

## Article

# Secure Autonomous Cloud Brained Humanoid Robot Assisting Rescuers in Hazardous Environments

Georgios Angelopoulos <sup>1,\*</sup> , Nikolaos Baras <sup>2</sup>  and Minas Dasygenis <sup>2,\*</sup> 

<sup>1</sup> Department of Information and Communication Technologies, Universitat Pompeu Fabra, 08002 Barcelona, Spain

<sup>2</sup> Department of Electrical and Computer Engineering, University of Western Macedonia, 501 00 Kozani, Greece; nbaras@uowm.gr

\* Correspondence: george.angelop@gmail.com (G.A.); mdasygenis@uowm.gr (M.D.)

**Abstract:** On 31 January 2020, the World Health Organization (WHO) declared a global emergency after the discovery of a new pandemic disease that caused severe lung problems. The spread of the disease at an international level drew the attention of many researchers who attempted to find solutions to ameliorate the problem. The implementation of robotics has been one of the proposed solutions, as automated humanoid robots can be used in many situations and limit the exposure of humans to the disease. Many humanoid robot implementations are found in the literature; however, most of them have some distinct drawbacks, such as a high cost and complexity. Our research proposes a novel, secure and efficient programmable system using a humanoid robot that is able to autonomously move and detect survivors in emergency scenarios, with the potential to communicate verbally with victims. The proposed humanoid robot is powered by the cloud and benefits from the powerful storage, computation, and communication resources of a typical modern data center. In order to evaluate the proposed system, we conducted multiple experiments in synthetic hazardous environments.

**Keywords:** assistive robot; cloud-brained robot; human-robot interaction; indoor-navigation; rescuer robot



**Citation:** Angelopoulos, G.; Baras, N.; Dasygenis, M. Secure Autonomous Cloud Brained Humanoid Robot Assisting Rescuers in Hazardous Environments. *Electronics* **2021**, *10*, 124. <https://doi.org/10.3390/electronics10020124>

Received: 17 November 2020

Accepted: 5 January 2021

Published: 8 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years, research in robotics has become one of the most popular fields in industry and academia, and with the recent popularity of cloud computing [1], there has been increased interest in applying similar concepts to robotics and autonomous systems. Cloud robotics [2] allows robots to take advantage of the rapid increase in communication bandwidth to offload tasks without rigid real-time requirements. This is of particular interest for mobile robots, where on-board computation entails additional power requirements, which may reduce the operating duration and constrain robot mobility.

Cloud-based robots have proven to have several applications and advantages over traditional networked-based robots. They can offload computation-intensive emerging applications to the cloud, such as artificial intelligence and machine learning. These types of robots are only required to be equipped with actuators, sensors and the minimum processing power necessary to communicate with the cloud and perform control and navigation tasks. As a result, the weight, cost and complexity of the robot are minimized and easier to acquire and maintain. The on-board software of the robots becomes simpler and requires fewer updates. Because server hardware on the cloud can be upgraded independently of the robots, the operational life of the robots is extended and enhanced with new features. Additionally, the cloud enables robots to access vast amounts of data: the robots can acquire information and knowledge to execute tasks through databases in the cloud. Their operation can be extended by only updating the software on the central server, without any need for individual local updates. Robots do not have to deal with

the creation and maintenance of such data. The cloud also enables robots to access shared knowledge and new skills: it provides a medium for robots to share information and learn new skills and knowledge from each other. The cloud can host a database or library of skills or behaviors that map to different task requirements and environmental complexities.

In the last few years, many research groups have been exploring the usage of cloud technologies in their applications. One example of a cloud-connected robot is Google's self-driving cars. Google's self-driving automobiles have shown that an unmanned car equipped with computer technology can travel safely from city to city [3]. Apart from that, a research group at Singapore's ASORO laboratory has built a cloud computing infrastructure to generate 3D models of environments, allowing robots to perform simultaneous localization and mapping (SLAM) much faster than by relying on their on-board computers [4]. A number of researchers have also been exploring the idea of using robots based on cloud computing infrastructure, with the aim of accessing a vast amount of processing power and data. Professor James Kuffner at Carnegie Mellon University has described the possibility of embracing the cloud in robots [5]. CPU-intensive tasks in robots can be offloaded from smaller and less power-consuming on-board computers to remote cloud servers.

Even though there are many advantages of cloud robotics and their applications, there are also some significant drawbacks that we have to address. Because the movement and motion of the robot depends on the installed sensors, internet connection and feedback from the controllers, it is sometimes very difficult for the robot to navigate an area. Also, due to the nature of cloud-based applications, cloud robots can be negatively affected by high latency or internet connection errors. Last but not least, these robots are developed by engineers that are not IT security experts, and due to the utilization of many communication protocols, they are vulnerable to malicious exploitation as a result of their larger area at risk of attack.

When a humanoid is performing in a hazardous environment, human–robot interaction, engagement and trust are crucial for the overall success of a search and rescue mission. Considering children's and adults' reactions when they saw a large robot-dog at a shopping center [6], we see that they started to develop positive emotions. More specifically, children were thrilled the first time they saw the robot-dog. After some time playing with it, they wanted to bring it back to their homes. Of course, a robot-dog has significantly fewer functions when compared to a humanoid robot, and the role it plays is different. Children's interactions (a victim in the case of an emergency) with a dog-robot would be different to their interactions with a humanoid robot, similarly to how they interact with other humans.

Robots have become increasingly central to our society, and human interaction with them is evolving from master–slave to peer-to-peer. Studies have also shown [7,8] that people tend to rely on automation that they trust and are reluctant to use equipment that they do not trust. Within the literature on trust in automation, complacency is conceptualized interchangeably as the overuse of automation, the failure to monitor automation and a lack of vigilance. For the optimal performance of a human–automation system, human trust in automation must be promoted.

In [9], trust is conceived to be an "attitude that an agent (automation or another person) will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability". The majority of research in trust in automation has focused on the relationship between automation reliability and operator usage, often without measuring the intervening variable of trust. The utility of introducing an intervening variable between automation performance and operator usage, however, lies in the ability to make more precise or accurate predictions with the intervening variable than without it. This requires that factors influence trust in automation in addition to automation reliability/performance. The three-dimensional (purpose, process, performance) model proposed by Lee and See [9], for example, presumes that trust (and indirectly, propensity to use) is influenced by a person's knowledge of what the automation is supposed to do (purpose), how it functions (process) and its actual performance. While such models seem plausible, support for the

contribution of factors other than performance has typically been limited to the correlation between questionnaire responses and automation use.

This research work is an extended version of the diploma thesis presented in [10]. It presents a cloud brained autonomous robot capable of assisting rescuers primarily in indoor hazardous environments. We have enhanced the research by including a better face detection algorithm, a solid indoor positioning system using Light Detection and Ranging (LiDAR) and a more robust system integration of all the software and hardware components.

