International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

The Fault Tolerant Smart Cradle

Mounira Bouzahzah[*1], Nourhane Herimi[2], Malak Kabouche[3]

Submitted:11/03/2024 **Revised**: 26/04/2024 **Accepted**: 03/05/2024

Abstract: Nowadays, artificial intelligence is present in all parts of our lives from work, to hospitals, to schools and even our home. Through this work, we will present a smart cradle for infant monitoring. This system uses AI algorithms and IOT tools to monitor the baby and detect its needs which in turn, would be a very important tool to help parents, babysitters and nurses in hospitals to monitor the baby and understand its needs using the cries that they make. In addition to that, our smart cradle is a robust system that can detect if one of the sensors used is unavailable or malfunctioning and proposes solutions to insure its service.

Our smart cradle incorporates various sensors, including temperature, humidity and audio sensors, along with a camera, to continuously monitor the infant's health and movements within the cradle environment. The microphone is the most important sensor in our system because it allows the detection of the baby cries and it sends the sound to our intelligent algorithm that analysis the sound and determines what the baby needs. The most important part in our smart cradle is the fault tolerance part that allows the overall system to work correctly and without breakdowns. As we know, sensors are electronic objects that can break down due to different causes such as power outages, so this module is responsible for handling any exceptions that may arise.

We use different techniques for fault tolerance mainly replication, exception handlers and learning algorithms to solve all detected faults. The fault tolerance part makes our smart cradle different from all proposed systems for baby monitoring.

Keywords: *Fault tolerance, Generative AI, Internet of Things, Monitoring system, Neural Networks, Qlearning algorithm.*

1. Introduction

Modern life needs a lot of facilities to help humans in accomplishing their goals. Within this paper, we will introduce a smart fault tolerant cradle that will help mothers, house keepers and nurses at hospitals to take care of babies. The main difficulty when someone is taking care of a baby is that they cannot fully understand the reason for which the baby is crying. In reality, there is no way for the baby to communicate with the external world. But recent studies in field of medicine have established a connection between the cries of the baby and its needs; this means that the baby's cries change in tone depending on its needs like hungry, uncomfortable or tired. Authors in [1] show the importance of accurately interpreting infant cries, which can help parents better understand and care for their babies.

To allow a kind of communication between the baby and its surrounding people, we used an intelligent neural network algorithm and generative artificial intelligence techniques to analyze the baby's cries and classify its needs into five different categories. We also used the Q learning algorithm to propose solutions for any problem detected regarding the functioning of the sensors.

The remainder of the paper is structured as follows: The relevant works that suggest smart cradles for baby monitoring are presented in Section II. Section III describes the general architecture of the fault tolerant smart cradle.

Section IV provides the details of the three parts that make up the monitoring system: The electronic part and the different sensors used, the baby cry sound analysis and part with the techniques used to determine the baby needs and some results given by the proposed algorithm. Finally, the fault tolerance part in which we explain the approach used and we give some result. Section V concludes the work, gives some comparative features within the existing works and describes our future works.

2. Related Works

The key motivation that pushes for the suggestion for a surveillance system for babies is the increasing number of working mothers in today's industrialized nations, which has made infant care a daily struggle for numerous families. Due to the high expense of living, working parents frequently don't have the time to give their infants the attention they need. Current alternatives, such as leaving infants with their grandparents or employing caregivers, nevertheless do not give parents the ability to constantly check on their infants' health in both regular and abnormal circumstances. The following can be reported from a number of works that offer assistance to parents:

1 Departement of mathmatics and computer science*
University Center Abdelhafid BOUSSOUF, Mila, Algeria.
ORCID ID: 0000-0002-4950-191X
2 Departement of mathmatics and computer science
University Center Abdelhafid BOUSSOUF, Mila, Algeria
ORCID ID: 0009-0005-8398-8521
3 Departement of mathmatics and computer science
university center Abdelhafid BOUSSOUF, Mila, Algeria
ORCID ID: 0009-0008-0634-8346
** Corresponding Author Email: m.bouzahzah@centre-univ-mila.dz*

An inventive Internet of Things-based baby monitoring system (IoT-BBMS) for a smart cradle is presented in Work [2]. This study is new because of the suggested IoT-BBMS automation system, which is made possible by three important contributions: first, the design and the fabrication of a smart baby cradle prototype with auto-swinging capability triggered by baby cries, along with an integrated web camera for vision monitoring and a musical toy for soothing babies. Second, the development of a new algorithm implemented on the NodeMCU microcontroller to carry out the necessary oversight and management duties based on sensor inputs and user commands. Third, the utilization of the NodeMCU Wi-Fi enabled microcontroller board and the Adafruit MQTT cloud server for retrieving data from various sensors and sending commands to control the cradle's actuators like the swing motor, fan and musical toy over the internet. The paper systematically follows a methodical design approach to propose, develop and successfully implement a first-of-its-kind IoT-enabled smart cradle solution. By leveraging modern technologies like Wi-Fi enabled microcontrollers, cloud platforms and sensor/actuator components, it introduces automated capabilities like baby cry detection, temperature control and remote accessibility over the internet, and integration of soothing mechanisms like auto-swinging and musical toy activation.

The smart cradle system overcomes major limitations of conventional baby monitoring systems, while offering enhanced convenience, safety, efficiency and peace of mind for parents. But it still does not deal with the cause of baby cries; it only tries to calm them down using music.

