Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

В. А. Вишняков

СПЕЦИАЛИЗИРОВАННЫЕ IoT-СЕТИ: МОДЕЛИ, СТРУКТУРЫ, АЛГОРИТМЫ, ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА

SPECIALISED IoT SYSTEMS: MODELS, STRUCTURES, ALGORITHMS, HARDWARE, SOFTWARE TOOLS

Минск БГУИР 2023

УДК 004.738:33
ББК 32.971.353+65.05
В55

Рекомендовано к изданию Советом учреждения образования
«Белорусский государственный университет
информатики и радиоэлектроники»
(протокол №3 от 28.10.2022)

Р е ц е н з е н т ы:

заведующий кафедрой учреждения образования
«Белорусская государственная академия связи»
доктор технических наук, профессор В. И. Курмашев;

почетный профессор учреждения образования
«Белорусский государственный университет
информатики и радиоэлектроники»
доктор технических наук, профессор Л. М. Лыньков

Вишняков, В. А.

В55 Специализированные IoT-сети: модели, структуры, алгоритмы, программно-аппаратные средства = Specialized IoT systems: Models, Structures, Algorithms, Hardware, Software Tools / В. А. Вишняков. – Минск : БГУИР, 2023. – 184 с. : ил.

ISBN 978-985-543-689-9.

Монография включает анализ проблем, модели, алгоритмы и программно-аппаратные средства специализированных сетей интернета вещей.

Рассмотрены результаты проектирования и моделирования сети интернета вещей, мониторинга качества продукции, анализа звуковой информации окружающей среды, а также технология выявления заболеваний легких на базе нейронных сетей.

Монография предназначена для специалистов в области инфокоммуникаций, может быть полезна студентам соответствующих специальностей, слушателям факультетов повышения квалификации, магистрантам и аспирантам.

УДК 004.738:33
ББК 32.971.353+65.05

ISBN 978-985-543-689-9

CONTENT

ABBREVIATION

ABS – Algorithm Based on Scenarios
ANN – Artificial Neural Networks
API – Application Program Interface
AWS – Amazon Web Services
BI – Business Intelligence
CLI – Command Line Interface
CI – Computational Intelligence
CoAP – Limited Application Protocol
CPSI – Collecting and Processing Sound Information
CC – Cloud Computing
DAS – Data Acquisition System
DDS – Data Distribution Service
DLP – Data Loss Prevention
DNS – Domain Name Server
DPRA – Delayed Power Increase Algorithm
DTW – Dynamic Time Warping
EA – Evolutionary Algorithms
eMTC – enhanced Machine-type Communication
FC – Fog computing
GA – Genetic Algorithms
GMM – Gaussian Mixture Models
GSM – Global System for Mobile Communications
HAM – Hierarchy Analysis Method
HDMI – high definition media interface
HMM – Hidden Markov Models
HOLA – Heuristic Channel Selection Algorithm
HTTPS – HTTP Secret
IaaS – Infrastructure as a Service
IoE – Internet of Everything
IoT – Internet of Things
IP – Internet Protocol
IPSO – Improved PSO
IETF – Internet Engineering Task Force
ISO – International Organization for Standardization
KNN – KNearest Neighbors
LPWAN – Low-power WAN
LTE – Long Term Evolution
MA – Memetic Algorithms
MAIOT – Multiagent IoT
MAS – Multiagent System
MASSI – Multiagent System of Sound Information
MLP – Multi-Layer Perceptron

MQTT – Message Queue Transport Telemetry
M2M – Machine-to-Machine Communication
NB-IoT – Narrowband Internet of Things
NCF – Near Field Communication
NGN – Next-generation Networks
OMG – Object Management Group
PaaS – Platform as a Service
PDF – Portable Document Format
RACH – Random Access Channel
RFID – Radio Frequency Identification
PSO – Particle Swarm Optimization
RTP – Real Time Protocol
QPSO – Quantum Particle Swarm Optimization
SaaS – Software as a Service
SMS – Smart Management System
SNMP – Simple Network Management Protocol
SQL – Structured Query Language
SSO – Single Sign-On
SVM – Support Vector Machine
SWIFT – Society for World-Wide Interbank Financial Telecommunications
TCP – Transmission Control Protocol
TKIP – Temporal Key Integrity Protocol
TLS – Transport Layer Security
TPM – Trusted Platform Module
UC – Ubiquitous computerization
USB – Universal Bus
URL – Universal Resource Locator
VCCA – Variable Category Clustering Algorithm
VoIP – voice over IP
VRML – Virtual Reality Markup Language
VBAS – Video business process as a Service
VSaaS – Video surveillance as a Service
WAN – Wide Area Network
WLAN – Wireless Local Area Network
WOT – Web of Things
WPA – Wi-Fi Protected Access
WSN – Wireless Sensor Network

Introduction

For the first time, the term «Internet of Things» (IoT) was used in 1999 by British engineer Kevin Ashton, who led the pioneering work on radio frequency identification networks. He proposed the presentation to the management of Procter & Gamble, which was devoted to how the mass introduction of radio frequency tags can change some markets, such as retail and logistics.

The IoT is the concept of a data transmission network between physical objects, called «things», which are equipped with built-in technologies and means designed to interact with each other or with the external environment in real time. Things are able to work in automatic mode, the administrator can manage them, including remotely. It is assumed that the organization of these networks is capable of reconstructing economic and social processes, as well as eliminating the need for human participation from part of operations and actions [1].

At the end of the 2000s, the IPSO Alliance organization appeared, which is aimed at developing and implementing solutions related to the Internet of things. In 2011, Gartner included the IoT in the list of the most promising developing technologies. In 2012, the whole world started talking about IoT. In January 2014, Google acquired Nest, a company engaged in the development of smart home systems, which became a landmark event and signaled that the future belongs to the Internet of Things.

Cloud platforms are platforms for cloud computing, which are ready–made software and hardware leased over the Internet for the deployment, development, testing of their applications. The first cloud platform appeared in 2006 – Amazon Web Services. Amazon has offered more than 50 different services in 14 geographical regions. Today, cloud platforms are more popular than ever. The main advantages of using cloud platforms are the speed of creating new applications, flexibility and scalability of the system [2].

Since the 2010s, the IoT networks have been a steady trend in information technology, thanks to the ubiquity of wireless networks, the constant growth of Internet connection bandwidth, the emergence of cloud computing, the development of technologies for machine-to-machine interaction and software-defined networks, a person has surrounded himself with a network infrastructure that helps him and solves tasks, which previously had to be solved independently. According to a variety of forecasts and reports from analytical companies, connected devices (machines, fitness trackers, solutions in the fields of smart cities, houses, etc.) will become more and more. At the same time, the concept of the IoT will expand. Already now, more than 10 years after the emergence of the IoT has become one of the main trends of high technologies.

The monograph is devoted to the development of IoT networks based on a cloud platform for product quality control, audio information analysis, voice (cough) detection.

Using the monograph, readers will get acquainted with the analysis in the field of IoT network, IoT cloud platforms, hardware-software IoT tools and using intelligent technologies (Chapter 1). They will study the processes of developing and modeling the IoT network of quality control systems (Chapter 2). They will get acquainted with the problems of building and programming the data base of the IoT network and product quality control (Chapter 3). They will study the developing and modeling the IoT network for the processing of sound information from the surrounding environment (Chapter 4). In chapter 5 readers can recognize with methods and tools of cough detection using deep leaning and neural nets technologies.

The monograph includes an analysis of the results in the field of construction and optimization of the IoT networks, as well as the author's developments obtained in the course of state-funded research, work with graduate students and undergraduates. The book is based on the author's work on scientific projects within the framework of the Informatization program in 2015–2022:

– «New technologies of information management and electronic marketing»;

– «Research and development of a distributed cloud computing and virtualization environment in order to introduce innovative information and communication technologies into the educational process.

In the process of writing the monograph, the articles material with my graduate student Bahaa Shaya (Chapter 4) was used. Chapter 5 is written with PhD student Bahaa Shaya.

The author thanks the reviewers of the Honored Professor of BSUIR, Doctor of Technical Sciences L. M. Lynkov and Professor, Doctor of Technical Sciences V. I. Kurmashev for a number of valuable comments and recommendations.

1 ANALYSIS OF INTERNET OF THINGS NETWORKS, PLATFORMS AND DATABASES

1.1 Overview of the Internet of Things

The Internet of Things (IoT) is a concept for the development of internet technologies aimed at automating and excluding human participation from most of the processes of the IT infrastructure [1–4]. The concept provides that most of the devices used by people will be equipped with microcontrollers for control and network interfaces for digital data transmission and communication among themselves. The RFID group defines IoT as a worldwide network of accessible objects whose unique addressing is based on standard communication protocols [5].

The following areas of IoT use can be distinguished: food production; industry; transport and transportation; assessment of air quality, noise and energy consumption; provision of information about traffic congestion; use in everyday life; street lighting (Figure 1.1).

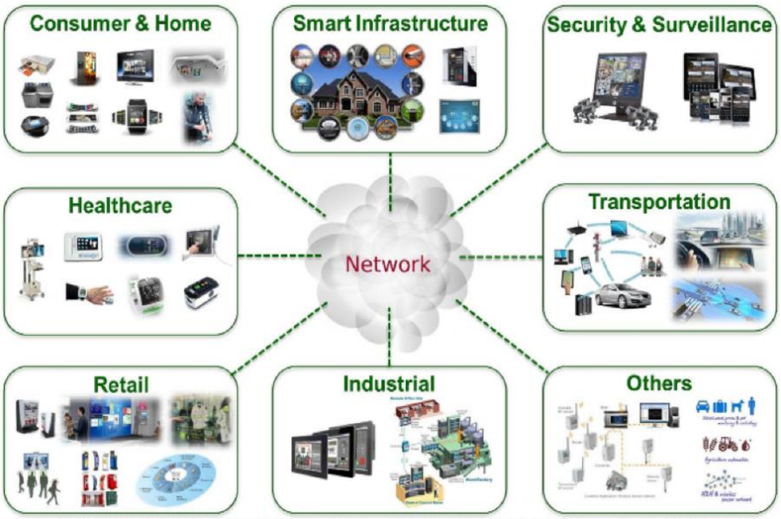


Figure 1.1 – Areas of use of IoT

From a technical point of view, IoT is a set of tools that provide the following features [5]:

– communication and interaction – objects can create a connection with each other, or with internet resources and update their state, using wireless technologies: Bluetooth, GSM, UMTS, Wi-Fi, ZigBee, etc.;

– addressability – in IoT objects are distributed in space and have a specific addressing;

– identification – clear association of data with a specific object and their extraction. With the help of RFID (Radio Frequency Identification) and NFC (Near Field Communication) technologies, it is possible to identify objects that do not have built-in energy resources;

– integrated information processing – objects are equipped with a microcontroller or processor for instant analysis and processing of information;

– probing devices generate data about the environment through sensors and change their state under its influence [6].

IoT allows companies to automate processes and reduce labor costs. Which in turn reduces the amount of waste, improves the quality of services provided, reduces the cost of the production process and logistics.

In 2009 the number of devices connected to the global Internet exceeded the number of inhabitants of the planet. It is predicted that the number of «things» will only increase with the widespread introduction of electronics. By 2021, the number of «things» has approached 50 billion, including kettles, smart watches, sockets, cameras, lamps, microwaves, refrigerators, air conditioners and other equipment.

According to the IoT analytics study, in 2021, the highest level of use of IoT technology was observed in transport, energy, industry, healthcare.

It is necessary to distinguish between the concepts of «Internet of Things» and «Internet thing». An internet thing means any device that:

– has access to the Internet for the purpose of transmitting or requesting any data;

– has a specific address in the global network or an identifier by which you can make feedback with the thing;

– has an interface for user interaction.

IoT have a single protocol of interaction, according to which any node of the network is equal in providing its services. On the way to the implementation of the IoT, there was a problem associated with the IPv4 protocol, the resource of free network addresses of which has almost exhausted itself. The preparation for the widespread implementation of the IPv6 protocol version solves this problem and brings the idea of IoT closer to reality. Each node of the IoT provides its own service, providing some kind of data delivery service. At the same time, a node of such a network can receive commands from any other node. This means that all IoT can interact with each other and solve joint computing tasks. IoT can form local networks united by a single service area or function [7].

To date, IoT finds its practical application mainly in the complex of M2M systems, in the near future, integrated sensor networks will be created on the basis of chipsets with very economical energy consumption and miniature RFID tags, and then cognitive networks («smart» networks based on knowledge).

M2M stands with machine-to-machine interaction. This means that there are several devices (at least two) exchanging data with each other. Here, the focus is not on the way data is transmitted, but on how it moves between «machines». M2M does not include the interaction of a machine with a person. As an example, we can consider a calculator. When used by humans, the M2M mechanism is not used. However, the calculator transmits the entered numbers to the database, respectively, M2M interaction occurs between the calculator and the database. The presence of the Internet is not necessary for M2M interaction [7].

The IoT receives a signal from the TV station, the interaction of the TV and the TV tower. When a smartphone is connected to a tower or to a Wi-Fi access point, M2M interaction occurs. Thus, M2M is the interaction of two devices via cable or wireless communication [8].

Consider the differences between M2M, IoT and IoE (Internet of Everything) [8] (Table 1.1).

Table 1.1 – Differences between M2M, IoT and IoE

Atributes	**M2M**	**IoT**	**IoE**
1	**2**	**3**	**4**
Size	M2M is a subset of IoT	IoT is a superset of M2M	IoE is a superset of IoT
Key components	M2M includes three key components: 1 Devices that generate or receive data from other devices. 2 Communication for efficient data transfer between devices and gateway. 3 An application designed to provide services in accordance with the requirements of the end user	The IoT consists of four key components: 1 Sensors or devices for generating or receiving data from other devices. 2 Communication, for transferring data to the Internet or between devices. 3 Data storage services for efficient data storage in a database or in the cloud. 4. Application for the provision of the intended service	IoE consists of four key components: 1 People, as end nodes, for the exchange of information, actions. 2 Things – devices that generate data or receive it from other devices. 3 Data used for the analysis and processing of information for making intellectual decisions and management mechanism

1	2	3	4
Type of communication	Between devices there is a point-to-point communication	IP network exists between devices due to the integration of communication protocols	IoE – this is a network connection of people, processes, data and things
Internet Requirement	M2M communication can exist without Internet	Devices in IoT in most cases require active Internet	Devices and their applications require active Internet access
The integration task	It is limited, since M2M uses the corresponding standard	It is more difficult due to the use of another communication standard in its solution	Higher degree of integration complexity compared to IoT

Components of IoT are represented as ABCDE: Analytics, BigData, Connection, Devices, Experience [9].

Analytics is the main link in the work of IoT, which combines the devices themselves, data from these devices and optimizes business processes.

BigData – there is information from devices – stored on the cloud. Makes it possible to automate existing processes or build new ones;

Connection is the channels through which devices receive and transmit information.

Devices connected to the system, which, depending on the tasks, must have the appropriate frequency of messages for correct operation.

Experience – work with the existing experience of solving problems using IoT, its analysis.

According to Forture Business Insights, the volume of the global IoT market in 2018 was $160 billion, by 2026 its volume will exceed $1.1 trillion. Rapid growth is associated with the widespread introduction of artificial intelligence and machine learning systems. The growth of the market is also facilitated by the growth in the number of users of «smart» devices, smartphones, as well as the growing demand for energy saving.

1.2 Basic Principles of IoT

Due to the rapid development of packet-switched networks and, the Internet in the early 2000s, the global telecommunications community first developed and then began to implement a new paradigm for the development of communications – next-generation networks (NGN). NGN technologies have already passed the evolutionary path of development from flexible switches (Softswitch) to multimedia communication subsystems IMS (IP Multimedia Subsystem) and wireless networks of long-term evolution LTE (Long Time Evolution). At the same time, it has always been assumed that the main users of NGN networks will be people and, consequently, the maximum number of subscribers in such networks will always be limited by the population of the planet Earth.

However, recently, significant development has been achieved in the methods of RFID (Radio frequency identification), WSN (Wireless Sensor Network), short-range communication NFC (Near Field Communication) and M2M (Machine-to-Machine communication), which, integrating with the Internet, allow provide simple communication of various technical devices, the number of which can be huge. Thus, there is currently an evolutionary transition from the «Internet of People» to IoT.

In general, IoT is understood as a set of various devices, sensors, devices connected to a network through any available communication channels using various protocols of interaction with each other and a single protocol for accessing the global network. The Internet is currently being used as a global network for Internet of Things. The common protocol is IP (Internet Protocol).

Most importantly, large companies have already begun to implement specific projects:

– provides the highest data transfer rate, but does not allow the equipment to be mobile;

– 3G/4G/5G. The coverage area of mobile networks covers all major cities of the planet and most of the other settlements. For example, 5G is capable of providing speeds up to 1-2 Gbps, which is almost as good as a wired connection;

– Wi-Fi, Bluetooth, Wi-Max and analogues. Data transmission at a short distance – in an apartment, house, office. Allow small appliances to connect wirelessly to the network;

– NFC, RFID and the like. The data stored in RFID tags is read or recorded via radio signals. Visually, the labels look like small labels attached to things;

– satellite Internet. With the launch of the Starlink network and its analogues, this way of communicating things can become one of the main ones, especially in remote regions where there is no mobile network coverage.

The leaders in the use of IoT in Russia are manufacturing, transport, and energy, which accounts for more than 50 % of the total IoT market. They are followed by the public sector associated with the creation of smart cities. Most of the IoT market in Russia and Belarus received three directions: production management; management and monitoring of transport by RFID, GPS, GPRS and geographic information system technologies are used; intelligent power systems [2].

IoT is based on three basic principles. Firstly, the ubiquitous communication infrastructure, secondly, the global identification of each object and, thirdly, the ability of each object to send and receive data via a personal network or the Internet to which it is connected.

The most important differences between the IoT and the existing Internet of people are:

– focus on things, not on the person;

– significantly more connected objects;

– significantly smaller object sizes and low data transfer rates;

– focus on reading information, not on communication;

– the need to create a new infrastructure and alternative standards.

The concept of next-generation NGN networks assumed the possibility of people's communication (directly or through computers) at any time and at any point in space. The concept of the IoT includes another direction – communication of any devices or things.

The concept of IoT and the term for it were first formulated by Kevin Ashton, founder of the Auto-ID research group at the Massachusetts Institute of Technology, in 1999 at a presentation for the management of the Procter company & Gamble. The presentation talked about how the comprehensive introduction of RFID radio frequency tags will be able to modify the logistics chain management system in the corporation.

The official definition of the Internet of Things is given in the ITU-T Recommendation Y.2060, according to which IoT is a global information society infrastructure that provides advanced services by organizing communication between things (physical or virtual) based on existing and developing compatible information and communication technologies.

«Things» here means a physical object (thing) or an object of the virtual (informational) world (a virtual thing, for example, multimedia content or an application program), which can be identified and combined through communication networks.

In addition to the concept of «thing», ITU-T also uses the concept of «device», which means a piece of equipment with mandatory communication capabilities and optional capabilities for sensing or sensing, actuating things, collecting, processing and storing data. It follows that ITU-T pays more attention to aspects of communications and interconnections than to IoT applications.

The diagram of the display of physical and virtual things is shown in Figure 1.2. It follows from the figure that virtual things can exist without their physical incarnations, while physical objects/things necessarily correspond to at least one virtual object. At the same time, the leading role is played by devices that can collect various information and distribute it over communication networks in various ways: through gateways and through the network; without gateways, but through the network; directly among themselves. Recommendation Y.2060 describes a different combination of the listed 10 connection methods. This indicates that ITU-T provides for the use of a variety of network technologies for IoT – global networks, local area networks, wireless self-organizing (ad-hos) and mesh (mesh) networks. These communication networks transfer the data collected by the devices to the corresponding software applications, also transmit commands from software applications to the devices [10].

It should be noted that things and related devices can have full-fledged control processors for data processing in the form of a «system-on-a-chip», including with its own operating system, an environmental sensing unit and a communication unit.

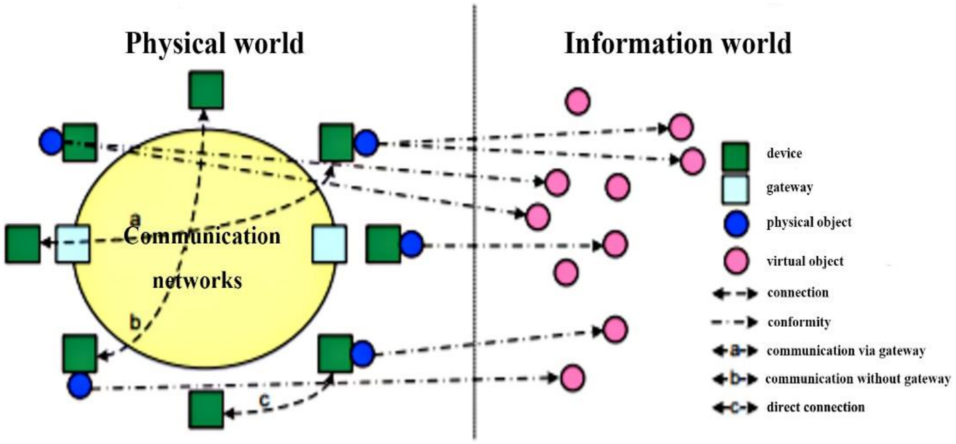


Figure 1.2 – Diagram of the display of physical and virtual things

1.3 Interaction of IoT with Promising Infocommunication Technologies

An important role in the formation and successful implementation of IoT is played by various promising information and communication technologies, such as big data, cloud technologies and ubiquitous computerization, with which IoT actively interacts. Currently, IoT finds its practical embodiment mainly in the form of M2M systems, in the near future integrated sensor networks will be created on the basis of chipsets with ultra-low power consumption and miniature RFID tags, and then cognitive networks («smart» networks based on knowledge).

Big Data. Until the beginning of the XX century, the amount of knowledge doubled every century, today the amount of knowledge of mankind doubles every 2-3 years. 70 % of all available information appeared after the invention of the Internet. The IoT radically increases the amount of data collected, which is a consequence of a huge number of information sources (primarily various sensors).

Giant sensor networks are already producing huge data streams that need to be able not only to store, but also to process, draw conclusions from them, make decisions and all this taking into account the inaccuracy of both the original data and processing procedures. In the late 2000s, an approach was formed for processing a large amount of data, which is called «Big Data» – a series of tools and methods for processing structured and unstructured data of huge volumes and a significant variety to obtain the necessary processing results.

As defining characteristics for big data, «VSD» are noted: volume (in the sense of the amount of physical volume), speed (in the sense of both the rate of increase and the need for high-speed processing and obtaining results), diversity (in the sense of the possibility of simultaneous processing of various types of structured and unstructured data) (Figure 1.3).

The main difference between big data and «ordinary» data is that it is impossible to process this data with traditional database management systems (DBMS) and solutions of the process intelligence class due to their large volume and diverse composition. Their other important property is the phenomenal acceleration of data accumulation and constant change. Such popular tasks as combining data obtained from different sources (Data Clearing, Data Megiddo, De-duplication) require special analysis methods in the case of inaccurate data, especially data of huge size. In this regard, a set of tools was developed, called «big data», allowing to work with data regardless of their type and volume.

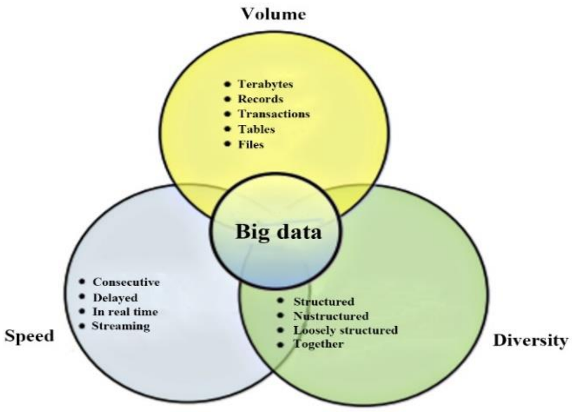


Figure 1.3 – Three main characteristics of big data

It is predicted that the introduction of big data technologies will have the greatest impact on information technologies in manufacturing, healthcare, trade, public administration, as well as in areas and industries where individual resource movements are recorded and where Internet of Things technologies can potentially be used.

Cloud computing (CC). Since the IoT generates BD, therefore, a natural question arises: where to store them and how to process them? The answer to this question is a promising infocommunication technology – cloud computing. Cloud computing involves renting services and resources for storing and processing data on a global network instead of its own infrastructure. CC systems should have five main characteristics: on-demand self-service, broadband network access, resource pool, the ability to quickly reconfigure or expand, and measurable maintenance.

There are four models for deploying cloud infrastructure (so-called «clouds»):

1 Private cloud is an infrastructure designed for use by one organization that includes several consumers (for example, divisions of one organization), possibly also by customers and contractors of this organization. A private cloud can be owned, managed and operated by both the organization itself and a third party (or some combination thereof), and it can physically exist both inside and outside the jurisdiction of the owner.

2 Public cloud is an infrastructure designed for free use by the general public. A public cloud can be owned, managed, and operated by commercial, scientific, and governmental organizations (or some combination thereof). The public cloud physically exists in the jurisdiction of the owner – service provider.

3 Hybrid cloud is a combination of two or more different cloud infrastructures (private, public or public) that remain unique objects, but are interconnected by standardized or private data transfer technologies and applications.

4 Community cloud is a type of infrastructure designed for use by a specific community of consumers from organizations that have common tasks (for example, missions, security requirements, policies, and compliance with various requirements). A public cloud can be co-owned, managed and operated by one or more of the community or third party organizations (or any combination thereof), and it can physically exist both inside and outside the jurisdiction of the owner.

Various CC services, generally referred to as XaaS (X as a Service), can be attributed to three main classes:

– infrastructure as a service (IaaS) – rental of server capacity and storage capacity of data centers;

– software as a service (SaaS) – rental of software that runs from the CC;

– platform as a service (PaaS) – renting a software development platform by collective or individual developers.

All other services of CC systems (for example, VBAS – video business process as a service or VSaaS – video surveillance as a service) can, one way or another, be attributed to the three above classes of cloud services.

Fog computing (FC) can also be used for the operation of IoT technologies. By «fog» is meant the approach of the «cloud» to the earth, in this case, «fog» is a kind of cloud services located not somewhere in inaccessible heights, but in the environment around us. In other words, Fog Computing is not an alternative, but an addition to Cloud Computing, and situations of their joint action may arise (for example, the execution of an analytical application), and in this case Cloud will provide a Fog service.

Fog computing complements cloud computing and ensures the interaction of smart things between themselves and cloud data centers in the form of a three-level hierarchical structure. The top level is occupied by thousands of cloud data centers that provide the resources necessary to perform serious, for example, analytical, IoT software applications. The level below contains tens of thousands of distributed control data centers, which contain the «intelligence» of Fog Computing, and at the lower level there are millions of computing devices of smart things.

Fog Computing can be defined as a highly virtualized platform that supports three main types of services that form M2M machine-to-machine communications: computing, storage, and networking. The task of Fog Computing is to ensure the interaction of billions of devices with each other and with cloud data centers.

The Fog Computing paradigm differs from Cloud Computing in a number of parameters:

– distribution of computing power and real time. Significant computing resources can be placed on the periphery of the network, and there should be no dependence on the coordinates of the place where the device is located, and at the same time, real time operation assumes a low level of delays in data exchange, besides, convergence of two control systems that have existed independently from each other for a long time can occur in Fog Switching business and technological systems;

– geographical distribution of components. The service distribution model in Fog Computing is less centralized than for clouds, and individual devices can be interconnected by data streams and provide each other with «heavy» services;

– a large amount of external data. Devices equipped with numerous sensors can generate huge amounts of data in real time;

– complex topology. Millions of geographically distributed nodes can create diverse and non-deterministic connections in advance;

– mobility and heterogeneity. The mobility of devices will require the use of alternative protocols, for example, the LISP routing protocol (Locator/ID Separation Protocol), which allows you to divide the functionality of IP addresses into two parts: host identifiers and routing locators. The concept provides for the installation of tunnel routers that will add LISP headers to information packets as they move through the network.

Ubiquitous computerization (UC). In 1991, Mark Weiser, a researcher at the Hegoh PARC laboratory, put forward the concept of a future world «richly and imperceptibly saturated with sensors, displays and computing elements connected to a single network and being integral elements of everyday objects». The physical possibility of implementing this concept appeared by the end of the 2000s with the total spread of cheap and miniature computing mobile devices, wireless networks and satellite navigation (Figure 1.4).

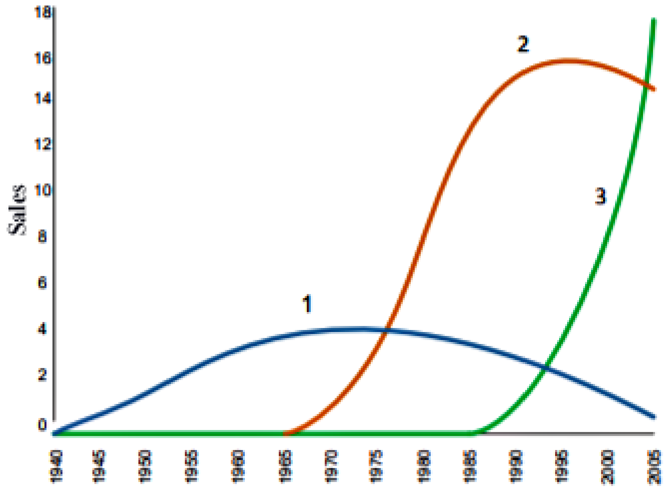


Figure 1.4 – Evolution of computing systems: mainframes (curve 1) – one computer, many users; personal computers (curve 2) – one computer, one user; ubiquitous computing (curve 3) – one user, many computers

1.4 Architecture of IoT Network

IoT consists of a set of various infocommunication technologies that ensure its operation. The architecture of IoT clearly demonstrates how various technologies are interconnected, and includes the following levels: things, gateways, networks, server [6, 10].

At the level of things, the devices themselves are executed directly. This is the lower level of the IoT architecture. It consists of «smart» objects connected to sensors that collect and process information in real time to achieve certain goals. For example, to measure temperature, acidity, density, location, etc. The development of microprocessors has led to a reduction in the size of hardware sensors and allowed them to be implemented everywhere.

As a rule, things, i.e. devices, have a connection to gateways connected to a local or global computer network. However, there are also «independent» devices that work on the basis of cellular operators' networks (connection is via Wi-Fi or Ethernet). At the same time, gateways are hubs that support a certain standard or protocol that provides a connection to things. As an illustration of an IoT system without a gateway, a schematic drawing is given describing the operation scheme of a GPS tracker with an NB-IoT module (Figure 1.5): The operator assigns an IP address to the device (or non-IP technology can be used) via NB-IoT (L1, L2) and gives things access to the Internet. The tracker itself can support the MQTT protocol, CoAP, or simply send data in UDP packets to the platform address specified in the settings via a public network. Sensors characterized by low power consumption and data transfer rate form wireless sensor networks (WSN – Wireless Sensor Network), which are actively gaining popularity, because they can contain a much larger number of sensors with battery support and include large areas. This is achieved by applying the mesh network topology. As an example, we can cite the ZigBee standard (IEEE 802.15.4), which is increasingly used in home automation systems using the «Smart Home» method;

At the levels of gateways and networks implement a wide range of tasks in IoT, it is necessary to ensure the collaboration of many different technologies and protocols. Access networks must provide the required values for the transmission of information on latency, bandwidth and security. This is necessary to connect diverse networks into a single network platform. Network gateways provide users and automated systems with network interaction with the end devices of the IoT infrastructure, through a supported communication standard. The gateway should provide the following functions: fixing a critical situation, followed by a local reaction; sending processed information to the server and receiving data for the configuration of the peripheral device; saving the collected data and information about the position of peripheral devices.

The service level (application level) contains a set of information services that provide automation of technological and business operations in IoT, namely, support for operational and business activities; analytical processing of information; interface for storing, entering and outputting data from an external system; application (API); information security; centralized control panel things [11].

At this level of the IoT architecture there are various types of applications for the relevant industrial sectors and spheres (Figure 1.6). Applications can be «vertical», specific to a particular industry, as well as «horizontal», used in various sectors of the economy.