The main novel contributions of this research are (a) a prototype cloud-based humanoid robot capable of navigating in hazardous environments to assist rescuers, (b) a low-cost implementation indoor navigation algorithm capable of avoiding obstacles, based on the Received Signal Strength Indicator (RSSI), (c) a secure cloud computing integration and (d) a modular architectural design, making future improvements easy and cost-effective.

The remainder of this paper is organized as follows. Section 2 presents related research. In Section 3, our materials and methods are presented. In Section 4, we showcase the performance evaluations. Finally, Section 5 concludes our paper.

## 2. Related Research

The combination of hazardous environments and cloud brained humanoid robots is rarely found in the related research literature. This may be attributed to the fact that robots are very complicated embedded systems that require high levels of expertise in embedded software and hardware. Not only there is not enough documentation for the robots to expand their capabilities, but also they bear a high acquisition cost.

As officials in countries around the world are currently racing to respond to a new wave of the COVID-19 pandemic, they have been implementing measures ranging from mass quarantines to robotic patient care to halt the spread of the disease. Due to the contagiousness of the pandemic, it is safer if human-to-human contact is minimized. Since robots are immune to infection, tech companies and researchers have stepped up to the challenge to bring more robots out to aid in this effort.

In [11], Evans presented the system technologies embodied in HelpMate in which hospital personnel (operators) place goods in specially designed shelves installed on the robot's platform, and then the operator selects a destination or destinations from the map displayed on the robot's GUI (Graphical User Interface). The robot then follows the build mission plan (the path to take). The robot starts to travel to the destination on a pre-planned path. If it meets an obstacle, the robot tries to go around it, and if there is no path around the obstacle, the robot waits until obstacles are removed. When goods are delivered, the person that receives them needs to confirm that the mission has finished.

Apart from that, the staff at Providence Regional Medical Center in Everett, Washington rely on a robot that can measure a patient's vital signs and act as a platform for video conferencing [12]. The robot, known as Vici, was developed by InTouch Health and has a high-definition screen and camera, allowing a doctor or a nurse to help a patient without physically meeting them. The robot is a platform on wheels with a built-in screen. These methods are not only ineffective but also inhumane for the patient, as robots like these look like a machine and cannot provide psychological aid.

In other hazardous environments, some robots, such as "Quince", developed by Chiba Institute of Technology in collaboration with Tohoku University and International Rescue System [13], "PackBot", developed by iRobot [14], and "Survey Runner", developed by TOPY Industries Ltd. [15], are deployed in the Fukushima Daiichi Nuclear Power Plant. These robots are crawler-type robots and are controlled by an operator who remains in a safe area. They measure temperature and radioactivity in hazardous zones and transmit video and sensory information back to the operator. They perform duties that are very important at the beginning of restoration work, but several tasks are challenging for them to execute.

Lytridis et al. presented a Nao robot as a solution to a remote education and support scenario during the pandemic [16]. In their paper, they demonstrated that the use of a social robot in conjunction with a human therapist yields better educational results, and they subsequently integrate robots into Greek Special Education. Despite the prominent results that this research shows, in the scenario, the human was not able to verbally interact with the humanoid robot in real-time; another drawback was that the robot was not able to act autonomously.

Autonomous navigation inside a dynamic hazardous environment that constantly changes from one point to another, while simultaneously avoiding objects, has many uses in several application domains, such as agriculture or the automated transfer of goods. Over the last decades, many researchers, scientists, hobbyists and students have proposed many different implementations [17–20]. In the past few years, the hardware and software available have improved greatly, and as a result, this field has attracted even more attention from the scientific community. Although we can find insightful approaches in some of the implementations, many of them have severe drawbacks such as high power requirements and computational costs. Using these option for an embedded system is impossible because of the aforementioned drawbacks.

In order to efficiently navigate within an environment, a humanoid robot must be equipped with a positioning mechanism. Here are the most common approaches found in the literature for determining the position of a robot:

1. GPS (Global Positioning System): On a mobile node, GPS is capable of calculating the exact coordinates. The main drawbacks are that it has an increased cost and is not power-efficient, losing Line of Sight (LoS) with the connected satellites can cause the robot to disconnect, and it is only suitable for outdoor locations.
2. Pedometers: A device that is capable of counting the steps of a moving person. It uses the movement of the hips in order to calculate steps. There are publications found in the literature that utilize this method for indoor navigation; however, they are not practical, and have very specific environmental requirements, such as the path-length and density of the specific network. This approach is also characterized by a high probability of error.
3. Ultrasound: A device (a) with a sensor that is capable of measuring the distance to another device (b) by utilizing the time the signal takes to travel between devices. The main drawback of this method is that the range of a sound signal is very small. Even though it has the potential to offer high degrees of accuracy, it is not cost-effective and bears a high implementation complexity.
4. Ultra-Wide Band (UWB) and Bluetooth Low-Energy Beacons (BLE): These are different technologies that have similar implementation procedures. Antennas or beacons are installed in multiple locations around the environment. The node is connected to multiple nodes, and using an algorithm, it can calculate its position. Even though the accuracy of both of these methods is very good, both of them bear a high implementation complexity and are not cost-effective, especially for the proposed humanoid robot. A mobile node with an ultrasonic sensor measures the distance to a node by exploiting the ultrasonic signal propagation time. However, the transmission range of an ultrasound signal is small, as it cannot propagate further than the radio-frequency wave. It also increases the size, cost and energy demand in each device. Therefore, even though the ultrasound-based localization approach can achieve high accuracy, it is not suitable for our project needs.
5. RSSI (Received Signal Strength Indicator): This represents the relationship between transmission and received power. It can be employed to compute the distance of separation between a transmitter and a receiver. This approach has been used in several mobility-aware MAC protocols. If there is a direct path between two nodes placed in the environment in which no signal interference and attenuation occurs, the received signal power  $P_r$  is related to the relative distance  $d$  between the transmitter and the receiver nodes with the inverse square law.

From the aforementioned research works, it is clear that there are multiple approaches for autonomous robot navigation. However, they usually come with high implementation complexity and are difficult to use in a small portable form. In the proposed work, we use an efficient algorithm along with inexpensive and reliable hardware. In contrast with other authors, we have developed a secure autonomous cloud brained humanoid robot for search and rescue operations with the ability to communicate with a victim. Our robot can recognize humans and immediately alert rescuers. We also have developed a secure web application as a control panel for rescuers. Our application contains information about the victim (a photo captured from the robot) and the real-time position of our humanoid automobile. As presented in Table 1, compared to other humanoid robots, the characteristics of the presented humanoid robot are crucial for providing emergency psychological aid to the injured in an emergency area.

