to the Arduino board, then the Arduino was connected to the computer using a USB cable. Fig. 2 shows the circuit connected, including the sensors and the Arduino.

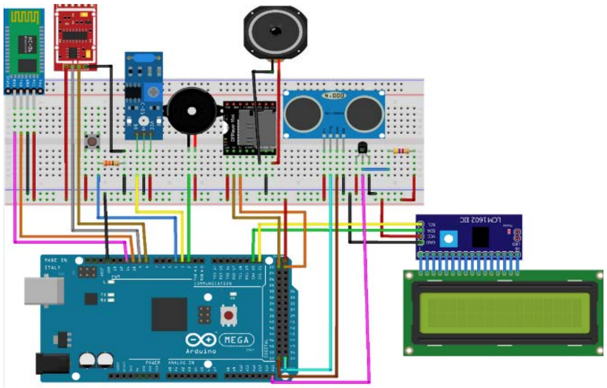


Fig. 2. The Circuit.

Table II shows the pins that are connected from the module to the pins on the Arduino of the circuit.

TABLE II. THE CIRCUIT OF ARDUINO

Module		**Arduino**
Temperature & Vibration	VCC	+ 5 V
	DO	D4
	GND	GND
Passive InfraRed	VCC	+ 5 V
	OUT	PIN2
	GND	GND
Ultrasonic	VCC	+ 5 V
	Echo	10~
	Trig	11~
	GND	GND
LCD Display	VCC	+ 5 V
	Echo	10~
	Trig	11~
	GND	GND

C. *Elevint Application*

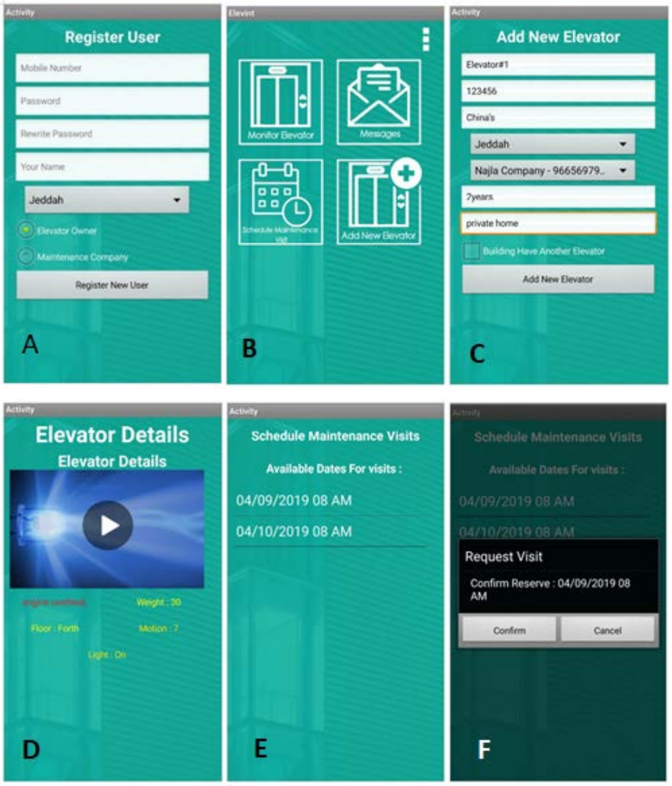
Fig. 3 and 4 show some pages from the Elevint application. Fig. 3 (A) shows the screen where a new user is registered, and after registering, the user can view screen (B) which shows the list of functions that suits the user of the application; whether they are an elevator owner, a maintenance company, or a technician. The functions provided for an elevator owner are: adding a new elevator, monitoring elevator data in real-time, sending messages to a maintenance company and scheduling a maintenance visit. Screen (C) shows adding a new elevator, Screen (D) provides a view of the elevator's real data, Screens (E-F) shows how to schedule a maintenance visit. In Fig. 4, screens (G-K) show how the user can view the maintenance history in detail. Screen (L) shows the message function which allows easy communication between the user and the maintenance company.