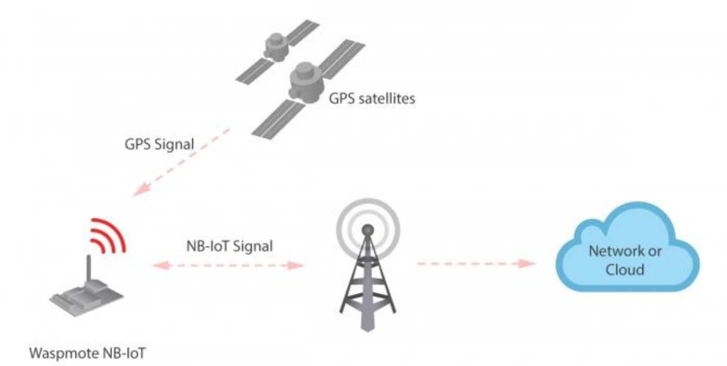


Figure 1.5 – GPS tracker with NB-IoT module

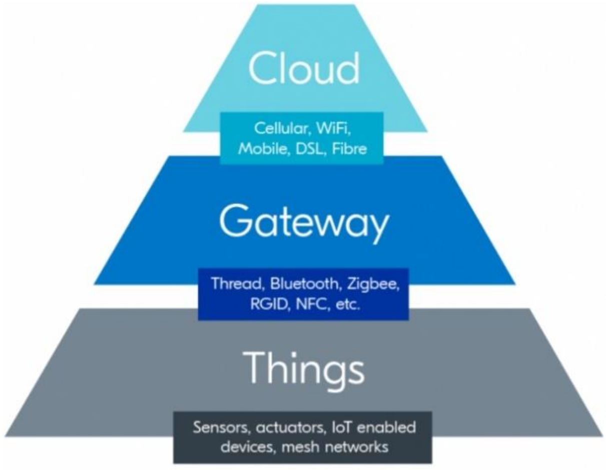


Figure 1.6 – Internet of Things architecture levels

A more detailed architecture is described below (Figure 1.7).

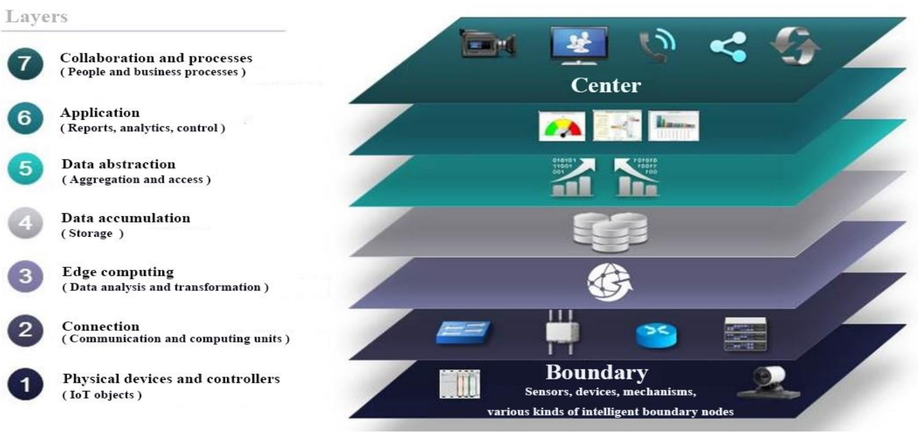


Figure 1.7 – Architecture of the IoT system

The first elements of the system include sensors, sensors, drives. Sensors are small devices or systems that are designed to understand and detect changes in the environment where they are located, as well as to further optimize information in their system. They have the unique ability to detect physical parameters, such as humidity or temperature, and then convert them into electronic signals. Drives are part of a machine that converts an electrical signal into physical actions. They play a major role as components of IoT networks.

The second element of the IoT system is Internet gateways and data collection systems. The Data Acquisition System (DAS) collects raw data from sensors and converts it from analog to digital format. DAS then collects and formats the data before sending it through the Internet gateway over wireless wide area networks (Wi-Fi, cellular) or wired wide area networks for the next stage of processing.

Wi-Fi is the most popular and versatile method used in data-driven technologies. Ethernet is equipment that supports fixed or permanent devices such as video cameras, game consoles, and security systems. Bluetooth is a widely used technology, suitable mainly for communication between devices at a short distance. NFC provides communication at a very short distance of 50 cm or less.

LPWAN (Global network with low power consumption), designed and built with the use of the IoT over long distances in mind. These low-power WAN devices can last for more than ten years while consuming low power consumption.

ZigBee is also an advanced wireless network technology that consumes a small amount of energy and can offer a small data sharing capability.

Cellular networks are ideal for communication on a global scale with great trust and reliability. For IoT, there are two broad levels of IoT in the cellular network:

– LTE-M is a long-term development of machines, providing very high-speed data exchange and uninterrupted direct cloud communication;

– NB-IoT as a narrowband, which offers a small data exchange using correspondingly low-frequency channels.

Also at this level, there are forr data transfer protocols:

1 Data Distribution Service (DDS) is a real-time messaging platform between machines in Internet of Things systems.

2 Advanced Message Queuing Protocol (AMQP) provides server-side protocols for servers via peer-to-peer data exchange.

3 The Limited Application Protocol (CoAP) defines protocols for limited devices that use low power consumption and little memory, such as wireless sensors.

4 Message Queue Transport Telemetry (MQTT) is a messaging protocol standard for low-power devices that use TCP/IP for seamless data transmission.

The third element of the IoT system: preprocessing and analytics, after the Internet of Things data is digitized and aggregated, it will need to be processed to further reduce the amount of data before it gets to the data center or to the cloud. The edge device can perform some analytics as part of the preprocessing. Machine learning can be very useful at this stage to provide feedback to the system and continuously improve the process, without waiting for instructions from the corporate data center or the cloud. Processing of this type usually takes place on the device in a location close to where the sensors are located, for example, in a switch cabinet on site.

Each device sends millions of data streams over the IoT network. Here, the data comes in different shapes, speeds and sizes. Separating important data from these large streams is a top priority that professionals should prioritize at this level.

After the data collection stage is completed, the selected data is extracted from big data for the application to optimize its business procedures. Here data abstraction follows the following path.

Collecting all data from all IoT and non-IoT systems (CRM, ERP and ERM), using data virtualization to Date of access from a single location, managing raw data in multiple forms, interaction between devices and architecture plays a crucial role at the processing level.

The fourth element of the IoT system: analysis in the cloud or data center. At this stage, data processing begins with powerful IT systems for analysis, management and secure storage of data. This usually happens in a corporate data center or in the cloud. A company can operate in different regions, and IoT data can be analyzed to identify key trends and patterns or identify anomalies.

The fifth element of the IoT system: at this stage, the processed data is sent to the system managers (users). From this data, you can learn trends, patterns and anomalies.

1.5 Common IoT Platforms

They provide the following functions for devices and software applications: start-up, maintenance, analytical processing, data storage and information security tools. Let's consider the most famous of the platforms from the world's leading IT giants [3, 4].

The Microft Azure IoT Suit platform supports both the already developed version of the network for modeling, and the ability to create new network solutions that meet the project specifications. Provides information security tools, the possibility of expansion and integration with existing or planned systems. The platform allows you to connect hundreds of devices from various vendors, process data from devices, receive analytical reports and use the information for further machine learning.

AWS IoT Soge (Amazon) is a platform on which you can create local networks or IoT applications, it supports special communication protocols, including custom ones, which allows communication between devices from different vendors. The component of the AWS IoT Devicemount platform supports the addition and management of external devices, their monitoring, adjustment and updating of their functioning. The AWS IoT Analytics platform component implements automatic processing and analytical calculations of large amounts of data from various IoT devices. The component of the AWS IoT Devicedefender platform supports the configuration of information security tools for IoT networks (authentication, encryption), allows you to create and adjust security policies, controlling the authentication and authorization of the device, providing a closed transmission channel (encryption).

The IBM platform supports remote control and monitoring of peripheral devices, secure transmission and secure storage of information from devices in a cloud environment, machine learning tools based on the use of artificial intelligence technologies.

The Google Cloud IoT platform includes a number of components with which the user creates IoT networks, such as: Cloud IoT Soge (a service for securely connecting, managing and receiving data from peripheral devices); Cloud Pub/Sub (a service for processing event data and implementing analytical processing of streams); Cloud Maschine Lightning Epdine (a service creates machine learning models when using data from IoT devices).

The platform from the company SAR is an environment for remote management and monitoring of all connected devices belonging to the IoT network. Peripheral devices can be connected directly or via a cloud service. Analytical tools allow you to process, systematize and study data received from sensors, meters and other IoT devices. The SAR platform provides a means of applying data in the IoT network to create artificial intelligence and machine learning applications.

The Oracle IoT platform (the world leader in the field of Oracle databases) supports the processing of a very large amount of data, which allows you to create large-scale IoT networks (tens of thousands of devices) with different mechanisms of Big Data technology. The platform uses basic information security mechanisms that protect IoT networks from various threats.

Cisco has created and is developing a platform for mobile IoT solutions based on cloud technologies. The Cisco service supports voice and data transmission, extensive configuration of IoT applications.

Half of the IoT platforms on the Russian market and EAEU (Eurasian Economic Union) countries are produced by domestic developers. But there are no comprehensive IoT platforms covering all known functions. From the point of view of system integrators and developers of smart devices, it is more convenient to work with one platform that would have a common programming interface for all or most functions, but so far there is no such platform. More than half of all market platforms are developed by domestic companies from the EAEU − Belarus, Russia and Kazakhstan (51 %). The second supplier of platforms to the EAEU market is the USA (23 %). The analysis of IoT platforms in the industry context reflects the overall dynamics of the market, the most competitive segments are the segments «Transport and Logistics» (42 %) and «Smart City» (32 %) [5].

Let’s consider an algorithm for modeling the IoT network using the IoT platform. Google Cloud IoT platforms create services taking into account the numerous requirements of users, and as a result, you can build a network quickly. This cloud platform has a significant advantage – the ability to independently simulate a network in a short time without involving a corporate IT service and additional information security tools. The generalized algorithm for creating a network on the Google Cloud IoT platform looks like this:

– sensors measure the parameters of processes (devices) that interact with the IoT platform using the SDK (Software Development Kit);

– devices send messages that are verified by the authentication and authorization service of the platform. In case of failure of verification, correction of device ids is necessary;

– information from the devices is sent to the gateway (Devicegate), while various network protocols can be used. Being transformed in the gateway, the information goes to the rules processing unit (communication with analytics) and in parallel – to the device storage unit (Deviceshadows);

– deviceshadows stores the current states of network peripherals for constant access to software applications. In the absence of communication with a separate device on the network, the Deviceshadow block executes commands from applications and, when the connection is restored, synchronizes the current state with the device;

– the rule handler, depending on the nature of the received data, performs (programmed) actions: saves data in the database, sends information to the network manager via SMS or e-mail about their receipt, calls HTTP API, sends data to the analytics system.

1.6 Google BigQuery Database

All organizations strive to get business information from their data. But it can be difficult to scale, store, and analyze this data as it grows rapidly. Google's corporate data warehouse, called BigQuery, was designed to make large-scale data analysis accessible to everyone [12].

BigQuery is designed to process huge amounts of data, such as log data from thousands of retail systems or IoT data from millions of vehicle sensors around the world. It is a fully managed and serverless data warehouse that allows to focus on analytics instead of infrastructure management. By design, BigQuery helps to avoid the data storage problem that occurs when you have separate teams in your company that have their own independent data storefronts. This can create significant friction between data analysis between teams and cause problems with data version control. Thanks to integration with Google Cloud's own identity and access control, you can assign read or write permissions to specific users, groups or projects and keep your confidential data secure while collaborating between teams. Working with data in BigQuery includes three main parts: storage, ingestion, requests.

Google handles everything else. BigQuery – it's a fully managed service, which means you don't have to configure or install anything. And you don't need a database administrator. You can simply log into your Google Cloud project from the browser and get started [12].

BigQuery storage. The data is stored in a structured table, which means you can use standard SQL to easily query and analyze the data. BigQuery is perfect for big data because it automatically manages all this storage and scaling operations. There are many ways to do this, as BigQuery is integrated with the rest of Google's data analytics platform. You can upload data directly from cloud storage or transfer data from cloud data stream. It can also be used to build an ETL (Extract, Transform, Load.) pipeline using Cloud Datafusion. You can also import data from various file formats.

BigQuery supports the same structured query language, or SQL, with which a large number of programmers familiar with relational databases compatible with ANSI work [12]. BigQuery can switch between standard SQL and dialects.

DML (Data Manipulation Language) operations INSERT, UPDATE and DELETE are currently supported only when using standard SQL. Another difference between these dialects is the way the tables are vertically joined. In standard SQL, the UNION operator and the keyword ALL or DISTINCT are used for this.

In its own SQL dialect, the functionality for vertical joining of tables is much broader. There is a special set of table substitution functions (Table Wildcard Functions). For a simple union, it is enough to simply list the names of the necessary tables or subqueries separated by commas.

Microft Power BI (Bussibess Intelligents), like most popular BI systems and spreadsheets, has been supporting integration with Google BigQuery out of the box since July 2017. The connector has rather poor capabilities: it does not know how to access saved views or send requests to BigQuery. So far, only flat tables can be pulled using the built-in connector:

1 Simba Drives. If you need to get data from Google BigQuery in a spreadsheet or a BI system that does not support integration out of the box, use the free Simba Drives. This driver supports all the necessary features, including switching SQL dialects. For details of the setup.

2 Language R. The R language is one of the most powerful tools for working with data. He can both receive data from Google BigQuery and record them. The bigrquery package is the most convenient for this. First, set the language to R. Also, for the convenience of working with R, recommend installing the integrated development environment RStudio. Launch RStudio and use the keyboard shortcut «Ctrl + Alt + Shift + 0» to open all the panels available in it. Most often, you will need the Soigse and Console panels.

Google BigQuery is a simple and at the same time powerful tool for storing and processing data. This is a cloud database with support for most DBMS functions. The service is much cheaper than maintaining, maintaining and administering a server for free databases (MySQL or PostgreSQL) [7].

1.7 Rationale for Choosing a MySQL DBMS

MySQL is a relational database management system (RDBMS). Where «relational» means that the data is stored as tables. One of the oldest, since it was written back in the 80s in C and C++ languages. Now it belongs to the company Oracle [13].

MySQL-type systems can be used by any sites or applications to store different data. For example, social networks store all information about users, their messages, pictures, videos. MySQL has an open source code, that is, you can change it or supplement it to suit your needs. Since the ability to download the software is available to anyone. The system is also compatible with many platforms – macOS, Windows, Linux, Ubuntu.

SQL is a structured query language that allows interaction between a software product and a database. There are many other RDBMS that also use the SQL language. In general, SQL is one of the most popular languages used in databases. Since with its help, all manipulations with data can be carried out efficiently. This is important for the speed and stability of the database.

Features of MySQL. This system is considered relatively cheap, compared to other paid RDBMS. It scales well, is considered quite flexible and easy to use. Plus, the system was originally designed to manage large databases. Therefore, now it is suitable for industrial operation, if you look at it from the point of view of the speed of work. It doesn't matter if you are doing heavy business analytics or need to store large amounts of ecommerce data.

Flexibility is provided by a large number of table options for data storage. You can select, for example, tables of the MuISAM type that support full-text search. Also InnoDB type tables that support transactions at the level of individual records. That is, you calmly choose what is right for your project.

Plus, the system supports many graphical interfaces – WorkBench, SequelPro, dbvisualizerand Navicat DB. Some of them are available only for a specific OS, some are commercial. But in any case, it is possible to choose your own comfortable format.

Useful MySQL functions:

– the possibility of cryptography (AES_ENCRYPT()/AES_DECRYPT()). This feature provides a high level of protection for algorithms such as AES and DES. They perform symmetric encryption and decryption of data. Therefore, they should be used when you want to increase the application's resistance to hacking;

– work with long text. The functions COMPRESS() and UNCOMPRESS() allow to store long text in the database without affecting performance. Also, long text affects the requirements for the amount of disk space, so it is better to compress it through the CMP. And when necessary, convert it back;

– the REGEXP function provides flexible matching with a regular expression template;

– another convenient feature of MySQL is full-text indexing of the fields VARCHAR and TECHT. For example, if you store articles or announcements in the database and want to provide the user with the ability to search. In general, you can install a search engine for such tasks, but for this you will have to create a separate database [8].

1.8 Analysis of Optimization Methods for IoT Networks

Network optimization is defined as a technology used to improve network performance, it is important because a large amount of data is loaded into the network from various devices and applications. Network optimization offers various benefits such as higher data transfer speeds, data recovery, elimination of redundant data, and increased application and network response time.

It will discuss the need for network optimization in IoT, various types of algorithms proposed by the authors to ensure network optimization in IoT in article [14]. Network optimization in IoT is attracting more and more attention due to the expectation of a significant increase in traffic from IoT things and objects. It is obvious to developers and operators that they must provide an effective solution for IoT networks in order to reduce the generated IoT traffic affecting other services on the network and use network resources efficiently. The traffic generated by IoT devices differs from cellular network traffic due to the heterogeneity of applications and device types. In addition, it is necessary to regulate IoT traffic to monitor the operation of IoT devices and its services. The IoT application generates less data, but the integration of devices into the application generates more traffic due to control plane messages. Consequently, this non-application traffic creates a significant additional load on the network. Thus, to overcome this burden, an effective mechanism is required for processing and optimizing the exchange of messages in the control plane from Internet of Things devices.

Classification of algorithms. The IoT network optimization task consists of many parts that will be combined using various combinations and methods that solve a certain type of network problem. There are two optimization methods: 1) the application of a known optimization structure to solve the problem; 2) the development of a new approach based on a heuristic method of solving the problem [15]. The above approaches are not mutually exclusive, but sometimes they are combined when the problem is too complex or known approaches give inadequate results.

The heuristic approach consists of: a) an algorithm that provides a faster approximation solution for a more complex problem, for example, convex optimization; b) a greedy approach that provides an optimal solution by making assumptions. Both of these approaches provide optimal solutions to complex tasks and provide performance close to optimal. Therefore, there will be no single algorithm that would provide an optimal solution to the problem of network optimization in IoT.

Algorithms based on particle swarm optimization (PSO). PSO is a computational method that optimizes a given problem by iteratively improving a possible solution taking into account a given quality. PSO originated on the basis of swarming behavior of animals, birds, etc. and their gregarious nature. Thanks to the unique structure demonstrated by them, the necessary information is provided that intelligence is not concentrated on individuals, but is distributed among many individuals of the group. PSO has gained a lot of popularity in recent years, and many research articles related to various optimization methods have been published using this method.

In [16], the authors proposed a PSO algorithm with orthogonal learning, which provides rapid route recovery after a path failure due to the mobility of the receiver node, and also provides an alternative path for effective path recovery using an orthogonal learning strategy. The result proves that the algorithm reduces communication overhead and increases the service life of the network. The authors in [17] used PSO to estimate the different level of transmission power required for each node, without creating disconnected areas in the sensor cluster. The final results show that when using PSO, the method saved more sensor energy compared to deploying conventional nodes with a single transmission power.

Energy efficiency is a critical issue in cluster capillary networks, where the process of selecting cluster heads (CHS) has a noticeable impact on network performance. In [18], a new QPSO scheme for selecting CHs is proposed, which increases energy efficiency and extends the service life of the network compared to evolutionary algorithms. In [19], the authors proposed an improved PSO (IPSO) to improve the accuracy of measurements using weight coefficients calculated using experimental modeling. The results obtained as a result of the experiment show that this algorithm combines the factors of weight, reliability of merging of information sources, redundancy of information and consolidation of hierarchical structure in uncertain merge scenarios. This fusion data will be extracted optimally by eliminating noise, interference, and this method reduces the energy consumption of the sensors. The authors in [20] considered a multi-purpose algorithm for optimizing a swarm of particles to increase the broker's profit while reducing the response time to a request and reducing the power consumption of a cloud broker.

Genetic Algorithms (GA). GA tries to assign a suitable value to a competing solution for a problem using natural evolutionary activity, as well as using the principle of survival of the fittest. GA can be used for both limited and unlimited optimized tasks.

In [21], an optimal routing algorithm, a k-means and GA clustering algorithm are proposed. Using the k-means clustering algorithm, you can achieve the best cluster header and cluster formation, and using GA, you can choose the optimal path. GA relies on the energy value of the cluster head and the path length, therefore, the resulting path obtained with GA will have greater reliability, higher speed and service life.

In [22], the authors proposed a GA-based clustering optimization method for limited networks, taking into account the basic IETF ((Internet Engineering Task Force)) standards for data transmission and the main interfaces, thanks to this, the battery level at the nodes, transmission energy and node processing capabilities can be improved. With the help of the main interfaces, energy consumption can be reduced, since fewer control messages are used in the communication process.

The authors in the article [23] proposed a heuristic-based genetic algorithm for selecting effective nodes for performing annotation of sensor data in the network. This method uses multi-purpose criteria to select the best candidates and selects sensors with the maximum amount of memory and energy level.

Genetic sorting algorithm without dominance II (NSGA-II). NSGA-II is a multi-purpose evolutionary algorithm based on non-dominance of sorting to reduce computational complexity, a non-elitism approach and the need to specify a shared access parameter [24]. Many researchers have chosen NSGA II to solve various multi-purpose optimization problems corresponding to various tasks. To save energy, various routing algorithms based on single-purpose optimization have been proposed, but the author in [25] proposed a multi-purpose evolutionary optimization algorithm that reduces network power consumption by optimizing sensor distribution. Song et al. in [26] combined quantum particle swarm optimization (QPSO) together with NSGA-II to increase the operational efficiency of industrial applications. This combined algorithm provides a better compromise between quality of service and energy consumption, and also improves network performance. In order to solve the optimization of energy consumption as a multi-purpose task instead of a single-purpose assessment, the authors in [27] proposed MOR4WSN based on NSGA-II, choosing an outstanding sensor distribution to maximize the service life of the network, as well as a method for optimizing the results.

Heuristic algorithms [15]. A heuristic algorithm is used to find a solution from a variety of possibilities and provides a relatively close solution to a complex problem in a simpler and faster way. There is a lot of literature available for network optimization based on heuristic algorithms. For example, in [28], the authors proposed the RPL routing protocol in the form of a reliable shortest path tree (RSPT), which increases the stability of network routing by taking into account the uncertainty present as a channel, and to solve the problem of the cost of a single arc, which is determined by acceptable values instead of the problem with a single value, they expanded the Heuristic algorithm based on scenarios (ABS).

The authors in [29] proposed computational intelligence (CI) to save energy and device resources by switching CI tasks from IoT devices to Cloud N, and optimized heuristics based on dominance sorting are used to save energy. The performance of all IoT network devices has improved with this method.

Kaustubh et al. in [30] proposed a heuristic channel selection algorithm (HOLA) that minimizes total energy consumption and also balances energy across the entire network. HOLA achieves this by transferring device data to smart devices calibrated according to factory settings.

The authors in [31] used LTE technology to provide coverage for various Internet of Things devices, and in order to make this technology resource-intensive and facilitate efficient communication, they proposed an LTE Random Access Channel (RACH) mechanism. This mechanism allows devices to access channels and reduce transmission power. The authors proposed a delayed power increase algorithm (DPRA), which is a heuristic approach.

An approach to biology-inspired heuristic algorithms. Biology-inspired algorithms are algorithms widely used for optimization and computational intelligence. Recently, many research papers have been published on network optimization in IoT to solve many problems. For example, the author in [32] proposed local recovery of 6LoWPAN using the Bio Inspired Artificial Bee Colony (ABC) routing protocol to reduce network overhead when detecting a route to a destination, since LOAD, MLOAD and AODV for the 6LoWPAN mesh network overload the network when detecting a network using a broadcast route request message. The authors in [33] have developed a new multi-object optimization algorithm based on a chaotic ant swarm (CAS). CAS use the chaotic behavior of individual ants and the self-organizing characteristics of an ant colony to determine the rules for choosing neighbors and convergence of the algorithm to reduce the error rate, the distance between generations and intervals.

Maciej et al. in [34] proposed hive surveillance for early warning of network intrusions using methods to reduce latency when detecting network intrusions. This method uses a joint bee decision-making method and a local information extraction algorithm to search for and use critical resources in their environment to enhance network intrusion detection.

The authors in [35] proposed a way to improve the communication between the Internet of macro/nano things, since their intermediate nodes refuse to interact with other nodes to save energy and reduce network overhead. To achieve this goal, the authors proposed a distributed model inspired by biology that uses a Voronoi-based collaboration strategy and a trust strategy to enhance collaboration between nanoscales.

Evolutionary Algorithms (EA). EA uses a population-based approach to a metaheuristic algorithm. The Expert Advisor provides an approximate solution to almost all types of problems, since it does not make assumptions when formulating a problem. Some of the EA-based works in [36], the authors proposed an optimal protected protocol with energy consumption and an improved algorithm for optimizing bacterial feeding for secure data transmission and energy savings when choosing a cluster head for data transmission between source and destination. This method is superior in throughput, energy and latency compared to previous methods.

The failure of host devices due to lack of energy is considered by the authors in [37]. They proposed a method that balances the energy consumption of devices placed outdoors, using an evolutionary approach based on games to select services. This method restricts the integration of devices through global interaction to select a service in the case of simultaneous applications.

The authors in [38] proposed a method for eliminating heterogeneity in the Internet of Things networks. They compared GA and Harmony Search (HS) in all aspects to show that traditional clustering methods will not lead to effective clustering.

Biying et al. [39], proposed an evolutionary algorithm for data classification to identify data and manage a huge amount of data from the Internet of Things. This algorithm performs sensitivity analysis to find the optimal solution and helps the neural network to rebuild itself to get rid of the tedious input problem.

Algorithms based on fuzzy logic. Fuzzy logic is used to determine partial truth, the true value of which is between the complex value of true and false. Linguistic variables controlled by membership functions and interference rules are used to obtain truth values. To detect an anomaly in the IoT traffic, the authors in [40] proposed a fuzzy logic interference applied to stationary Poisson or self-similar traffic of the Internet of Things network. They proposed a modified sliding window and a modified stochastic approximation for detecting anomalies in traffic.

The authors in [41] proposed a variable category clustering algorithm (VCCA) using fuzzy logic applied to a local Internet of Things network to select the network that has the greatest network capabilities. To achieve this, VCCA uses a fuzzy inference system, which uses a rule-based variable to select variables of lower complexity and to provide higher scalability among cluster variables. According to the authors, this algorithm outperforms network performance in terms of power saving, latency, bandwidth, and network life.

The authors in [42] propose a fuzzy logic method applied to a special automotive network to create a smart car IoT application. The authors claimed that the proposed algorithm optimizes the performance of the V2V network in double digits in terms of data transfer between the access point and devices.

Yijun et al. in [43] proposed an encryption scheme based on fuzzy identification to protect data during transmission. The proposed scheme provides better security and shorter public parameters. The authors suggested that this scheme is suitable for secure data transmission in an IB environment.

Automation of energy consumption in the case of energy-intensive industrial equipment is a difficult task. To achieve these goals, the authors in [44] proposed a fuzzy integrated assessment method that tracks and helps optimize the energy required by energy-intensive equipment for industrial use, as well as assesses the operational level.

Stochastic algorithms. Stochastic algorithms used for optimization purposes use random variables that consist of random constraints or functions to solve stochastic problems. In [45], the authors proposed a model for ensuring reliable data transmission of the Internet of Things in a wireless communication environment. To achieve this goal, they proposed an improved algorithm for distributed stochastic routing, which reduces the delay in the delivery of IoT data while increasing the packet delivery coefficient by applying the Markov chain concept to the proposed model.

To establish a connection between base system and IoT devices to support a massive IoT network in a cellular network, the authors in [46] proposed a random access channel (RACH) model. It uses a new spatio-temporal model taking into account traffic to analyze the proposed effect of the RACH model in the IoT network. This model helps to integrate and analyze different types of IoT devices at different times to achieve an optimized network goal.

Due to the higher complexity of the IoT network, conventional security countermeasures cannot be applied directly. To achieve this goal, the authors in [47] proposed a stochastic game net (SGN) model for IoT security. This model has improved privacy, integrity, and accessibility over traditional methods.

In order to provide reliable and efficient wireless access to data generated by IoT in cellular networks, Mohammad et al. In [48], a model based on geometry and queuing theory was proposed. This model solves the scalability problem imposed by IoT in the cellular network and provides an efficient way to transfer data.

Memetic Algorithms (MA). MA uses an evolutionary or population strategy to improve problem finding. In [49], the authors proposed a new intelligent transport platform based on authentication. The framework is used to reduce real-time traffic, waiting time, processing time on toll plazas and transportation problems through the use of MA. In this framework, MA plays an important role in computing an optimized single-point solution using data collected from the object, agent, and some third-party services.

The authors in [50] proposed MA in combination with local and global optimization to solve the inverse problem. This algorithm helps to provide a better solution for a structural condition monitoring application that requires higher computations, a delay-resistant and faster response system.

Other algorithms. In addition to the general type of algorithms described above, there are many others that help optimize the Internet of Things network. In [51], the authors proposed a Bayesian approach to the network model for detecting intrusions into the Internet of Things networks. This model has great opportunities for dynamic identification of the main nodes to provide the best security function that can be achieved using historical data.

The authors in [52] proposed an algorithm based on artificial intelligence to form clusters, select an optimized route and perform multipath routing to achieve a better quality of service. In [53], a Markov chain model was proposed to increase the service life of IoT devices using energy harvesting. This model defines a set of policies for battery level management and increases the speed of detection of IoT devices.

A game theory-based routing protocol was proposed in [54], which is responsible for choosing the best transition for forwarding data packets in an opportunistic IoT. This protocol is superior to it in terms of reducing the number of transitions, message loss and overhead.

The authors in [55] used a lexicographic optimization approach to ensure energy efficiency and QoS when choosing IoT services.

The main network parameters supported by various types of algorithms. Let's compare the above types of algorithms with different network parameters to provide different network parameters supported by these types of algorithms:

1 Load Balancing: Network load balancing during data routing plays an important role to extend network life. Taking into account the multipath metric in routing will help ensure reliable data transmission with a lower probability of node failure in the network.

2 Maximizing Network Life: Load balancing options and node failure management along with energy efficient routing help maximize network life. Since the battery life or energy of the nodes is limited, the mechanism must limit or balance various network parameters in order to maximize the service life of the network. Most of the algorithm helps to maximize the lifetime of the network.

3 Failure management: Connection failure occurs due to the failure of nodes in the network, leads to signal degradation and shortens the service life of the network. Therefore, the mechanism should minimize connection failures to ensure reliable communication.

4 Communication quality: This parameter provides QoS for communication. In the case of multipath propagation, the path is checked and the data is forwarded along an efficient path to reduce payload retransmissions and predicted delays. This option partially helps to maximize network life, as it reduces packet retransmission.

5 Energy efficiency: Algorithms should provide an energy-saving mechanism to minimize the energy consumption of nodes, which is an important part of the Internet of Things. Energy saving can be considered in various aspects, such as routing, shortening the work cycle, overload control and many others. Most of the algorithm provides an energy-saving strategy to maximize network life.

6 Heterogeneity: IoT is a combination of different types of devices and services, data from these devices have a heterogeneous form. Heterogeneity is taken into account according to many factors, such as different manufacturers, different types of hardware and software, different protocols, etc. The algorithm must support heterogeneous environments and facilitate interaction between different protocols. But most of the algorithm does not support heterogeneity.

1.9 IoT Hardware and Software Development Tools

The process of developing and implementing IoT systems is a sequence of interrelated activities, starting from the project planning stage and ending with the use of the system [56]. The structure of the IoT development program includes: goal definition; preliminary surveys and resource assessment; IoT solution strategy; selection of the IoT reference architecture; design solution; monitoring development; system implementation; quality assurance procedures for IoT systems; data management [57].

The planning process can be divided into three main stages. The first stage involves identifying the needs and defining the goals of the IoT. After defining the goals, you can decide what data is needed and how it will be used. Then the solution strategy, IoT reference architecture and technology platforms are determined. The second stage includes the development of the IoT project. The third stage includes the actual implementation of the IoT project.

IoT design strategies. When developing IoT systems, two main strategies are used: hybrid and model [57]. Hybrid assumes that the IoT system is developed by combining existing services and is based on well-known web development tools and approaches (prototyping), used for non-critical applications.

A model-based approach for the design and development of systems based on IoT things was proposed in [58]. Tools are also used, for example, the Internet of Things Modeling Language (ThingML) [59], which provides a customizable code generation structure. It can be configured for specific languages, middleware, operating systems, libraries, and systems [60].

In [61], two main stages of the methodology based on the model were identified: 1) identification of requirements and 2) formalization of requirements. The first stage is based on consultations and discussions with domain experts to gather information about the functionality of the system and identify requirements. Functional requirements can be presented using UML diagrams describing the functions of the IoT system.