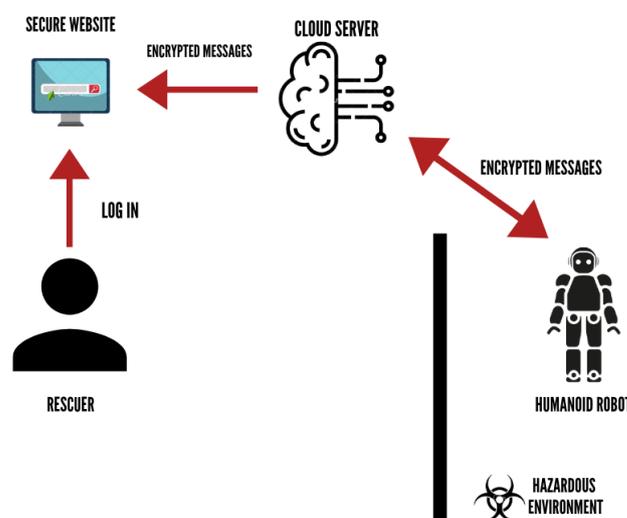
**Table 1.** A comparison of state-of-the-art robots used in hazardous environments.

Robot	Height	Weight	Anthropomorphism	Acquire New Skills through Cloud	Autonomous Movement	Price	Target Environment
Quince [13]	110 cm	45 kg	NO	NO	NO	\$140,000	Nuclear environment
Packbot [14]	88.9 cm	14.3 kg	NO	NO	NO	\$100,000	Many applications
HelpMate [11]	137.16 cm	260 kg	NO	NO	YES	\$70,000	Pandemic/hospital
Vici [12]	156.7 cm	54 kg	NO	NO	NO	\$100,000	Pandemic/hospital
Atlas [21]	150 cm	80 kg	YES	NO	NO	N/A	Search and rescue tasks
Spot [22]	84 cm	25 kg	NO	NO	YES	\$74,500	Many applications
NAO (A Nao robot that integrates the proposed system)	58 cm	4.5 kg	YES	YES	YES	\$10,000	Many Applications

### 3. Methodology

#### 3.1. System Architecture

The proposed system architecture of this research project is presented in Figure 1. The system consists of one humanoid robot that communicates in an encrypted manner with a cloud server and a secure website to inform rescuers. Our proposed architecture is a system in which an Internet connection is mandatory in order for our humanoid robot to detect and communicate with the victim. In the following subsections, we detail the hardware and software components, as well as the communication and encryption between the device and the server.



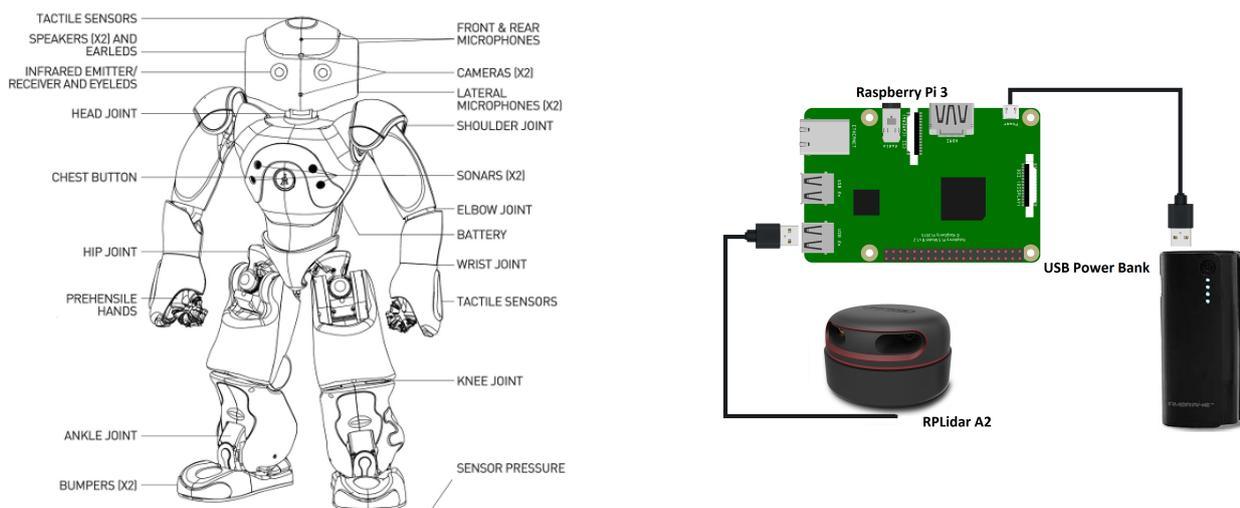
**Figure 1.** In the proposed messages system architecture, our robot offloads processing tasks to a cloud server.

### 3.2. Hardware: Components and Integration

Currently, the cost of a humanoid robot is a disincentive for labs or research teams that cannot build reliable legged robots. Researchers and engineers must devise functional robots, and specialized companies must manufacture them to ensure good industrial integration. For our research, we had at our disposal the academic version of NAO model H25. Some of the features of the robot include an on-board programmable computer with an x86 AMD Geode CPU with a 500 MHz clock, 256 MB of SDRAM and 1 GB of flash memory, WiFi and Local Area Network (LAN) connections. The height of the robot is 58 cm, and its weight is 4.3 kg. The running time of the robot is around 90 min before its 21.6 V Li-ion battery is depleted. It has two cameras with a resolution of  $1280 \times 960$  that can record up to 30 frames/s, two hands with self-adapting gripping abilities but with a single-engine that controls the three fingers of each hand, so they cannot be moved independently. It has force-sensitive sensors on its hands and feet to perceive contact with objects, light-emitting diodes on the eyes and body, four microphones to identify the source of sounds and two loudspeakers with which it communicates. It has 25 degrees of freedom in the joints, allowing the movement of head, shoulders, elbows, wrists, firm, waist, legs, knees and ankles independently.

Due to Nao's design, our humanoid robot cannot perform in every situation. However, it can perform with ease in situations with biological or chemical or, in some cases, physical hazards, such as in a poor air quality environment. Air does not necessarily have to be contaminated to be dangerous. Low oxygen or high dust levels can be fatal. Apart from that, Nao can perform in an extreme decibel level environment; prolonged exposure to excessive noise can damage or destroy hearing capabilities on humans, or cause even more severe problems.

Our implementation of an autonomous humanoid robot consists of two main components, besides the Nao robot: a Raspberry Pi 3B+ and a Slamtec RPLiDAR2 module. For the detection of surrounding obstacles, we used LiDAR—a Light Detection and Ranging module. LiDAR is a surveying method that continuously sends pulsed light into every direction ( $360^\circ$ ) and measures the reflected pulses at each angle with a sensor. The distance of each nearest obstacle is calculated based on the time and wavelength of the received pulse. LiDAR was suitable for our project needs because it can operate in low-light situations, which are very common in hazardous environments, so it was crucial that the proposed robot could operate in these conditions without problems. In order to detect surrounding obstacles and be able to navigate within the room, all of these captured measurements needed to be processed and seeded to our algorithm. The Raspberry Pi 3 B+ was used for this operation, which is an inexpensive, small and robust single-board computer (SBC). The Raspberry Pi's main advantage is that it has high customizability, mainly because of the plenty (40) of general purpose input–output (GPIO) pins that it offers. The power requirements of the Raspberry Pi and the high availability of software made it one of the best candidates for our purposes. We intentionally decided to use separate power sources for the Raspberry Pi and the Nao robot so that, in an extreme case in which something were to go wrong and in which the battery of the Nao robot were wholly discharged, we could still send commands to the Raspberry Pi and receive information about the surrounding environment. According to our experiments, the components' average power consumption during operation was 3.7 Watts for the Raspberry Pi, 1.9 Watts for the RPLiDAR module, and 45 Watt for the Nao robot. The total cost of the proposed prototype was 10,383 \$, with Nao being the most expensive part at 9999 \$, followed by the RPLiDAR A2, priced at 299 \$, and finally the Raspberry Pi 3B+ at 55 \$. The power source for the Raspberry Pi and LiDAR cost 30 \$. Figure 2 depicts the Nao robot, its components and the RPLidar A2 module along with the Raspberry Pi 3B+.