D. *Detection Rules*

The rules for detecting the various faults are of the form:

if condition then action(s) {else action(s)}

Note that the part between curly brackets indicates that it can be excluded. If more than one action is included, then actions will be numbered. Tables III to VI shows our rules for detecting a number of faults. The first row in each table indicates the goal of the detection rule, the second row specifies the used sensors, and the third row shows the detection rule. Note that these detection rules are not fixed rules, but rather heuristic gathered from experts in the field. The threshold values used in the rules were determined through the performed tests on the elevator model.



Register User

Mobile Number
Password
Rewrite Password
Your Name

Jeddah

● Elevator Owner
○ Maintenance Company

Register New User

Add New Elevator

Elevator#1
123456
China's
Jeddah
Najla Company - 96656979..
2years
private home

□ Building Have Another Elevator

Add New Elevator

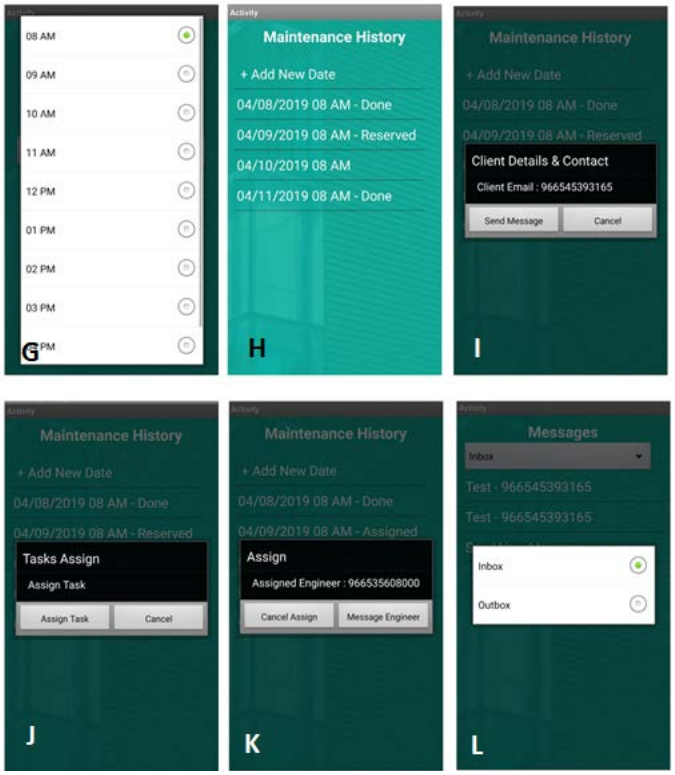
Elevator Details
Elevator Details

engine overheat
Weight : 30
Floor : Forth
Motion : 7
Light : On

Schedule Maintenance Visits
Available Dates For visits :
04/09/2019 08 AM
04/10/2019 08 AM

Request Visit
Confirm Reserve : 04/09/2019 08 AM

Fig. 3. App Pages for Maintenance Company Users - Part 1.



08 AM
09 AM
10 AM
11 AM
12 PM
01 PM
02 PM
03 PM
PM

Maintenance History
+ Add New Date
04/08/2019 08 AM - Done
04/09/2019 08 AM - Reserved
04/10/2019 08 AM
04/11/2019 08 AM - Done

Client Details & Contact
Client Email : 966545393165

Tasks Assign
Assign Task

04/09/2019 08 AM - Assigned

Assign
Assigned Engineer : 966535608000

Messages
Inbox
Test - 966545393165
Test - 966545393165

● Inbox
○ Outbox

Fig. 4. App Pages for Maintenance Company Users - Part 2.

Table III demonstrates rule #1, which detects if the elevator's engine is overheated. For instance, when the temperature exceeds 60 degrees Celsius in the engine room, it indicates the possibility of a fault in the elevator engine and therefore, the LCD screen and the Monitor Elevator screen on the Elevint application will show the exact temperature, and an SMS will be sent to the owner of the elevator indicating that an overheat engine fault is possibly detected in an elevator, giving its serial number. Note that notification_rule#6 is demonstrated in Table VII and will be explained in the next section. Otherwise, if the temperature of the engine is a normal value, it will appear on the LCD screen and will be displayed in the Elevint application. We have tested "overheated engine" faults by directing a lighter on the temperature sensor and the result was an SMS message sent to the mobile number of the owner registered on Elevint.