The second stage is aimed at structuring requirements and developing models to describe the interconnection of system processes. At this stage, it is possible to identify sublevels with specific tasks related to the design of Internet of Things systems, such as coordination, integration, big data management, etc. The main advantage of the model-oriented approach is platform-independent modeling, which allows you to generate code for specific IoT platforms.

Design tools. Mashup tools allow you to perform very rapid prototyping and use the transformation, transformation and consolidation of data from one or more services to achieve the goals of the project. They also allow you to connect various services to create new processes. In addition, some mashup tools, such as Clickscript [62], WotKit [63], Paraimpu [64], can provide modeling tools and support interaction between different platforms. The tools can be effective in describing the system architecture, message flow (activity diagrams), and deployment. Mashup-based development includes three phases.

Phase 1: Study the development environment to determine the most appropriate tools currently available to achieve the goals of the IoT project.

Phase 2: Choosing a platform for integration between services and remote platforms. For example, this phase can be performed using an open source computing notebook (Jupyter Notebook) [65].

Phase 3: Data processing and manipulation. IoT data collection can include five alternative IoT session layer protocols: AMQP, CoAP, DDS, MQTT, and XMPP.

The simplest IoT reference architecture includes three layers: sensor, network and application. A more complex architecture can contain from five to seven layers: device, network, processing, application, business, management and security.

IoT devices can be connected either directly to the network or through a gateway, which allows devices to interact with each other, as well as with cloud services and applications [66]. Moving up, the components become more complex, and their connectivity increases [67]. Let's consider the composition and purpose of components in a five-tier IoT architecture:

1 Sensors, actuators, intelligent devices and embedded systems are components of the sensor layer. They collect data from various physical, human and natural environments and provide temporary storage of this data. They provide two main capabilities: collecting and analyzing data from the environment.

2 IoT gateways (hubs) are devices that connect the components of the first layer to networks. Integrated into automotive engines, heating, ventilation and air conditioning systems, using the data obtained and their pre-processing, built into concentrators, allow final IoT projects to adapt to the behavior of the environment and optimize the efficiency of functioning.

3 Network and cloud services provide the infrastructure for IoT operation. They can be either publicly accessible (to the public) or partially accessible (protected by the organization's firewall). These network services provide Internet connectivity to hubs, as well as provide the cloud computing power needed to collect, store and analyze large amounts of data from many end devices.

4 Data centers (cloud platforms) includes the most technologically complex components of the Internet of Things. There, intelligent processing of the received data is performed and appropriate decisions are made.

5 Advanced services (applications) allow you to collect and analyze data from various platforms and provide extensive interactive functions for end users.

Sensor networks are an important component of systems based on the Internet of Things. The feedback IoT system consists of sensors, processing centers and actuators. Data is collected using sensors. Then these data are transmitted through hubs, sensor networks to cloud platforms, where they are processed, and appropriate decisions are made based on the results of processing. The executive mechanisms perform system operations based on the decisions made.

Development boards and IoT platforms for prototyping. In many cases, the IoT development process involves prototyping using a single board or combinations of them that best meet the desired goal. IoT development boards contain standard communication tools, sensor interfaces and general-purpose I/O connections, so that the developer can integrate sensors, actuators, communication tools and a microelectromechanical system to create a prototype of his IoT network.

The most popular IoT platforms for prototyping are Arduino, Raspberry Pi, ESP8266 and Spark Core [68]. Raspberry Pi is ideal for studying server projects of the Internet of Things, it can connect to both a local network and Wi-Fi networks. However, it is not recommended to use Raspberry Pi for projects where there is a custom printed circuit board, or for integrating Raspberry Pi into a finished product. Arduino is ideal for recording sensor data and controlling actuators using commands sent to the server by another client. ESP8266 is best used in client applications such as data logging and control of actuators from online server applications.

The Spark Core platform is ideal for both server and client functions. It can be used to record sensor data to the cloud Spark.io or to receive commands from the cloud. There are several basic criteria for choosing a suitable IoT platform for an IoT project [69]:

– type of connection (Wi-Fi or cellular communication) for an IoT system;

– type of service: some services are purely connection platforms, while others are complete solutions offering hardware, software and connectivity options;

– service life of the project: how long will the IoT system work;

– tariff plan: does the supplier offer an effective tariff plan;

– security/privacy: how the IoT platform solves security problems. Cloud platform gateways should offer SSL or DTLS encryption;

– managed integration / API access: how does the provider integrate cellular modems, SIM cards, device diagnostics, firmware updates, cloud connections, security, application layer, etc.;

– redundancy and disaster recovery: does the cloud platform provider have a dedicated infrastructure for data processing and is it backed up;

– IoT system: relationships between platform services IoT;

– data access: does the service meet the needs of integrating data received through the IoT platform with the current cloud service;

– hardware: does the vendor offer any off-the-shelf applications, developer kits, or starter packs for your use case;

– device management: how the provider allows you to track, segment and manage IoT devices;

– frontier intelligence: the IoT platform should be able to expand from cloud to fog and support new topologies for decentralized computing;

– updates: as a provider, it allows you to remotely send updates and fix bugs on devices.

IoT project development tools on the Russian market [70]. It is necessary to divide two big directions: IoT-communication and IoT-services. IoT-communication is communication and services that allow this communication to be managed. For example, you purchased the required number of SIM cards from the operator, inserted them into M2M devices (counters, ATMs or cash registers) and you can control the costs and availability of devices using the M2M monitoring platform.

In this market segment, the most capacious industries are transport and logistics, although the level of penetration is quite high. They also include finance and housing and communal services, the development of which is largely stimulated by the state.

The market of IoT services is implemented differently. Business applications are being developed that work with the data collected by things. Correctly collect, process, analyze, visualize and help automate processes by setting up the logic of things with or without human participation. Each industry has its own meters and sensors, its own things, its own data collection and exchange protocols, and most importantly, its own business logic.

The Smart Housing and Communal Services project collects data from electricity, heat, gas, and water meters, the product is intended for management companies and developers. The «Online Water Monitoring» solution is intended exclusively for water utilities. And in the logistics industry, housing and utilities solutions do not scale – there are other sensors and other business logic of applications.

Another example: the Video Analytics product has developed more than 20 computer vision scenarios: for example, to recognize visitors, to read the number of the car to open the barrier, etc.

The most capacious in terms of revenue and quite complex for digitalization with the help of IoT technologies is «Industry 4.0». It requires deep industry expertise, knowledge of production processes, the ability to calculate the return on investment better, because its projects are individual and expensive.

From the point of view of deploying the technology on the infrastructure of a telecom operator, NB-IoT is an attractive and promising technology, since it uses the infrastructure and equipment of an existing network. The technology is energy efficient and operates in a licensed spectrum, which guarantees protection against interference. Its deployment requires minor changes in the core of the network and in its radio part. Since the NB-IoT technology works in the LTE standard, it was only necessary to update the software and activate the technology on the network infrastructure - base stations and the reference network.

Types and features of hardware platforms [71]. Let's consider hardware platforms for the development of IoT (Figure 1.8). At the same time, it should be noted that some of these platforms are focused only on one aspect of the IoT (for example, SigFox focuses on connecting devices), while others are kind of all-in-one platforms (for example, Particle.io), providing a comprehensive solution for the development of the Internet of things. Of course, the list of such platforms is far from complete – there are a lot of them now, but we tried to choose the most popular of them:

1 *Particle.io.* Particle.io – this is one of the most complete integrated platforms of the Internet of Things. It is a platform that offers solutions for the development of Internet of Things equipment, connectivity, cloud devices and applications. Particle.io produces a wide range of products for the development of IoT equipment both for «fast» prototypes and for production at the DFM level.

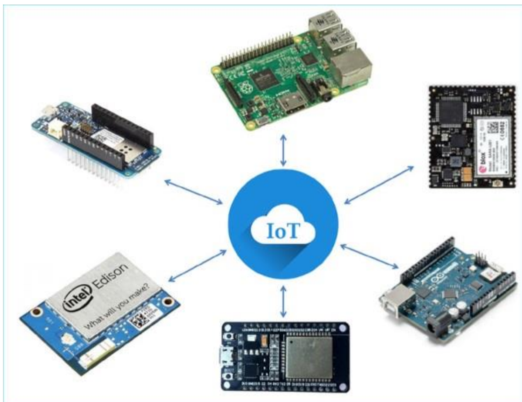


Figure 1.8 – Hardware support for IB development

Creating an IoT product in it begins with connecting devices to the Internet, and all particle microcontroller boards support communication via Wi-Fi, cellular communication (2G/3G/LTE) or mesh network (Figure 1.9). Some of these boards support several types of communication at once.

Figure 1.9 – Type of board Particle.io

2 *Platform microcontrollers Particle.io*. They are controlled by a special operating system that allows the developer to easily integrate devices with the device cloud and particle applications. As a rule, their devices and communication modules come with CE and FCC certificates, which reduce the cost of certification when the product is ready for mass production. Their open source boards are provided with a good level of technical support, which, of course, is very convenient for developing products based on this platform. Many developers like the platform Particle.io due to the complex nature of its services. This ensures that get support at every stage of your product development without worrying about compatibility.

3 *Espressif ESP8266 and ESP32 boards.* The range of Espressif and AI thinker products is perhaps the best for the platform Particle.io. Since the release of the ESP8266-01 Wi-Fi chip a few years ago, ESP8266-based chips and boards have become one of the most preferred chipsets for Wi-Fi-based IoT devices. These modules are quite cheap, consume little electricity and are easy to use. This attracts the increased attention of hardware developers to them. Following the ESP8266 module, Espressif introduced the ESP32 module, which is already built on the basis of a dual-core microcontroller and has a wider set of functionality compared to the ESP8266 module, but costs only slightly more (Figure 1.10).

ESP chips have great flexibility and can be used as Wi-Fi modules, connected to other microcontrollers or used in standalone modes without the use of additional microcontrollers.

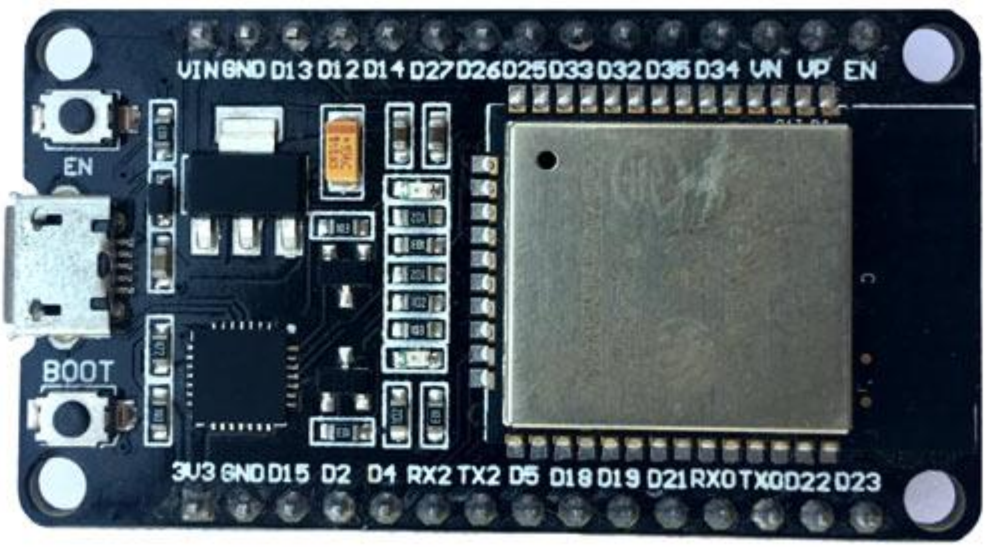


Figure 1.10 – View of the ESP32 module board

They have small form factors and simplify the implementation of functions necessary for the IoT, such as wireless firmware updates OTA technology. The presence of development boards, such as NodeMCU and some other third-party ESP-based boards, allows developers to familiarize themselves with the capabilities of this platform in detail before using it in projects. Just like other platform-based boards Particle.io the ESP8266 boards come with FCC and CE certification to reduce the overall cost of certifying the device after manufacture.

ESP cards provide one of the most reliable, dedicated Wi-Fi interfaces, including several protocols that support IoT technology, for example, the ESP touch protocol, which allows the device to safely and seamlessly access Internet via Wi-Fi networks. The ESP8266 boards are also quite easy to learn and can be used with any microcontrollers to create IoT projects.

4 *Intel IoT development boards*. Intel is one of the largest players in the market of semiconductor elements, so it is not surprising that among their products there are boards with functions that allow using IoT. Although they have discontinued support for some older boards, some of these boards are still used for rapid prototyping by developers of IoT projects. It is not surprising that one of the main features of such boards from Intel is their huge computing capabilities. One of the most popular Intel boards in the IoT is the Intel Edison computing module (Figure 1.11).

The Intel Edison module uses a 22-nm Intel SoC module (system-on-a-chip – system on a chip), which includes a dual-core dual-threaded Intel Atom processor with a frequency of 500 MHz and a 32-bit Intel ® Quark microcontroller operating at a frequency of 100 MHz.

Figure 1.11 – Intel Edison Module

However, this module and most other boards, such as Intel Curie and Intel Galileo, have been discontinued. Currently, the most popular platform for the development of Internet of Things hardware from Intel is the Internet of Things development kit Up Squared groove, which is a platform designed specifically to meet the stringent requirements of industrial Internet of Things applications.

5 *Development boards from Adafruit.* Adafruit is one of the largest online stores of electronic components. This company also develops libraries for working with the components that it sells (displays, address LED strips, sensors, and much more). Those who have come across the Arduino platform are aware of the important role that Adafruit plays in the Arduino eco-system.

In addition to development boards, as the platform Particle.io (Figure 1.12), Adafruit provides cloud services for devices with client libraries for all major IoT hardware development platforms, powerful APIs, beautiful dashboards and a universal secure IoT platform. The main difference between Adafruit and Particle is how their products are designed. Adafruit.io focused on the developer community. This solution is ideal for prototyping projects. Platform Particle.io, on the other hand, has a more pronounced commercial connotation of its products.

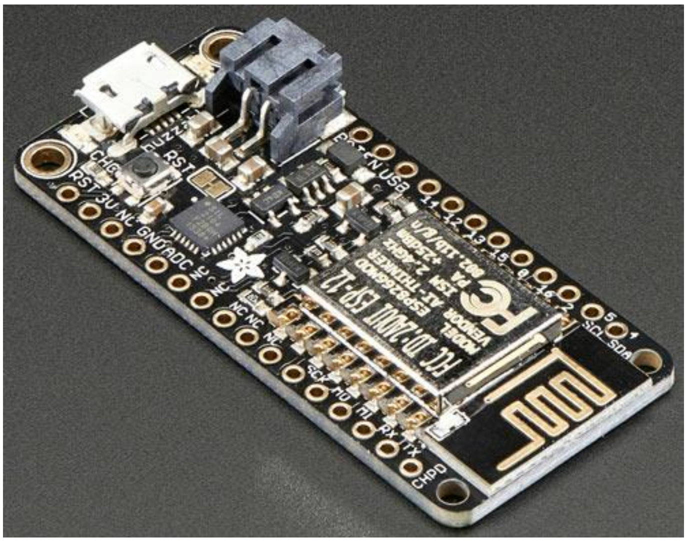


Figure 1.12 – Module view Particle.io

6 *Arduino IoT product line*. Due to the simplicity of programming and ease of integration into various embedded systems, the Arduino board is suitable for everyone who is engaged in the development of various electronic projects.

Early versions of Arduino boards were mainly general-purpose microcontrollers that connected to Internet using GSM and Wi-Fi modules, but as the concept of the IoT began to develop more and more, boards with special functions supporting the Internet of Things began to be developed.

These include boards such as Arduino 101 (developed jointly with Intel), MKR1000, Arduino WiFi Rev 2 and MKR Vidor 4000, which is the first Arduino board based on an FPGA chip, each of these boards was created taking into account IoT technologies, and they all have different functions that make them more suitable for specific solutions, that is, each of these boards is ideal for use in its specific niche.

The Arduino WiFi Rev 2 board comes with an IMU (Inertial Measurement Unit), which makes it convenient for creating drone-based applications (Figure 1.13).

Figure 1.13 – View of the Arduino WiFi Rev 2 board

As well as Adafruit and Particle.io, Arduino also has a cloud service designed to be used by some Arduino boards, including MKR1000, Arduino Yun/Yun Shield and Arduino 101/WiFi Shield 101. Arduino Device Cloud (cloud.arduino.cc) offers manufacturers a simple tool to connect their devices to the Internet and requires a minimally short process settings for accessing the Arduino cloud.

7 *Raspberry Pi.* Although the Raspberry Pi board is more of a general-purpose device, it would be unfair to ignore the contribution of the Raspberry community to the development of a number of products and projects of the IoT. However, Raspberry Pi boards are quite expensive and complex (the exception is the recently appeared Raspberry Pi Pico board) (Figure 1.14) in order for them to be expediently used for simple connection of any sensors or actuators, but they are used as data aggregators, hubs and gateways of devices in IoT projects.

Figure 1.14 – View of the Raspberry Pi Pico board

In order to attract the attention of IoT developers who would like to use Raspberry Pi in their products, Raspberry Pi computing modules were put into production.

The Raspberry Pi 3 computing module (Figure 1.15) contains the insides of the Raspberry Pi 3 board (BCM2837 processor and 1 GB of RAM), as well as a 4 GB eMMC flash device (equivalent to an SD card in Pi) operating at a processor frequency of 1.2 GHz, all integrated on a small board of size 67,6×31 mm, which fits into a standard DDR2 SODIMM connector (the same type of connector that is used for laptop memory). Such functionality makes Raspberry products suitable for use as gateways in projects requiring high data processing speed.

A good feature of all the platforms discussed in this article is their open source code, which means they have good support from the developer community.

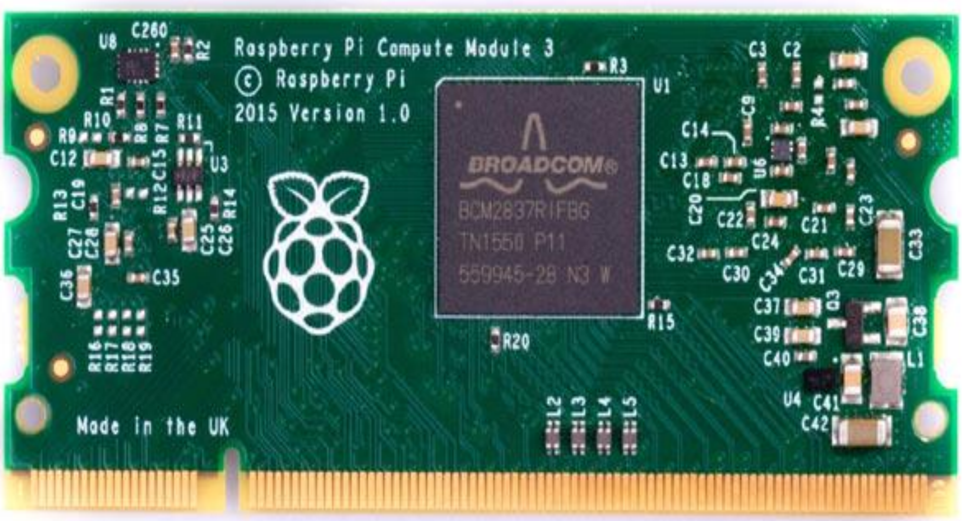


Figure 1.15 – View of the Raspberry Pi 3 board

1.10 Structure of Component Interaction and Models Access in IoT Networks

The interactions of devices with each other within the same sensor network form a mesh topology. A self-organizing network with mesh topology allows to implement the following features [73]:

– creation of zones of continuous information coverage of a large area;

– scalability of the network, that is, an increase in the area of the coverage area and the density of information flows in the self-organization mode;

– use of wireless transport channels for communication in the »everyone with everyone»;

– resistance of the network to the loss of individual elements.

The tasks solved by IoT things require the use of computing resources of the cloud, which can functionally be of two types: cloud computing and fog computing, where the former provide on-demand access to some common computing resources, for example, data networks, servers, storage devices applications and services, the latter, being a kind of cloud services, are implemented by local sensor networks and are located around the «cloud». Such an organization of IoT networks involves the decentralization of data processing by transferring part of the data processing work from the cloud to the computing resources of the fog.

The analysis of infocommunication technologies that ensure the functioning of the IoT networks has shown their heterogeneous nature. The connection of technologies with each other is reflected in the IoT architecture, which includes four functional levels.

The level of interaction with the environment is the lowest level of the IoT architecture, including sensor devices and sensor networks (NU). NU implement the connection of the physical and virtual (digital) worlds, providing the collection and processing of information in real time. Most NU require aggregation via gateways.

The network layer provides transportation of large amounts of data generated by numerous IoT systems at the first level and consists of a network infrastructure that is created by integrating heterogeneous networks into a single network platform.

The service level contains a set of information services that automate technological business operations in IoT: support for operational and business activities, various analytical information processing, data storage, information security, business process management, and others.

The application layer includes various types of applications for the relevant industrial sectors and fields of activity, such as energy, transport, trade, medicine, education and others. Applications can be «vertical» when they are specific to a particular industry, horizontal, that can be used in various sectors of the economy.

One of the issues of the organization of the IoT is the development of methods of interaction between the NU (internet things). In practice, three access methods are implemented: direct, through a gateway, through a server.

In the case of direct internet access, things must have their own IP address or a network alias by which they can be accessed from any client application. The interface with such things is usually made in the form of a web resource with a graphical interface for management via a web browser. It is possible to use specialized software. The disadvantages of this method are obvious:

– the need to have a fixed address on the network, which depends on the Internet service provider of such things. Another way out of the situation is to use an alias (a network alias of an IP address), which requires the internet service to constantly contact a special server with a request to update the network address by an alias;

– the limit of connections to the device is caused by the poor quality of IoT connection, as well as their weak computing resources. This problem is solved by including high-performance equipment in IoT and connecting things to a stable source of internet connection. This makes it necessary for such a thing to consume more energy and often forces such things to be stationary, powered by permanent sources of electricity.

Access to NU through a gateway is a more rational way of organizing interaction and completely replaces the method of direct access, if necessary, the organization of communication of wireless sensor networks or IoT network with the global Internet. Most wireless sensor network standards do not support the IP protocol, using their own communication protocols. This feature makes it necessary to have a device for relaying messages from the sensor network to the Internet for protocol compatibility. The disadvantages of this approach are the same as in the case of direct access, but they extend to the gateway.

Access to NU through the server implies the presence of an intermediary between Internet things and the user. Such an intermediary is a server whose main functions include: receiving messages from IoT and transmitting them to users; storage of received information and its processing; providing a user interface with the possibility of two-way exchange between the user and NU.

This access method is the most rational and frequently used, since it allows to transfer the load of processing user requests from NU to a centralized server, thereby unloading a weak radio communication channel of IoT, transferring the load to wired communication channels between the server and users.

The analysis of IoT technology allowed us to identify the specifics of information interaction in IoT networks and to conclude that it is impossible to apply models and algorithms of traditional computer networks to IoT networks.

These circumstances made it possible to formulate the task of the dissertation research as the task of developing models and algorithms that contribute to the selection of optimal modes of information interaction in IoT networks.

When solving this problem, it is proposed to use modeling tools to evaluate the functional dependence between a set of parameters P that define the Internet of Things network quantitatively and probabilistic-temporal characteristics H, describing information interaction qualitatively, i.e. formally

$$H = f(P).$$

This formulation makes it possible, under different conditions, to solve all three types of modeling tasks necessary for the design of IoT networks:

– direct – evaluate the characteristics with the available IoT network parameters;

– reverse – with the specified permissible values of characteristics, design an IoT work from elements with the appropriate parameters;

– settings – build models based on the proposed methods and algorithms to identify the functional relationship between the sets of *P* and *H*.

The set of *P* parameters includes flow and structural metrics:

– each sensor device is characterized by the intensity of self-identification in the IoT space, periods of activity and passivity;

– each node of IoT things is set by performance parameters;

– the structure of the links corresponds to the mesh topology;

– the elements of the SU-node and node-node connections are set by the time of transmission of the call in the forward and reverse directions; the values of the probabilities of loss of communication.

The network is built as an integration of «foggy» (cluster) and cloud computing, the number of which is theoretically unlimited. Several routers and gateways are located in one cluster, which are connected to the main information channel using an optical or electrical cable or via a radio channel using broadband access systems (Figure 1.16).

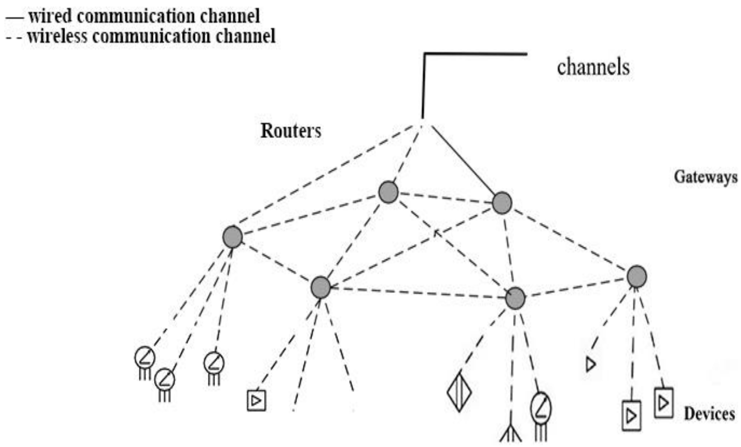


Figure 1.16 – General view of the mesh-cemu structural organization

The gateways in the cluster are connected to each other via a radio channel. Protocols implementing transportation functions in mesh-cemu are based on creating a routing table with monitoring of the state of the transport channel and support for dynamic routing, which allows each of them to send traffic along the optimal route between neighboring nodes. If any of the nodes fails, traffic is automatically redirected to another route, which guarantees not only its delivery, but also delivery in the shortest possible time. In conditions of sudden traffic surges both inside the network and on its borders, the problem can be solved by installing new gateways within the cluster, the integration of which occurs automatically. The increase in coverage areas is reduced to adding new SU and gateways to the network, which also happens automatically, thereby scaling IoT network. It is proposed to use modeling tools to evaluate the probabilistic-temporal characteristics of IoT network. Models should be based on original methods and algorithms that take into account the specifics of information interaction in IoT.

Models for assessing the characteristics of information interaction in IoT are proposed in the work [73]:

1 A simulation model of information interaction in IoT based on a multi-agent approach. The model was created in the AnyLogic simulation system and functions as follows: requests come from network agents stochastically. If an agent passes validation and is certified to receive/transmit information and transmits a request for the first time since it was detected on the network, then it is added to the «map» of IoT network to identify it in space. After processing and storing the information in memory, a control command is sent from the server to the agent, to which the agent must send a notification of receipt.

During the passage of all stages of information interaction, the following situations may arise:

– the agent's request for data transmission is rejected due to collisions of data sources or the route for transmission has not yet been built;

– multiple addition of the same agent to the map. The problem is solved by removing the object from the map if no response is received from it after the specified time;

– «looping» – sending the same command to the agent after disconnecting/ connecting. If the agent cannot execute the transmitted command (expressed in the absence of data that is expected to be received in response), the system will offer to execute this command again until it is executed, or after a certain number of attempts, information about which is stored in memory. The request is successful – the information interaction took place.

2 Access models in fog computing with the resolution of collisions of data sources in the time domain, implementing respectively the modes of polling, interrupts, multiple access.

Each NU is assigned an ID in accordance with its IP address as part of the IoT. Next, for all SU, the generator of random positive integers RPI plays the value of the time t of the start of transmission from the range $[L, R]$

$$t = r \times \Delta T, \tag{1.1}$$

where r – a random number, $r \in$ RPI; ΔT – the sampling interval of the system time of the model.

The central node of the «fog» establishes a connection with the NU for which min (t_i), $i = 1, n$. Let *ID* this NU is *k*. For the rest of the NU the new transmission start time is determined by the formula

$$J_i \, \Delta T = J_{i-1} \Delta \, T + \; p_{ak} \; \Delta T \; + \; r \, \Delta T \, , \tag{1.2}$$

where J_i – the starting point of the next data transfer; J_i – the start point of data transmission of the previous (k – 1)-th NU; p_{ak} – the number of report points of the time required for data transmission – a constant value, depends on the length of the data packet; *r* is a number specifying the number of reference points:

– to determine the time spent on occupying the slot *k*-th NU when implementing the polling mode;

– to determine the access time when implementing interrupt mode;

– to determine the random delay relative to the end of data transmission (k – 1)-m NU when implementing multiple access mode.

Expression (1.2) is the main implementation of the access simulation model in FC.

The proposed simulation models are invariant to the number of sensor devices, the format of the data packet, and the average transmission time.

3 Probabilistic model of establishing information interaction in IoT with mesh topology.

According to the information interaction model, the estimate of the time of information interaction in the IoT network is presented as the sum of

$$t_{ic} = t_{sc} + t_{tc} \, ,$$

where t_{ic} – the time of information interaction; t_{sc} – the time required to establish a connection for information interaction; t_{tc} – the time of data transmission over the established information interaction connection.

To evaluate the t_{sc} a probabilistic model of establishing a connection in the network and NU. It is proposed to evaluate the t_{tc} using the queuing systems apparatus.

The probabilistic model of establishing a connection in an IoT network with a mesh topology allows us to evaluate the absolute and probabilistic characteristics of information interaction. The model takes into account the conditions corresponding to the real process of information interaction – the dynamic status of channels and access points, a limited number of repeated attempts to establish connections, the availability of alternative routes. To estimate the data transmission time, it is proposed to use the Laplace-Stieltjes transformation apparatus (TLC).

In IoT networks, *K* independent data streams of various intensity classes L_k, $k = 1,..., K$ are simultaneously transmitted between its various elements. The duration of data transmission of each class is independent random variables with any kind of distribution я $B_k(i)$, characterized by

$$M_k = C / l_k ,$$

where M_k – the bandwidth of the node point, in packets/s; C – the bandwidth of the node point, in bits/s; l_k – the length of the k-th class data packet, in bits, $l = 1,\ldots, K$.

In accordance with the properties of the TLC, the duration of transmission of k-class data from the source to the addressee is defined as

$$\beta_k(t) = \prod_{d=1}^{N} \beta_d(t), \qquad (1.3)$$

where $B_k(t)$ – the duration of the data packet's stay in the d-th element of the route.

Distribution (1.3) is a function of the intensity vectors of the receipt and maintenance of data packets per route element

$$B_k(t) = f(L_k, M_k).$$

The first central moment of the TLC allows to determine the average data transmission time for the established information interaction. The probabilistic meaning of the TLC allows you to evaluate the probability of data delivery.

Conclusion on Chapter 1

1 Components of the IoT are represented as ABCDE: Analytics, BigData, Connection, Devices, Experience. IoT is based on three basic principles: 1) the ubiquitous communication infrastructure; 2) the global identification of each object; 3) the ability of each object to send and receive data via a personal network or the Internet to which it is connected.

2 The Fog Computing paradigm differs from Cloud Computing in a number of parameters: distribution of computing power and real time; geographical distribution of components; a large amount of external data; complex topology – mobility and heterogeneity. The mobility of devices will require the use of alternative protocols, for example, the LISP routing protocol (Locator/ID Separation Protocol), which allows you to divide the functionality of IP addresses into two parts: host identifiers and routing locators. The concept provides for the installation of tunnel routers that will add LISP headers to information packets as they move through the network.

3 The architecture of the Internet of Things clearly demonstrates how various technologies are interconnected, and includes the following levels: things, gateways, networks, server. At the level of things, the devices themselves are executed directly. At the level of gateways and networks implement a wide range of tasks in IoT, it is necessary to ensure the collaboration of many different technologies and protocols. The service level

(application level) contains a set of information services that provide automation of technological and business operations in IoT, namely, support for operational and business activities; analytical processing of information; interface for storing, entering and outputting data from an external system; application (API); information security; centralized control panel things.

4 There are certain data transfer protocols: Data Distribution Service is a real-time messaging platform between machines in Internet of Things systems. Advanced Message Queuing Protocol provides server-side protocols for servers via peer-to-peer data exchange. The Limited Application Protocol defines protocols for limited devices that use low power consumption and little memory, such as wireless sensors. Message Queue Transport Telemetry (MQTT) is a messaging protocol standard for low-power devices that use TCP/IP for seamless data transmission.

5 IoT cloud platforms are the foundation of IoT network and provide the following functions for devices and software applications: start-up, maintenance, analytical processing, data storage and information security tools. The main IoT platform are: Microft Azure IoT Suit, AWS IoT Soge (Amazon), Google Cloud, IBM, Oracle and others.