**Figure 2.** Hardware configuration: The Nao robot, its components and the connection scheme of the RPLidar A2 module and Raspberry Pi 3B+.

### 3.3. Software: Cloud Server

One of the contributions of our research is that, for the first time, we propose the usage of a cloud server that performs the intensive computations for face detection, decision making, control and natural language processing. We coined the term “Thin-Robot”, derived from the term “thin-terminals” in cloud computing jargon, because the robot does not require a state-of-the-art CPU, due to the offloading of the decision making to another computer. A low-cost CPU interfaced with sufficient sensors can be utilized, reducing the TCO (total cost of ownership). This is precisely the mantra of cloud computing.

The server, which is located in the cloud, is the brain of the proposed system. Security is fundamental for our operation, so our robot communicates with the server via an encrypted channel and sends the captured measurements. Initially, the robot is activated and a communication channel is formed. Then, the rescuing process starts with the appropriate signal from the rescuer–user.

At first, the robot constantly checks the environment for surrounding obstacles and sends aggregated sensor readings back to the server. With these values, the server decides whether the robot should move forward or it is necessary to use the camera. If there is an obstacle in front of the robot, the server sends an encrypted message to Nao, forcing it to use the camera and capture a photo. The robot sends the encrypted photo following the last demand from the server using socket programming. After that, if the server does not detect a person on the image, it sends the necessary movement action in order to avoid the obstacle to Nao. This procedure is repeated unless the server detects a person in the image or the rescuer deactivates the robot.

### 3.4. Software: Communication

We used socket programming because our system needed to use a reliable protocol with a small communication overhead to avoid high-latency responses. Socket programming is the fundamental technology behind communications in Transmission Control/Internet Protocol (TCP/IP) networks. A socket is one endpoint of a two-way link between two programs running on a network. The socket provides a bidirectional communication endpoint for sending and receiving data with another socket. Socket connections typically run between two different computers on a local area network (LAN) or across the Internet, but they can also be used for interprocess communication on a single computer. Our architecture was designed in such a way that is feasible to be deployed even in non-routable networks consisting of RFC 1918 Private Internets [23], because the robot performs outbound queries only and no port forwarding is required. To create a socket, we used the `socket.socket()` function available in the `socket` module. Once we obtained

the socket object, the required functions to create the client (the humanoid) and server (the cloud) programs could be used.

Another significant difference from other systems is the connection between the Nao robot and our server in the cloud. In the proposed research, we utilize the Wi-Fi protocol. Wi-Fi consists of multiple network technologies of the IEEE 802.11 standard. It is often used for LAN and Internet access. The number of connected devices on the LAN as well as the characteristics of the specific Wi-Fi network determine the data rates. The main advantage that open Wi-Fi networks offer is that they are very common in modern cities. We can use them to communicate with our servers located in the cloud and further decrease the implementation cost of the project. Compared to other solutions, such as GSM or LoRa, Wi-Fi is less expensive, requires minimal hardware (such as antennas) and is easier and faster to implement.

As with all Internet communications, the communication between the Nao robot and the cloud server suffers from latency. The amount of latency (expressed in milliseconds) is mainly determined by the physical location of the server and the strength of the Wi-Fi signal of the Nao robot. Poor Wi-Fi signals and network congestion can be responsible for packet loss—a phenomenon whereby data packets do not reach their destination. In these cases, the packets have to be re-transmitted, causing even higher amounts of delay. In our experiments, the location of the cloud server was in the same city as the Nao robot, and the latency was in the range of 15–25 ms with 0% packet loss, using a bandwidth of 3000 kbps (2200 kbps for video, 300 kbps for control and 500 kbps for audio).

### 3.5. Software: Encryption

In order to develop a secure system, we established an encrypted communication channel between our humanoid robot and the server, using the popular AES-128 encryption scheme.

AES (Advanced Encryption Standard) is a block cipher that uses a symmetric key, operating in various lengths of 128 bits, 192 bits and 256 bits. A cipher block typically consists of two paired algorithms: one to encrypt at the side of the sender and one to decrypt at the receiver's side. Because the AES cipher uses a symmetric key, both the sender and the receiver know and use the same key. This key is also known as a private key. With the message  $P$  and the key  $K$  as input, the encryption algorithm forms the cipher-text  $C$ , as mentioned in Equation (1).

$$C = E_K(P) \quad (1)$$

This shows that the cipher-text  $C$  is produced by using encryption algorithm  $E$ , as a function of the plain text  $P$ , with the specific function determined by the value of the key  $K$  [24]. Only the receiver that possesses the correct key is able to invert this transformation process and regenerate the original plain text.

The inverse transformation that is performed using a decryption algorithm  $D$  as a function of the cipher-text  $C$  [24] is presented in Equation (2).

$$P = D_K(C) \quad (2)$$

The AES algorithm is an iterative cipher comprising computational rounds for both encryption and decryption. For every additional 32 bits in the cipher-key, the number of rounds is increased by one [25]. An AES 128 bit cipher is used in the proposed design.

### 3.6. Software: Face Detection

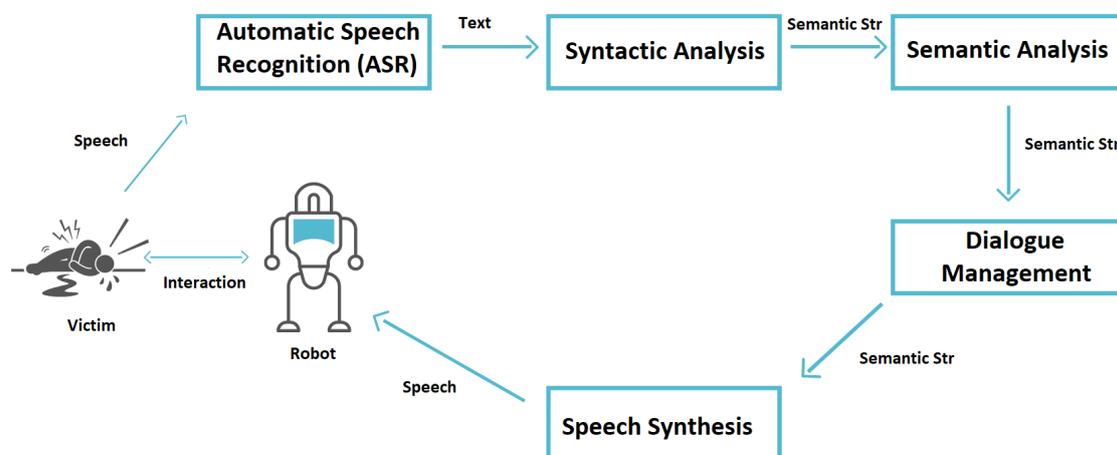
Face detection is a fundamental research problem in the field of computer vision and pattern recognition. The most classic face detection method is the Viola and Jones face detection method, proposed by Viola and Jones in 2001 [26], which uses simple Haar features and a cascade AdaBoost classifier for face detection to achieve efficient and real-time performance for face detection.