The authors in [3] propose a system incorporates various sensors, including temperature, humidity and ultrasonic sensors, along with a camera, to continuously monitor the baby's condition and movements within the cradle environment. This system highlights its ability to monitor the baby's health, specifically focusing on temperature changes. The temperature sensor integrated into the cradle continuously senses and monitors any abnormal fluctuations in the infant's body temperature. If any significant deviations from normal temperature ranges are detected, the system promptly alerts the parents through a notification. This feature ensures close monitoring of the baby's temperature, which is crucial for maintaining their health and comfort.

The main issue with this work is that it relies on one essential sensor to detect baby temperature but if this sensor breaks down or gives erroneous results, the whole system fails.

In [4], an additional smart cradle is suggested. This system uses machine learning to enable remote baby monitoring through the use of an Android app and an innovative Internet of Things (IoT) based cradle system. The core functionality of the proposed system revolves around one key feature. The smart cradle incorporates an automated swinging mechanism for the cradle itself, triggered by the detection of a baby's cry through a noise sensor equipped with a microphone. When the baby starts crying, the microphone picks up the sound and converts it into an electrical signal, prompting the cradle to initiate a swinging motion in an attempt to soothe the baby. However, if the baby's cries persist for an extended duration despite the swinging, the system escalates the situation by activating an alarm buzzer and sending an alert notification to the parent's smartphone. This notification essentially conveys that the cradle's swinging motion alone is insufficient to calm the baby and human intervention is required to address the baby's needs.

This work tries to treat the baby’s cries through the automated swinging mechanism but if it fails, it needs human intervention to calm the baby.

After the study of all these monitoring systems, we decided to propose a fault tolerant smart cradle based on two main ideas, which are:

First, the use of smart algorithm to analyze the baby’s cries and to determine what the baby needs; this function is realized using a set of sensors that collect data and a machine learning algorithm.

Second, the use of a fault tolerance approach to make sure that the system will not fail. The fault tolerant approach is based on replication and exception handlings.

3. The global architecture of the proposed system

The objective of this system is to make use of the technology of connected objects and artificial intelligence to develop a system allowing parents to remotely monitor their children without system failures. The fault tolerant smart cradle is composed of the following parts:

• A baby cry detection mechanism that identifies the baby's needs at a given moment using an intelligent classification algorithm, we name it the voice analyses part.

• Another mechanism that allows real-time monitoring of the child and their environment using several sensors capable of measuring and monitoring the humidity and the temperature of the mattress in order to support the decisions made by the classification algorithm, we called it the electronic part.

• A third mechanism, based on replication and a learning algorithm, guarantees the system's fault tolerance.

• Finally, the user interface (web application) that facilitates communication between the system and the user (parents, nannies, nurses, etc.).

Fig.1. shows the architecture of our system.

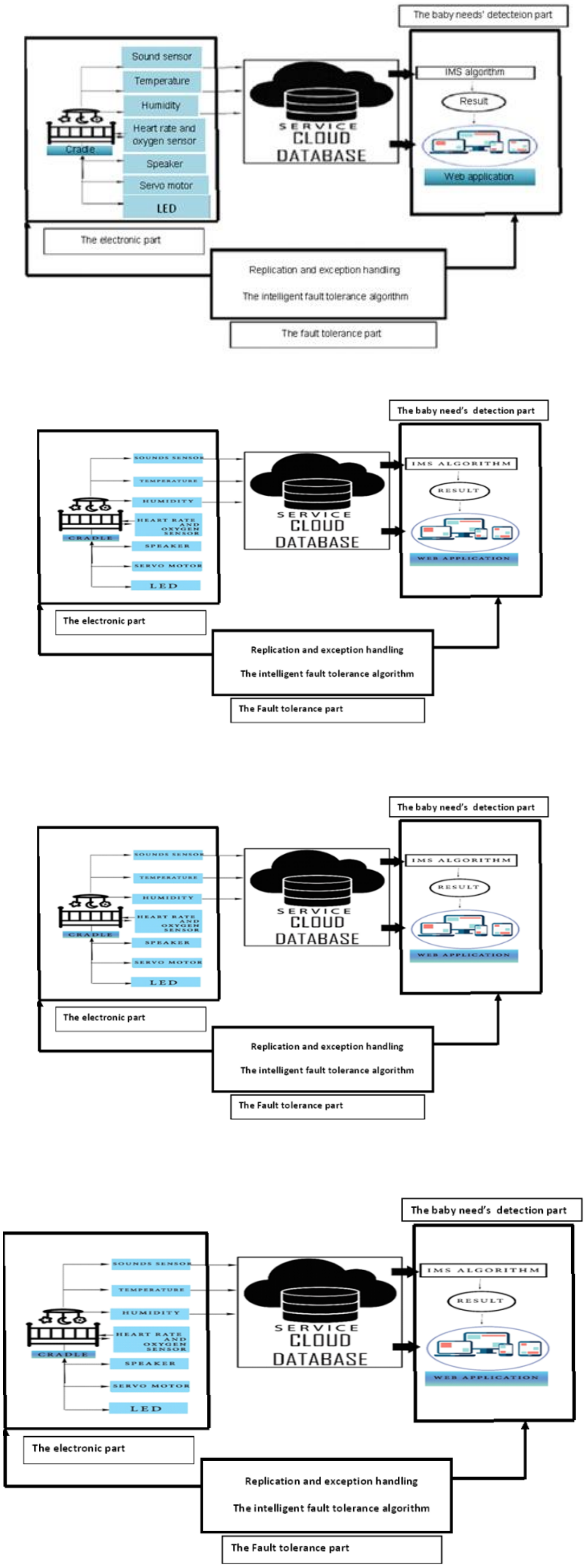

Fig. 1. The fault tolerant smart cradle general architecture

4. The detailed description of the fault tolerant smart cradle

From the increasing challenges faced by working parents in maintaining a continuous watchful eye on their babies' behavior and needs, especially after grueling work hours. This smart cradle system is proposed to bridge this gap and assist parents in ensuring their babies' well-being even when they are not physically present. As we mentioned in the section before, the proposed smart cradle has four main parts that we will give their detailed description through this section.