6 BigQuery is designed to process huge amounts of data, such as log data from thousands of retail systems or IoT data from millions of vehicle sensors around the world. It is a fully managed and serverless data warehouse that allows to focus on analytics instead of infrastructure management. By design, BigQuery helps to avoid the data storage problem that occurs when you have separate teams in company that have their own independent data storefronts. This can create significant friction between data analysis between teams and cause problems with data version control.

7 MySQL is a relational database management system (RDBMS). Where «relational» means that the data is stored as tables. One of the oldest, since it was written back in the 80s in C and C++ languages. Now it belongs to the company Oracle. MySQL-type systems can be used by any sites or applications to store different data. MySQL has an open source code, that is, you can change it or supplement it to suit your needs.

8 The observation of optimization methods in IoT: the application of a known optimization structure to solve the problem; the development of a new approach based on a heuristic method of solving the problem s for IoT networks are executed. Such optimization algorithms as heuristic, based on particle swarm optimization, genetic, heuristic, evolutionary, based on fuzzy logic, stochastic, mimetic, based on artificial intelligence are discussed.

9 Hardware platforms (bourdes) for the development of IoT networks such as Particle.io, Espressif ESP32, Intel IoT, Adafruit, Arduino and Raspberry Pi where analyzed and discussed.

10 A self-organizing network structure with mesh topology is considered. Models for assessing the characteristics of information interaction in IoT are discussed.

References

1 What is IoT and What Should be Known About it [Electronic resource]. – Access mode : https://habr.com/ru/company/otus/blog/549550. – Date of access : 12.04.2022.

2 What is the «Internet of Things» [Electronic resource]. – Access mode : https://msk.tele2.ru/journal/article/what-is-internet-of-things. – Date of access : 12.04.2022.

3 Trends in Automation of Dairy Enterprises [Electronic resource]. – Access mode : https://dairynews.today/news/tendencii_avtomatizacii_molochny h_predprijatij.html. – Date of access : 12.04.2022.

4 The system of technological monitoring of milk production on the farm [Electronic resource]. – Access mode : https://cyberleninka.ru/article/n/sistema-tehnologicheskogo-monitoringa-proizvodstva-moloka-na-ferme-krs/viewer. – Date of access : 12.04.2022.

5 Internet of Things in Industry [Electronic resource]. – Access mode : https://controlengrussia.com/internet-veshhej/klyuchevy-h-tehnologij. – Date of access : 12.04.2022.

6 Roslyakov, A. V. Internet of Things: textbook / A. V. Roslyakov, S. V. Vanyashin, A. Yu. Grebeshkov. – Samara: PGUTiI, 2015. – 115 p.

7 What is M2M [Electronic resource]. – Access mode : https://mcs.mail.ru/blog/chto-takoe-m2m-i-kak-mashiny-obshchayutsya-drug-s-drugom. – Date of access : 04.18.2022.

8 What is the Internet of Things [Electronic resource]. – Access mode : https://trends.rbc.ru/trends/industry/5db96f769a7947561444f118. – Date of access : 22.04.2022.

9 Introduction to the concept of the Internet of Things [Electronic resource]. – Access mode : https://nag.ru/material/38920. – Date of access : 12.04.2022.

10 Internet of Things or IoT [Electronic resource]. – Access mode : https://janberg.by/internet-veshhej-ili-iot. – Date of access : 04.15.2022.

11 Maturity of the IoT concept [Electronic resource]. – Access mode : https://ozlib.com/950115/tehnika/zrelost_kontseptsii_sostavlyayuschih_tehnologiy. – Date of access : 12.04.2022.

12 In monitoring system decisions [Electronic resource]. – Access mode : https://1-engineer.ru/internet-veshchey-obzor-resheniy-v-sistemakh-monitoringa. – Date of access : 15.04.2022.

13 Protocol MQTT [Electronic resource]. – Access mode : https:// habr.com/ ru/post/463669. – Date of access : 20.04.2022.

14 Network optimizations in the Internet of Things: A review / N. N. Srinidhi [et al.] // Engineering Science and Technology, an International Journal 22, 2019. – P. 1–21.

15 Saati, T. Decision-making. Method of hierarchy analysis / T. Saati. – M. : Radio and Communications, 2003. – 320 p.

16 An immune orthogonal learning particle swarm optimisation algorithm for routing recovery of wireless sensor networks with mobile sink / Y. Hu [et al.] // Decision-making. Method of hierarchy analysis / Int. J. Syst. Sci. 45 (3), 2014. – P. 337–350.

17 Particle Swarm optimization implement ation for minimal transmission power providing a fully-connected cluster for the internet of things / G. L. Da Silva Fré [et al.] // International Workshop on Telecommunications (IWT), 2015. – P. 1–7.

18 Qpso-based energy-aware clustering scheme in the capillary networks for internet of things systems / L. Song [et al.] // IEEE Wireless Communications and Networking Conference (WCNC), 2016. – P. 1–6.

19 Sung, W. T. IoT system environmental monitoring using IPSO weight factor estimation / W. T. Sung, C.-C. Hsu. – Sens. Rev. 33 (3), 2013. – P. 246–256.

20 Multi-objective optimization in cloud brokering systems for connected internet of things / T. Kumrai [et al.] // IEEE Int. Things J. 4 (2), 2017. – P. 404–413.

21 Dhumane, A. V. An optimal routing algorithm for internet of things enabling technologies / A. V. Dhumane, R. S. Prasad , J. R. Prasad. – Int. J. Rough Sets Data Anal. (IJRSDA) 4 (3). 2017. – P. 1–16.

22 Gacn: self-clustering genetic algorithm for constrained networks / J. Martins [et al.] // IEEE Commun. Lett. 21 (3), 2017. – P. 628–631.

23 A genetic algorithm-based solution for efficient in-network sensor data annotation in virtualized wireless sensor networks / I. Khan [et al.] // 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2016. – P. 321–322.

24 A fast and elitist multi-objective genetic algorithm: NSGA-ii / K. Deb [et al.] // IEEE Trans. Evol. Comput, 6 (2), 2002. – P. 182–197.

25 Adapting NSGA-ii for hierarchical sensor networks in the IoT / A. Rodriguez [et al.] // Procedia Comput. Sci. 61, 2015. – P. 355–360.

26 QoS-Aware Energy-Efficient Cooperative Scheme for Cluster-Based IoT Systems / L. Song [et al.] // IEEE Syst. J. 2017. – P. 25–32.

27 Rodriguez, A. Energy optimization in wireless sensor networks based on genetic algorithms / A. Rodriguez, P. Falcarin, A. Ordonez // SAI Intelligent Systems Conference (IntelliSys), 2015. – P. 470–474.

28 A scenario based heuristic for the robust shortest path tree problem / I. A. Carvalho [et al.] // IFAC-PapersOnLine, 49 (12), 2016. – P. 443–448.

29 Verma, A. Computational intelligence based heuristic approach for maximizing energy efficiency in internet of things / A. Verma, S.Kaushal, A. K. Sangaiah // Intelligent Decision Support Systems for Sustainable Computing, 2017. – P. 53–76.

30 Dhondge, K. Hola: Heuristic and opportunistic link selection algorithm for energy efficiency in industrial internet of things (IIoT) systems / K. Dhondge, S. Shorey, J. Tew // 8th International Conference on Communication Systems and Networks (COMSNETS), 2016. – P. 1–6.

31 Power efficient RACH mechanism for dense IoT deployment / S. Shailendra [et al.] // IEEE International Conference on Communications Workshops (ICC Workshops), 2017. – P. 373–378.

32 Ismail, N. H. A. 6LoWpan local repair using bio inspired artificial bee colony routing protocol / N. H. A. Ismail, R. Hassan // Procedia Technol. (ICCEI-2013), 11, 2013. – P. 281–287.

33 Huang, J. A novel bio-inspired multi-objective optimization algorithm for designing wireless sensor networks in the internet of things / J. Huang, J. Sens. – 2015. – P. 1–16.

34 Hive oversight for network intrusion early warning using diamond: a Bee-inspired method for fully distributed cyber defense / M. Korczynski [at al.] // IEEE Commun. Mag, 54 (6). 2016. – P. 60–67.

35 Raz, N. R. A Bio-Inspired model for emergence of cooperation among nanothings / N. R. Raz, M.-R. Akbarzadeh // Iranian Conference on Intelligent Systems (ICIS), 2014. – P. 1–6.

36 Reddy, P. K. An evolutionary secure energy efficient routing protocol in internet of things / P. K. Reddy, R. Babu // Int. J. Intell. Eng. Syst, 10 (3), 2017. – P. 337–346.

37 An Evolutionary Game Approach on IoT service selection for balancing device energy consumption / J. Na [et al.] // in IEEE 12th International Conference on e-Business Engineering, 2015. – P. 331–338.

38 Hamza, K. S. Evolutionary clustering for integrated WSN-RFID networks / K. S. Hamza, F. Amir // 10th International Conference on Informatics and Systems, 2016. – P. 267–272.

39 Zhang, B. Y. Data classification in internet of things based on evolutionary neural network / B. Y. Zhang. – Adv. Mater. Res. 659. 2013. – P. 202–207.

40 Abnormal traffic detection in networks of the internet of things based on fuzzy logical inference / S. Ageev [et al.] // XVIII International Conference on Soft Computing and Measurements (SCM), 2015. – P. 5–8.

41 Variable-categorized clustering algorithm using fuzzy logic for internet of things local networks / J. H. Kwon [at al.] // Multimedia Tools Appl, 2017. – P. 1–20.

42 Choi, J.-Y. Design and performance analysis of cost-optimized handoff scheme based on fuzzy logic for building smart car IoT applications / J.-Y. Choi, J. Jeong // Int. Inf. Inst. (Tokyo), 18 (10), 2015. – P. 339.

43 Fully secure fuzzy identity-based encryption for secure IoT communications / Y. Mao [et al.] // Comput. Stand. Interfaces, 2016. – P. 117–121.

44 Fuzzy Comprehensive Evaluation Method for Energy Management Systems Based on an Internet of Things / Y. Li [et al.] // IEEE Access, Volume: 5, 2017. – P. 312–322.

45 Ali, Z. A stochastic routing algorithm for distributed IoT with UnReliable wireless links / Z. Ali, J.-Y. Abbas, F. Y. Li // IEEE 83rd Vehicular Technology Conference (VTC Spring), 2016. – P. 1–5.

46 Random access analysis for massive IoT networks under a new spatio-temporal model: a stochastic geometry approach / N. Jiang [et al.] // IEEE Transactions on Communications. Volume: 66, Issue: 11, November 2018. – P. 5788–5803.

47 Kaur, R. Security in IoT network based on stochastic game net model / R. Kaur, N. Kaur, S. K. Sood // Int. J. Network Management, 2017. – P. 11–19.

48 Spatio-temporal stochastic modeling of IoT enabled cellular networks: scalability and stability analysis / M. Gharbieh [et al.] // IEEE Trans. On Commun., Vol. 65, 2017. – P. 3588–3600.

49 Kuppusamy, P. Novel authentication based frame work for smart transportation using IoT and memetic algorithm / P. Kuppusamy, B. Kalaavathi // Asian J. Res. Soc. Sci Humanities 6 (10), 2016. – P. 674–690.

50 Kus, W. Memetic inverse problem solution in cyber-physical systems / W. Kus, W. Mucha // Adv. Tech. Diagn., 2016. – P. 335–341.

51 Sun, F. Bayesian networks for intrusion dependency analysis in water controlling systems / F. Sun, C. Wu, D. Sheng // J. Inform. Sci. Eng. 33, 4, 2016. – P. 1249–1263.

52 Long, N. T. Research on applying hierarchical clustered based routing technique using artificial intelligence algorithms for quality of service of service based routing, internet of things and cloud computing / N. T. Long, N. D. Thuy, N. D. Hoang // Spec. Issue Qual. Serv. Serv. Based Routing 3 (6–1), 2015. – P. 1–8.

53 Tunc, C. Markov fluid queue model of an energy harvesting IoT device with adaptive sensing / C. Tunc, N. Akar // Perform. Eval. 111, 2017. – P. 1–16.

54 A game theoretic context-based routing protocol for opportunistic networks in an IoT scenario / S. J. Borah [et al.] // Comput. Netw., 129 (2), 2017. – P. 572–584.

55 Energy-centered and QoS-aware services selection for internet of things / M. E. Khanouche [et al.] // IEEE Trans. Autom. Sci. Eng. 13 (3), 2016. – P. 1256–1269.

56 Vishnyakou, U. A. Means and examples of the development of Internet of Things systems / U. A. Vishnyakou. – Bulletin of Communications, 2022. – № 5. – P. 54–57.

57 Zhang, Y. Study on the Fire IoT Development Strategy / Y. Zhang, J. Yu // Procedia Engineering. – 2013. Vol. 52. – P. 314–319.

58 Prehoferand, C. From Internet of things mashups to model-based development / C. Prehoferand, L. Chiarabini // IEEE 39th Annual Computer Software and Applications Conference. – September 2015. – P. 499–504.

59 Fleurey, F. ThingML: A Generative Approach to Engineer Heterogeneous and Distributed Systems / F. Fleurey // IEEE International Conference on Software Architecture Workshops (ICSAW). – April 2017. – P. 185–188.

60 Dias, J. State of the Software Development Life-Cycle for the Internet-of-Things / J. Dias, H. Ferreira // 2019 [Electronic resource]. – Access mode : https://arxiv.org/pdf/1811.04159. – Date of access : 25.12. 2021.

61 Mezghani, E. A Model-Driven Methodology for the Design of Autonomic and Cognitive IoT-Based Systems: Application to Healthcare / E. Mezghani, E. Expósito, K. Drira // IEEE Transactions on Emerging Topics in Computational Intelligence. – June 2017. Vol. 1 (3). – P. 224–234.

62 Guinard, D. A resource oriented architecture for the web of things / D. Guinard, V. Trifa, E. Wilde // Internet of Things (IOT). – November – December 2010. – P. 1–8.

63 Blackstock, M. IoT mashups with the WoTKit / M. Blackstock, R. Lea // 3rd International Conference on the Internet of Things (IOT). – October 2012. – P. 159–166.

64 Pintus, A. Paraimpu: a platform for a social web f things / A. Pintus, D. Carboni, A. Piras // 21st International conference companion on World Wide Web. ACM. – April 2012. – P. 401–404.

65 Rule, A. Exploration and Explanation in Computational Notebooks / A. Rule, A. Tabard, J. Hollan // CHI Conference on Human Factors in Computing Systems – CHI ’18. – April 2018. – P. 1–12.

66 Geber, A. Simplify the development of your IoT solutions with IoT architectures / A. Geber, S. Cansal // IBM Developer. – August 2017. – P. 3–8.

67 Burkit, F. A Strategist’s Guide to the Internet of Things / F. Burkit // 2014 [Electronic resource]. – Access mode : https://www.strategy-business.com/article/00294?gko=a9303. – Date of access : 27.12. 2021.

68 Lee, J. How to Choose the Right IoT Platform: The Ultimate Checklist / J. Lee // Apr., 2018. [Electronic resource]. – Access mode : https://hackernoon.com/how-to-choose-the-right-iot-platform-the-ultimate-checklist-47b5575d4e20. – Date of access : 30.12. 2021.

69 Top 10 selection criteria to choose your IoT platform [Electronic resource]. – Access mode : https://iotify.io/top-10-selection-criteria-for-your-iot-cloud-platform. – Date of access : 29.12. 2021.

70 Internet of Things in Russia 2021 [Electronic resource]. − Access mode : https://www.cnews.ru/projects/2021/internet_of_things. − Date of access: 15.01.2022.

71 Top hardware platforms for Internet of Things (IoT) projects 2021 [Electronic resource]. – Access mode : https://microkontroller.ru/esp8266-projects/top-apparatnyh-platform-dlya-proektov-interneta-veshhej-IOT. − Date of access : 09.15.2022.

72 Elizarov, M. A. Models and algorithms of information interaction in the Internet of Things networks : abstract of the dissertation of cand. of techn. Scienc. : 05.13.01 / M. A. Elizarov ; SPb. state univerxity. − St. Petersburg, 2017. − 18 p.

2 MODEL, STRUCTURE AND DEVELOPMENT OF IoT NETWORK FOR PRODUCTION QUALITY MONITORING

2.1 Model and Structure of IoT Network for Product Quality Monitoring

In the livestock complex of the agro-industrial sector of agriculture, the trend of introducing modern technologies is coming, opening up the IoT for quality control of dairy products, which makes it easier for farms to track the quality of milk and optimize processes on a dairy farm. This means that it is possible to track the health and condition of cows using sensors, information from which is sent online to the computers of the dairy complex. In this section, we will talk about automation of work in the field of agriculture, namely, quality control of dairy products, which will correspond to the level of automation 4.0 [1–12].

International experience shows that the use of new technologies increases the efficiency of farms by 15–25 %. In the Republic of Belarus, the introduction of innovations is only gaining momentum. The share of domestic enterprises that use modern technologies is about 10 %, but their number may multiply in the next 3–5 years.

A multi-agent approach will be used to line up the IoT network model for monitoring milk quality (MMQ). For this purpose, there are many agents of milk quality sensors, storage of quality indicators, converters, agents of processing milk quality indicators to obtain an analysis of the conclusion, agents of monitoring these indicators and conclusions [13]:

$$\mathrm{IoT}_{ccm} = \{PA_m, SC, CP, MA_i\}, \tag{2.1}$$

where IoT_{ccm} – the IoT network model; RA_m – multiple sensors (from milk quality analyzers on farms); SC – a set of converter agents (network gateways of converters); CP – a cloud platform for storing milk quality indicators (quality indicator storage agents and processing agents); MA_i – monitoring agents (mobile devices for monitoring milk quality indicators). Such model can be presented in the following form (Figure 2.1) [1–3]. This model includes portable milk quality analyzers from each of the controlled farms – A_1–A_n. As a rule, analyzers send the received data to a computer or printer via a serial port, which corresponds to the automation 3.0 level. In the given structure, these indicators are sent by microcontrollers to the gateway, which is necessary for converting and transmitting to the cloud platform (CP) the identified milk quality indicators [14].

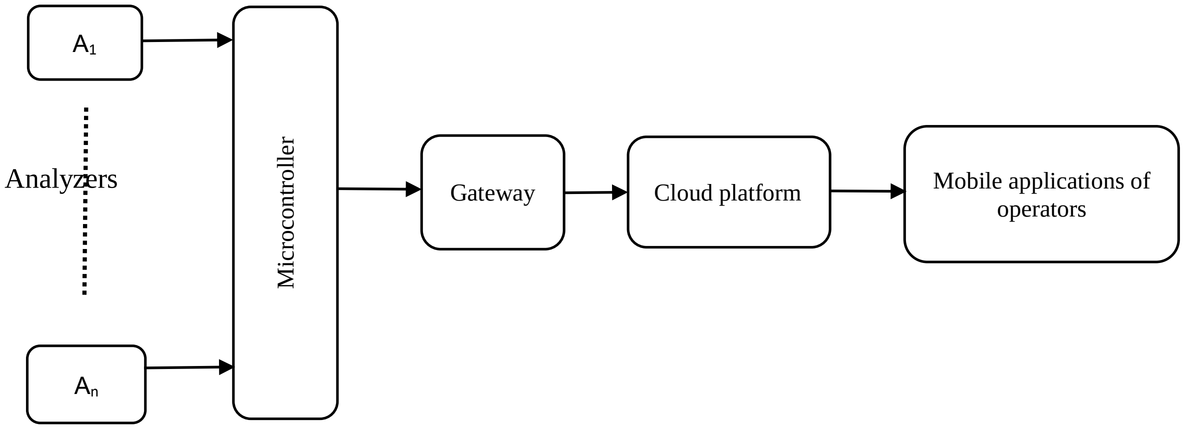


Figure 2.1 – The model of the IoT MQM network

The milk analyzer is a device for determining the characteristics of the quality of milk and products based on it (fat fraction, density, acidity of samples, lactose, sample temperature, and others). This device ensures the environmental friendliness of the studied component, the indicators are read without reference to chemical reagents. Analyzers are used on farms, as well as food industry enterprises, milk reception points and during research work in the field of food industry. Such milk analyzers have high accuracy. This characteristic allows you to make a sufficient number of measurements in the required time. Currently, the use of analyzers in farms and dairy processing plants is gaining popularity. The maximum set of measured parameters is as follows: fat; protein; dry matter; density; lactose; added water; sample temperature; freezing point; salts; pH, conductivity [8, 19].

In the Republic of Belarus there are milk analyzers of own production: ultrasonic analyzer Lactan 1–4 Mini version; Clover-2.

Lactane 1–4 Mini version is able to determine the mass fraction of fat, SOMO, added water and density in milk samples of various states: whole, fresh, canned, pasteurized, normalized and skimmed milk. The time interval of measurement, on average, is 3 minutes, which is twice as fast as using the traditional method of analyzing indicators, safer, more economical and requires less human involvement in the process. The accuracy of determining the parameters of milk quality at the same time fully meets the requirements of standard methods. The analyzer allows you to determine the six most important parameters in 180 seconds without the use of chemical reagents, such as: protein, fat, SOMO, density, temperature and mass fraction of added water in samples of milk of various types: whole fresh, canned, pasteurized normalized, skimmed, reconstituted milk and long-term storage milk.

Clover-2 provides an express assessment of the percentage of fat, protein, skimmed milk powder and density in one sample of fresh whole, canned milk or cream. Despite the presence of a variety of types of measurements, unskilled personnel can also carry out their verification, this is due to the fact that the Clover-2 milk quality analyzer is easy to use for any operator. The principle of operation of the device is to pour a sample of milk for measurement and drain it after measurement. Selecting a sample and pouring it into the sample receiver is a quick process that takes only a few seconds. The process of measuring the quality of milk or cream takes 2.5−3.5 minutes. Milk at room temperature is measured in 2.5 minutes, and chilled − in 3.5 minutes. The operator observes all the necessary information on the indicator of the device. The measurement results are indicated in digital form with a sampling rate of 0.01 %.

The characteristics of analyzers from Bulgaria of the Lactoscan series can be cited as imported analogues. They can be used to measure fat, solids, density, protein, lactose, salts, water content, temperature, freezing point, pH, conductivity, as well as the total solids content of the same sample immediately after milking, collection and during processing; somatic cell counters for the detection of clinical and subclinical mastitis; temperature-regulating devices for various types of tests; highly sensitive test strips for detecting adulteration of neutralizers, hydrogen peroxide and urea adulteration of raw milk, which work in an efficient and reliable way. With high accuracy and speed, Lactoscan portable ultrasonic milk analyzers are competitive with Foss Electric, Delta Instruments and Bentley milk analyzers, which have a much higher price, which is their direct advantage. Minimal energy consumption and the absence of consumables make the Lactoscan milk analyzer an attractive option for the dairy industry. The low operating costs of the costs and the low price make the Lactoscan milk analyzer suitable for dairy farms, dairy enterprises, milk sampling centers and laboratories.

Ekomilk is a model range of ultrasonic milk quality analyzers of Bulgarian production. Devices of this series have the following additional features: connecting a pH electrode to measure the activity of hydrogen ions in the test, monitoring the falsification of the whole milk under study by the conductivity parameter, correcting the calibration of the analyzer by introducing correction values. Ekomilk analyzers are equipped with a self-diagnosis system with the output of the corresponding errors on the display, have an RS-232 connector for connection to a personal computer and the ability to connect a compact thermal printer.

Analyzers can use different interfaces and protocols, which makes it difficult to connect and poll them with infrastructure tools in a cloud environment. In our case, the information from the analyzers can come through parallel or serial ports. Therefore, gateways-converters are needed to interact with the equipment of the cloud platform. Let's consider possible solutions. One of them represents equipment from MOXA, which has been creating communication solutions for more than 30 years and uses its best practices in the field of Ethernet to COM port converters. She has developed a solution for connecting devices with a COM port to cloud environments. MOXA offers solutions for connecting devices directly to the cloud environment. Let's consider some such converters.

The converters of the NPort IA5000A-I/O, the NPort IAW5000A series and the MGate 5105-MB-EIP gateway support integration with Alibaba Cloud IoT Platform, Azure IoT Hub or with a private cloud via the MQTT protocol. The NPort IA5000A-I/O and NPort IAW5000A-I/O converters allow you to transmit not only raw data from the COM port, but also to control the built-in discrete I/O channels. Data is transmitted over the MQTT protocol in JSON format. In IoT, there is no single universal protocol for the integration of physical objects. Therefore, to create a network of physical devices, it is necessary to purchase all components from the same manfacturer [15].

For the Internet of Things, technologies of wireless networks with low power consumption of different ranges are used. That is, the key factor for choosing a network connection standard is the range of the network and how much energy it consumes. Data transmission protocols are the basis of any radio communication. It depends on them what kind of network topology, routing, addressing, data security will be.

Internet of Things systems use [6, 7]:

− local and personal networks (WLAN − Wireless Local Area Network, and WPAN − Wireless Personal Area Network). This includes short-range (small and medium-range) networks, protocols such as Wi-Fi, 6LoWPAN, Thread, ZigBee IP, Z-Wave, ZigBee, BLE 4.2 (Bluetooth Mesh), WirelessHART, MiWi;

− energy-efficient global networks LPWAN (Low-power Wide Area Network). This includes technologies for transmitting small data over long distances: LoRaWAN, SIGFOX, CIoT, 4G LTE, 5G, NB-IoT and some others.

Let's take a closer look at LPWAN (Low-power Wide-area Network), a wireless technology with ultra-low power consumption. It is one of the options for the implementation of autonomous wireless devices operating in energy-efficient long-range networks, recommended for this Internet of Things product quality monitoring network. LPWAN is a new type of wireless networks designed to transmit telemetry data from various devices over long distances for applications that are undemanding to data transfer speeds. The principle of operation of LPWAN is similar to cellular networks. LPWAN uses the «star» topology, where each device directly transmits data over the radio channel to the base station. The station receives signals from all devices within its radius of operation and retransmits the received data to the server using an available communication channel: Ethernet, cellular communication, satellite communication. The server processes and archives data, and also provides data to users. To ensure global access, users receive data via the Internet or via cellular communication. At the moment, there is no single standard for data transmission in LPWAN networks, and various manufacturers offer their own solutions, the characteristics of which differ significantly in individual parameters. To provide users with access to LPWAN networks, manufacturers, by analogy with cellular operators, deploy their own networks and provide services for connecting to these networks. Users install devices equipped with radio modems that support the appropriate wireless communication standard, connect these devices to the operator's network and Date of access from these devices.

Wireless sensors operating in LPWAN networks are, in fact, classical analog sensors and/or discrete sensors coupled with a measuring converter that converts measurements into physical quantities of controlled parameters and digitizes them. The wireless sensors also include a radio modem for working in LPWAN networks and a battery pack for powering the device. If necessary, the devices can be equipped with GPS/GLONASS modules or other global positioing systems (Figure 2.2) [16].

The difference from the classic solutions is that the device is in ultra-low power consumption mode most of the time in the so-called «sleep» mode. This mode, and is activated for a short period of time, which is literally a few seconds. In active mode, the device receives data from the connected sensors, performs the necessary self-diagnosis procedures, including, in particular, measuring the remaining battery charge level and transmits the received data to the base station.

Depending on the frequency of transmission of telemetry information, due to the long periods when the device is in «sleep» mode and insignificant periods of activity, wireless sensors can operate on a single battery for a significant period of time. Some manufacturers claim that the duration of operation of such devices on a single battery with a survey period once an hour reaches 10 years. In fact, any existing sensors can be used as an analog sensor in LPWAN devices. However, often, the use of standard sensors that are not adapted to the harsh conditions imposed by the ultra-low power consumption mode can lead to a significant reduction in battery life. Therefore, manufacturers offer their own solutions that are fully optimized for low energy consumption of sensors. At the moment, manufacturers offer a large set of such wireless sensors that allow you to control a wide range of parameters:

temperature, humidity, pressure, concentration of various substances, wind speed and direction, precipitation, the level of various types of radiation, etc.

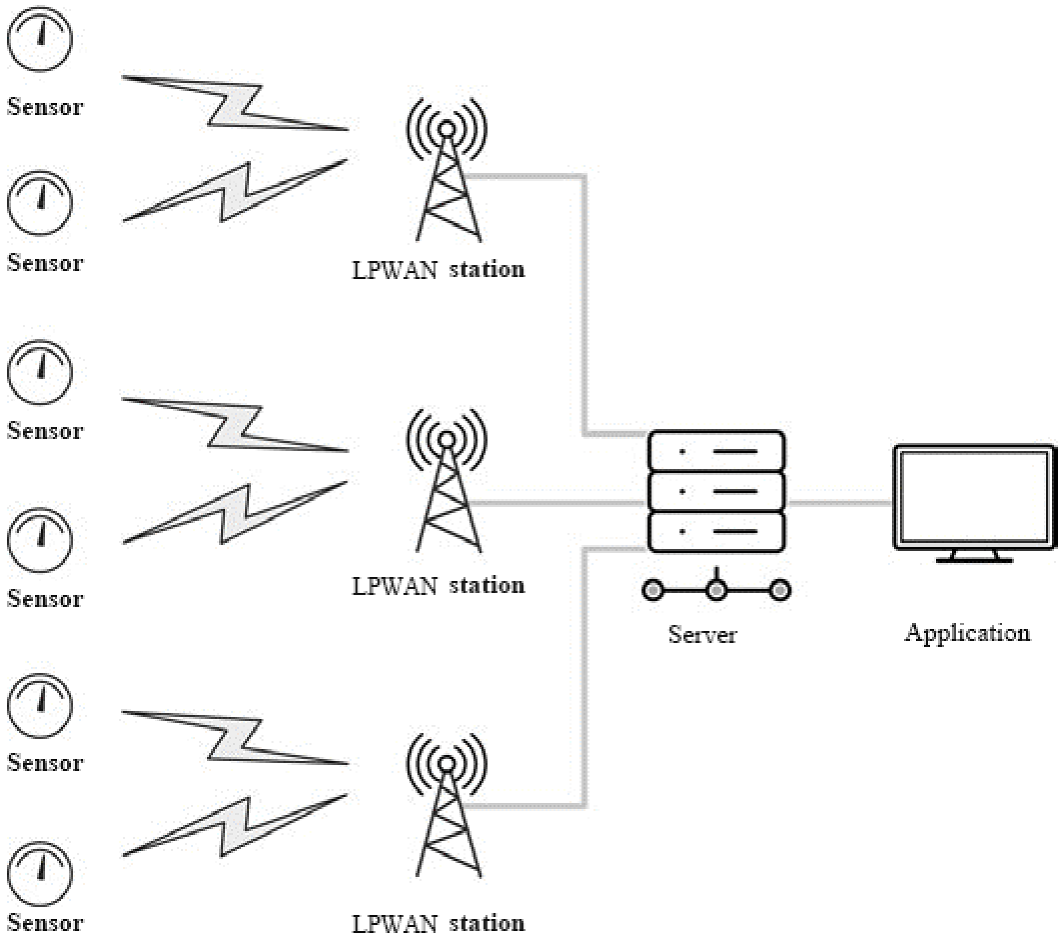


Figure 2.2 – Architecture of LPWAN networks

2.2 The Structure of IoT Network Based on a Cloud Platform

This section discusses the structure of the Internet based on the Google cloud platform. Google Cloud Platform (GCP) is a set of cloud services provided by Google that run on the same infrastructure that Google uses for its end – user products, such as Google Search and YouTube. In addition to management tools, a number of modular cloud services are also provided, such as cloud computing, data storage, data analysis and machine learning. The infrastructure for working with the Internet of Things (IoT) is also based on it [8, 9].

Cloud IoT Core is a fully managed service that allows you to quickly and securely connect, configure and receive data from a large number of devices. Using Cloud Pub/Sub, the Core can combine data from decentralized devices into a single global system. In combination with other Google Cloud services, IoT Core offers a comprehensive solution for collecting, analyzing and visualizing IoT data in real time. This, in turn, allows you to create multifunctional models that optimize or predict data (Figure 2.3) [10].