The field of face detection has achieved significant progress in recent years. However, high-performance (in real-time) face detection still remains a very challenging problem [27]. The face detection method based on convolutional neural networks (CNNs) has made a breakthrough in this area and has become one of the most promising techniques. In our work, we have implemented a very popular neural network proposed by Zhang et al. [28], the MTCNN (Multi-task Cascaded Convolutional Neural Networks), with 98% accuracy—much better than the Viola and Jones algorithm. MTCNN is a convolutional neural network that uses multi-level network cascading and multi-task training for face detection. It consists of three stages: in the first stage, it produces candidate windows quickly through a shallow CNN; it then refines the windows to reject a large number of non-faces windows through a more complex CNN; finally, it uses a more powerful CNN to refine the result and output facial landmarks positions. In order to achieve a cloud-based humanoid robot, the proposed implementation, including face detection, was performed on the cloud server since this process requires a significant amount of computing power.

### 3.7. Software: Dialog Management with the Victim

In our case, it was essential to design a humanoid robot to interact with and ask for important information from the victim; for instance, their health status. Therefore, we used a frame-based conversational system, also called a form-based system [29]. The dialog management architecture is depicted in Figure 3.

When the system receives the input, it first has to recognize the words in the input. Thus, Nao can convert audio to text by applying Google Speech API. The next action, syntactic parsing, gives a detailed analysis of the syntactic structure, which can be very beneficial when the system has to distinguish between similar sentences. When pattern matching is used, the input sentence only needs to contain some keywords to be considered a match. The system is equipped with scripts that contain a sequence or a set of keywords. The sentence is examined for keywords, and if all keywords are represented and no other script matches, it is considered a match, and a response can be generated automatically. For a system to be able to participate satisfactorily in a dialog, it has to be capable of recognizing and analyzing different dialogs; it can, for example, receive a request, a question or a reply. The dialog model of the system works as a guide and helps the system to characterize the input from the user and to perform an appropriate response to the input. Lastly, for speech synthesis, we applied Nao's text-to-speech function locally, optimizing the upstream data to the robot.



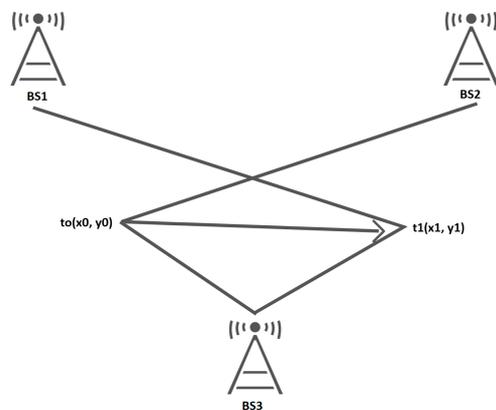
**Figure 3.** The dialog management architecture that we have implemented into our robot, which handles text-to-speech and speech-to-text vocal communication.

### 3.8. Software: Indoor Positioning System

To calculate the robot’s position inside the building, and for it to be able to navigate, it was necessary to implement an indoor positioning system (IPS). The usage of dedicated extra equipment increased the cost and the complexity of our project; because of this, we decided to use the existing WiFi infrastructure for navigation. Even though Ultra-Wide Band (UWB) and Bluetooth Low-Energy Beacons (BLE) offer higher accuracy, Wi-Fi-based IPS approaches have a lower cost, which was very important for our project. In a previous research work [30], we experimented with RSSI (Received Signal Strength Indicator) and its applications in robot positioning and navigation. The experimental results were very promising, and so we decided to implement this approach for the proposed humanoid robot in this paper as well. The typical scenario that we considered consisted of an indoor location with multiple WiFi stations from different locations broadcasting a signal. Modern workplaces already have multiple WiFi stations; therefore, this typical scenario is not far from reality. Furthermore, compared to UWB and BLE implementations, Wi-Fi-based IPS does not require the installation of extra equipment. In order to calculate the exact position of the robot at any given time, it is necessary that the robot must be in contact with at least three WiFi stations. The relative physical position of all WiFi stations, the starting position and the robot’s ending position are considered to be known. The Received Signal Strength (RSS) is the actual signal strength at the receiver, and it is measured in milliwatts (mW) or decibel milliwatts (dBm). The higher the RSS value, the smaller the distance between the transmitter (Tx) and the receiver (Rx). The RSSI distance calculation generally uses the logarithmic distance path loss model [31], which is expressed in Equation (3).

$$RSSI = -10n \log \left( \frac{d}{d_0} \right) + A + X_\sigma \tag{3}$$

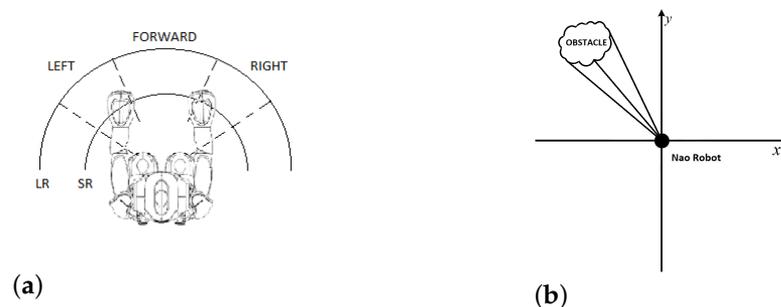
where  $d$  is the distance between the receiver and the transmitter,  $n$  is a path loss parameter related to the specific wireless transmission environment,  $X_\sigma$  is the random Gaussian distribution variable, and  $A$  is the RSSI with a distance of  $d_0$  from the transmitter. More obstacles lead to a larger value of  $n$ . RSSI’s approach only allows us to calculate the distance of each robot relative to each WiFi Broadcast Station (BS). To calculate the exact position of the robot, we used the triangulation technique proposed in [32]. At the robot’s location, if more than 3 BSs were broadcasting, the 3 BSs with the strongest signal were selected. Finally, to calculate the current direction of the robot, a short memory buffer was used. The proposed algorithm compares the robot’s current position to the previously calculated position every few seconds and determines the robot’s current direction. For example, at  $t_0$ , the robot has coordinates  $(x_0, y_0)$ , and at  $t_1$ , the coordinates are  $(x_1, y_1)$  (Figure 4). Using the proposed method, we were able to calculate the current direction of the robot.