4.1. The electronic part

This part represents the set of material used to build our smart cradle. In addition to the wooden cradle, which is a simple baby bed, we have the electronic part that is formed by group of sensors such the audio sensor, the speaker, the temperature-humidity sensor and the camera.

The audio sensor or the microphone and the camera are the most important sensors in this system. The first sensor (microphone) is used to listen to the infant's crying and to send the signal to be analyzed by the intelligent algorithm in order to specify the baby's need. The camera helps the parents to monitor their babies at any time. The temperature and humidity sensors are used mainly to confirm the baby's needs detected by the intelligent algorithm. For example if the algorithm decides that the baby wants their diaper the changed, this need will be confirmed by the measurement made by the humidity sensor. We use, also, a speaker to play some music in order to calm the baby until the parents arrive. Figure2 shows some sensors used in our project.

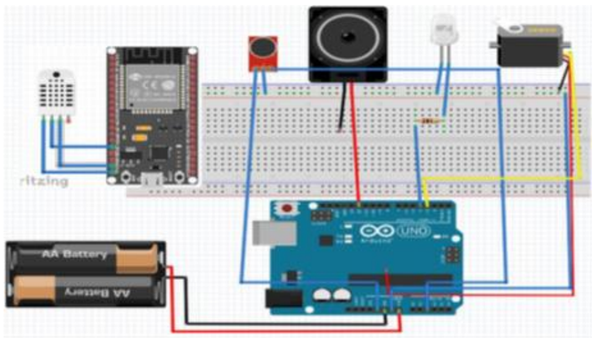

Fig. 2. Component circuit diagram

4.2. The baby's needs detection part

Perhaps the most common problem faced by parents when dealing within a new born baby is the inability to understand the baby's needs or know why the baby is crying. Within this work, we try to build a communication channel between the parents and their babies using the proposed voice analyses algorithm.

Given how little language they can use, one of an infant's main forms of communication is crying. However, failing to properly understand the reasons behind an infant's cries can lead to anxiety, frustration and even the possibility of abusive actions by caregivers who are desperately trying different methods to calm and soothe the infant.

Having an automatic system, which analyzes the baby's cry and translates it, can really help parents and caregivers to quickly calm the baby.

In reality, there are many works that deal with this subject; we notice here the research of Yesy Diah Rosita [5]. The article proposes a system for automatically classifying the sound of an infant's cry into one of three categories: hunger, discomfort or fatigue, using a neural network.

The authors of [1] present a thorough survey of current studies on cry signal processing and categorization tasks for infants. It provides a thorough rundown of all the phases that are involved in this field, starting with data collection and ending with pre-processing, feature extraction, feature selection and machine learning classification models.

This study helps us to decide concerning the algorithm and the architecture, we choose to classify the baby's cries.

4.2.1. The Neural Networks Model

Though this section we will describe and give the details of the part responsible for the automatic analysis of the audio. We use the neural networks solutions described in [6], [7] and [8] here.

A. Multi-Layer Perceptron Model

A supervised learning algorithm called MLP trains on a dataset to learn a function $f: R^m \rightarrow R^o$, where m is the number of input dimensions and o is the number of output dimensions. It can learn a non-linear function approximator for either regression or classification given a set of features $X = x_1, x_2, \ldots, x_m$ and a target y. In contrast to logistic regression, it is possible to add one or more hidden layers - non-linear layers - between the input and output layers. Artificial neurons arranged in a sequential fashion are layered into multiple layers to form MLPs. Because every neuron in one layer is linked to every other neuron in the layer above it, information flows from the input layer to the output layer unidirectional [6].

In this instance, audio data is used to train the model for classification. It is made up of multiple dense layers that are sequentially ordered and totally connected. The objective of the model is to predict the correct class label given an input feature vector that was collected from audio data. Figure 3 displays the architecture of the model.

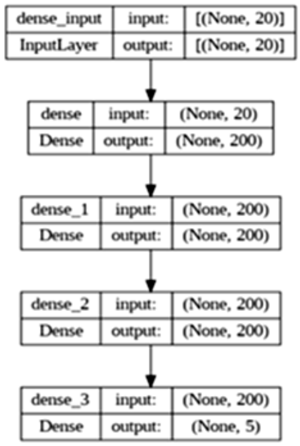

Fig. 3. The model architecture

The first layer of the model is the input layer, which is responsible for accepting the feature vector. The size of this layer is determined by the number of features in the input data.

Three hidden layers, each with 200 units, are present after the input layer. The rectified linear unit (ReLU) activation function is used by these hidden layers. ReLU gives the model nonlinearity, which makes it possible for it to identify intricate patterns in the audio data.

The output layer is the last layer in the model and its number of units is the same as the number of classes in the target variable. The softmax activation function is employed in this layer. Indicating the possibility that the entry belongs to each class, it generates a probability distribution between the different classes.