TABLE III. DETECTION RULE #1

***detection_rule#1*: Overheat Engine**	
Goal	Detect a possible fault in the elevator's engine (*e.g.*, fire)
Sensors	Temperature
Rule	**if** *temperature* > 60 ° **then** 1) *Print the temperature on the LCD screen* 2) *Display the temperature in the application* 3) *Send SMS notification to users according to* ***notification_rule#6*** *with fault* = 'engine overheat' **else** 1) *Print the temperature on the LCD screen* 2) *Display the temperature in the application*

Table IV shows rule #2, which detects if there are people trapped in an elevator. For instance, if Motion = High and Light = Off, that means there are people trapped in the elevator. Thus, the LCD screen and in the Elevint application will show: "Last Motion: 3 seconds ago", and Lighting: On. An SMS will also be sent to the elevator owner indicating that people may be trapped in the elevator with its serial number. Otherwise, the data will be displayed on the LCD screen and in the Elevint application. We have tested faults from type "Trapped People" by turning off the light and doing movements in front of the motion sensor and the result was an SMS message sent to the mobile number of the owner registered on Elevint.

Table V demonstrates rule #3, which detects if the elevator is stuck between floors. For instance, if the distance between floors is detected to be 15.5 cm and the vibration is 0, then the LCD screen and the Monitor Elevator screen on the Elevint application will show the message: "Floor: elevator is stuck", and an SMS will be sent to the owner of the elevator indicating that the elevator, giving its serial number, may be stuck. Otherwise, the data will be displayed on the LCD screen and on the Elevator Monitor in the Elevint application. We have tested "elevator stuck" faults by moving the elevator and stopping it between two floors and the result was an SMS message being sent to the mobile number of the owner registered on Elevint.

TABLE IV. DETECTION RULE #2

***detection_rule#2*: Trapped People**	
Goal	Detect possible trapped people in the elevator
Sensors	Motion and Light
Rule	**if** *motion = high* && *light = OFF* **then** 1) *Print last detected motion timing on the LCD screen* 2) *Display last detected motion timing in the application* 3) *Print light status on the LCD screen* 4) *Display light status in the application* 5) *Send SMS notification to users according to* ***notification_rule#6*** *with fault* = 'trapped people in the elevator' **else** 1) *Print last detected motion timing on the LCD screen* 2) *Display last detected motion timing in the application* 3) *Print light status on the LCD screen* 4) *Display light status in the application*

TABLE V. DETECTION RULE #3

***detection_rule#3*: Stuck Elevator**	
Goal	Detect if the elevator is stuck between floors
Sensors	Distance and Vibration
Rule	**if** (!((*distance >= 0 cm && distance <= 15 cm*) || (*distance > 15 cm && distance <= 25 cm*) || (*distance > 25 cm && distance <= 35 cm*) || (*distance > 35 cm && distance <= 45* cm && *vibration = 0* **then** 1) *Print 'elevator is stuck' on the LCD screen* 2) *Display 'elevator is stuck' in the application* 3) *Send SMS notification to users according to* ***notification_rule#6*** *with fault* = 'stuck elevator' **else** 1) *Print current elevator floor on the LCD screen* 2) *Display current elevator floor in the application*

Table VI demonstrates rule #4, which detects if the elevator is overloaded. For instance, if the weight is 1200 g, then the LCD screen that is attached on the elevator and the Monitor Elevator screen on the Elevint application will show the message: "Weight: elevator overloaded", and an SMS will be sent to the owner of the elevator indicating that the elevator, giving its serial number, is overloaded. Otherwise, the data will be displayed on the LCD and on the Elevator Monitor in the Elevint application. We have tested faults from type "overloaded elevator" by putting a weight on the weight sensor and the result was an SMS message sent to the mobile number of the owner registered on Elevint.