**Figure 4.** To calculate the direction of the robot, we computed the coordinates according to the Received Signal Strength Indicator (RSSI) levels  $x_0, y_0, x_1$  and  $y_1$ .

### 3.9. Software: Obstacle Detection

The LiDAR module that we used (RPLidar A2) is capable of capturing 8000 measurements each second. The degrees in which measurements were taken ranged from  $0^\circ$  to  $360^\circ$ . The maximum distance in which an obstacle could be detected is 18 m. The LiDAR module returned these measurements in a point cloud. Each measurement reflected an obstacle and was described by the following numbers: distance, signal strength and degrees. Based on each point's degrees, we categorized the measurements into four classes: left, right, back and forward. For each class, each of these measurements was divided into the following categories: short range (SR) and long range (LR). Measurements that fell between 0 cm and 115 cm were placed in the short range category, and measurements between 116 cm and 1800 cm were placed in the long range category (Figure 5). SR measurements were more heavily weighted compared to the LR measurements.



**Figure 5.** In (a), we see LiDAR detecting an obstacle. In (b), collected data are categorized based on the measurement location.

## 4. Performance Evaluation

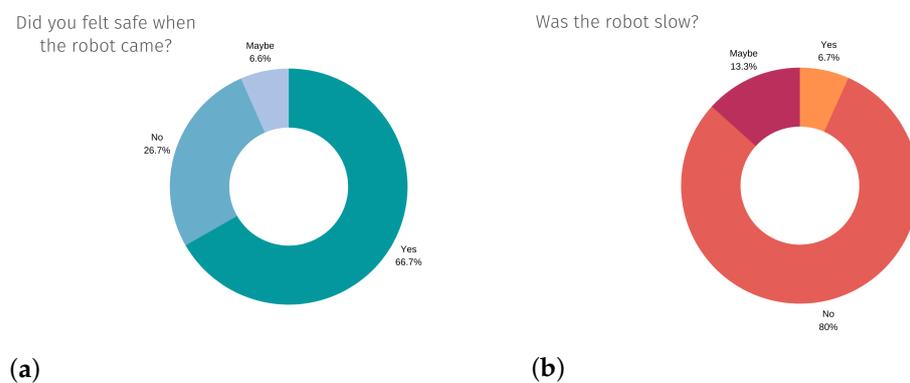
In order to evaluate the proposed humanoid prototype, we had to perform several experiments. However, during that time, because of the COVID-19 pandemic, we were skeptical about conducting experiments and possibly transmitting the disease. An interesting study [33] revealed that human coronaviruses such as Severe Acute Respiratory Syndrome (SARS) coronavirus, Middle East Respiratory Syndrome (MERS) coronavirus or endemic human coronaviruses (HCoV) can persist on inanimate surfaces such as metal, glass or plastic for up to 9 days, but can be efficiently inactivated by surface disinfection procedures with 62–71% ethanol, 0.5% hydrogen peroxide or 0.1% sodium hypochlorite within 1 min. The Nao robot mainly consists of plastic, which meant that without disinfection procedures, it was possible for the robot to carry the virus for a maximum of 9 days. Bearing that in mind, we programmed the humanoid robot to avoid close contact with victims.

Apart from that, according to the Centers for Disease Control and Prevention (CDC) and the World Health Organization, the risks of transmission of coronavirus from an item such as the proposed humanoid robot are very low. This is because the virus is generally thought to be spread most often by respiratory droplets that are transmitted by sneezing and coughing.

As an initial test, we created a simulated disaster environment in a 27 m<sup>2</sup> room. Then, we recruited 50 volunteers—42 of which were undergraduate students and eight of which were Ph.D. students from the University of Western Macedonia, with an average age of 22—to each perform an evacuation scenario. Even though all the participants of the experiment had engineering backgrounds and we understand that human bias may have altered the results, we believe that the conducted experiments are a good indicator of the effectiveness of the proposed prototype. The experiment took place in Greece in two phases at the end of the pandemic wave on April 2020 and at the beginning of the second pandemic wave on September 2020. Our results indicated that all the victims followed the instructions of our robot. However, this simulation was low-stress and straightforward, meaning that victims may react differently in real hazardous environments. In real disasters, many people will

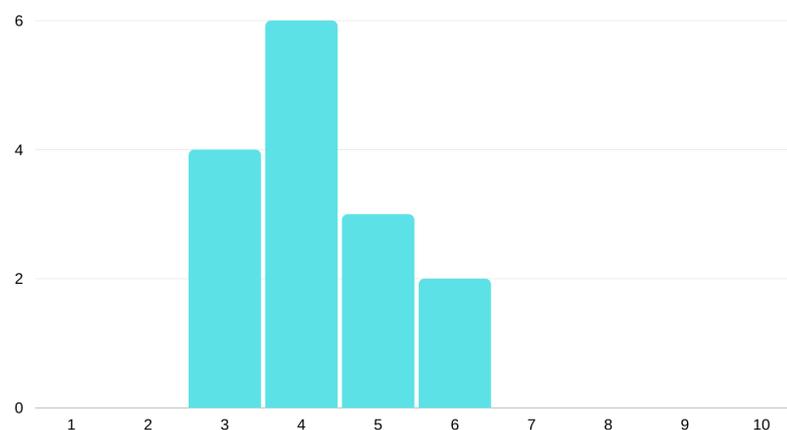
risk personal injury and even death to find members of their group [34]. After the test finished, we received feedback from our volunteers about the experiment. We asked three essential questions about their emotions when the robot approached them:

- “Did you feel safe when the humanoid robot arrived?” (Figure 6a)  
In total, 66% of the respondents described that they felt safe when the robot came; 26% of our volunteers answered no.
- “Was the robot slow?” (Figure 6b)  
In total, 80% of our volunteers answered no; 6.7% of the answers described the robot as slow.
- “From 1 to 10, how well did the environment simulation resemble a realistic hazardous environment?” (Figure 7)  
The majority of our volunteers graded the environment between 4 and 6. So, as shown by the answers, this simulation was low-stress and simple.



**Figure 6.** (a) Question about invulnerability; (b) question about reaction time.

From 1 to 10, how well did the environment achieved to simulate a hazardous environment?



**Figure 7.** Question about the room.

These findings raise an interesting question: How distant is the era in which autonomous robots, such as the prototype we propose here, can blend in with human rescuers in real catastrophic events, cooperating with them in order to save more lives? Investigating this further would be a fruitful avenue for future research that would bring benefits to society. In summary, we found that humans tended to feel safe with the humanoid robot.

As depicted in Figure 8, with the obstacle ordinance, our robot communicated with the system 13 times (points marked from A to M) via an encrypted channel in order to decide how to move inside the room. The system was able to detect the victim within 1 min and 49 s in this specific environment.

As we have discussed, security is fundamental for our system. Thus, in order for the user–rescuer to view information from the website, the first step is to log in with the given credentials. As long as the user inputs the correct credentials in the login page, they can view the control panel and monitor the operational status and view the room, as shown in Figure 8.