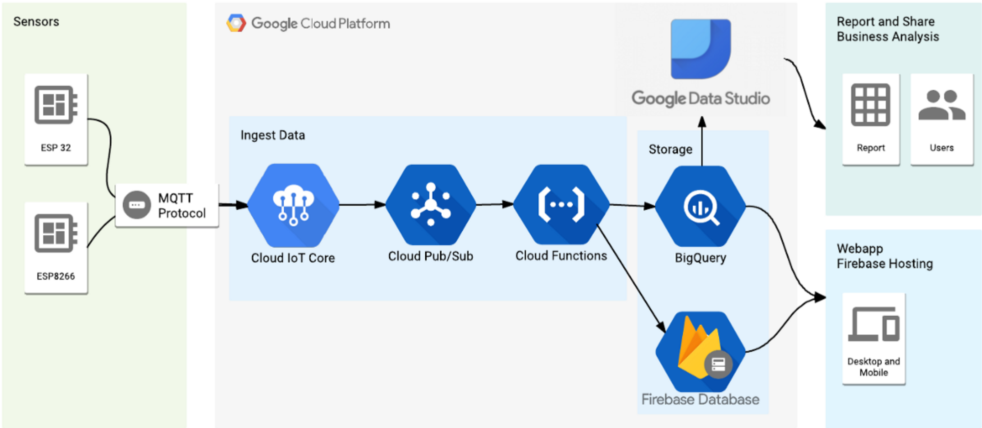


Figure 2.3 – Google cloud platform block diagram

Cloud IoT Core supports the usual MQTT and HTTP protocols, which allows customers to use many existing devices to create their IoT systems. This service works in Google's infrastructure, which automatically scales in real time, providing, if necessary, the ability to connect a million devices.

Cloud Pub/Sub is an asynchronous messaging service that separates the services that create events from the services that process events.

Google Cloud Functions is a runtime environment for creating and connecting cloud services. With the help of cloud functions, you can write simple single-purpose functions that are attached to events originating from the cloud infrastructure and services. The function is triggered when the monitored event is triggered. The code runs in a fully managed environment. There is no need to create any infrastructure or worry about managing any servers.

Firebase is a platform for developing mobile and web applications. It provides developers with a variety of tools and services that help them develop high-quality applications, expand their user base and earn more profit.

After the product was acquired by Google in 2014, Firebase quickly turned into a multifunctional giant – a platform for developing mobile and web applications, the status of which it retains to this day.

In this structure of the Internet of Things for milk quality control, smartphones act as sensors (Figure 2.4). Data such as acidity, temperature, density, freezing point, bacterial contamination and somatic cells are recorded. Then this data is processed on the server and given to the administrator. The data transmission technology used is LPWAN, since wireless sensors operating in networks are analog sensors and/or discrete sensors coupled with a measuring converter that converts measurements into physical quantities of controlled parameters and digitizes them. Data is transmitted over the MQTT protocol in JSON format.

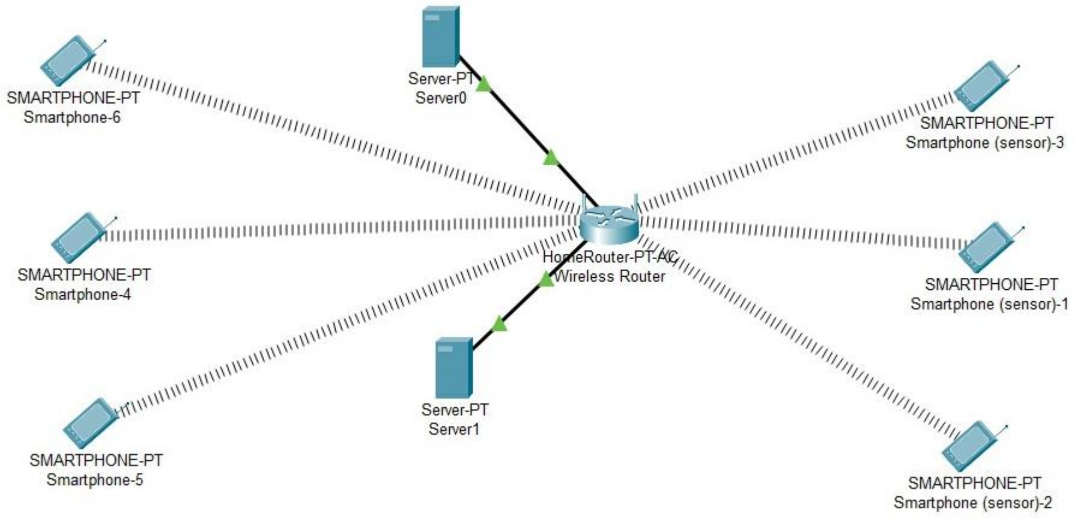


Figure 2.4 – Structure of the IoT network

MQTT is a publisher-subscriber (pub/sub) messaging protocol. The original version was published in 1999 by Andy Stanford-Clark of IBM and Arlen Nipper of Cirrus Link. They viewed MQTT as a way to maintain communication between machines in networks with limited bandwidth or unpredictable connectivity. Thus, the MQTT protocol is needed for streaming data between devices with limited CPU power and/or battery life, as well as for networks with expensive or low bandwidth, unpredictable stability or high latency. That is why MQTT is known as the ideal transport for IoT. It is built on the TCP/IP protocol, but there is a branch MQTT-SN to work over Bluetooth, UDP, ZigBee and in other IoT networks other than TCP/IP.

The principle of operation of the protocol is as follows: a communication system built on MQTT consists of a publisher server, a broker server and one or more clients. The publisher does not require any settings for the number or location of subscribers receiving messages. In addition, subscribers do not need to configure for a specific publisher. The system may have several brokers distributing messages (Figure 2.5) [17]. MQTT provides a way to create a hierarchy of communication channels. Whenever the publisher has new data to distribute to customers, the message is accompanied by a delivery control note.

Higher-level clients can receive each message, while lower-level clients can receive messages related to only one or two basic channels «branching off» at the bottom of the hierarchy. This facilitates the exchange of information ranging in size from two bytes to 256 megabytes.

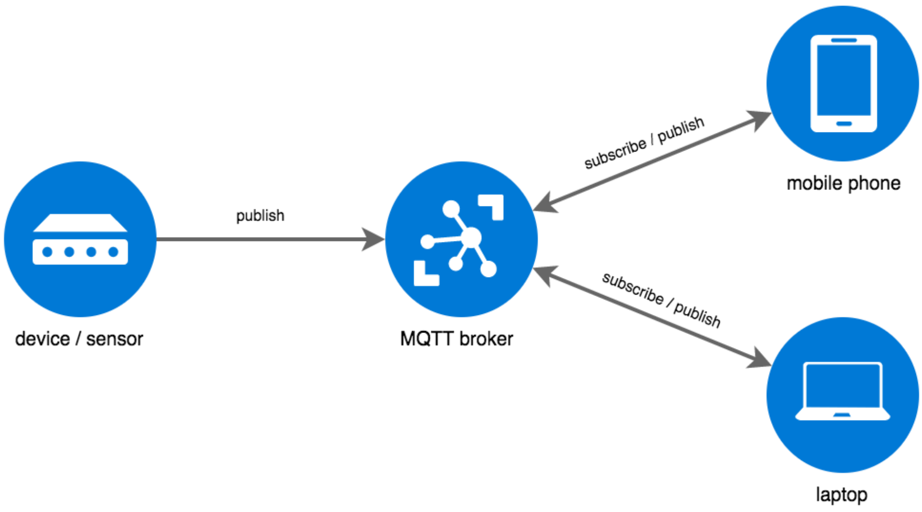


Figure 2.5 – The principle of operation of the MQTT protocol

2.3 The Algorithm of IoT Network for Product Quality Monitoring

In this section, we will consider the algorithm of the Internet of Things for monitoring the quality of milk. To do this, you need to develop an application that consists of a sensor analyzer (mobile application), a Firebase cloud platform on which data is stored, and a client application.

The application for the sensor should look like this: when you enter the application, the main page displays a table with the following milk data: acidity, temperature, density, freezing point, bacterial contamination, somatic cells. The data depends on the type of milk [9].

In a cloud environment, a server containing databases, an API, and a client application is involved. The database stores the data received from farms, the recorded characteristics of milk. Then the data is sent to the server using an API request, which compares the current milk indicators with the range of indicators of this type of milk and this data is stored in the database [14].

The client application reflects on the screen the recorded and received milk indicators through the mobile application of the operator-administrator. On each of the mobile devices with the application installed, displaying information about monitoring the quality of milk from the database, you can see the removed indicators, also daily at 12:00 a newsletter with current information about the quality characteristics of milk arrives to the email addresses of registered users. The algorithm for creating IoTs like this:

1 Sensors measure the parameters of processes or devices that interact with the IoT platform using development tools (SDK).

2 Devices send messages that are verified by the authentication and authorization service of the platform. In case of unsuccessful verification, correction of device IDs is required.

3 Information from devices is sent to the gateway (Device Gateway), various network protocols can be used. Being transformed in the gateway, the information arrives at the processing unit, where communication with analytics is carried out and in parallel to the device storage unit (Device Shadows).

4 Device Shadows stores the current states of network peripherals for constant access to software applications. If there is no connection with a separate device on the network, the Device Shadow block executes commands from applications, and when the connection is restored, it synchronizes the current state with the device.

5 The rule handler, depending on the nature of the incoming data, performs the following actions: saves data in a database, sends information via SMS or e-mail to the network manager about their receipt, calls the HTTP API, sends data to the analytics system, etc.

6 Applications use this data to monitor and manage devices using the API (application interface).

7 Information about all devices is stored on the IoT platform [12].

The application page is presented in the form of a table with milk quality indicators, the values should not exceed the critical ones (Table 2.1).

Table 2.1 – Critical data of product quality control

Indicators/Quality	Top grade	First grade	Second grade	«Extra»
Acidity, °T	16 to 18	From 16 to 18	16 to 20	16 to 18
Freezing point, °C	0.520–0.515	0.515–0.510	More 0.510	Less 0.520
Temperature, °C	4–6	4–6	6–8	Less 4
Density, kg/м^3	More 1028	More 1027	Less 1026	More 1028
Bacterial contamination, thous/cm^3	100,1–300	300,1–500	500,1–4000	Before 100
Somatic cells, thous/cm^3	300,1–500	300,1–750	750,1–1000	Before 300

Data is sent to the cloud at the click of a button, as well as automatically every day at 12:00. The cloud processes the data (compares it with the maximum allowable). The cloud also stores a database with information about the users of the application, about the operators and about the data received from the sensor. And, depending on the result of the comparison, messages are sent.

2.4 Development of the Using Case and Class Diagrams

The use case diagram is a diagram that describes which functionality of the software system being developed is available to each group of users. This diagram is used in the design of software systems. The actors and use cases and the relationships between them are displayed. The use case diagram acts as a conceptual model of the system in the process of its development and design [15].

The purpose is as follows: the designed software system is presented in the form of so-called use cases, with which the interaction of external entities is shown. The actor can be any object, subject, or system that interacts externally with the simulated software system, for example, a person, a technical device, a program, or any other system acting according to the developer's idea. The use cases determine the set of actions that are performed by the system when interacting with the actor [18].

The purpose of the use case is to fix a certain fragment of the behavior of the designed system without specifying the specifics of the implementation of this functionality. In this sense, each use case corresponds to a separate service provided by the simulated system at the request of the actor, namely, it defines one of the use cases. The final sequence of actions should be indicated. It means that after the end of processing the actor's request by the system, there should be a return to the initial state of readiness to execute other subsequent requests. The use case diagram contains a finite set of use cases that define all possible aspects of the expected behavior of the system. The use of use cases at all stages of work on the project allows not only to achieve the required level of unification of designations to represent the functionality of subsystems and the system as a whole, but also is a powerful means of consistently clarifying the requirements for the designed system based on their iterative discussion with all interested specialists.

There are various relationships between the elements of the diagram that describe the interaction of instances of some actors and use cases with instances of other actors and options. A single actor can interact with multiple use cases.

Figure 2.6 shows a diagram of use cases for the Internet of Things network being developed. This use case diagram shows the following actors: administrator, supplier (farm) and consumer. Each of them has the opportunity to register or log in to the system, as well as view the email newsletter.

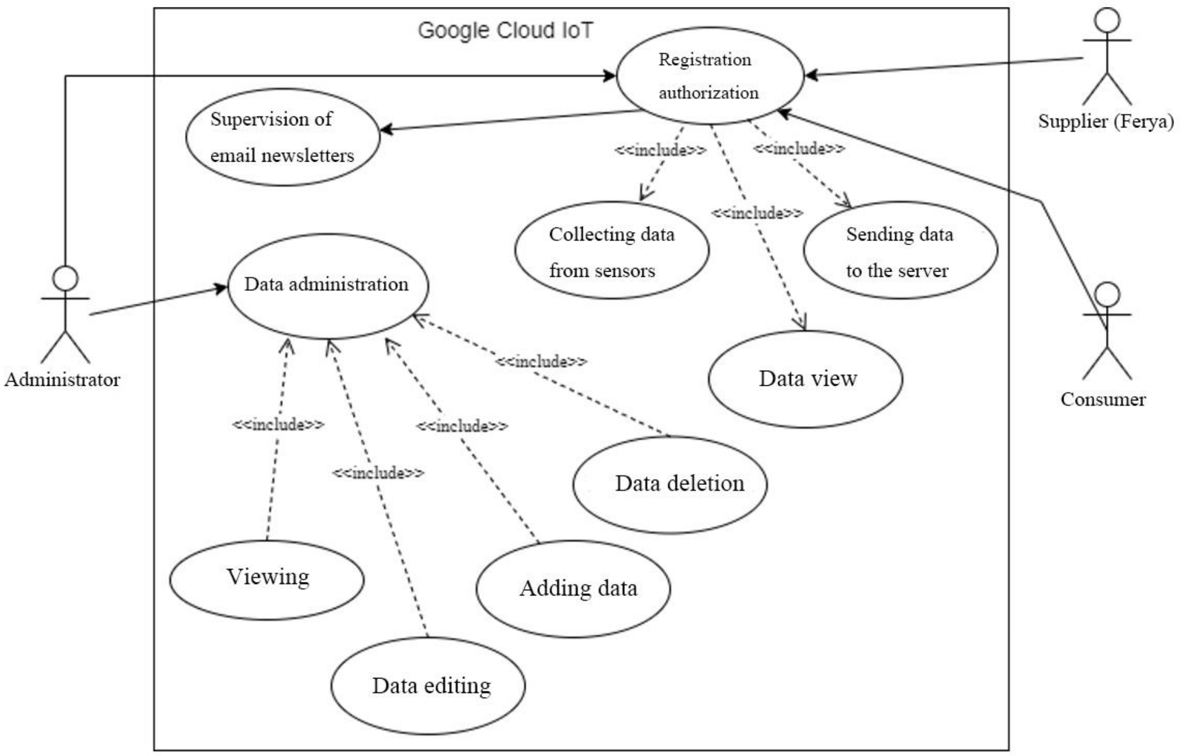


Figure 2.6 – Diagram of use cases

The administrator has access to actions such as editing data, adding and adding data, these actions are related to the database.

The administrator (farm) and the consumer can view the data taken from the sensors, update the data and send the updated data to the server. The rectangle contains the actions that are carried out on the cloud platform.

Class diagram is a structural diagram of the UML modeling language that demonstrates the general structure of the hierarchy of classes of the system, their cooperation, attributes (fields), methods, interfaces and relationships between them. It is widely used not only for documentation and visualization, but also for construction through direct or reverse engineering (Figure 2.7).

All entities that the programmer is going to work with must be represented by class objects in the program. Each class should have only one purpose and a uniquely meaningful name that will be associated with this purpose [15].

The class diagram is directly related to the algorithm of the application in section 2.3. The user registers, then logs in. If a regular user has logged into the system, then he can view data on the quality of milk, if an administrator, then deleting, adding and viewing data is available.

This diagram shows 7 classes: RegistrationForm, LoginForm, Userform, Adminform, ViewDataForm, DeleteDataForm, AddingDataForm.

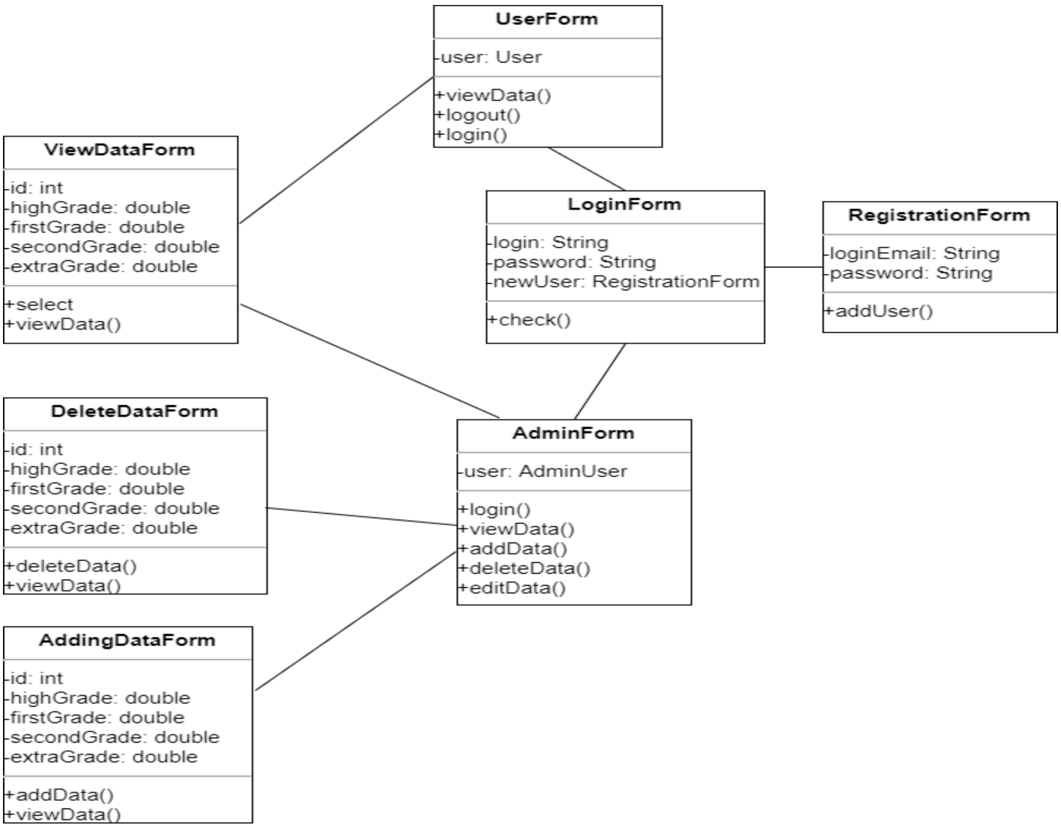


Figure 2.7 – Class diagram

The RegistrationForm class contains the loginEmail, password attributes. The operation of the class is addUser. LoginEmail and password type String (Figure 2.8).

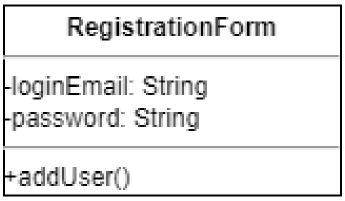


Figure 2.8 – Class RegistrationForm

The LoginForm class contains the loginEmail, password, and newUser attributes of the String type. The class operation is check (Figure 2.9).

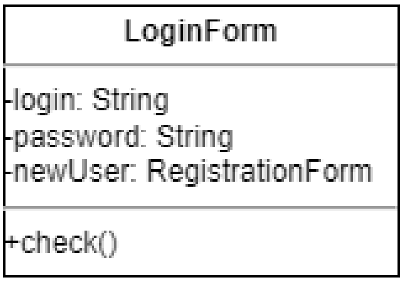


Figure 2.9 – LoginForm class

The UserForm class contains the loginEmail, password, and newUser attributes of the String type. The class operation is ViewData, logout, login (Figure 2.10).

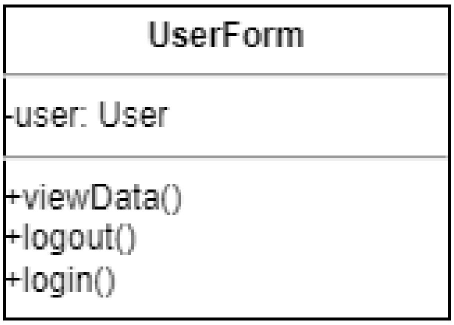


Figure 2.10 – UserForm Class

The AdminForm class contains the user attribute. The class operation is login, ViewData, addData, deleteData, editData (Figure 2.11).

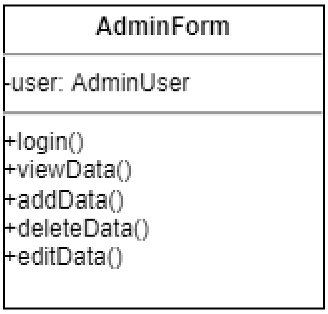


Figure 2.11 – AdminForm Class

The ViewDataForm class contains id attributes of type integer, highGrade, firstGrade, secondGrade, extraGrade of type double. The class operation is select, ViewData (Figure 2.12).

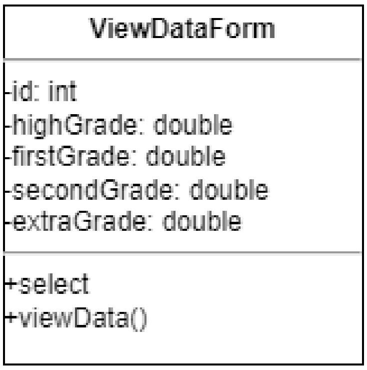


Figure 2.12 – ViewDataForm Class

The DeleteDataForm class contains id attributes of type integer, highGrade, firstGrade, secondGrade, extraGrade of type double. The class operation is deleteData, ViewData (Figure 2.13).

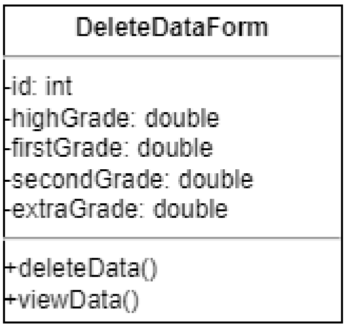


Figure 2.13 – DeleteDataForm class

The AddingDataForm class contains id attributes of the integer, highGrade, firstGrade, secondGrade, double type. The class operation is addData, ViewData (Figure 2.14).

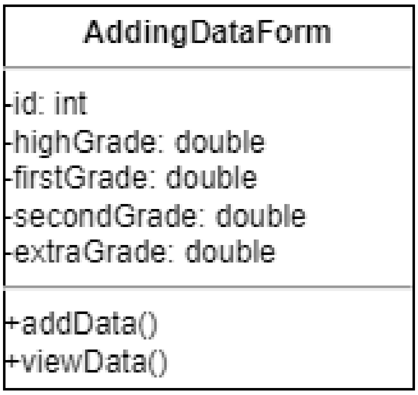


Figure 2.14 – AddingDataForm class

2.5 Simulation of Sensors on Smartphone

An important decision at the stage of software product implementation is the choice of development technology. Based on the requirements for the mobile application being developed and the algorithm of operation, it is necessary to decide on the choice of programming language.The developed mobile application for quality control of dairy products should work on the Android operating system. The programming language chosen was Java, which is considered the official development language for Android. Java is supported by Google's Android Studio development environment and allows you to create an installation file of the application being developed for a smartphone. The development environment for the Android operating system should have an intuitive interface, logical and consistent work in the program code editor, work should be aimed at working with both code and graphics, clear instructions for work.

Android Studio is an application development environment for devices based on the Android operating system. Each next update brings an increase in the speed of information processing, acceleration of compilation of the finished project and many new features for maximum compatibility with the new Android OS APIs. Among the advantages are the flexibility of the development environment, a large set of functions, a development process that adapts to the user. During the creation of applications for the Android operating system, the software user can observe changes in the project in real time. The program has a built-in emulator that allows you to check the correct operation of the application on devices with different screens, with different aspect ratios.

Google's Firebase has been selected as a cloud platform. It is intuitively easy to use, makes it possible to establish communication between the parties in real time, special security tools are provided, and a variety of authentication functionality. These advantages are very important, because users of the application must be authorized by email. Popular features of the Google Firebase platform include databases, authentication, push notifications, analytics, file storage and much more.

Sensors that monitor the physical properties and condition of milk on a particular farm provide innovative ways to improve the quality of dairy products. The use of sensors optimizes and facilitates work on farms. As sensors in this development, a smartphone is used, which generates data that is subsequently sent to the cloud platform. Several sensors are used in the development.

The implementation of the data sensor generator for the freezing point is given:

```
public float getFreezingPointRandomNumForHighGrade() {
float max = -0.520f;
float min = 0.515f;
return (float) ((Math.random() * (max - min)) + min);
}
```

The implementation of the somatic cell data sensor generator is given below:

```
public int getSomaticCellsRandomNumForHighGrade() {
int max = 300;
int min = 500;
return (int) ((Math.random() * (max - min)) + min);
}
```

The implementation of the temperature sensor generator is given below:

```
public int getTemperatureRandomNumForHighAndFirstGrade() {
int max = 4;
int min = 6;
return (int) ((Math.random() * (max - min)) + min);
}
```

The implementation of the milk acidity indicator generator is shown below:

```
public int getAcidityRandomNumForHighFirstExtraGrade() {
int max = 16;
int min = 18;
return (int) ((Math.random() * (max - min)) + min);
}
```

The implementation of the milk density indicator generator is shown below:

```
public int getBacterialContaminationRandomNumForHighGrade() {
int max = 100;
int min = 300;
return (int) ((Math.random() * (max - min)) + min);
}
```

The implementation of the milk density indicator generator is shown below:

```
public int getDencityRandomNumForHighExtraGrade() {
int max = 1028;
int min = 7000;
return (int) ((Math.random() * (max - min)) + min);
}
```

In our case, virtual sensors are used, which provide simplified data. The device may include several implementations of the same type of sensors.

When entering the simulator application of the product quality control sensor, a table without values appears. To generate sensor indicators, click on the «Generate» button. After that, a pop-up message «Generated!» will appear (Figure 2.15). Next, you need to send the generated data to the server, click the «Send» button (Figure 2.16).

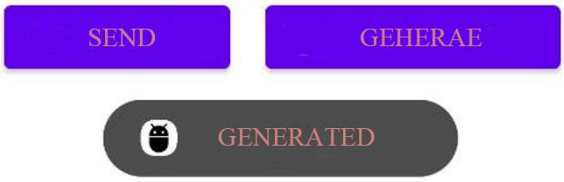


Figure 2.15 – Buttons on the sensor

	Top grade	First grade	Second grade	Extra
Acidity	17	16	16	17
Temperature	326	632	836	272
Density	4009	2100	514	1128
Freezing point	0	0.270348	1.55443	-0.272602
Bacterial contamination	160	321	1314	13
Somatic cells	429	0	999	146

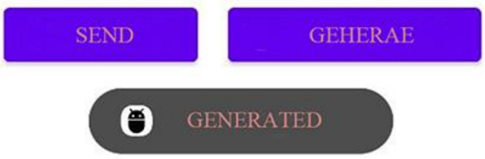


Figure 2.16 – Generated milk quality indicators

Implementation of data generation on the above sensors will be described:

```
JSONObject jsonObject = new JSONObject();
jsonObject.put(«highGrade», highGrade);
jsonObject.put(«firstGrade», firstGrade);
jsonObject.put(«secondGrade», secondGrade);
jsonObject.put(«extraGrade», extraGrade);
String postBody = «{ \»highGrade\»:\»« + highGrade + «\», \»firstGrade\»:\»« + firstGrade + «\», \»secondGrade\»:\»« + secondGrade + «\», \»extraGrade\»:\»« + extraGrade + «\»}»;
```

```
try {
    postRequest(postUrl, postBody);
} catch (IOException e) {
    e.printStackTrace();
}
JSONObject jsonObject1 = new JSONObject();
jsonObject1.put(«highGrade», highGrade1);
jsonObject1.put(«firstGrade», firstGrade1);
jsonObject1.put(«secondGrade», secondGrade1);
jsonObject1.put(«extraGrade», extraGrade1);
String postBody1 = «{ \»highGrade\»:\»« + highGrade1 + «\», \»firstGrade\»:\»« + firstGrade1 + «\», \»secondGrade\»:\»« + secondGrade1 + «\», \»extraGrade\»:\»« + extraGrade1 + «\»}»;
try {
    postRequest(postUrls, postBody1);
} catch (IOException e) {
    e.printStackTrace();
}
```

Using the example of the method of obtaining bacterial contamination data, we will show their introduction into the sensor fields:

```
private String getContent() throws IOException {
 BufferedReader reader = null;
 InputStream stream = null;
 HttpsURLConnection connection = null;
 try {
  URL url = new URL(«https://borshes.store/api/acidity/read.php»);
  connection = (HttpsURLConnection) url.openConnection();
  connection.setRequestMethod(«GET»);
  connection.setReadTimeout(10000);
  connection.connect();
  stream = connection.getInputStream();
  reader = new BufferedReader(new InputStreamReader(stream));
  StringBuilder buf = new StringBuilder();
  String line;
  while ((line = reader.readLine()) != null) {
     buf.append(line).append(«\n»);
  }
  return (buf.toString());
} finally {
```

```
            if (reader != null) {
                reader.close();
            }
            if (stream != null) {
                stream.close();
            }
            if (connection != null) {
                connection.disconnect();
            }
        }
    }
```

Let's describe the method of sending data to the server using the «send» button:

```
        JSONObject jsonObject10 = new JSONObject();
        jsonObject10.put(«highGrade», somaticCellsHighGrade);
        jsonObject10.put(«firstGrade», somaticCellsFirstGrade);
        jsonObject10.put(«secondGrade», somaticCellsSecondGrade);
        jsonObject10.put(«extraGrade», somaticCellsExtraGrade);
        String postBody11 = «{ \»highGrade\»:\»« + somaticCellsHighGrade + «\», \»firstGrade\»:\»«
                + somaticCellsFirstGrade + «\», \»secondGrade\»:\»« + somaticCellsSecondGrade +
                «\», \»extraGrade\»:\»« + somaticCellsExtraGrade + «\»}»;
        try {
            postRequest(postSomaticCellsPoint, postBody11);
        } catch (IOException e) {
            e.printStackTrace();
        }
        JSONObject jsonObject11 = new JSONObject();
        jsonObject11.put(«highGrade», temperatureHighGrade);
        jsonObject11.put(«firstGrade», temperatureFirstGrade);
        jsonObject11.put(«secondGrade», temperatureSecondGrade);
        jsonObject11.put(«extraGrade», temperatureExtraGrade);
        String postBody12 = «{ \»highGrade\»:\»« + temperatureHighGrade + «\», \»firstGrade\»:\»«
                + temperatureFirstGrade + «\», \»secondGrade\»:\»« + temperatureSecondGrade +
                «\», \»extraGrade\»:\»« + temperatureExtraGrade + «\»}»;
        try {
            postRequest(postTemperature, postBody11);
```

```
        Toast.makeText(getApplicationContext(), »Сгенерировано!»,
Toast.LENGTH_SHORT).show();
      } catch (IOException e) {
        e.printStackTrace();  } }
```

2.6 Smartphone and Cloud Platform Communication

Thanks to the rapid development of microelectronics, communication channels, Internet technologies, the topic of smart agriculture is becoming more and more relevant. Solutions are coming to the market that turn farms into complex information systems controlled from anywhere in the world using a smartphone. Moreover, knowledge of programming languages is no longer required for human-machine interaction, information is presented in a form understandable to any user.

In order to make a smart farm out of an ordinary farm, it is necessary to install sensors that measure various parameters and controllers that perform calculations in accordance with sensor measurements and embedded logic that issues commands for subsequent devices. In the simulated IoT network, a smartphone acts as sensors for monitoring product quality. Next, the connection of the smartphone with the cloud platform will be described. Sensors and actuators, as a rule, are connected wirelessly to a controller that combines all these devices into a single network and controls them.

The server process is a key component that performs all the basic work on automating the information processes that form the basis of a smart farm: receiving and processing sensory data, issuing control actions depending on the logic laid down. In our case, milk quality control involves sending an email to the administrator and users. The purpose of the server process is to interact with sensors, execute logical rules, receive and process commands from the graphical interface (Figure 2.17).

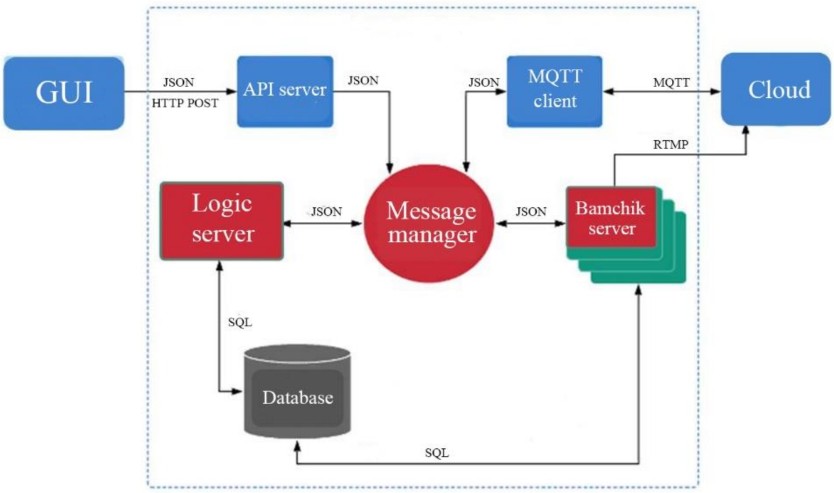


Figure 2.17 – Server process

The cloud communicates with the platform via the HTTP protocol. HTTP is a widespread data transfer protocol, originally designed to transfer hypertext documents (that is, documents that may contain links that allow you to organize a transition to other documents). The HTTP protocol assumes the use of a client-server data transmission structure. The client application forms a request and sends it to the server, after which the server software processes this request, forms a response and transmits it back to the client. After that, the client application can continue to send other requests that will be processed in the same way. The API of many software products also implies the use of HTTP for data transmission – the data itself can have a format, for example, XML or JSON. Data from the sensor is transmitted to the server at the touch of a button.