E. System Notification Rules

In our application, we send SMS notifications to users when certain events take place according to the rules of the form:

when event then action(s)

TABLE VI. DETECTION RULE #4

***detection_rule#4*: Overloaded Elevator**	
Goal	Detect if the elevator is overloaded
Sensors	Weight
Rule	**if** *weight >= 1000 g* **then** 1) *Print lift overloaded on the LCD screen* 2) *Display current weight in the application* 3) *Send SMS notification to users according to* ***notification_rule#6*** *with fault = 'elevator is overloaded'* **else** 1) *Print current weight on the LCD screen* 2) *Display current weight in the application*

where an event is an event generated by the various users of the application, such as selecting a choice, sending or receiving messages, booking an appointment, or assigning tasks, and action is a call to the function *sendSMS*, which takes two arguments, the receiver of the SMS and the message itself. Table VII below shows one notification rule, rule #6 which captures the case when a fault is detected.

TABLE VII. NOTIFICATION RULE #6

***notification_rule#6*: Detect Fault**	
Goal	Informing a maintenance company and an elevator owner about a detected fault.
Event	*a fault is detected*
Action	1) *sendSMS* (elevator owner, "Hi *owner_name*, a fault has been detected in the elevator with the serial number: *serial_no*, possible fault: *fault.*") 2) *sendSMS* (maintenance company, "Hi *mainComp_name*, a fault has been detected in the elevator with the serial number: *serial_no*, possible fault: *fault.*")

IV. EXPERIMENTAL RESULTS

We tested our fault detection rules by creating manually 10 faults for each of the 5 rules and calculating the results using the classification metrics from the information retrieval using the formulas below:

$$\text{precision} = \frac{TP}{TP+FP} \quad (1)$$

$$recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1 = 2 * \frac{precision*recall}{precision+recall} \quad (3)$$

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (4)$$

where *TP* is the number of correctly detected real faults, *FP* is the number of detected faults which were wrong (*i.e.*, not fault), *FN* is the number of undetected faults which were true (*i.e.*, failed to detect) and *TN* is the number of undetected faults which were not fault (*i.e.*, normal case).

Fig. 5 shows the results for all five detection rules, in which, the precision, calculated from (1), was 1 for all rules, while the recall, calculated from (2), was 1 in two rules only. For rules 1, 4 and 5 the recall scores were 0.86, 0.8, 0.9 respectively. The average precision for all rules is 1 and the average recall is 0.91. The F1 measure from (3) was also calculated and the result is 0.95. Finally, the accuracy from (4) was calculated and the result is 0.98.

Fig. 6 shows the testing of Detection Rule #4 and Fig. 7 shows the elevator model that was used to evaluate Elevint.

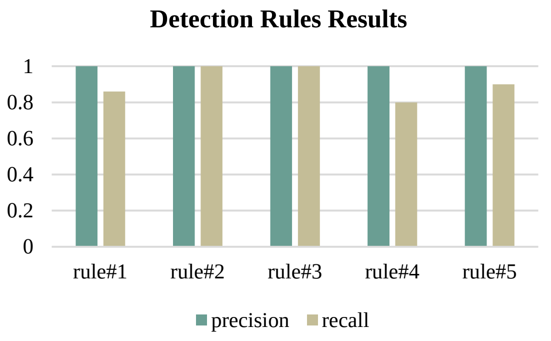


Fig. 5. Elevint's Detection Rules Results.

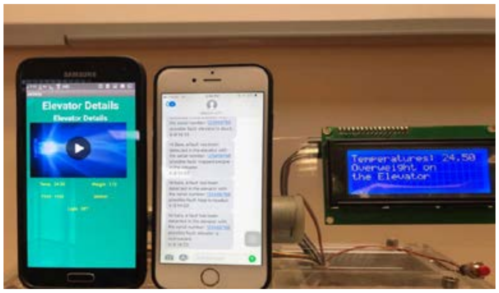


Fig. 6. Testing Detection Rule #4.