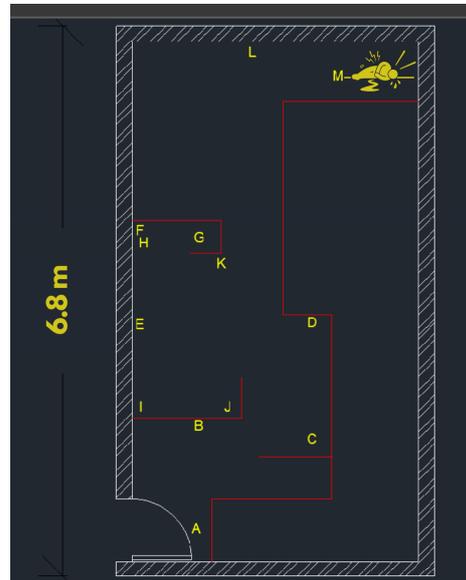
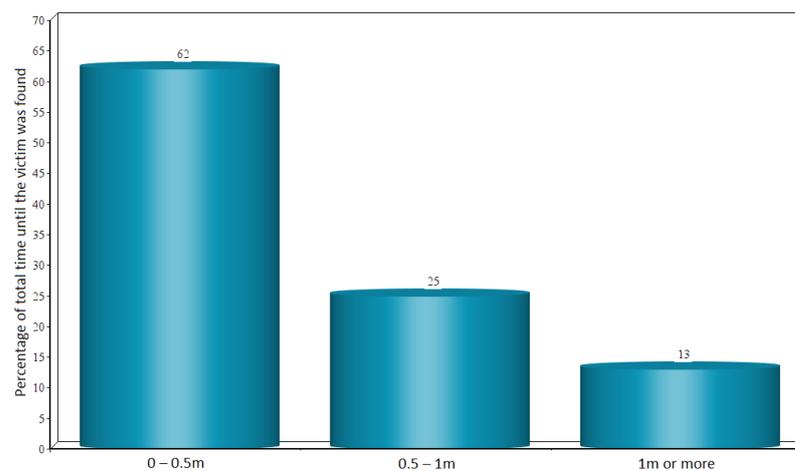


Figure 8. The room we used for experiments.

Since disaster robots are capable of being in excessively dangerous environments, they must not be accessed by an external entity. An unauthorized entity may take control of a robot which has been deployed to disconnect a nuclear platform. This can cause a hindrance to the disconnection process. To explore our system’s safety, we performed an experiment in which a penetration tester tried to hijack our robot. From this experiment, we discovered some security holes in our system. Firstly, the penetration tester cracked our Wi-Fi password using AirSnort [35]. Then, using the technique of the man-in-the-middle attack [36], the tester detected our robot’s IP, and finally, the penetration tester connected with the robot and disabled it. Having a robust encryption mechanism on wireless access points and a strong password for our humanoid robot would prevent unwanted users from joining our network simply by being nearby. A weak encryption mechanism could allow an attacker to brute-force their way into the network and begin a man-in-the-middle attack; the stronger the encryption implementation, the safer the system.

As mentioned above, the robot communicated with the cloud server in order to send feedback and receive new instructions. In this particular environment, the robot communicated 24 times to decide how to move in the room. Moreover, the system detected the victim in an average time of 1 min and 48 s. During that time, the actual position of the robot was very close to the detected position of the robot (Figure 9) using the RSSI method mentioned in Equation (3). Even though our experiment was performed in synthetic low-stress conditions, we could derive useful and comparable conclusions because our procedure was on a par with similar research works such as [37,38], who reported that it is very challenging to effectively simulate a plausible environment.



**Figure 9.** Error rate of the detected position of the robot compared over the operation time.

## 5. Conclusions

In the near future, millions of humanoid robots are expected to be deployed in the field and make instant decisions in critical situations involving the life and death of humans. Such machines must have the possibility to immediately acquire new, essential skills and information by connecting their “brains” to a computer network. This paper proposes a novel, fast and secure system that utilizes a cloud server as a brain connected to a humanoid robot with minimal communication latency. Such a system gives the capability to the humanoid to move autonomously and detect survivors in emergency scenarios and additionally to interact with victims. Our research has been materialized into a prototype robot, and several experiments have been performed to validate its usefulness. The implications of the findings and the critical discussions in this paper make it evident that more emphasis should be placed on research in this domain, as the death rate of humans in hazardous environments should be minimized. This work is the first to propose the usage of a cloud humanoid robot in hazardous environments and opens an area with multiple possibilities that researchers should delve deeper into. During our research, many challenges have emerged, such as the latency of communication between the robot and the cloud server or the shortcomings of the weak WiFi signal in some spots. This robot is the first step in establishing the cloud paradigm in the robot era, but more research is required to transform this prototype into a consumer device.

Furthermore, considering the satisfying results that this study produced, in the future, we plan to enhance the software with the aim of creating an architecture which leverages the combination of an ad-hoc cloud [39] formed by machine-to-machine (M2M) communications among participating robots with a cloud-enabled infrastructure using machine-to-cloud (M2C) communications. In this scenario, multiple robots with form a swarm, establishing ad-hoc communications between each other and relaying data to a cloud server, which will analyze all inputs and distribute tasks, leading the cohort of devices. In the future, we plan to experiment with different components other than the Nao and the Raspberry Pi to further decrease the manufacturing cost and make the robot accessible to more people. Because the proposed methodology utilizes the power of the cloud to process and analyze data, we can experiment with many low-cost components that would otherwise be impossible to use.