In order to understand the relationship between the input characteristics and the class labels, the model modifies the weights and biases of the dense layers during the training phase. In the output layer, the probability predictions for every class are produced using the softmax activation function.

In summary, this deep learning model is specifically designed for audio classification tasks. By training on labeled audio data, it learns to associate input audio features with their corresponding class labels and can then make predictions on new, unseen audio samples.

B. Dataset Description

The dataset is used to analyze cry patterns of various cry reason classes and train a model to predict cry reasons for unseen data. Through this work, we have defined five cry classes which are:

1. Belly pain
2. Burping.
3. Discomfort.
4. Hunger.
5. Tiredness.

The dataset represents a great challenge in our work, because a complete open access dataset doesn't really exist. So we had to use an incomplete one because we had limited time. So as it is shown in figure 5 we have imbalanced data.

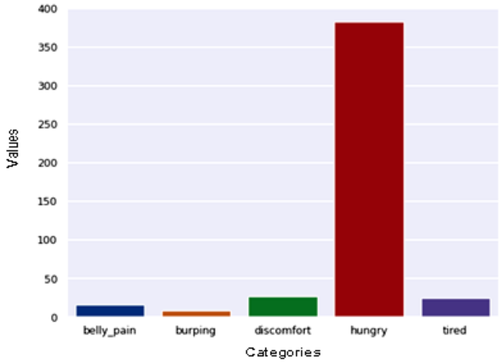

Fig. 4. The distribution of the dataset

The following plot illustrates the over fitting in the presence of imbalanced data:

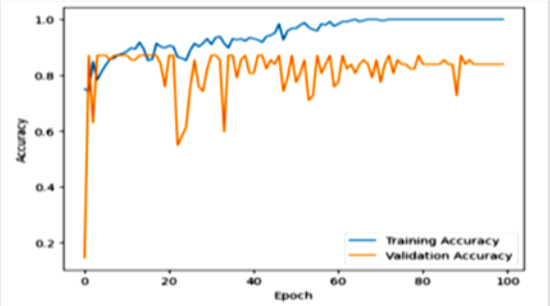

Fig 5. training and validation accuracy

In the plot above, the training accuracy continues to improve over time, as the model becomes increasingly adept at fitting the majority class. However, the validation accuracy initially increases, but then starts to decline. This divergence between training and validation accuracy indicates over-fitting, where the model fails to generalize well to unseen data. It may have memorized the majority class patterns, but struggles to correctly classify the minority class or perform well on new examples. As a solution to this problem we use Data augmentation techniques.

C. Data augmentation

A class of techniques called "data augmentation" uses an existing dataset to generate fake data. Little differences in the original data that shouldn't have an impact on the model's predictions are frequently present in this created data. Additionally, synthetic data can be used to depict combinations of distant examples that would be challenging to infer otherwise [9].

Data augmentation is a method that uses pre-existing data to create modified copies of a dataset, thereby artificially expanding the training set. It entails introducing small adjustments to the dataset or producing new data points via deep learning. It is among the best interfaces for affecting how deep neural networks are trained.

Ramyachitra et al [10] present the Synthetic Minority Over-sampling Technique (SMOTE), a cutting-edge solution to the problem of an uneven class distribution. For the minority class, SMOTE creates synthetic examples rather than just copying samples that already exist. The process entails improving every single minority class sample and generating synthetic instances along the line segments that link the minority class sample to any or all of its k nearest neighbors [10].

We use the SMOTE augmentation technique to create fake samples for the minority class. In addition to enabling the model to learn from a more representative dataset, this balances the distribution of classes.

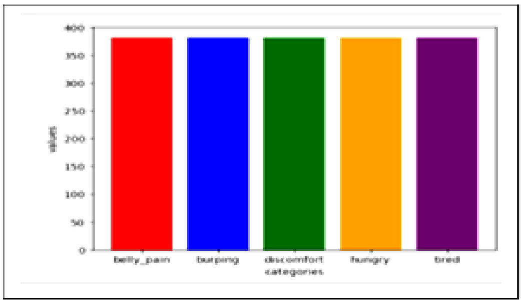

Fig. 6. Distribution of the dataset after augmentation

After applying the SMOTE data augmentation technique to balance the data, significant improvements have been observed in the training and validation accuracy. The following figure shows these improvements.

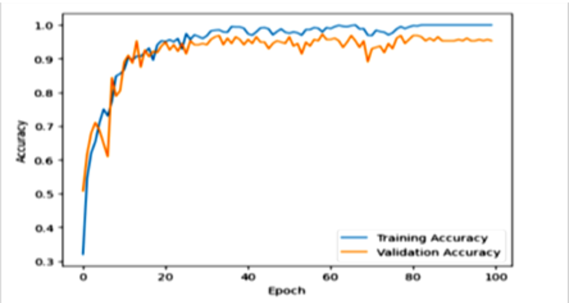

Fig. 7. training and validation accuracy

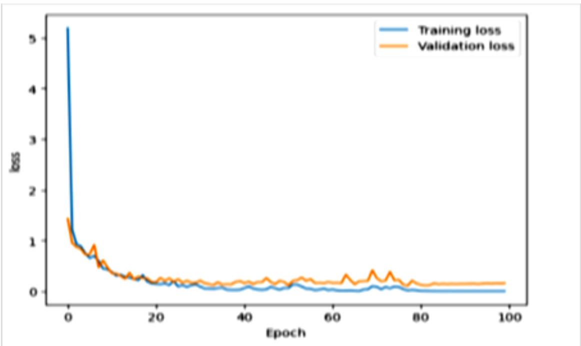

Fig. 8. training and validation loss.