2.7 Modeling of the IoT Network Based on Cloud Platform

LPWAN-based IoT solutions are actively used in agriculture. Previously, IoT devices were connected via networks of cellular operators – 2G, 3G, 4G. But due to constant synchronization, excessive data transfer speeds and online operation, the batteries of the devices were quickly discharged. This is how relatively new LPWAN networks got a boost in development, where the cost of connection is ten times lower than in GPRS or 3G. LPWAN equipment works without recharging for up to ten years, and the communication range, depending on the terrain conditions, exceeds 15 km. Experts attribute at least 16 technologies to LPWAN, among which NB-IoT, LoRaWAN, SigFox, LTE-M are leading. There are two types of LPWANs – operating in licensed and unlicensed frequency bands. The former belong to mobile operators, the latter are operated privately. Operators are building LPWAN on 4G networks using NB-IoT and LTE-M technologies, which are optimized for IoT (periodic transmission of small amounts of data).

LTE-M is not represented on the Belarusian market, but is developing in North America. NB-IoT is called promising due to the security and warranty of the service, LTE data protection algorithms, global roaming, and a wide coverage area. NB-IoT is universal for many scenarios where latency and data transfer speed are critical, according to representatives of the operator. But there are also disadvantages – earlier operators announced a limited range of NB-IoT client equipment and relatively high cost of modules. Special development sites were even created to solve the problem.

To communicate with the server, an API (application programming interface) is used – this is a set of tools and functions in the form of an interface for creating new applications, thanks to which one program will interact with another. This allows you to extend the functionality of the product and link it with others, as well as make development more secure, for example, when trying to authorize.

POST is used to create a new object in a set of objects. The response to the request will be sent in JSON format. To develop the Internet of Things on the Firebase platform, sensors are needed, in our case, emulators that measure milk quality indicators that interact with the IoT platform using development tools. Further the devices send messages that are verified by the authentication and authorization service of the Firebase platform. In case of failure of verification, correction of device ids is necessary. Information from the devices is sent to the gateway, and various network protocols can be used. Next, the information is sent to the rules processing and device storage unit, which stores the current states of the network peripherals for constant access to software applications. If there is no connection with a separate device on the network, the Device Shadow block executes commands from applications and, when the connection is restored, synchronizes the current state with the device.

The rule handler performs programmed actions depending on the nature of the received data: saves data in a database, sends information via SMS or e-mail to the network manager about their receipt, calls the HTTP API, sends data to the analytics system, etc. Further, information about all devices is stored on the Firebase platform.

Application for monitoring and results. After all the high-quality milk data is generated by the smartphone and sent to the cloud platform, it is necessary to register and authorize in the client application, which is accessed via the Internet. Data from the data collection devices are sent to the Firebase server, and the customer receives his personal account in which all data is available. The key advantages of such software are reliability, the absence of initial license costs, the speed of deployment of the working environment and the convenience of working (in any place where there is Internet). To begin with, we will find a folder with a client application on the desktop of the smartphone and go into it (Figure 2.18).

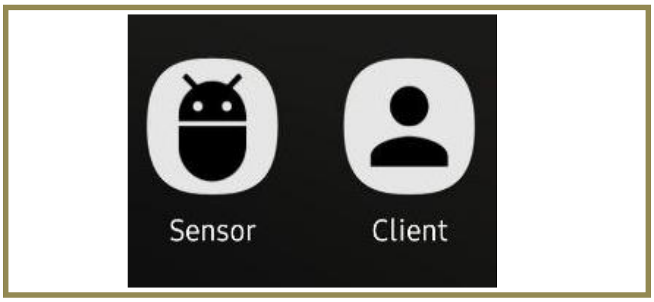


Figure 2.18 – Client application and sensor

Any unregistered user cannot log in to the application. Therefore, it is necessary to register, to do this, click the appropriate button.

Registration is necessary to create a stable connection between the user and the application. The application gets the opportunity to interact with its visitors directly.

In the upper left corner you can see the inscription «Client», this indicates that the application is a client.

To register, after clicking the appropriate button, you need to enter your email address, where messages with milk quality indicators will be sent, and a password (Figure 2.19).

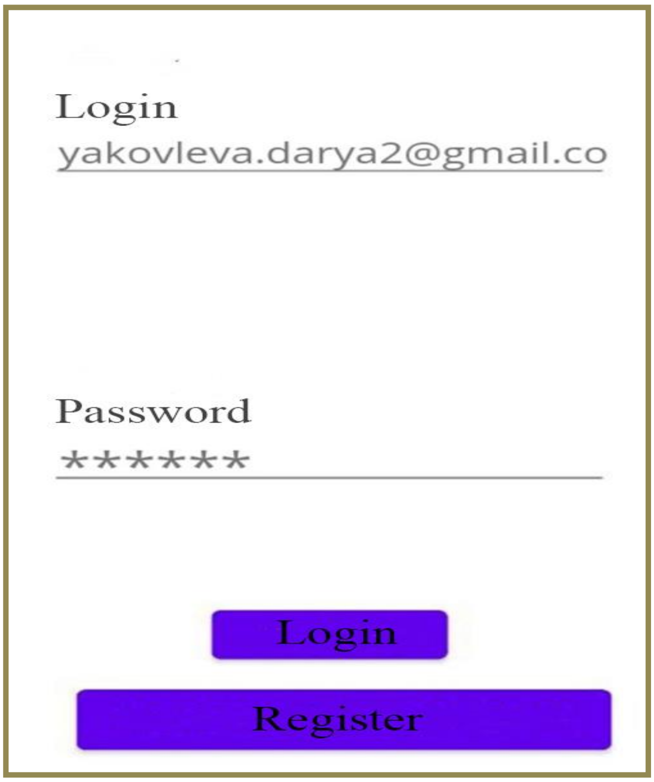


Figure 2.19 – Registration and authorization window

Then you need to log in, that is, enter your email address and password. If everything was successful, a pop-up will appear (Figure 2.20).

Figure 2.20 – Successful authorization

After authorization, a table without data will appear in front of the user. When you click the «Update» button, a table with the current milk indicators appears on the screen (Figure 2.21). The data in the client application matches the data generated by the indicator sensor.

Every day at 12:00, all registered users receive an SMS newsletter to the email address specified during registration, which displays milk indicators by varieties for the current day.

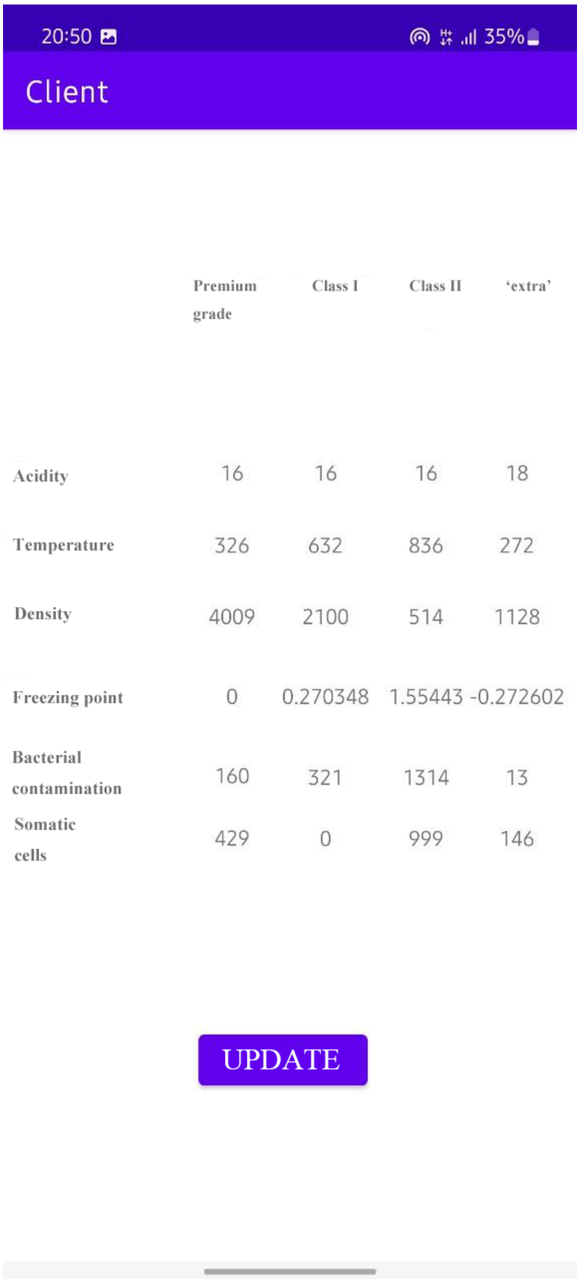


Figure 2.21 – Client application after update

Hosting is used for mailing beget.com . The contents of the messages received by e-mail are shown in Figure 2.22.

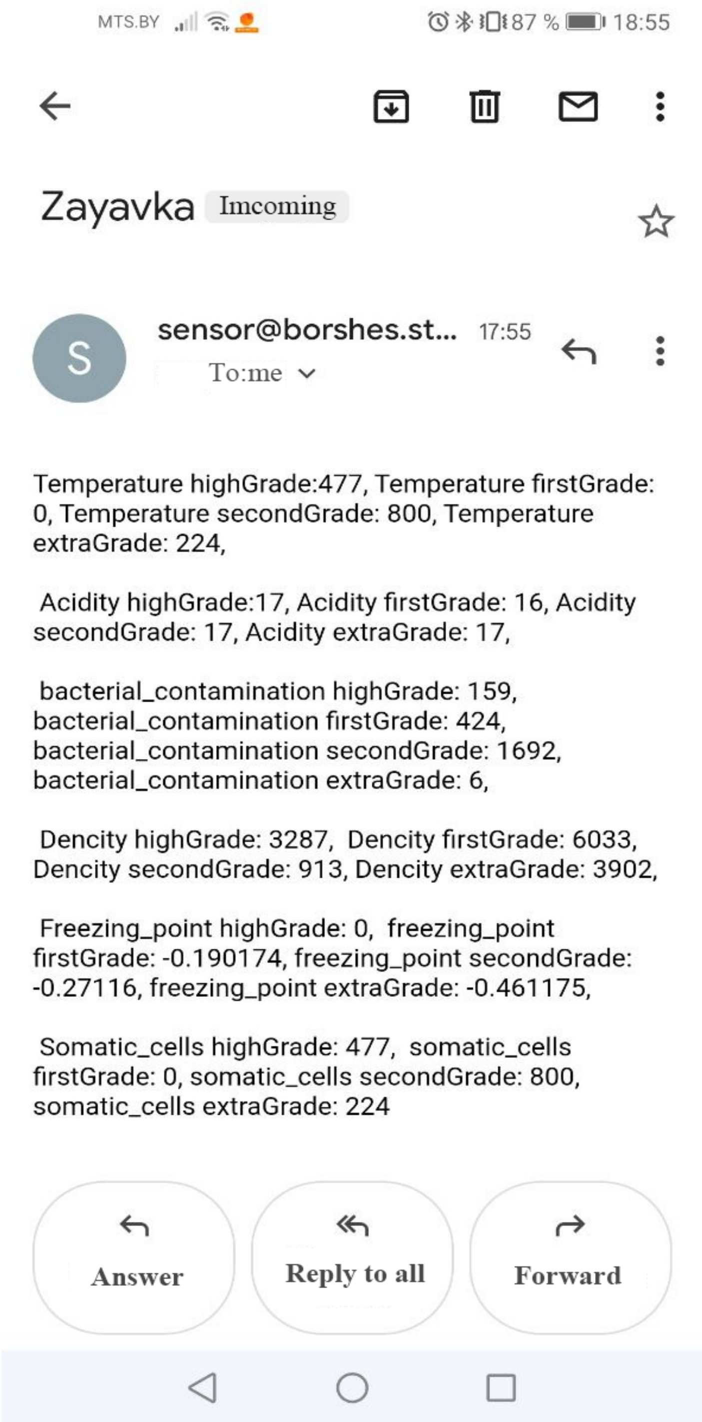


Figure 2.22 – Content of SMS mailing

2.8 Software Product Testing

Software testing is the process of evaluating a software product in order to find any errors. Checks whether the developed software meets the specified requirements and identifies any defects in the software in order to produce a quality product. The correspondence of the real result with the expected behavior of the program is also checked. Testing is carried out at every stage of creating a software product [17].

Globally, all types of testing, depending on the goals pursued, can be divided into groups:

– functional;

– non-functional;

– related to changes.

Functional testing is based on functions, features, interaction with other systems and can be presented at all levels of testing: modular, integration, system and acceptance. Functional types of testing consider the external behavior of the system. Includes such types of testing as security and interaction testing

Non-functional testing checks how the system works: describes the tests necessary to determine the characteristics of the software, which can be measured in various quantities. For example, performance testing and usability.

After making the necessary changes, such as fixing bugs/defects, the software should be tested again in order to confirm that the problem has actually been solved. Such types of testing as smoke and regression testing can be attributed [18].

Test case «Checking the installation of the Sensor on a mobile device».

Initial condition: the Android version on the mobile device must be at least 12.

Test case execution:

– open Explorer on a mobile device;

– open the Download folder where the Sensor apk file of the Sensor application is located and install this file.

Expected result:

– the process of installing the Sensor apk file is underway;

– a pop-up message about the successful installation of the application appears on the screen of the mobile device;

– among the installed applications, the «Sensor» application appears on the desktop.

According to the test results, there was a complete coincidence of the real result with the expected one.

Test case «Checking the removal of the Sensor application from a mobile device».

Initial condition: the presence of the sensor application on the mobile device.

Test case execution:

– open the list of mobile device applications;

– find sensor among the list of applications;

– click the «Delete application» button and confirm the action in the corresponding pop-up window.

Expected result:

– the application is being uninstalled;

– a pop-up message appears on the screen of the mobile device stating that this application has been deleted;

– absence of remote among applications.

Complete coincidence of the real result with the expected one.

Test case «Checking the launch of the Sensor application».

Initial condition: the presence of this application on a mobile device.

Test case execution:

– open the list of applications installed on the mobile device;

– launch this mobile application.

Expected result:

– the Sensor mobile application is being launched;

– a table without data and the «Generate» and «Send» buttons appear on the screen of the mobile device.

Complete coincidence of the real result with the expected one.

Test case «Verification of Sensor data generation».

Initial condition: the presence of the Sensor application on the mobile device.

Test case execution:

– open the Sensor application in the list of mobile device applications;

– click the «Generate» button.

Expected result:

– data generation occurs;

– the appearance of a pop-up message «Generated».

A complete match of the real result with the expected one.

Test case «Display of generated data».

Initial condition: the presence of the Sensor application on the mobile device and the successful generation of data on the sensor.

Test case execution:

– open the Sensor application in the list of mobile device applications;

– click the «Generate» button;

– click the «Send» button.

Expected result:

– data generation occurs;

– the appearance of a pop-up message «Generated»;

– the appearance of the generated data on the screen.

Complete coincidence of the real result with the expected one.

Test case «Checking the installation of the Client application on a mobile device».

Initial condition: the Android version on the mobile device must be at least 12.

Test case execution:

– open Explorer on a mobile device;

– open the Download folder where the Client apk file of the client application is located and install this file.

Expected result:

– the process of installing the Client apk file is underway;

– a pop-up message about the successful installation of the application appears on the screen of the mobile device;

– the Client application appears among the installed applications on the desktop.

According to the test results, there was a complete coincidence of the real result with the expected one.

Test case «Checking the removal of the Client application from a mobile device».

Initial condition: the presence of an application for customers on a mobile device.

Test case execution:

– open the list of mobile device applications;

– find Client among the list of applications;

– click the «Delete application» button and confirm the action in the corresponding pop-up window.

Expected result:

– the application is being uninstalled;

– a pop-up message appears on the screen of the mobile device stating that this application has been deleted;

– absence of remote among applications.

Complete coincidence of the real result with the expected one.

Test case «Checking the launch of the Client application».

Initial condition: the presence of this application on a mobile device.

Test case execution:

– open the list of applications installed on the mobile device;

– launch this mobile application.

Expected result:

– the Client mobile application is being launched;

– a registration and authorization window appears on the screen of the mobile device with the «Log in» and «Register» buttons and fields for entering email and password.

Complete coincidence of the real result with the expected one.

Test case «Client application registration verification».

Initial condition: the presence of a client application on a mobile device, the password must contain at least 4 characters, the email address must contain the «@» sign and the by, com, ru domain.

Test case execution:

– open the Client application on a mobile device;

– click the «Register» button;

– enter your email address and password;

– press the «Log in» button.

Expected result:

– a pop-up message appears about the successful registration of the user;

– if the e-mail address or password data is entered incorrectly during registration, as required by the initial conditions, the message «Invalid login or password» is displayed;

Complete coincidence of the real result with the expected one.

Test case «Verification of authorization in the Client application».

Initial condition: the presence of a client application on a mobile device, the presence of a registered user account.

Test case execution:

– open the Client application on a mobile device;

– enter your email address and password;

– press the «Log in» button.

Expected result:

– when you click the «Log in» button, a pop-up message about successful authorization and a table without data appears;

– if the e-mail address or password data is entered incorrectly during registration, as required by the initial conditions, the message «Invalid login or password» is displayed;

– if the e-mail address or password data is entered incorrectly during authorization, an error message is displayed.

Complete coincidence of the real result with the expected one.

Test case «Checking data in the Client application».

Initial condition: the presence of the Client application on a mobile device, the presence of a registered user account.

Test case execution:

– log in to the client application;

– log in to the app by entering your email address and password;

– when you click the «Log in» button, a pop-up message about successful authorization and a table without data appears;

– click on the «Update» button to update the data.

Expected result:

– successful authorization to the Client application;

– after clicking the «Refresh» button, the data generated by the sensor should appear.

Complete coincidence of the real result with the expected one.

Conclusion on Chapter 2

1 Within the framework of «Automation 4.0» is proposed to use the IoT technology for remote monitoring and control of milk quality of dairy farms distributed throughout the district. To implement the milk quality control IoT network, it is proposed to use a cloud platform (a cloud-hosted managed service that acts as a message center for two-way communication between an IoT application and devices).

2 The model of such a IoT network based on multi-agent technology is presented. The structure of this IoT network is proposed, which includes milk analyzers, gateways-converters and a cloud structure, in which the server platform is rented. The server database stores milk quality indicators based on critical control points. These indicators can be monitored from the mobile devices of specialists. In the future, the IoT network will be able to solve the issues of optimizing these indicators. The most popular cloud platforms are considered. The 4th generation LTE network using the technology for the IB-NB – IoT network was chosen as the network for transmitting information from dairy farms to the cloud environment.

3 The Google Cloud IoT platform allows to create IoT networks using machine learning on end devices and implementing services such as Cloud IoT Core (collects data published in Cloud Pub/Sub for further analysis), Cloud IoT Edge (for securely connecting edge devices to the cloud). The procedure for connecting sensors to the Google Cloud IoT platform is presented, including: configuring the local environment and installing the necessary components; creating an account; connecting a virtual device and viewing telemetry.

4 The algorithm of work for monitoring the quality of milk is described. Class diagrams and usage case diagrams are presented. The simulation of sensors on a smartphone is performed and its connection with a cloud platform is described. A client application and a milk quality control sensor application have been developed. Testing of these applications was also performed at the end of the development process, the test results were presented in the form of test cases.

References

1 Visniakou, U. A. Organization of management and structure in local networks internet of things / U. A. Visniakou, A. H. Al-Masri, S. K. Al-Haji // System Analyses and application informatics. – 2020. – № 2. – P. 11−16.

2 Visniakou, U. A. Structure, network protocols of the internet of things for quality production control / U. A. Vishniakou, B. H. Shaya, A. H. Al-Masri, S. K. Al-Haji // Siberian Journal of Science and Technology. – 2020. – № 4. – P. 478−482.

3 Visniakou, U. A. Model and structure of the network internet of things for monitoring milk quality / U. A. Visniakou, A. H. Al-Masri, S. K. Al-Haji // System Analyses and application informatics. – 2021. – № 1. – P. 39–44.

4 Vishniakou, U. A. Technologies of intelligence multiagent information processing with blockchain for management systems / U. A. Vishniakou, B. H. Shaya, A. H. Al-Masri, S. K. Al-Hajj // Materials of the 9th international scientific and technical conference OSTIS-2019. – Minsk : BSUIR. – 2019. – P. 311–314.

5 Visniakou, U. A. Technology of adaptive intelligence multiagent processing for information defense system design / U. A. Vishniakou, A. H. Al-Masri, S. K. Al-Haji // Int. scien. conf. «Coding and Signal Digital Processing». – Minsk : BSUIR. – 2019. – P. 99–103.

6 Vishniakou, U. A. Specialists training in area of networks IoT / U. A. Vishniakou, H. A. Al-Masry, K. S. Al-Hajj // Materials of the IV International Scientific Conference «Informatization of Education and E-Education Methodology: Digital Technologies in Education» 6–9 October 2020 : at 2 h. Part 1 / sub. ed. N. V. Noscova. – Krasnoyarsk, Sib. feder. univ. – 2020. – P. 403–406.

7 Visniakou, U. A. Automation tools for the development of internet of things network / U. A. Vishniakou, B. H. Shaya, A. H. Al-Masri, S. K. Al-Haji // Int. scientific seminar Telecommunications: Systems and Networks, Algebraic Coding and data security». – Minsk : BSUIR, 2020. – P. 13–16.

8 Visniakou, U. A. Simulation of internet of things network on cloud platform base / U. A. Vishniakou, A. H. Al-Masri, S. K. Al-Haji // Int. scien. conf. «Coding and Signal Digital Processing» − Minsk : BSUIR. – 2021. – P. 65–68.

9 Visniakou, U. A. IoT components for production quality monitoring / U. A. Vishniakou, A. H. Al-Masri, S. K. Al-Hajj // «Telecommunications: networks and technologies, algebraic coding and data security» materials of the International Scientific and Technical Seminar, Minsk, November – December 2021. – Minsk : BSUIR. – P. 65–68.

10 Trends in automation of dairy enterprises [Electronic resource]. – Access mode : https://dairynews.today/news/tendencii_avtomatizacii_molochnyh_predprijatij.html. – Date of access : 12.04.2022.

11 The system of technological monitoring of milk production on the farm [Electronic resource]. – Access mode : https://cyberleninka.ru/article/n/sistema-tehnologicheskogo-monitoringa-proizvodstva-moloka-na-ferme-krs/viewer. – Date of access : 12.04.2022.

12 Internet of Things in industry [Electronic resource]. – Access mode : https://controlengrussia.com/internet-veshhej/klyuchevy-h-tehnologij. – Date of access : 12.04.2022.

13 Vishnyakou, U. A. Model, structure and components of the Internet of Things network for controlling dairy farms / U. A. Vishnyakou // Problems of infocommunications. – 2020. – № 2. – P. 36–40.

14 Review of solutions in monitoring systems [Electronic resource]. – Access mode : https://1-engineer.ru/internet-veshchey-obzor-resheniy-v-sistemakh-monitoringa. – Date of access : 04.15.2022.

15 Precedent diagram [Electronic resource]. – Access mode : https://github.com/kolei/PiRIS/blob/master/articles/5_1_1_10_uml_use_case.md. – Date of access : 12.04.2022.

16 Butch, G. UML language. User's Guide / G. Butch, D. Rambo, I. Jacobson. – M. : DMK Press, 2006. – 120 p.

17 Types of testing [Electronic resource]. – Access mode : https://bugza.info/vidy-testirovaniya. – Date of access : 08.05.2022.

18 Vishnyakou, U. A. Platforms for the development of Internet of Things networks / U. A. Vishnyakou // Vesnik suvyazi. – 2021. – № 3. – P. 62–65.

19 Milk quality analyzers [Electronic resource]. – Access mode : https://www.csm.brest.by/analizatory-kachestva-moloka. – Date of access : 04.25.2022.

3 MODEL, STRUCTURE AND DESIGN OF IoT NETWORK DATABASE

3.1 Rewire of Milk Analyzers in the IoT Network

To creating IoT network models for monitoring milk quality (MMQ) we use a multi-agent approach [4]. In this multi-agent structure, the multiplication of agents of milk gesture sensors, agents of transformations, agents of feeding bloodiness indicators, agents of processing showed the pledge of milk for taking the oath, agents of monitoring these pointers and the killed. We will represent this multi-agency model as a four components (see formula (2.1)) [15].

On the basis of this model, the detailed structure of IoT network control the quality of milk is designed. It includes portable analyzers of milk gestures from each from controlling farms. More often, the resulting analyzers are output to a computer or printer via a port. In our structure, these indicators are handed over to gateways-converters – *SC* of IoT. The *SC* scoundrels for the transformation and transfer to the IoT cloud platform *CP* with the highest indicators according to the milk quality.

The server in the CP contains a database, an API – a program for working with the database and a client application. In the database of feedings, semi-closed data from farms are obtained, and nominal similar features of milk. These data will be sent to the server via the API. On cloud server there are files with a logical solution for the working the received data from sensors. The client applications use IoT server as a means of displaying the removed and obtained results according to the milk parameters. According to the mobile devices (MD) of farm specialists installed when connecting, calling, information of interest from the cloud database through the applications.

The milk analyzer is a structure for determining the characteristic feature of milk gestures and ego-based products. This structure does not evaporate chemically reagents, which invalidates the environmental friendliness of the component under study. Milk analyzers are applicable to farms, undertaken by the pajama industry, milk reception points and during scientific and investigative work in the pajama industry area.

Milk analyzers have high-precision accuracy and invite you to act sufficiently measurable in the required time. In the coming time, analyzers were inherited and became normal processes at milk processing plants.

Maximum fold of measured milk indicators: acidity (T); temperature – (°C); density (kg/m^3); freezing point (°C); bacterial obsession (thousand/cm^3); somatic curses (thousand/cm^3).

Consider distant swollen milk analyzers used in the Republic of Belarus [12].

The ultrasonic analyzer Lactane 1–4 half Mini (Figure 3.1) select a massive proportion of acorns, SOMO, extracted water and density in the rehearsal of whole, fresh, conservative, pasteurized, normalized and devalued milk. The average measurement time will be 3 minutes, that is, 2 times faster than with the manifestation of the traditional method of analysis, harmless, economical and durable. The accuracy of determining the parameters of milk hardness with this completeness of the bustle of the required standard methods [12].

Figure 3.1 – Ultrasonic analyzer lactane 1–4 chubby mini

Lactane analyzer 1–4 full 220 (Figure 3.2) call in 180 seconds without the manifestation of chemical reactive actions, select six important parameters – white, acorn, sumo, density, temperature and volume fraction of added water in samples of all fresh, conservative, pasteurized normalized, normalized, skimmed, waxed milk, long-term nutrition milk [12].

Figure 3.2 – Lactane analyzer 1–4 ispolnenie 220

The Clover-2 milk analyzer (Figure 3.3) provides an express assessment of the percentage of acorns, whites, skimmed milk residue (SOMO) and density in one rehearsal of fresh whole, conservative milk or Cream.

Recklessness on multiple functions, the Clover-2 milk quality analyzer is easy to use, what the caller spends on it is measured by unqualified personnel. The main operations on accessories are closed in the volume, that is, the sample is filled in for measurement and exfoliated after measurement.

Figure 3.3 – Clover-2 milk analyzer

To buy a sample and stock up on ee in the sample receiver, – all this was part of the read seconds. Processes measured by qualitative indicators of milk or cream are interested in 2.5–3.5 minutes. Milk room temperature is measured in 2.5 minutes, and chilled – in 3.5 minutes. the accessories indicator is displayed at the expense of the necessary operator information. Indication of the resulting measured output in digital form with a sampling rate of 0.01 % [12]. As foreign analogues, brought the characteristics of analyzers from Bulgarii serii Lactoscan (Figure 3.4). They can be used to measure acorns, solid particles (SOMO), densities, whites, lactose, salts, water content as a percentage, temperature (°C), disturbance points, pH, conductivity.

Figure 3.4 – Lactoscan milk analyzer

As well as the total solid eternities of the same sample immediately after milking, collection and during processing; somatic curses for the removal of clinicians and subclinical mastitis; thermoregulatory installations for various types of tests; highly artistic test strips for highly artistic falsifications of neutralizers, distortions of hydrogen and urea falsifications of raw milk, which works in effective abilities [13].

Mastering high-precision precision and speed, Lactoscan portable ultrasonic milk analyzers are competitive with Foss Electric, Delta Instruments and Benntley milk analyzers, which have a higher price. Minimal energy consumption and lack of consumables make the Lactoscan milk analyzer attractive for the dairy industry. Low operating cost costs and low price make Lactoscan milk analyzer profitable for dairy farm, dairy enterprise, milk selection center and laboratory.

Ekomilk (Figure 3.5) is a model range of ultrasonic milk quality analyzers of Bulgarian production. Accessories of this series will master the following additional features: attaching a pH electrode to measure the activity of ionic hydrogen in the test sample (displays both in pH and in significant milk acidity °T), control of falsification of the studied full milk according to the conductivity parameter, correction of graduated analyzers through the entered repair values. Ekomilk milk analyzers are equipped with system self-diagnostics with the output of cellular errors on the display, they have RS-232 connectors for inclusion in a personal computer and the ability to turn on a compact thermal printer [14].

The final decapitation may turn out to be a variety of interfaces and protocols that will get pregnant for inclusion and forgiveness by their average infrastructures in the cloud environment. In our case, the information from the analyzers can act as parallel or subsequent ports. Therefore, the necessary gateways are transformers for borrowing with emerging cloud platforms. Let's consider possible solutions [15].

Figure 3.5 – Ekomilk ultrasonic milk quality analyzer

One of the its presented tools from the MOHA company, which has been interested in the composition of utility solutions for more than 30 years and, providing its experience in the field of transforming the Internet into a COM port. She has developed a solution to enable a device with a COM port to cloud environments. MOHA offers a solution to enable the soma system in a cloud environment. Consider some non-mechanical taxi transformers.

Npourt IA5000A-I/O, Npourt IAW5000A-I/O series transformers and Mgaute 5105-MB-EIR gateway will support integration with Alibaba Cloud IoT Platform, Azure IoT Hub, Firebase, Google Cloud IoT or S Honorary cloud via MQTT protocol [13]. Transformers Nport IA5000A-I/O and Nport IAW5000A-I/O ring transmitting not only raw daytime with som Porta, but also control the built channels of discrete wwoda-výRvoda. Data is transferred via the MQTT protocol to JSON formats. In IoT, there is no single universal protocol for the integration of physical objects. Therefore, to create physical installations, we will innocuously select all components from the same manufacturer. The data from the transformative will be processed into a cloud platform, one of which details will be considered below.

The communicator technology recommended for this Internet farm monitoring network requires the cover of a significant gap, it calls LPWAN (Low-Power Wide-agea Network) – an energy-saving long-range installation). In the network, by transmitting information with dairy farms on a cloudy environment, I choose four generations, which has already been tested and well advertised in the republic. From somewhere, it turned out to be for Internet clients. But it has NB-IoT technologies in its composition, it is within LTE, has a low future data transmission, but a big cover – thanks for the manifestation of LTE-network mobility [15].

3.2 Selection and Designing of the Structural Scheme of the Database

Database Management Systems (DBMS) – meeting of software and linguistic tools of general or special purpose that hinder the management of the creation and allocation of databases [16]. All modern DBMS have:

– visual means of creating tables, form, reports;

– access to server DBMS data;

– nostrils attached, high-level functions of web browsers and scaling of data to the Internet;

– object-oriented language;

– SQL support.

MySQL is a kind of relational database management system. MySQL has two-level licensing. MySQL can be distributed in accordance with the terms of the GPL license. For some reason, under the terms of the GPL, esli kakaya-libo programs include MySQL source codes, this is what dolgna is distributed under the GPL license. This can be spent with the plans of developers who do not want to disclose the source texts of their programs. For such cases, a pre-planned commercial license, which also deprives honest service support.