**Author Contributions:** M.D. was the supervisor of this manuscript; G.A. and N.B. contributed equally to the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** The experiments of this study were conducted at the University of Western Macedonia. All the students participated in this experiment have given their signed consent, according to the University's Regulation on Ethics (found in: <https://bit.ly/2LqtSXX>).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Foster, I.; Zhao, Y.; Raicu, I.; Lu, S. Cloud Computing and Grid Computing 360-Degree Compared. In *2008 Grid Computing Environments Workshop*; IEEE: Piscataway, PA, USA, 2009; Volume 5. [CrossRef]
2. Wan, J.; Tang, S.; Yan, H.; Li, D.; Wang, S.; Vasilakos, A.V. Cloud robotics: Current status and open issues. *IEEE Access* **2016**, *4*, 2797–2807. [CrossRef]
3. John, M. Google Cars Drive Themselves, in Traffic. Available online: <https://www.nytimes.com/2010/10/10/science/10google.html> (accessed on 9 October 2010).
4. Arumugam, R.; Enti, V.R.; Bingbing, L.; Wu, X.; Baskaran, K.; Kong, F.F.; Kumar, A.S.; Meng, K.D.; Kit, G.W. DAVinCi: A cloud computing framework for service robots. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, Anchorage, AR, USA, 4–8 May 2010; pp. 3084–3089.
5. Kuffner, J. Cloud-enabled humanoid robots. In *Proceedings of the 2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Nashville, TN, USA, 6–8 December 2010.
6. Weiss, A.; Bernhaupt, R.; Lankes, M.; Tscheligi, M. The USUS evaluation framework for human-robot interaction. In *Proceedings of the 23rd Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour, AISB 2009*, Edinburgh, UK, 6–9 April 2009; pp. 11–26.
7. Muir, B.M.; Morayi, N. Trust in automation experimental studies of trust and human intervention in a process control simulation. *Ergonomics* **1996**, *39*, 1905–1922.
8. Stephan, L.; Mundy, M.; Tan, G. The dynamics of trust: comparing humans to automation. *J. Exp. Psychol. Appl.* **2000**, *6*, 104.
9. Lee, J.D.; See, K.A. Trust in automation: Designing for appropriate reliance. *Hum. Factors J. Hum. Factors Ergon. Soc.* **2004**, *46*, 50–80.
10. Angelopoulos, G. Secure Autonomous Cloud Brained Humanoid Robot for Search and Rescue missions in Hazardous Environments. Diploma Thesis, Department of Informatics and Telecommunications Engineering, University of Western Macedonia, Kozani, Greece, 2018.
11. A Man Diagnosed with Wuhan Coronavirus Near Seattle Is Being Treated Largely by a Robot. Available online: <https://cnn.it/348Sv1T> (accessed on 6 August 2020).
12. Evans, J.M. HelpMate: An autonomous mobile robot courier for hospitals. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, Munich, Germany, 12–16 September 1994; Volume 3, pp. 1695–1700. [CrossRef]
13. Nagatani, K.; Kiribayashi, S.; Okada, Y.; Tadokoro, S.; Nishimura, T.; Yoshida, T.; Koyanagi, E.; Hada, Y. Redesign of rescue mobile robot quince. In *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Kyoto, Japan, 1–5 November 2011.
14. Axelrod, B.; Huang, W.H. Autonomous door opening and traversal. In *Proceedings of the 2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, Woburn, MA, USA, 11–12 May 2015; pp. 1–6.
15. Overview of the Robot survey Runner. Available online: <https://bit.ly/2JdlFSz> (accessed on 6 August 2020).
16. Lytridis, C.; Bazinas, C.; Sidiropoulos, G.; Papakostas, G.A.; Kaburlasos, V.G.; Nikopoulou, V.A.; Holeva, V.; Evangelidou, A. Distance Special Education Delivery by Social Robots. *Electronics* **2020**, *9*, 1034. [CrossRef]
17. Bimbraw, K. Autonomous Cars: Past, Present and Future—A Review of the Developments in the Last Century, the Present Scenario and the Expected Future of Autonomous Vehicle Technology. In *Proceedings of the ICINCO 2015—12th International Conference on Informatics in Control, Automation and Robotics*, Colmar, France, 21–23 July 2015; Volume 1, pp. 191–198. [CrossRef]
18. Memon, Q.; Ahmed, M.; Ali, S.; Memon, A.R.; Shah, W. Self-driving and driver relaxing vehicle. In *Proceedings of the 2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, Rawalpindi, Pakistan, 1–2 November 2016; pp. 170–174.
19. Flämig, H. Autonomous Vehicles and Autonomous Driving in Freight Transport. In *Autonomous Driving*; Springer: Berlin/Heidelberg, Germany, 2016. [CrossRef]
20. Blaga, B.-C.-Z.; Deac, M.-A.; Al-doori, R.W.Y.; Negru, M.; Danescu, R. Miniature autonomous vehicle development on raspberry pi. In *Proceedings of the 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, 6–8 September 2018. [CrossRef]
21. ATLAS Robot. Available online: <https://www.bostondynamics.com/atlas> (accessed on 6 August 2020).
22. SPOT Robot. Available online: <https://www.bostondynamics.com/spot> (accessed on 6 August 2020).
23. Rfc 1918: Address Allocation for Private Internets. Available online: <https://bit.ly/2QKCl8a> (accessed on 6 August 2020).
24. Stallings, W. *Cryptography and Network Security: Principles and Practice*; Pearson Publications: London, UK, 2006.

25. Daemen, J.; Rijman, V. AES proposal: rijndael. *AES Proposal Doc.* **1999**, *9*, 1–45.
26. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; p. I-I.
27. Zhang, S.; Chi, C.; Lei, Z.; Li, S.Z. RefineFace: Refinement Neural Network for High Performance Face Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [[CrossRef](#)] [[PubMed](#)]
28. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [[CrossRef](#)]
29. Goddeau, D.; Meng, H.; Polifroni, J.; Seneff, S.; Busayapongchai, S. A form-based dialogue manager for spoken language applications. In Proceedings of the Fourth International Conference on Spoken Language Processing, ICSLP '96, Philadelphia, PA, USA, 3–6 October 1996; Volume 2, pp. 701–704.
30. Baras, N.; Nantzios, G.; Ziouzos, D.; Dasygenis, M. Autonomous Obstacle Avoidance Vehicle Using LIDAR and an Embedded System. In Proceedings of the 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCASST), Thessaloniki, Greece, 13–15 May 2019; pp. 1–4. [[CrossRef](#)]
31. Li, G.; Geng, E.; Ye, Z.; Xu, Y.; Lin, J.; Pang, Y. Indoor Positioning Algorithm Based on the Improved RSSI Distance Model. *Sensors* **2018**, *18*, 2820. [[CrossRef](#)] [[PubMed](#)]
32. Mahiddin, N.A. Indoor position detection using wifi and trilateration technique. In Proceedings of the International Conference on Informatics and Applications (ICIA2012), Kuala Terengganu, Malaysia, 3–5 June 2012.
33. Kampf, G.; Todt, D.; Pfaender, S.; Steinmann, E. Persistence of coronaviruses on inanimate surfaces and their inactivation with biocidal agents. *J. Hosp. Infect.* **2020**, *104*, 246–251. [[PubMed](#)]
34. Sime, J.D. Affiliate behaviour during escape to building exits. *J. Environ. Psychol.* **1983**, *3*, 21–41.
35. Aircnort: Wep Key Cracking Tool. Available online: <https://bit.ly/2vPUJVR> (accessed on 6 August 2020).
36. Man in the Middle. Available online: <https://bit.ly/2QKWjiW> (accessed on 6 August 2020).
37. Robinette, P.; Li, W.; Allen, R.; Howard, A.M.; Wagner, A.R. Overtrust of robots in emergency evacuation scenarios. In Proceedings of the 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Christchurch, New Zealand, 7–10 March 2016; pp. 101–108. [[CrossRef](#)]
38. Salem, M.; Eyssel, F.; Rohlfing, K.; Kopp, S.; Joubin, F. To err is human (-like): Effects of robot gesture on perceived anthropomorphism and likability. *Int. J. Soc. Robot.* **2013**, *5*, 313–323.
39. McGilvary, G.A.; Barker, A.; Atkinson, M. Ad Hoc Cloud Computing. In Proceedings of the 2015 IEEE 8th International Conference on Cloud Computing, New York, NY, USA, 27 June–2 July 2015; pp. 1063–1068. [[CrossRef](#)]