D. The results

The confusion matrix, displayed in Figure 9, is a table that summarizes the model's predictions against the real labels or ground truth to illustrate how well a classification model performs.

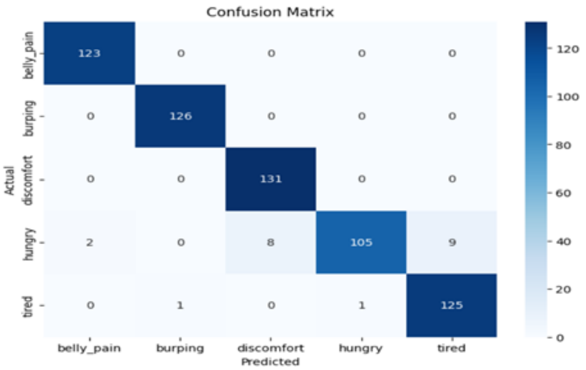

Fig. 9. Confusion matrix for the proposed NN model with augmentation

Achieving an accuracy of 0.97 on the dataset, these results indicate that our model is performing at a high level and making accurate predictions.

The macro average and weighted average accuracy scores of 0.97 suggest that our model demonstrates consistent performance across all classes, regardless of their support or sample size. This is a positive outcome as it indicates that our model is not biased towards particular class and is effectively capturing patterns and characteristics from all classes.

Table. 1. Classification report after using SMOTE technique

	precision	recall	f1-score	support
belly_pain	0.98	1.00	0.99	123
burping	0.99	1.00	1.00	126
discomfort	0.94	1.00	0.97	131
hungry	0.99	0.85	0.91	124
tired	0.93	0.98	0.96	127

To obtain a thorough knowledge of our model's performance, it is crucial to take into account evaluation measures like precision, recall and F1 score in addition to high accuracy scores. These measures shed light on how well our model can predict both positive and negative outcomes, and they are especially helpful when working with unbalanced datasets.

E. The web application

Within this work we propose the development of a web application that can be installed on a terminal to be used by the parents or other persons such as caregivers and nurses. The web application is used to display information concerning the baby's health such as the temperature, heart beat and the oxygen percentage; it also gives notifications about the baby's needs to their parents.

Figure 10 shows the UML use cases diagram which presents the main functionalities of the system

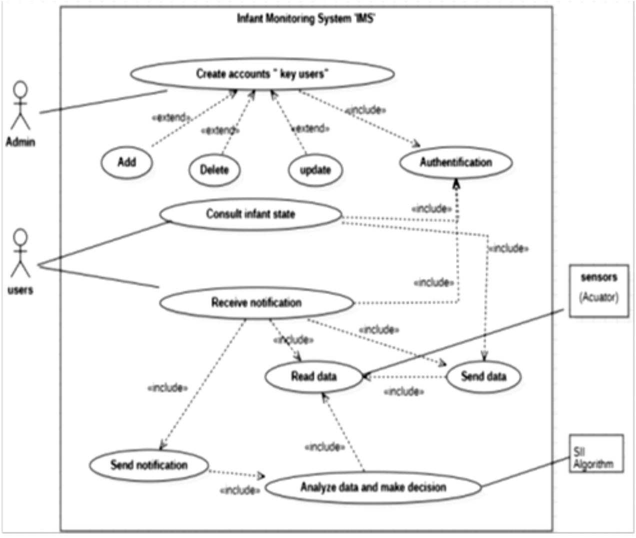

Fig. 10. Use case diagram

The interface of the web application is presented in fig. 11.

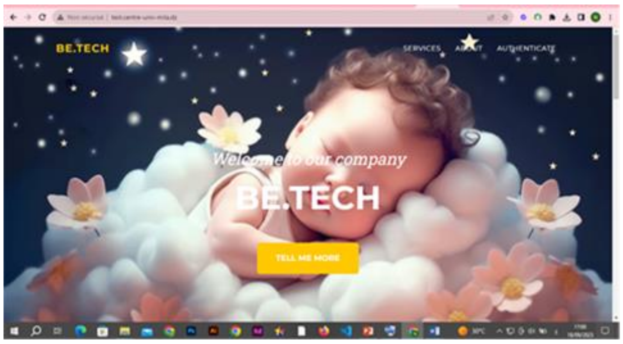

Fig. 11. The web application interface

4.3. the Fault tolerance part

Implementing fault tolerance IoT systems poses several challenges. The small size and high usage of devices of the IoT network often lead to frequent failures, which can have catastrophic consequences mainly when dealing with critical systems.

There are many works that deals within fault tolerance IOT systems such as the paper [11] that presents a novel Internet of Things (IoT) based architecture for remote patient health monitoring that focuses on critical features like scalability, fault tolerance, low power consumption and support for various healthcare services. The core communication infrastructure is built on 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks), which provides energy efficient and reliable data transmission for low-power IoT devices.

So, the paper presents a full IoT system architecture for reliable and energy-efficient remote health monitoring using 6LoWPAN, with novel mechanisms for fault tolerance and scalability. But this mechanism is not used with all types of systems.

The goal of the paper [12] is to improve the monitoring and the control reliability by addressing the issue of predictive maintenance of IoT networks in continuous-time. It presents a powerful predictive hybrid control system based on two major innovations: it models long-term state changes using continuous-time Markov chain analysis and it leverages ideas from cooperative game theory for dispersed nodes.