MySQL is ported to a large platform: Linux, mas OS X, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2003, WinCE, Windows Vista, Windows 7, Windows 8, Windows 10 and Windows 11. MySQL imeet API for Delphi, C, C++, Java, Perl, PHP, Python, libraries for language platforms.NAT.

Physical design is the creation of database schemas for a specific DBMS. The specifics of a particular DBMS may include restrictions on named volumes of databases, limited to the types of data being pulled. Krome ToGo, a specific specific DBMS for physical forecasting INCL. the choice of solved data connected to the physical power environment (the choice of disk memory management methods, the division of the database into files and structures, the method of data access), the creation of indexes.

Table designs are created on this stage. The specified table should get its own unique identifier (name). User ID can be is to language of DBMS choosing. The name, type and scale of the field are determined. The first key is selected, new handles are added and the field is indexed (for a bistro search) in the index of the tables.

In order to talk about their completeness obligations during operations to add, change or delete data in the parent and receiving tables. The following conditions may occur: absence of checks; prohibition of operations; cascading highly operational operations to restore or delete daily records in indecent all tables. In addition, during the development of physical models, the necessary resources were evaluated: RAM, speed and memory capacity for feeding data (minimum space on the hard drive).

Whole databases of representations in an objective form of meetings of autonomous materials, systematized in such a way that these materials can be found and processed using an electronic high-strength machine. For the convenience of DB implementation, tools are provided for their familiarization. Data feeding takes place on the Beget hosting.com and the Firebase cloud platform from Google Corporation.

PhpMyAdmin appeared in the daytime graduation project. PhpMyAdmin is a visual database design tool that integrates database design, modeling, creation and operation into a single seamless environment for database systems. The interface of this toolkit is shown in Figure 3.6 [16].

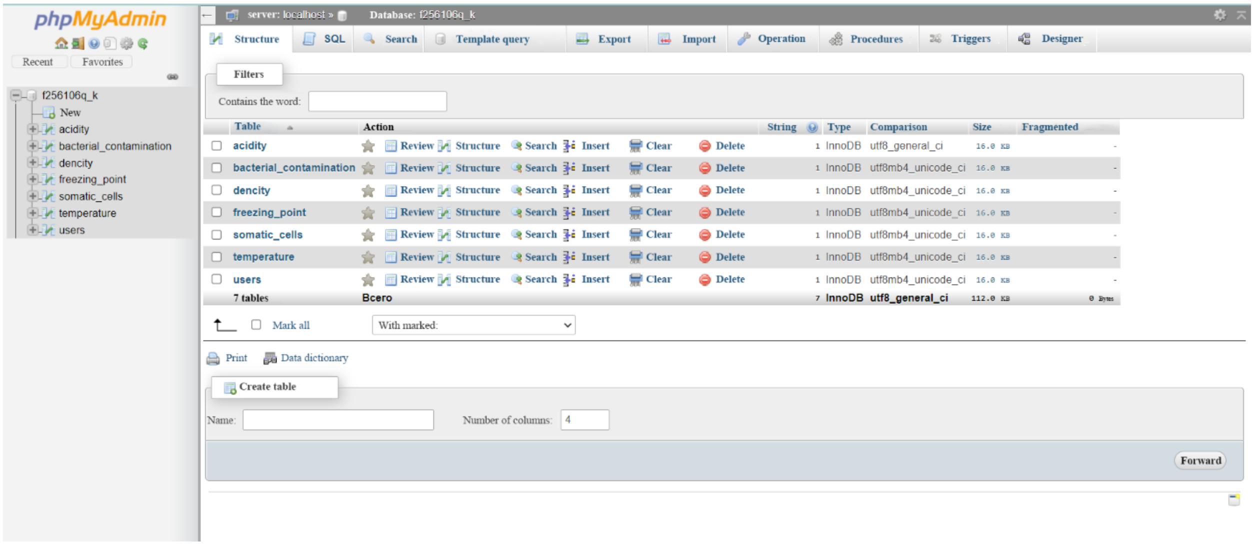


Figure 3.6 – PhpMyAdmin tool interface

It thanks the phpMyAdmin tool for designing a database of data for the Internet network. This is a tool that will be used to create databases and ee exported to a DOOM file, which means that they remember databases to resurrect databases to another DBMS.

When developing the server honor, in this they work with the database, deployed PHP technologies. As a tool for the development of the server side of the client, the PhpStorm development environment is attached and the sensor drops out.

3.3 Desing of Variant Development DB Diagrams

Diagram of variation of manifestation. The diagram is a variation of the manifestation of functional mobility and requires systems with the manifestation of actors and variants of manifestation. Options for using model services, tasks, functions, witch debt system. The options for the manifestation of functional mobility are high-level and this is how the user will call to the system. Variants of the manifestation of the basic concepts of Unified Modeling language [17].

The diagram consists of variants of manifestation, persons or various thinks that cause functions, call actors, and elements that open up variants of manifestation for implementation. Use case diagrams explore dynamic operating systems. It models how to implement a common loan with the system, to improve its operation. Use case diagrams open up for visualization of new, witch relationships with an honest system.

Assistance options for representing the high-level function and how the user should be treated with the system. A variant of the manifestation of the function of systems, components, packages or classes. It is indicated by an oval shape with the name variant evolution written inside the oval shape. Distinctive features in UML are illustrated in Figure 3.7.

Figure 3.7 – UML variation

Actor. He finds himself drawn into a diagram of manifestation options. An actor is a community that lends itself to the system. The user is a good example of an actor. An actor is a community that initiates a using variant in outside of the field of evaporation. It is can be a loving element that can initiate a loan with the case of manifestation. One subject can be connected with multiple variants standing out in the system. The designation of actors in UML is shown in Figure 3.8 [17].

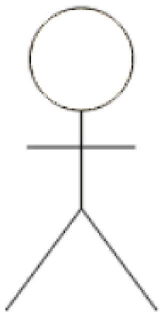


Figure 3.8 – UML actor

Diagram of the variant development of the software product introduced in Figures 3.9 and 3.10 for user and administrator.

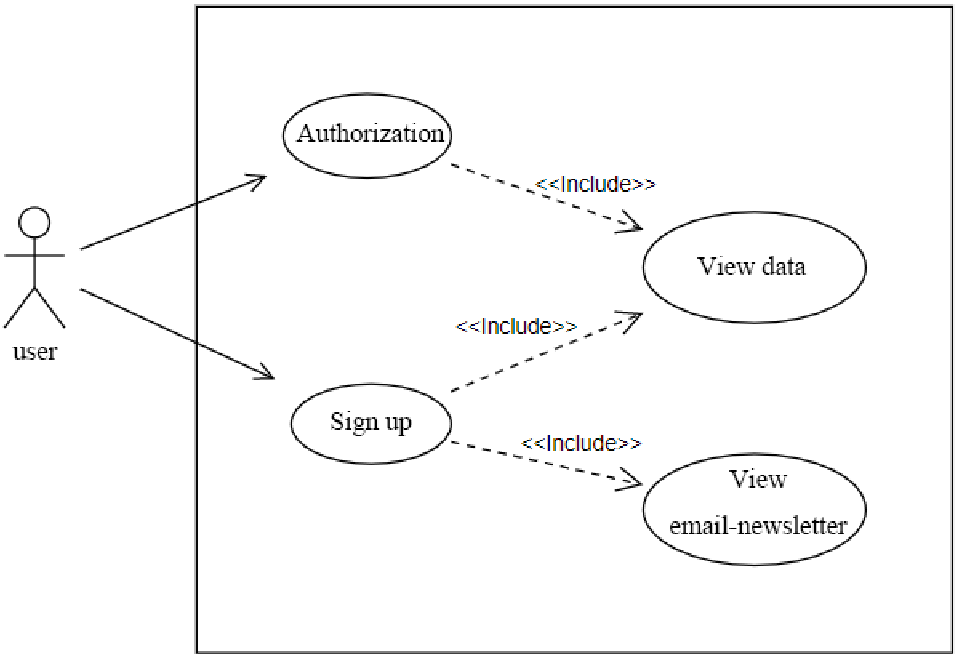


Figure 3.9 – Diagram of the variant development for user

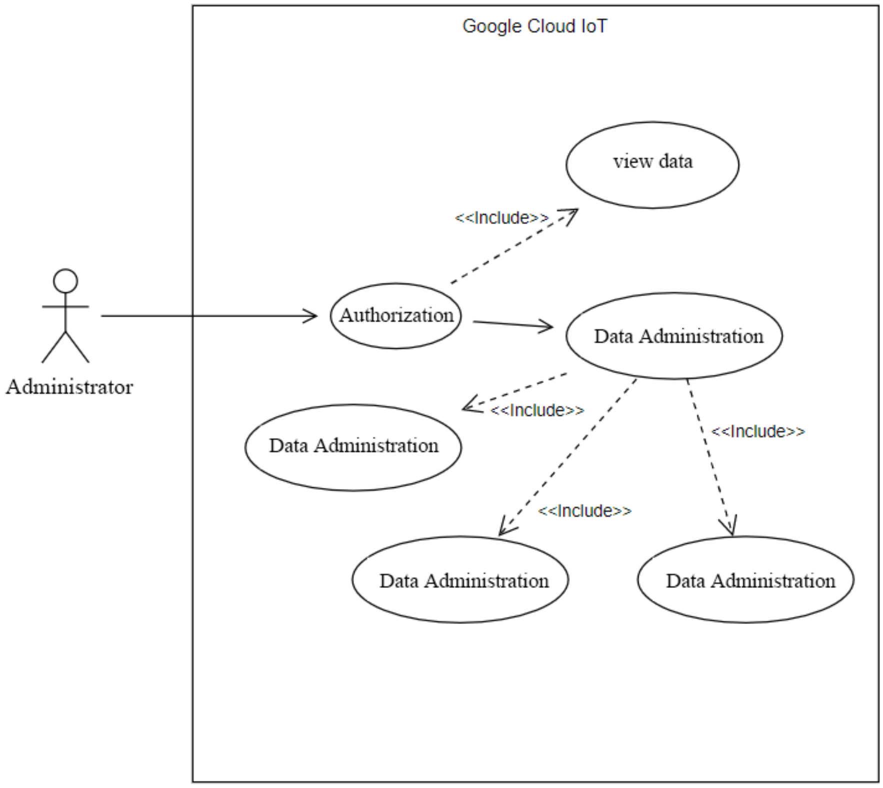


Figure 3.10 – Diagram of the variant development for administrator

The diagrams show the following actors: sensor and client attachment. The user of the client application can register or authorize. Through API-request is a proposal passing a proposal to the server honor of API. With the help of API on the client application, it is possible to get the last data. The sensor on API-request sends data to the database. Also, with the help of the dairy farm operator sensor will be able to manage the last data, if they turn out to be false. Through the SMTP server passes the email sending communicated with these former associates.

3.4 Class Diagrams of Database

Class diagram select the types of objects in the systems and the different types of domestic that interact between them. It gives a general idea of the attachment. The modeling method can work practically with all objective-oriented methods. A class can have its own reviews or inherit from other classes. The class diagram will help create code for software development. The advantage of class diagrams [18]:

– diagram of classes shows data models for eye integrated information systems;

– it provides a horizon of how the attachment is structured before learning the actual code. Something that can easily shrink during objectification;

– it helps to better understand the general attachment schemes;

– call up detailed diagrams that show the code needed for programming;

– it is usefull for developers and other interested parties.

The main elements of a UML class diagram are the name classes, attributes, and operations. The class name is required only in the graphical representation of the class. It is amplified in the highest section. A class is a plan of objects that can have single relationships, attributes, operations, and semantics. The class is pushed back in the vision of the brahmagon, involving the ego name, attributes and operations in remote compartments. When presenting a class, we will not be going to follow the rules (Figure 3.11):

– class name is always a long time to do with stuck letters;

– the name class should always be in the center of the first compartment;

– the class name should always be written in bold;

– the name of the abstract class should be written in italics.

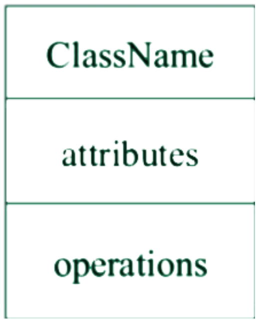


Figure 3.11 – Block diagram of the class from diagram classes

Attributes. The attribute is called the entire class, which is the description of the object being modeled. In the class diagram, this component is in the mood to hear below the names section.

Operations. Grace operations are working with attributes. The section with operations goes below the sections with attributes [18].

Class diagram structure of DB is shown in Figure 3.12. It has seven classes: ClientAppForm, SensorForm, ApiForm, DataWorker, AddingDataForm, ViewDataForm, DeleteDataForm. Concidere their structures.

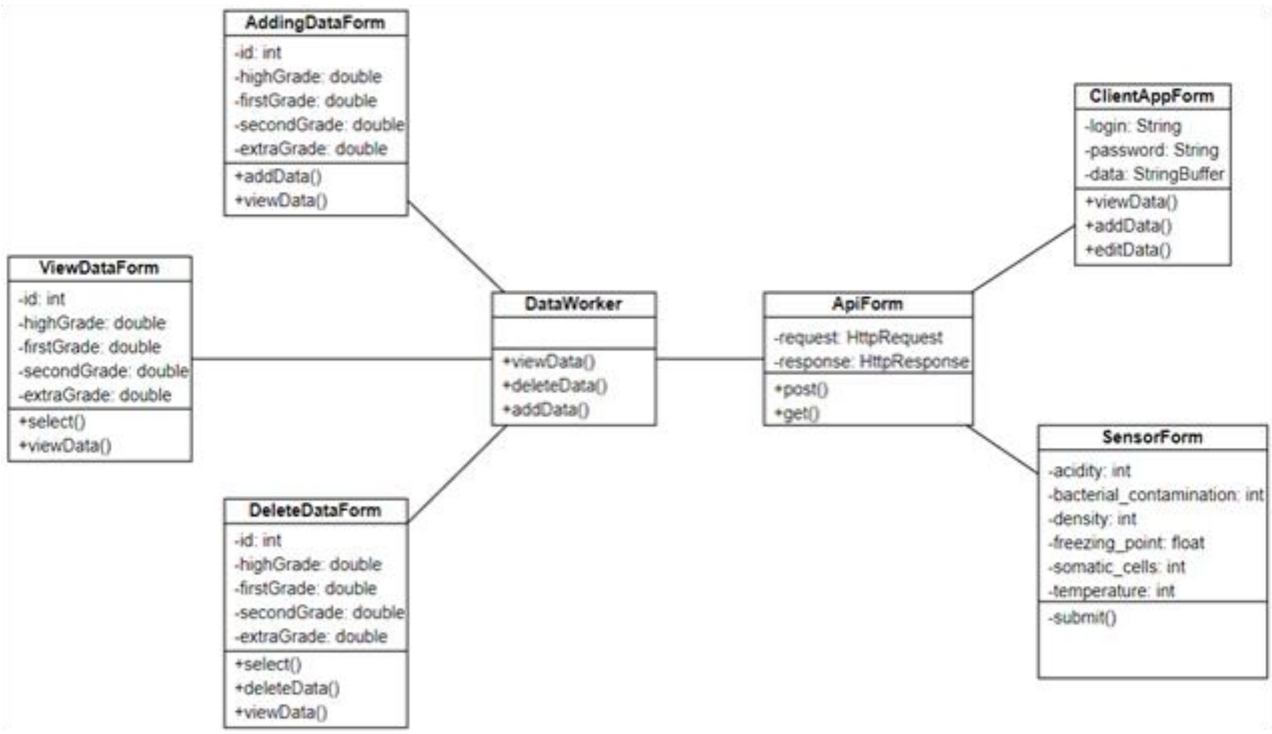


Figure 3.12 – Class diagramm structure of data base

Class ClientAppForm (Figure 3.13) has private attributes: login, password, data. It has public methods: viewData(), addData(), editData(). Two last depend from client. If he is administrator, he can delete and change of data.

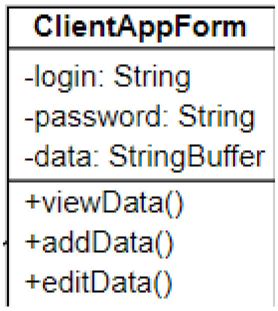


Figure 3.13 – ClienAppForm Class

SensorForm class (Figure 3.14) has class private attributes ocidity, bacterial_contaminatiown, density, freezing_point, somaratic_cells, temperature. Also contains a private submit() method.

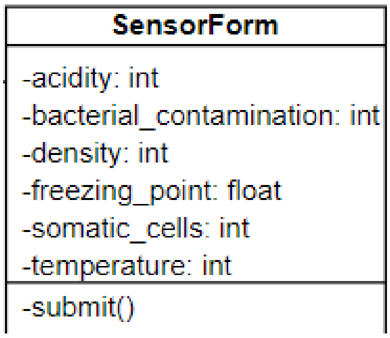


Figure 3.14 – SensorForm Class

The ApiForm class (Figurre 3.15) contains the private attributes request **response**. It also contains public methods Post() and Get(). This class serves in high-quality utilities, which can be transferred to others for forgiveness to the server part.

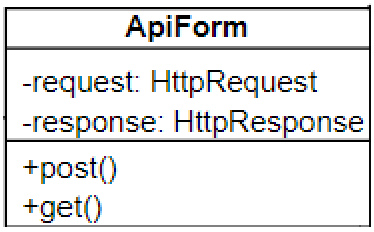


Figure 3.15 – ApiForm Class

The Dataworker class (Figure 3.16) contains publicnye metody ViewData (), deleteda (), adddata (). This class has become abstract. Its methods override classes that inherit from that DataWorker class. Successor classes: AddingDataForm, ViewDataForm, DeleteDataForm.

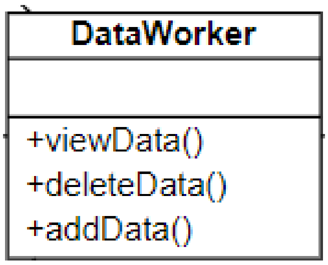


Figure 3.16 – DataWorker Class

The AddingDataForm class (Figure 3.17) contains private attributes id, highgrade, firstgrade, secondgrade, extradrade. It also contains the adddata() and ViewData() methods. The functional dannogo klassa is available to only operator administrator.

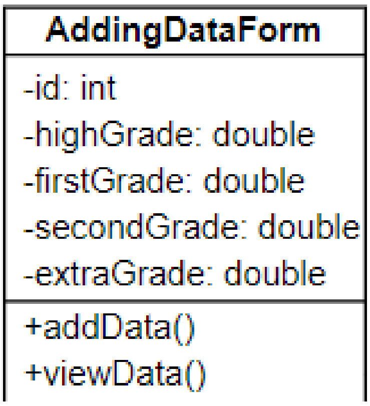


Figure 3.17 – AddingDataForm Class

The ViewDataForm class (Figure 3.18) contains private attributes id, highGrade, firstGrade, seconGgrade, extraGrade. It also contains the Select() and ViewData() methods. The functional of this class is available to all recipients.

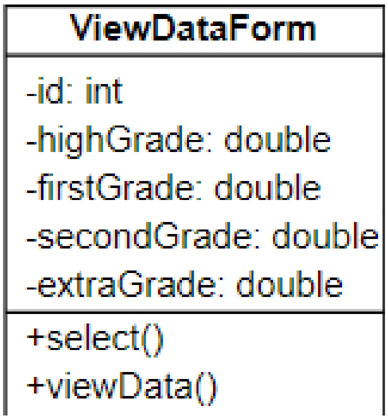


Figure 3.18 – ViewDataForm Class

The DeleteDataForm class (Figure 3.19) contains private attributes id, highGrade, firstGrade, secondGrade, extraGrade. It also contains the

Select(), deletteda(), ViewData() method. The functional of this class is available to the only operator administrator. The physical diagram of the structural database are given in Figure 3.20.

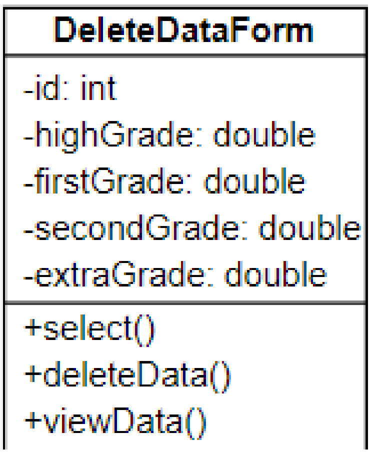


Figure 3.19 – DeleteDataForm Class

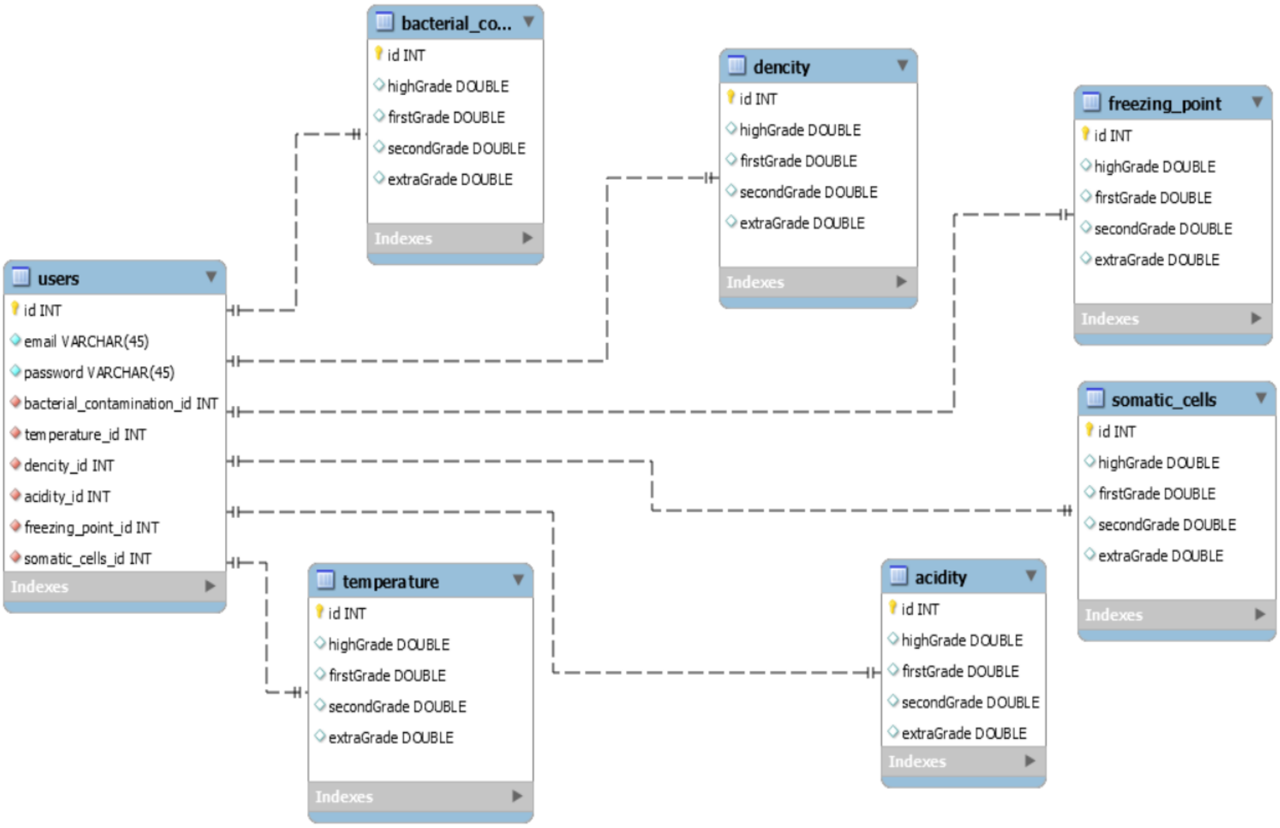


Figure 3.20 – The physical diagram of the structural database

3.5 Rationale for Choosing a Google Cloud IoT Cloud Platform

Google Cloud Platform (GCP) is a set of cloud services provided by Google that run on the same infrastructure that Google uses for its products intended for end users, such as Google Seagsh and YouTube. In addition to management tools, a number of modular general services are also provided, such as cloud computing, data storage, data analysis and machine learning. The infrastructure for working with the IoT is also based on it.

IoT means connecting things or specific devices to the Internet using network devices or routers, accessing them from a remote location and enabling them to communicate with each other and exchange data. This technique also has an autonomous function with which you can control any device without human intervention. Google Cloud Platform (GCP) is a set of cloud computing services provided by Google and running on the same platform that Google uses inside its end user products, such as YouTube and Google Search. Some of the GCP services offered by Google are [19]:

– Google BigQuery;
– Google Cloud DataFlow;
– Google App Epdipe;
– Google Cloud DataLab;
– Google Cloud SQL.

One of the most important services provided by Google in the field of the IoT is the Google Cloud IoT. Google Cloud IoT is a fully managed service that allows us to securely and easily connect, manage and receive data from devices connected to the Internet. In addition, it also allows other Google Cloud Platform services to collect, process, manage and visualize IoT data in real time.

Cloud IoT Soge is one of the services provided by Google in the field of the Internet of Things. This is useful when we have connected thousands of devices to the Internet, and at this time we need a well-scalable and managed service that is the core of cloud IoT. One of the best features of the IoT cloud core is that it automatically provides load balancing and data scaling. The two main components of Cloud IoT are:

– Device Manager – helps to register devices in services, and also provides a device authentication mechanism. It also supports a logical configuration for each device and can be used for remote device management from the cloud;

– Protokol Bridge – this is a way for a device to access the Google cloud or connect to it using some standard protocols, such as HTTP and MQTT. With this, we can use existing devices with minimal firmware changes.

The best part of using Google Cloud Platform is that it delivers unique offers for big data, artificial intelligence and IoT services. In addition, scalability is the main task when creating a project, therefore, thanks to its serverless architecture, GCP meets the requirements of the project [20].

Unlike the strategies of the IoT Amazon and Microsoft, which offer means to extend their cloud functions to a compatible local server, Google does not promote

Google Cloud expansion on the territory. Instead, the Cloud IoT element provides publishing and subscription interfaces for linking IoT telemetry with cloud applications, and Google supports various protocols that provide communication between the IoT peripheral device in the room and Google Cloud. However, together with Cloud IoT, a gateway is provided to support the autonomous operation of devices with limited resources. The gateway can replace the device when interacting with other Cloud IoT components, and then synchronize when the device is available. The Stackdriver monitoring allows you to create dashboards, alert and escalation flows and reports in real time or in summary form, as well as to set thresholds and report them. Stackdriver Logging provides logging of connections, traffic and errors, and also allows users to set thresholds and report exceptions based on these parameters. This is due to the Cloud Pub/Sub journals for publishing/subscribing to traffic and displaying the correlation between devices and events [21].

The Cloud Pub/Sub interface connects events with any Google analytics tools (Cloud Dataflow, BigQuery, Cloud Bigtable, ML and Google Dat Studio), as well as with third-party analytics or artificial intelligence applications that can use Cloud Pub/Sub. Google also offers a rich set of tools and applications using artificial intelligence and machine learning. They are available for IoT applications, but are not explicitly integrated with Google IoT offerings.

Although Google has some digital twinning tools at the business process level (a good example is its recent supply chain digital twin), it has not yet proposed digital duplication of IoT applications or IoT infrastructure.

As in the case of peripheral computing, Google may depend on relationships with external organizations to support a broader platform of digital IoT twins, but this has not yet been announced [21].

3.6 Data Processing on the Platform

The data received in Cloud IoT then goes to Pub/Sub, it is ideal for processing incoming IoT messages, and then allows subsequent systems to process them. The received data must be saved, so BigQuery cloud database on the Firebase platform is used, as well as the hosting of <url>. This hosting is used in the graduation project not only to optimize the SMTP server, but also for effective email distribution to all registered users. Tabulation takes place in the phpMyAdmin of the hosting <url>, then the database is exported. The database is imported to BigQuery, which reduces the time to create tables in BigQuery.

Creating the table in phpMyAdmin is shown in Figure 3.21.

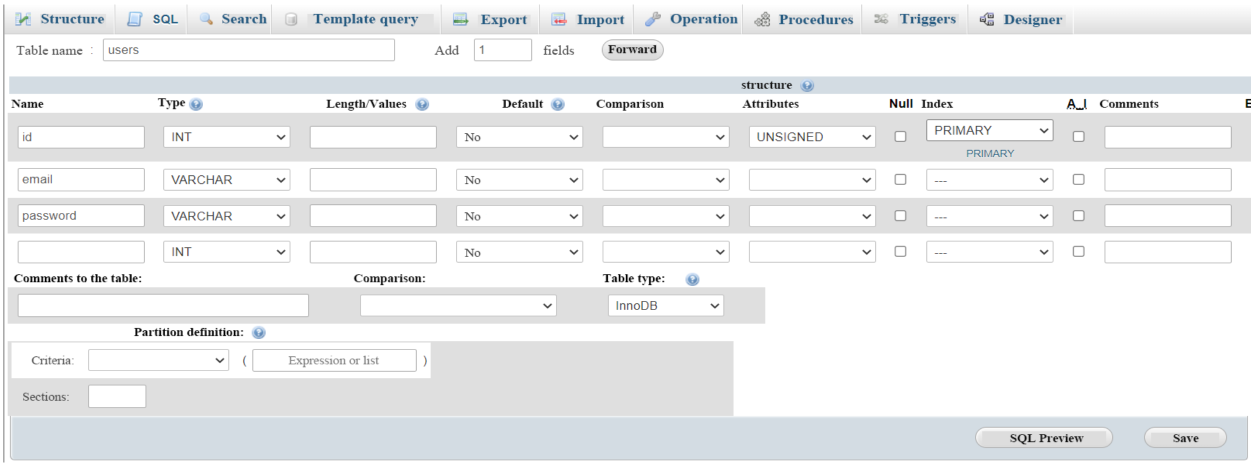


Figure 3.21 – Creating the isegs table in phpMyAdmin

Next, the created tables with already created users are displayed. Tables have been created, DB is exported and imported into the BigQuery database. The imported user table in the BigQuery DB is shown in Figure 3.22.

id	email	password
22	shedko.egor@inbox.ru	qweasd
47	yakovleva.darya2@gmail.com	qweasd

Figure 3.22 – Table with created users

The data in the BigQuery database is stored in the form of JSON (JavaScript Object Notation) in the First Database, from where the work with the data in BigQuery takes place (Figure 3.23). The table in Firebase is shown in Figure 3.24.

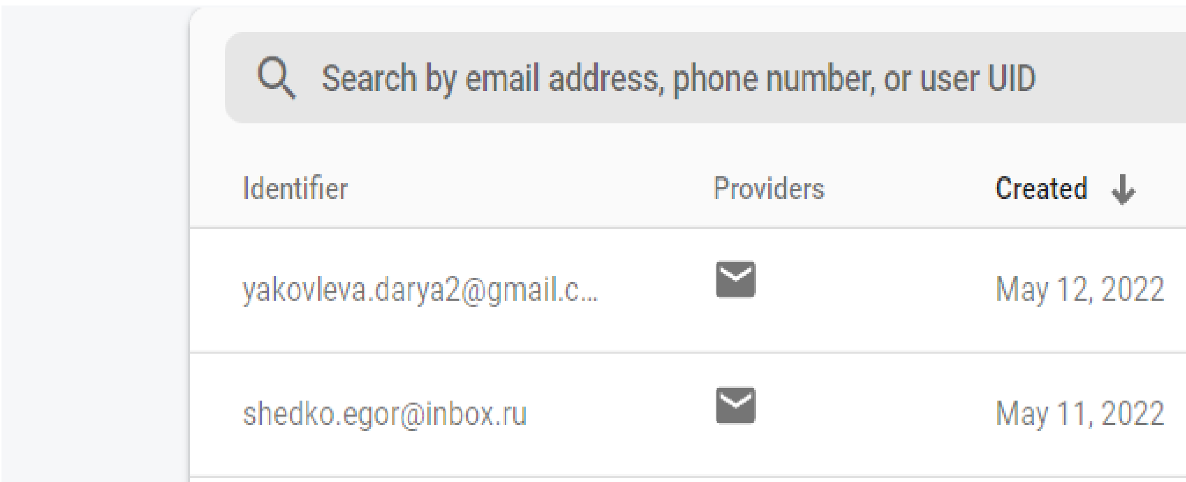


Figure 3.23 – Added users in the BigQuery database on the Firebase platform

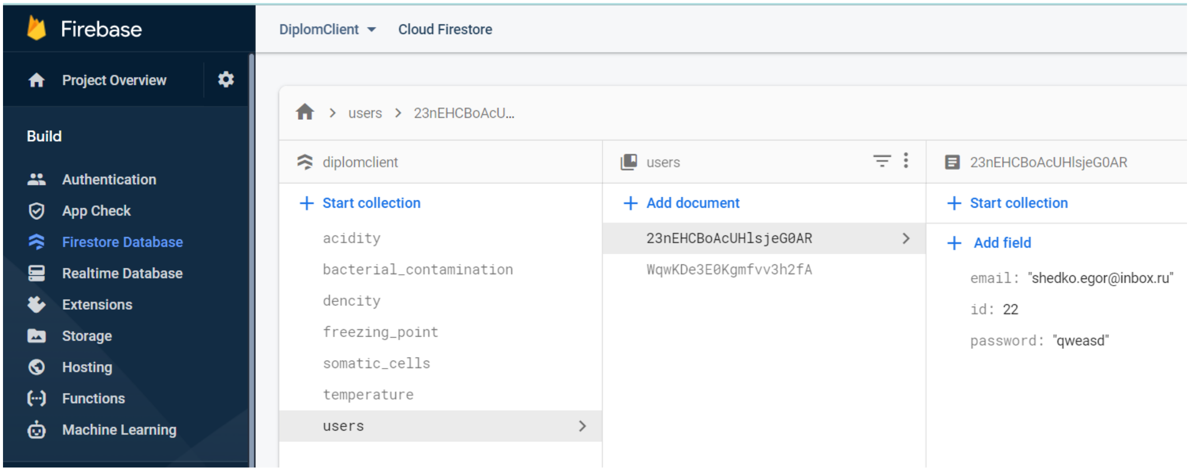


Figure 3.24 – A table with the data of the first user in the First Database

The data stored in the database can be conveniently viewed, modified or deleted. Only the network administrator can work with them.