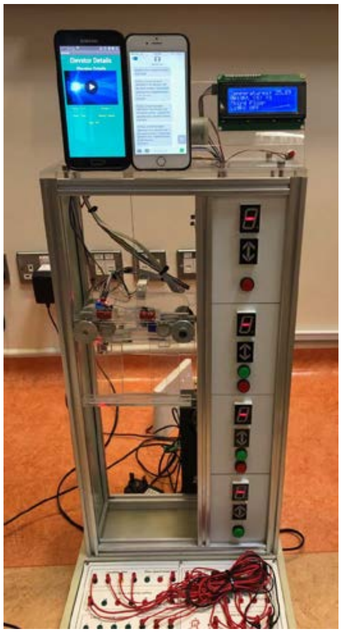
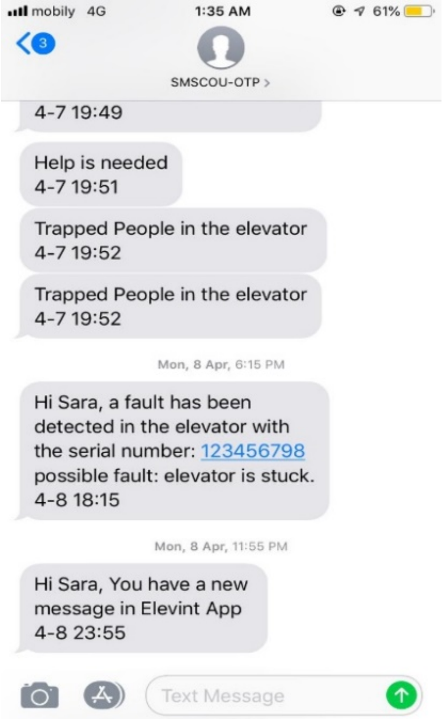


Fig. 7. Elevint's Model.

Fig. 8 shows the SMS that was sent to the owner due to the testing of Detection Rule #3.



4-7 19:49

Help is needed
4-7 19:51

Trapped People in the elevator
4-7 19:52

Trapped People in the elevator
4-7 19:52

Mon, 8 Apr, 6:15 PM

Hi Sara, a fault has been detected in the elevator with the serial number: 123456798 possible fault: elevator is stuck.
4-8 18:15

Mon, 8 Apr, 11:55 PM

Hi Sara, You have a new message in Elevint App
4-8 23:55

Fig. 8. Testing Detection Rule #3.

V. Conclusion

Undoubtedly, elevator failures can cause significant delays and threats to humans' lives if not handled quickly and efficiently. A main cause of the delay is the manual approach taken to report elevator breakdowns. The problem with most state-of-the-art approaches is that they concentrate on monitoring and detecting faults without providing automatic means for notifying users when these faults happen. Moreover, their fault detection mechanisms do not rely on rules specified by experts in the field. Furthermore, none of the previous approaches offered means for managing elevators and facilitating the communication between elevator owners, maintenance companies and engineers. As opposed to previous approaches, this paper has proposed Elevint, an IoT, cloud-based mobile application that provides managing, monitoring, fault detection and notification services. Specifically, through Elevint, elevator owners can request a maintenance visit and monitor their elevators' data in real-time, while maintenance companies can assign engineers to specific faults or scheduled visits, and view elevators' data and maintenance history. Moreover, all users can communicate with each other through Elevint via messaging. Furthermore, faults are detected by applying heuristic rules that compare the current conditions with severity levels determined by experts. Finally, when a fault is detected, elevator owners and maintenance companies are automatically notified. Testing Elevint on an elevator model showed promising results. In the future, we aim to increase the number of detected faults by adding more detection rules, and to test our system on real elevators. We also intend to improve the system further by offering fault prediction rather than fault detection using AI techniques.

References

[1] Perkel, C. (2016). Trapped in a Stuck Elevator? The Problem is Worsening in Canada, Experts Warn. Retrieved from TheStar: https://www.thestar.com/news/canada/2016/07/21/trapped-in-a-stuck-elevator-the-problem-is-worsening-in-canada-experts-warn.html (last visited: October 22, 2021).

[2] Hou, K., Tao, S., But , M., and Cho, K. (2016). Smart Elevator. 1-18.

[3] Lueth, K. (2016). The 10 Most Popular Internet of Things Applications Right Now. Retrieved from IoT Analytics: https://iot-analytics.com/10-internet-of-things-applications/ (last visited: October 22, 2021).

[4] Solanki, K. (2018). A Comprehensive Study on Smart City: Concept and Limiting Factors. International Journal of Advanced Research in Computer Science, 9(2).

[5] Shafique, M., & Rafiq, M. (2019). An Overview of Construction Occupational Accidents in Hong Kong: A Recent Trend & Future Perspective. Applied Sciences, 9(10), 2069.