In detail, the core challenge is that any malfunctioning IoT sensor node that collects incorrect temperature data can negatively impact the monitoring and the control efficiencies in smart IoT infrastructure. But within this work, the authors deal only with temperature sensors' problems and do not provide any solutions for other types of sensors.

Having a fault tolerant Smart cradle is our goal and to achieve it, we decide to consider mainly two possible issues:

- The sensor does not output any response.
- The sensor outputs faulty results.

In order to treat these two types of issues we propose the use of two fault techniques inspired from those proposed for distributed systems which are: replication and exception handling.

4.4.1 The replication technique

This technique allows the duplication of systems components in order to provide redundancy and resilience against system failure. Material replication refers to the use of multiple devices to fulfill the same function.

We prefer to use this technique to treat the first type of faults, which means, the case where the sensor does not response. Generally, the material replication is a good solution when dealing with systems that uses fewer numbers of devices. In the case of the smart cradle, we propose the use of two sensors that perform the same tasks. The first sensor is considered as the active sensor, while the second is known as the passive replica. When the active sensor works properly the system always takes its response into consideration. But if this latter fails to transmit information, the system uses the information sent by the second sensor to solve the problem and sends a notification to the user in order to repair the active sensor. Thus, the system is still able to function properly despite the failure of the active sensor. Concerning this work, we decide, also, that only two sensors must be replicated which are the camera and the sound sensor since they are the most important ones. We do not replicate all the sensors to avoid raising the cradle's cost. The detection of the sensor failure is concerned as an exception and it is treated using a specific handler.

4.4.2 The exception handling technique

Exception handling is a fault tolerance technique that allows systems to respond appropriately to unusual hardware or software errors. Exception handling is typically used to identify, diagnose and resolve software and system errors. As IoT (Internet of Things) networks become increasingly prevalent, it is important that we propose a technique able to detect and to handle errors in order to grantee fault tolerance in such systems. In the case of the smart cradle, we used exception handling techniques to solve the second type of issues taken in consideration in this work; this means that we use this technique to solve the case when the sensor gives faulty results

To achieve our goal and avoid the system failure, we decide to use solutions from the field of the reinforcement learning and memory based learning, we will propose an intelligent learning algorithm for exception handling.

4.4.3 The intelligent fault tolerance algorithm

Our algorithm is divided into three main parts:

- The decision part.
- The reinforcement learning part using the Qlearning algorithm.
- The memory based learning using a tree.

The organigram of this method is shown in the following figure.

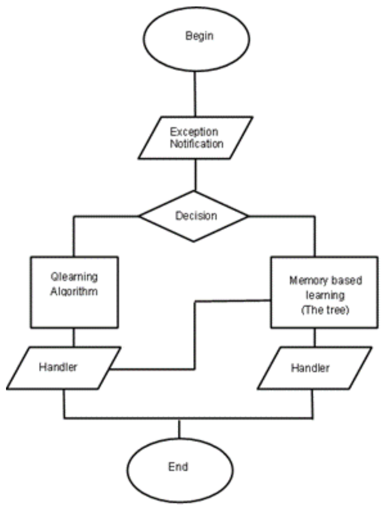

Fig. 12. The intelligent fault tolerance algorithm

As the graphic illustrates, the intelligent algorithm's primary function is to resolve system exceptions that arise when a sensor transmits inaccurate data. The cradle must validate all of the data transmitted by the sensors. To handle exceptions, this method employs two distinct approaches: one involves modifying the Qlearning Algorithm to handle exceptions, which has not been encountered previously. And the other involves organizing its prior outcomes into a tree structure to address exceptions. This tree has been encountered previously.

2.3 The fault tolerance algorithm and Exception handling

The fault tolerance method consists of three main sections, as demonstrated in the previous section: the decision portion, the reinforcement learning part (using an updated version of the Q learning algorithm) and the experiences part. We will go over each of the aforementioned sections in this section.

2.3.1 The decision step:

The fault tolerance algorithm receives the detected exception and processes it. This algorithm must first determine whether the exception is new or has already been handled. The algorithm must look through the experiences tree for the observed exception in order to complete this phase. The agent chooses to utilize the Qlearning algorithm if the research function provides a false result; if not, the fault tolerance algorithm applies its prior knowledge to resolve the exception. A thorough explanation of these two options can be found in the sections that follow.

2.3.2 The Q learning algorithm and exception handling

Reinforcement learning is a learning methodology that enables the system to learn by making mistakes and selecting the right course of action when a unique situation is identified. The Q learning algorithm, which selects the optimal course of action to take in a given state based on a value known as the reward, is one of the most popular algorithms for reinforcement learning.

The Q Learning algorithm has been in use since 1989 [13]. According to the author in [14], the Q Learning algorithm selects the optimal action A to be carried out in a specific state S based on a value known as the reward $\vec{Q}$ such that Q: S A and Q(S, A) is the outcome reward when the action a is carried out for the state s.

Concerning the smart cradle, we try to control the values given by the sensors to avoid any faulty notification that comes from sensors functioning problems. For example, the baby's temperature must be less than 43°, if the sensor gives a value equal to 60° the system will consider it as an exception and must use a handler to solve the problem.

The intelligent algorithm must associate with each pair (exception, handler), a calculated value called the reward of the handler chosen to solve the exception detected. The information: exception, handler and reward in a table called the value table.

The expected reward or future gain is represented by the value Q (e, h), when the fault tolerance algorithm employs handler h to resolve exception e. The formula below is used to determine this future reward:

$$Q(e,h) = Q(e,h) + \propto (r + \gamma \max[Q(e',h) - Q(e,h)])$$

- The learning rate parameter, denoted as α, lies within the interval [0, 1].
- The value of r, which represents the immediate reward, is determined by the fault tolerance mechanism following the use of the handler to address the detected exception.
- The handler h is employed and capable of resolving the issue.
- The state that results from using handler h is represented by e' and e is the exception detected state.
- The actualization factor is represented by the constant γ. It makes it possible for the clever algorithm to choose the optimal reward. The γ value falls inside the interval [0, 1]; hence, it must be close to 0 in order for the intelligent fault tolerance algorithm to seek an instant reward. γ is a constant representing the actualization factor. It allows the intelligent algorithm to determine the best reward. The γ value is including in the interval [0, 1]; if the intelligent fault tolerance algorithm wants an immediate reward γ must be near 0.
- In actuality, we adjust Q learning to meet the requirements of our application. It is assumed that the handlers' addresses need to be arranged in a table, with the initial values Q_0 (e, h) being expected to be 0. There are a limited number of exceptions that the system can identify.

The Boltzmann distribution is typically used by the Qlearning method to select an action (a handler for our application) [15]. However, we suggest that a table with the addresses of the handlers be created and that access to the handlers be granted progressively in order to make the approach easier to use.

We suggest that the values of the immediate reward r, be placed in the range [0, 1]. If r = max, r, it indicates that the handler has solved the exception; if r = 0, it has failed to complete the task. We suggest connecting the instant reward to the outcome of using the same sensor again. This implies that r must be close to 1 if the same sensor is used again and yields an acceptable result. However, r will have a reward close to 0 if the value is still incorrect.

Finally, we can say that when the system detects that one of the sensors is sending erroneous or illogical information, it sends a notification to the intelligent fault-tolerance algorithm to resolve the current exception for the first time, the algorithm must choose the most suitable handler from among those in the table. But, unfortunately, the Q-Learning algorithm cannot capture the regularities of the environment, this fact implies that the algorithm cannot detect cases of similar exceptions and also it cannot avoid exponential explosion. To deal with this problem, the intelligent fault tolerance algorithm has to use its historical results. The best solution in this case is the use of a memory inspired by those used in memory-based learning.

2.3.3 The memory based learning and exception handling

Since at least 1910, memory-based learning [16] has been a popular approach to education. The idea behind this approach is to store past information or experiences in a memory and use that memory to look up similar occurrences in the past while making decisions in the future.

In regards to the smart cradle, we employ a tree derived from the McCallum memory [17] in order to prevent exponential growth and identify the unclear states of exceptions. The root alone was used to make this tree initially. In order to be used if a similar exception is detected in the future, the intelligent fault tolerance algorithm must insert the exception detected (e), the handler (h) that was selected and the table containing the Q (e, h) values in the tree, when it uses the Q Learning Algorithm part to solve an exception detected using one of the handlers already present in the table.

As a result, when our intelligent algorithm encounters a new pair (exception, handler), it first learns their relationship and stores the case's information in the tree. In the event, an exception signal is received later; the intelligent fault tolerance algorithm will select to solve the exception using the Q Learning Algorithm if the search function returns a negative value. If, however, the search is positive and the exception is found in the tree, it indicates that the exception has already been handled. When this exception arises for the first time, the learning agent must employ the same handler that was chosen for it. The issue is resolved if the handler successfully handles this exception, which indicates that the immediate reward value, $r = 1$. If not, the suggested algorithm determines that this exception is unclear and that there may be hidden states that distinguish this exception from those that have been handled previously. The Q Learning Algorithm must be applied to this exception by the intelligent fault tolerance algorithm; however, in this instance, the table containing the Q (e, h) values will be initialized using the same table saved with the exception e. The handlers are selected based on the base of the Q (e, h) value; the handler with the highest value will be selected initially. The suggested approach adds a new node with a new exception and handler when the exception is resolved, updating the tree. Additionally, the table for the Q function needs to be added. If an exception is considered "new," it has not yet been handled by the intelligent fault tolerance algorithm and has an instantaneous reward of 0 when it is handled by all of the handlers that are available. We suggest notifying the user via notification so that he can report the malfunctioning sensor.

In order to test our intelligent fault tolerance algorithm, we send random temperature values to the monitoring system and when exception are detected the proposed algorithm uses the previous described steps to solve the exceptions.

The handlers in this algorithm's knowledge base are responsible for resolving anticipated exceptions. In order for the handlers' addresses to be used in the event that exceptions are found, they are stored in table H.

Table2. Handlers' addresses

@h1	@h2	@h3	…	……					@hn

Using a random function, the intelligent fault tolerance method chooses an exception from the exception set. The maximum value of the immediate reward is represented by the constant variable r_{max}. We are interested in the measurement of time during this test. In order to address the system's provided exceptions, we computed the time token using the suggested algorithm. To calculate the response time, we need to retrieve the "date" system variable in milliseconds. The graph below illustrates how exception solving times decrease when an exception is recognized and treated similarly to another exception. This implies that the intelligent fault tolerance algorithm will learn the environment policy and disregard the time token for exception handling after handling a large number of exceptions.