[6] Propmodo. (2015). IoT Connected Elevators May Save Office Workers Years of Waiting, Retrieved From: https://www.propmodo.com/iot-connected-elevators-may-save-office-workers-years-of-waiting/ (last visited: November 30, 2021).

[7] Wang, F., Wang, D., & Liu, J. (2011). Utilizing Elevator for Wireless Sensor Data Collection in High-Rise Structure Monitoring. Paper Presented at the Proceedings of the Nineteenth International Workshop on Quality of Service, 1-9.

[8] Jiang, H., Shi, Y., & Qi, L. (2015). Design of Elevator Monitoring and Alarm System Based on WiMAX. Proceedings of the 2015 International Conference on Electrical, Computer Engineering & Electronics, 92-96.

[9] Zarikas, V., & Tursynbek, N. (2017). Intelligent Elevators in a Smart Building. Paper presented at the Future Technologies Conference 2017.

[10] Salim, O., & Akin, E. (2017). IoT Application for Fault Diagnosis and Prediction in Elevators. International Journal of Innovative Research in Science, Engineering & Technology, 6(4), 5737-5742.

[11] Zhang, Y., Sun, X., Zhao, X., & Su, W. (2018). Elevator Ride Comfort Monitoring and Evaluation Using Smartphones. Mechanical Systems & Signal Processing, 105, 377-390.

[12] Suárez, A., Parra, O., & Forero, J. (2018). Design of an Elevator Monitoring Application Using Internet of Things. International Journal of Applied Engineering Research, 13(6), 4195-4202.

[13] Olalere, I., Dewa, M., & Nleya, B. (2018). Remote Condition Monitoring of Elevator's Vibration and Acoustics Parameters for Optimized Maintenance Using IoT Technology. Paper presented at the 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE).

[14] Ming, Z., Han, S., Zhang, Z., & Xia, S. (2018). Elevator Safety Monitoring System Based on Internet of Things. International Journal of Online Engineering, 14(8), 121-

[15] Li, J. (2018). Design and Implementation of Elevator Internet of Things Security Control System based on Cloud Computing.

[16] Zhou, Y., Wang, K., & Liu, H. (2018). An Elevator Monitoring System Based on The Internet of Things. Procedia Computer Science, 131, 541-544.

[17] Huang, Q., Cao, J., & Sun, R. (2020, August). Design and Implementation of an Elevator Power Failure Warning System. In *Journal of Physics: Conference Series* (Vol. 1621, No. 1, p. 012050). IOP Publishing.

[18] An, Z., Liu, S., Bai, D., Wang, N., & Yu, X. (2021). Intelligent Monitoring System of Elevator Internet of Things. In IOP Conference Series: Earth & Environmental Science (Vol. 791, No. 1, p. 012127). IOP Publishing.

[19] An, Z., Bai, D., Huang, Y., Ning, W., Deng, Y., Gan, N., & Liu, S. (2021). Building Elevator Safety Monitoring System Based on the BIM Technology. In Journal of Physics: Conference Series (Vol. 1939, No. 1, p. 012026). IOP Publishing.

[20] Shen, L. J., Lukose, J., & Young, L. C. (2021). Predictive Maintenance on an Elevator System using Machine Learning. Journal of Applied Technology & Innovation (e-ISSN: 2600-7304), 5(1), 75.

[21] Guo, Y., Liu, Y., Zhang, X., & Wang, G. (2021). The Real-Time Elevator Monitoring System Based on Multi-sensor Fusion. In *Journal of Physics: Conference Series* (Vol. 2010, No. 1, p. 012182). IOP Publishing.

[22] Bai, D., An, Z., Wang, N., Liu, S., & Yu, X. (2021). The Prediction of the Elevator Fault Based on Improved PSO-BP Algorithm. In *Journal of Physics: Conference Series* (Vol. 1906, No. 1, p. 012017). IOP Publishing.

[23] Gupta, S., Tyagi, S., & Kishor, K. (2022). Study and Development of Self Sanitizing Smart Elevator. In *Proceedings of Data Analytics & Management* (pp. 165-179). Springer, Singapore.