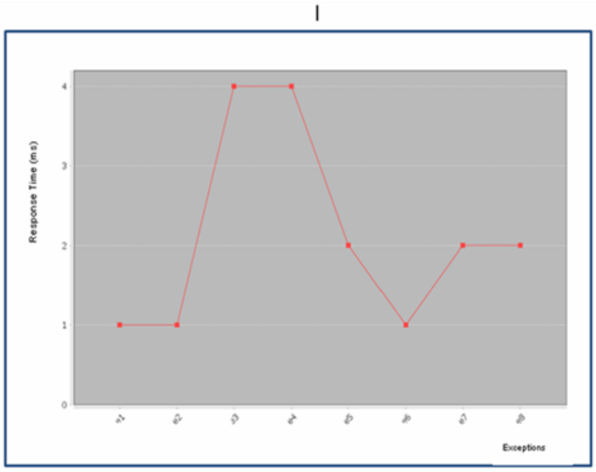

Fig. 13. The success rate graph

5. Conclusion and future works

Through this paper we describe a project called "the smart fault tolerant cradle". In reality, building a smart cradle is not a new idea, there are lots of projects that deal within the same subject because that type of tool helps parents to better care and understand their babies.

Our project differs from all similar ones because it is based on baby cry sound analysis to determine the needs of the baby. There exist many researches that deal with the subject of baby cry sound analysis and classification but it is the first time that such work is associated within a smart cradle. Our goal, at this level, is to give our project more importance and to create a communication channel between the parents and their babies. The results obtained are very important mainly because we have built a data set using generative artificial intelligence and the used algorithm can detect easily the baby's need.

The most powerful part of the proposed smart cradle is the fault tolerance one, because for the first time the fault tolerance techniques used for distributed systems are used to insure the functionalities of our cradle. So, the fault tolerance part uses two main techniques which are: replication and exception handling.

We introduce concepts from reinforcement learning and memory based learning to propose an intelligent fault tolerance algorithm that treats the faults that occurs within the smart cradle. To test this algorithm, we create simulation cases of faults and we notice that the algorithm sometimes take time to solve a problem for the first time but if the fault is repeated or a similar case the algorithm gives the handler in a neglected time.

Concerning our future works, we will try to choose other artificial intelligent techniques to tolerate failures and also to benefit from the crowdsourcing solutions to search for more exception's handlers instead of giving notification to the user when the algorithm fails to solve the problem, because the main aim of this work is to restrict the human interfere.

References

[1] Ji, C., Mudiyanselage, T. B., Gao, Y., & Pan, Y. (2021). A review of infant cry analysis and classification. *EURASIP Journal on Audio, Speech, and Music Processing*, *2021*(1), 8.

[2] Jabbar, W. A., Shang, H. K., Hamid, S. N., Almohammedi, A. A., Ramli, R. M., & Ali, M. A. (2019). IoT-BBMS: Internet of Things-based baby monitoring system for smart cradle. *IEEE Access*, *7*, 93791-93805.

[3] Kannan, P., Devaraj, A., Rajan, B. P. T., Swathira, P. K., Jayaranim, S., & Shebna, V. (2021). IoT Based Smart Cradle Using PI. In *Recent Trends in Intensive Computing* (pp. 716-720). IOS Press.

[4] Kumar, V. S., Pullagura, L., Kumari, N. V., Pooja Nayak, S., Devi, B. P., Alharbi, A., & Asakipaam, S. A. (2022). Internet of Things-Based Patient Cradle System with an Android App for Baby Monitoring with Machine Learning. *Wireless Communications and Mobile Computing*, *2022*.

[5] Rosita, Y. D., & Junaedi, H. (2016, October). Infant's cry sound classification using Mel-Frequency Cepstrum Coefficients feature extraction and Backpropagation Neural Network. In *2016 2nd International Conference on Science and Technology-Computer (ICST)* (pp. 160-166). IEEE.

[6] Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923*.

[7] Zinemanas, P., Rocamora, M., Miron, M., Font, F., & Serra, X. (2021). An interpretable deep learning model for automatic sound classification. *Electronics*, *10*(7), 850.

[8] Gokhale, P., Bhat, O., & Bhat, S. (2018). Introduction to IOT. *International Advanced Research Journal in Science, Engineering and Technology*, *5*(1), 41-44.

[9] Khandpur, R. S. (2017). *Telemedicine technology and applications (mHealth, TeleHealth and eHealth)*. PHI Learning Pvt. Ltd..

[10] Ramyachitra, D., & Manikandan, P. (2014). Imbalanced dataset classification and solutions: a review. *International Journal of Computing and Business Research (IJCBR)*, *5*(4), 1-29.

[11] Gia, T. N., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. (2015, April). Fault tolerant and scalable IoT-based architecture for health monitoring. In *2015 IEEE Sensors Applications Symposium (SAS)* (pp. 1-6). IEEE.

[12] Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. M. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. *Energies*, *11*(12), 3430.

[13] Watkins, C. J. C. H. (1989). Learning from delayed rewards.

[14] Sutton, R. S., & Barto, A. G. (1999). Reinforcement learning: An introduction. *Robotica*, *17*(2), 229-235.

[15] Tesauro, G., & Kephart, J. O. (2002). Pricing in agent economies using multi-agent Q-learning. *Autonomous agents and multi-agent systems*, *5*, 289-304.

[16] Lin, J. H., & Vitter, J. S. (1992, July). A theory for memory-based learning. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 103-115).

[17] McCallum, A. K. (1996). *Reinforcement learning with selective perception and hidden state*. University of Rochester.