3.7 Procedure for Transmitting Sensor Data

This subsection shows how sensor data is generated by the Cor-envir-ment board and transmitted to the IOT via the MQTT protocol. Sensor data is automatically published in the Pub/Sub IoT Sog. The data published in Pub/Sub automatically triggers a cloud function that processes the data and stores it in BigQuery. The data is uploaded to the server. To work with the server, a developed API is used, which transmits data on request [20].

The process of sending data from the sensor is as follows. When you click on the generate button, an icon is displayed on the sensor that the data was generated using the power of the board Cor Environmental sensor Bord. The response from the server is displayed on the icon (Figure 3.25).

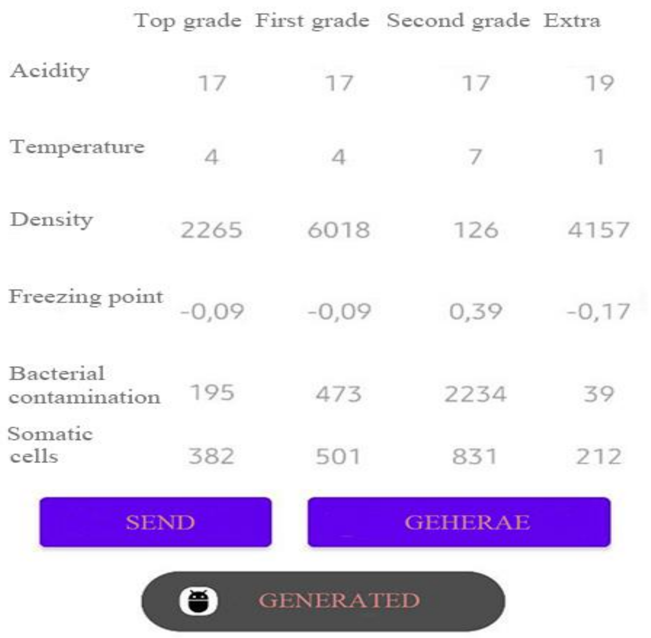


Figure 3.25 – Response after clicking the «Generate» button

After clicking on the Send button, the RT method is triggered, which displays the data on the sensor screen and via the MQTT protocol to the cloud database. The sent data is displayed on the sensor (Figure 3.26) and at the request of the client on the client application (Figure 3.27). After poisoning the data to the cloud server, cloud computing begins, based on the logic of PHP files sent to the server (Figure 3.28).

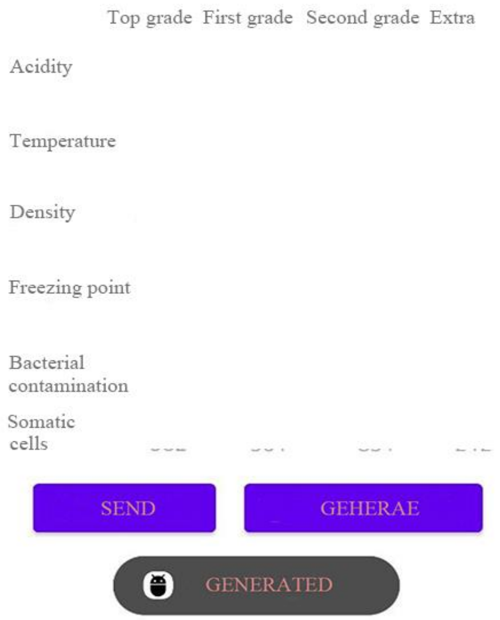
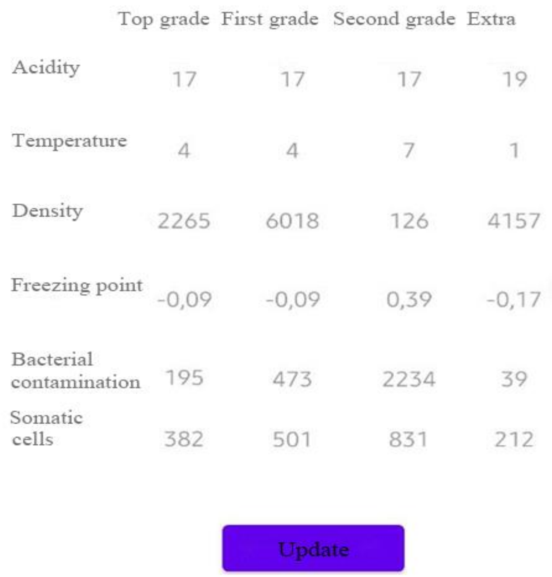


Figure 3.26 – Displaying data after clicking the Send button
with further editing to the database



	Top grade	First grade	Second grade	Extra
Acidity	17	17	17	19
Temperature	4	4	7	1
Density	2265	6018	126	4157
Freezing point	-0,09	-0,09	0,39	-0,17
Bacterial contamination	195	473	2234	39
Somatic cells	382	501	831	212

Figure 3.27 – Data display on the client application

Figure 3.28 – Sending a request to the server to communicate with the database

The server response displays «dataadd» Figure 3.29 shows the data in the database.

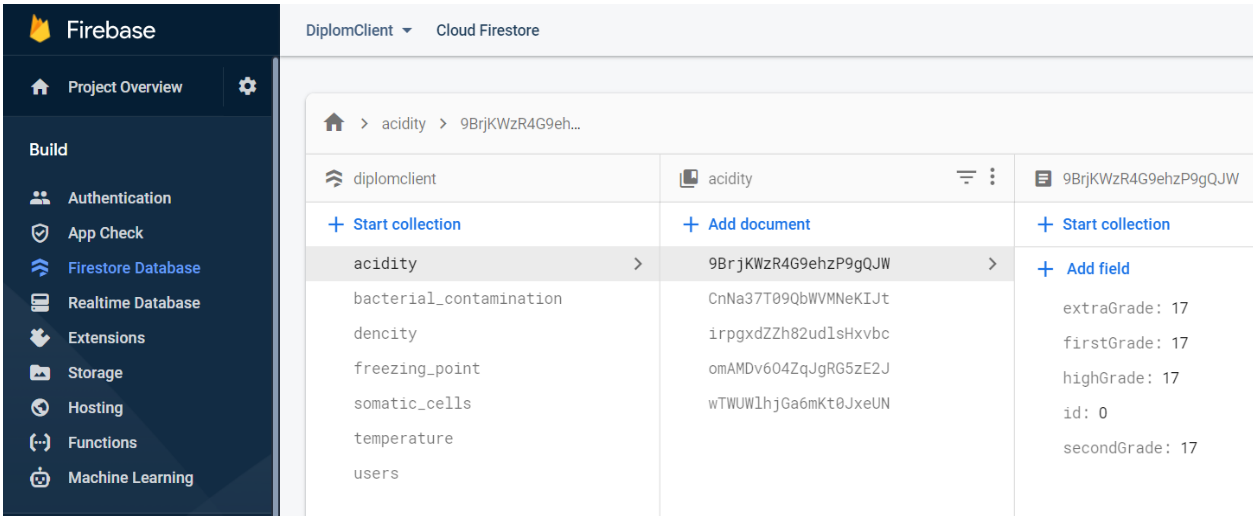


Figure 3.29 – Updated data of the first row of the acidity table

Also, the data is synchronously changed on the hosting of <url> (Figure 3.30).

id	highGrade	firstGrade	secondGrade	extraGrade
0	17	17	17	17

Figure 3.30 – The Acidity table after the update

Email notifications are sent out thanks to the SMTP server, which is built into the hosting of <url>. Thanks to him, at 12:00 every day, a newsletter is sent to all users who are in the «isegs» table. The positive effect is that this service is provided free of charge. In the case of using email newsletters from Cloud IoT, the service is paid. Thus, you can save money by using this software product.

3.8 Programming of the Milk Indicators Database

SQL was chosen as the programming language for developing a database of a distributed multi-agent milk quality assessment system.

To manage MySQL databases, we will use the phpMyAdmin tool [22]. phpMyAdmin is a program written in PHP and designed to manage a MySQL server over the world Wide Web. phpMyAdmin supports a wide range of operations on MySQL. The most frequently used operations are supported with the help of the user interface (database management, tables, fields, links, indexes, users, rights), at the same time it is possible to directly execute any SQL query.

Starting from version 3.0.0, phpMyAdmin joined the GoPHP5 initiative and has stopped supporting writing code for legacy versions of PHP and MySQL; version 3 and later require PHP 5.2 and MySQL 5. When used with legacy versions of PHP and MySQL, select the previous, but still supported, 2.x release branch, which can be found on the download page.

PhpMyAdmin has been released as stable code for more than ten years, for more information about the project and its history, see the main page. Its features:

– intuitive web interface;

– view and delete databases, tables, views, fields and indexes;

– create, copy, delete, rename and modify databases, tables, fields and indexes;

– server management, databases and tables, with tips for configuring the server;

– execution, revision and saving of any SQL expression, including batch requests;

– managing MySQL users and their privileges;

– working with stored procedures and triggers;

– support for importing data from CSV and SQL;

– administration of multiple servers;

– generation of visual database schemas in PDF format;

– creating complex queries using the template query function;

– global or partial database search;

– transformation of data into any format using a set of intended functions, such as displaying BLOB data in the form of an image or a download link.

As a server, we will use two servers: one is virtual, located on the Firebase platform, the second is also virtual, registered on the hosting of <url>. At the first stage, all operations concerning databases take place. On the second – all operations on working with the database, as well as email-mailing. Hosting byget.com contains the Apache server, MySQL, PHP script interpreter, phpMyAdmin and other add-ons designed for web development under Windows.

SQL is a language focused specifically on relational databases (RDBMS). It does a lot of work that would need to be done when using a universal programming language, for example, Java. To create a database on Java, it would be necessary to start from scratch. It would be necessary to create an instance of Conn (connection), then create an instance of Stat (assertion), which should be called when composing any SQL query. Also, in order to make an SQL query with unknown variables, it is necessary to use an instance of a pre-statement (prepared statement), which further increases the amount of code written [24].

There are two types of SQL: interactive and nested. For the most part, both forms work the same, but are used differently. Interactive SQL is used to function directly in the database to produce output for use by the customer.

Nested SQL consists of SQL commands placed inside programs that are usually written in a different language. This makes such programs more powerful and effective. However, admitting these languages, one has to deal with the SQL structure and data management style, which requires some extensions of interactive SQL. Passing SQL commands to nested SQL is passable («pass off») for variables or parameters used by the program in which they were nested.

In both interactive and nested SQL forms, there are numerous parts, or sub-divisions [24]:

– DDL (Data Definition Language) – the so called schema description language – consists of commands that create objects (tables, indexes, views, and so on) in the database;

– DML (Data Manipulation Language) is a set of commands that determine which values are represented in tables at any given time;

– DCD (Data Management Language) consists of tools that determine whether the user is allowed to perform certain actions or not.

These are not different languages, but sections of SQL commands grouped by their functions.

Not all types of values that can occupy table fields are logically the same. The most obvious difference is between numbers and text. It is not possible to put numbers in alphabetical order or subtract one name from another. Since systems with a relational database are based on connections between fragments of information, different types of data should clearly differ from each other, so that the corresponding processes and comparisons can be performed in them. In SQL, this is done by assigning each field a data type that indicates the type of value that this field can contain. All values in this field must have the same type.

Unfortunately, the definition of these data types is the main area in which most commercial database programs and the official SQL standard do not always coincide. The ANSI SQL standard recognizes only text and number type, while most commercial programs use other special types. Such as, DAT (DATE) and TIM (TIME) are in fact almost standard types (although their exact format varies). Some packages also support types such as MONEY and BINARY. MONEY is a special calculus system used by computers. All information in the computer is transmitted in binary numbers and then converted into other systems so that we can easily use them and understand them).

ANSI defines several different types of number values, the differences between which are quite subtle and sometimes they are confused.

The complexity of ANSI numeric types can be explained, at least in part, by the ability to make nested SQL compatible with a number of other languages.

Two types of ANSI numbers, INTEGER and DECIMAL (which can be abbreviated as INT and DEC, respectively), will be adequate for our purposes, as well as for the purposes of most practical business applications. Naturally, the INTEGER type can be represented as DECIMAL, which does not contain any digits to the right of the decimal point.

The type for the text is CHAR (or character), which refers to a line of text. A field of type CHAR has a certain length, which is determined by the maximum number of characters that can be entered in this field. Most implementations also have a non-standard type called VARCHAR (variable number of characters), which is a text string that can have any length up to a maximum defined by the implementation (usually 254 characters). The CHARACTER and VARCHAR values are included in single lines as «text». The difference between CHAR and VARCHAR is that CHAR must reserve enough memory for the maximum string length, while VARCHAR allocates memory as needed.

Character types consist of all printable characters, including numbers. However, the number 1 is not the same as the symbol «1». The symbol «1» is only another printed fragment of text that is not defined by the system as having a numeric value of 1. For example, 1 + 1 = 2, but «1» + «1» does not equal «2». Character values are stored in the computer as binary values, but are shown to the user as printed text. The conversion follows the format defined by the system used. This conversion format will be one of two standard types (possibly with extensions) used in computer systems: in ASCII code (used in all personal and small computers) and EBCDIC code (Extended Binary-Decimal Information Exchange Code) (used in large computers). Certain operations, such as alphabetizing the values of a field, will change along with the format.

Keywords are words that have a special meaning in SQL. They can be commands, but not text or object names.

SQL has certain special terms that are used to describe it. Among them are such words as query, sentence and predicate, which are the most important in the description and understanding of the language, but do not mean anything independent for SQL.

Commands, or sentences, are instructions that you use to access the SQL database. Commands consist of one or more separate logical parts called sentences. Sentences begin with the keyword for which they are named, and consist of keywords and arguments. Arguments complete or change the meaning of a sentence.

Commands, or sentences, are instructions that you use to access the SQL database. Commands consist of one or more separate logical parts called sentences. Sentences begin with the keyword for which they are named, and consist of keywords and arguments. Arguments complete or change the meaning of a sentence.

Objects are structures in the database that are named and stored in memory. They include basic tables, views (two types of tables), and indexes. To show you how teams are formed, we will do it with examples. There is, however, a more formal method of describing commands using standardized conventional designations. We will use it in later chapters for convenience in order to understand these conventions in case you encounter it in other SQL documents.

Square brackets «[]» will indicate the parts that may not be used, and ellipsis «...» indicate that everything preceding them can be repeated any number of times. The words indicated in angle brackets «< >» are special terms that explain what they are.

Queries are probably the most commonly used aspect of SQL. A query is a command that you give to your database program, and which tells it to output certain information from tables to memory. This information is usually sent directly to the screen of the computer or terminal you are using, although, in most cases, it can also be sent to the printer, stored in a file (as an object in computer memory), or presented as introductory information for another command or process.

During the implementation the project, there were seven tabs in the development of the database, using the toolkit [20, 21].

Tables are created by the CRITICAL TABLE command. This command creates an empty table – a table without rows. The values are entered using the INSERT DML command. The CRITICAL TABLE command basically defines the name of the table, in the form of a description of a set of column names specified in a certain order. It also defines data types and column sizes. Each table must have at least one column. Let's create tables for the following milk quality indicators:

1 Milk acidity index. Measured in Turner degrees (°T). The structure of the Acidity table is shown in Figure 3.31.

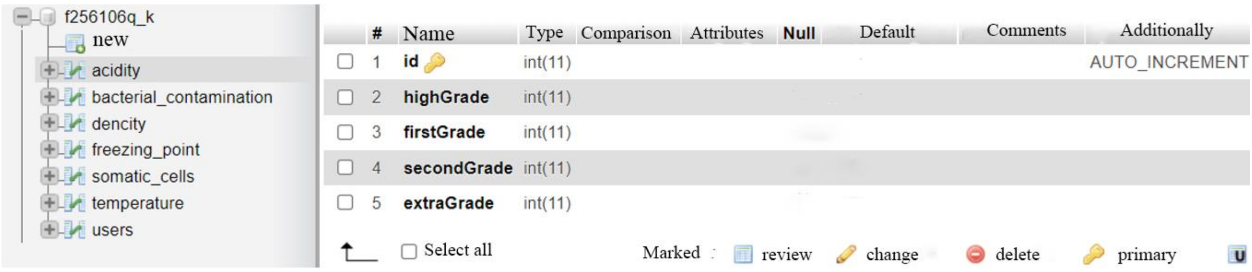


Figure 3.31 – Structure of the Acidity table

2 Iindicator of bacterial contamination of milk. It is measured in thousands per cubic centimeter (thousand/sm^3). The structure of the table Bactrial_contamination is displayed in Figure 3.32.

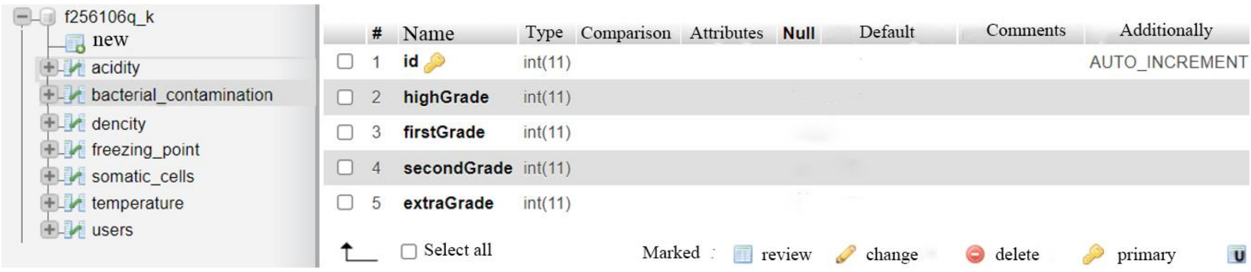


Figure 3.32 – Structure of the table Bactrial_contamination

3 Density indicator. It is measured in kilograms per cubic meter (kg/m^3). The structure of the Density table is shown in Figure 3.33.

4 Freezing point temperature indicator. Measured in degrees Celsius (oS). The structure of the Freshing_point table is shown in Figure 3.34.

5 An indicator of the volume of somatic cells. It is measured in thousands per cubic meter (thousand/m^3). The structure of the table is shown in Figure 3.35.

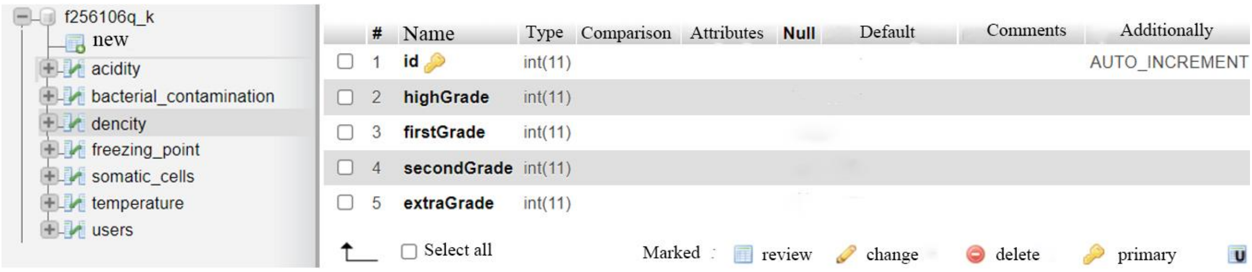


Figure 3.33 – Structure of the Density table

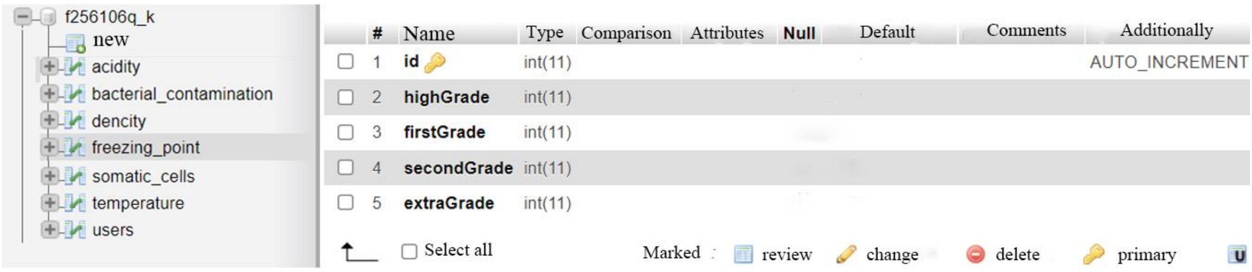


Figure 3.34 – Structure of the Front_point table

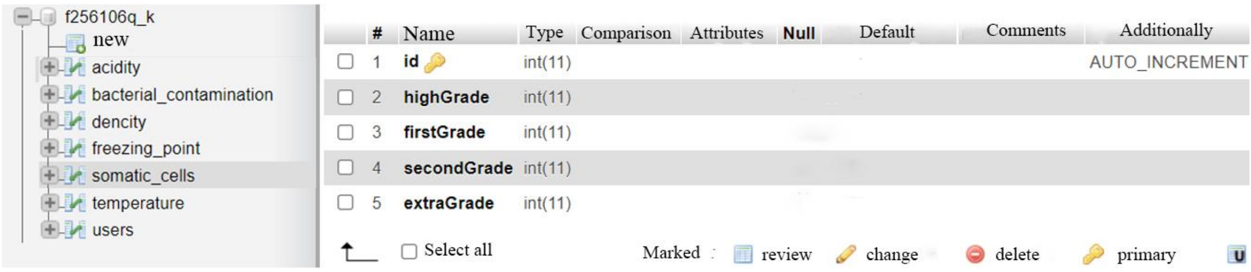


Figure 3.35 – Structure of the Somatic_cells table

6 Milk temperature indicator. Measured in degrees Celsius (oS). The structure of the table is shown in Figure 3.36.

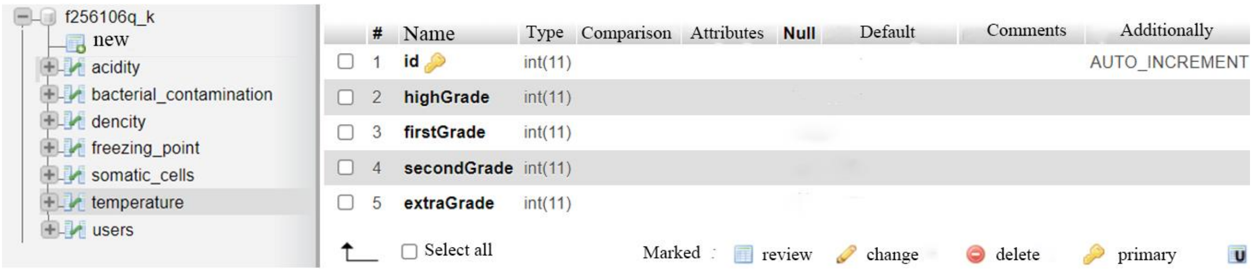


Figure 3.36 – Structure of the Temperature table

The data type INT is specified for the «id» field. The INT data type is the main type of integer data in the MySQL database. For the tables «acidity», «bactrial_contamination», «density», «soctic_cells», «temperature», the data type INT is specified for the fields «highGrade», «firstGrade», «secondGrade», «EXCEL». For the same fields, the data type FLOAT is set in the «front_point» table. The FLOAT data type stores high-precision numbers stored in floating-point form.

The NOT NULL limit is set for all fields. By default, the column can contain NULL values. The NOT NULL constraint ensures that the column does not accept NULL values. This applies a field to always contain a value, which means that you cannot insert a new record or update a record without adding a value to this field.

Separately, the keyword AUTO_INCREMENT must be specified for the «id» field. AUTO_INCREMENT allows you to create a unique number automatically when a new entry is inserted into the table. Often this is the primary key field that we would like to create automatically every time a new record is inserted. The

UNSIGNED attribute has also been added, it is optional for numeric data types in the MySQL DBMS, which means that a field will be created for storing non-signed numbers (greater than or equal to zero). That is, negative numbers, fields marked with the UNSIGNED flag, will not be able to store. This doubles the maximum allowed stored volume of the field.

Three fields have been created in the «isegs» table, as shown in Figure 3.37.

The NOT NULL limit is set for all fields. For the «id» field, the INT data type is specified and the AUTO_INCREMENT keyword is added.

The data type VARCHAR is specified for the «email» and «password» fields.

#	Name	Type	Comparison	Attributes	Null	Default	Comments	Additionally
1	**id**	int(11)						AUTO_INCREMENT
2	**email**	varchar(30)	utf8mb4_unicode_ci		Нет	*Нет*		
3	**password**	varchar(30)	utf8mb4_unicode_ci		Нет	*Нет*		

Figure 3.37 – Table of «isegs»

This data type represents a variable length string. The length of the stored string is also indicated in parentheses, for example, VARCHAR(30). However, unlike CHAR, the stored string will take up exactly as much space as necessary.

Using the CONSTRINT operator, you can specify a name for the foreign key constraint. To create a foreign key constraint, after the SIGN KEY, the table table that will represent the foreign key is specified. And after the key layer REFERNCES, the name of the linked table is indicated, and then in parentheses the name of the linked column to which the foreign key will point. After the REST statement, there are expressions ON DELETE and ON UPDATE, which specify the action when deleting and updating a row from the main table, respectively.

Unique indexes (UNIQUEINDEX) work almost the same way as primary keys. However, there can be only one primary key, and there are as many unique indexes as you want. It can create a primary key to an already existing table using several commands.

To create a foreign key constraint, after the SIGN KEY, the table table that will represent the foreign key is specified. And after the key layer REFERNCES, the name of the linked table is indicated, and then in parentheses the name of the linked column to which the foreign key will point.

The INSERT operator is used to fill in the database tables. The INSERT operator inserts new rows into an existing table. The form of this company is INSERT. VLUES inserts lines according to the exact values specified in the command [6].

3.9 Testing the IoT Database Based on a Cloud Platform

Software testing is the process of analyzing software and related documentation in order to identify defects and improve the quality of the product [25]. At the moment, there are the following types of database testing, which are divided into two groups:

– structural testing;

– functional testing;

– non-functional testing.

Structural data testing involves checking all those elements inside the data warehouse that are used primarily for data storage and that end users cannot directly manipulate.

Functional testing is divided into the following types:

– component or modular;

– integration;

– system;

– acceptance;

− functional;

− security testing;

− interaction testing.

Non-functional testing is divided into the following types:

− load testing;

− stress testing;

− stability and reliability testing;

− volume testing;

− installation testing;

− failure testing and recovery;

− configuration testing,

− localization testing;

− requirements testing.

Based on the above types of testing, a list of types of tests that will be conducted during the testing of the developed database was determined. The list includes:

– structural testing;

− functional testing [26].

During structural testing, the following will be checked:

– is the mapping of database fields and columns on the internal interface compatible with these mappings on the external interface;

– checking the correspondence of the length and naming of fields and columns, databases in accordance with the requirements;

– checking for any unused, unmapped database tables or columns;

– compatibility check.

During functional testing of the database, the following will be checked:

– integrity and consistency of data in the database;

– login and user data security.

Test case «Checking the compatibility of field and column mappings on the internal interface with these mappings on the external interface».

Initial conditions: the client application is installed on the mobile device and access to the database is open.

Stages of execution:

– open the app;

– enter your email and password;

– click on the «Register» button.

Expected result: the string format data matches the format of the columns «email» and «password». The actual result fully corresponds to the expected result.

Test case «Checking the correspondence of the length and naming of the fields and columns of the database in accordance with the requirements».

Initial conditions: the client application is installed on the mobile device and access to the database is open.

Stages of execution:

– open the app;

– enter user data;

– click on the «Log in» button;

– open the «isegs» table in Firebase.

Expected result:

– the «Authorization was successful» window appeared;

– the data in the table is displayed in the required cells, thereby proving that the form of the length and naming of fields and columns meet the requirements.

The actual result fully corresponds to the expected result.

Test case «Checking for any unused, unmapped database tables or columns».

Initial conditions: access to the database is open, the sensor application and the client application are installed.

Stages of execution:

– open the sensor database and application;

– run the script for generating milk indicators by clicking the «Generate» button;

– open the client application;

– enter user data;

– click the «Register» button;

– update the database and view the changes.

Expected result after updating the database:

– new data in all cells of tables with indicators;

– a new user has appeared in the «isegs» table and these cells are filled with the data entered by the user.

The actual result fully corresponds to the expected result.

Test case «Compatibility check».

Initial conditions: an arch file uploaded to a mobile device and the Android version is at least 12.

Stages of execution:

– open the file manager and open the arch file on the download path set by the user personally;

– install the ark file on your mobile device;

– open the sensor app;

– press any button from the app;

– open the client application;

– enter user data;

– click on the «Log in» button.

Expected result:

– the ark file has been installed;

– by clicking on the «Send» button of the sensor, the data was displayed, which means the stable operation of the application and database;

– by clicking on the «Log in» button of the client application, a table and the «Update» button were displayed, which means stable operation of the application and database.

The actual result corresponds to the expected result.

Test case «Checking the integrity and consistency of data in the database». Initial conditions: a client application is installed on a mobile device and open access to the database.

Stages of execution:

– open the client application;

– fill in the «email» and «password» fields with the data;

– click on the «Register» or «Log in» button if the user is registered;

– open the «isegs» table of the database.

Expected result:

– if enter an incorrect value in the «email» field, namely, the @ sign is not specified or a space is added, then the «Invalid login or password» window is displayed, which indicates that the database is protected from incorrect values;

– when entering a value in the «password» field consisting of three characters, or there is a space in the entered value, the «Invalid login or password» window is displayed, which indicates that the database is protected from incorrect values;

– after updating the database, incorrect user data was not saved in the table.

The actual result corresponds to the expected result. Thus, the integrity and consistency of the data is observed.

Test case «Login verification and user data security». Initial conditions: installed client application.

Stages of execution:

– open the app;

– enter login;

– enter the password;

– click the «Register» or «Log in» button.

Expected result:

– if the login or password is entered incorrectly, the «Invalid login or password» window is displayed;

– when you enter only a login or only a password, the «Error» window is displayed;

– with the correct login and password, the «Authorization was successful» window is displayed if the «Login» button was pressed, or the «Registration was successful» window if the «Registration» button was pressed.

Data security is provided by Google Cloud cloud platform. Google data processing centers use unique equipment, as well as special operating and file systems with an increased level of protection. Each of these components is optimized for maximum security and performance. All equipment is under the control of Google specialists. All data stored in Cloud Platform services and transmitted between the client and Google, as well as between data centers, is encrypted. In 2013, to protect against the latest cryptanalysis algorithms, Google doubled the length of the RS encryption keys to 2048 bits and began to change them regularly, thereby setting a new industry standard. The safety of user data is ensured [27].

The actual result corresponds to the expected result.

To shut down the application and exit it to the desktop, it is necessary to press the «Back» or «Home» button on the navigation system panel in any section of the application.

Conclusion on Chapter 3

1 The structure of the network of IoT for milk quality control has been developed. It includes portable milk quality analyzers for taking indicators from each of the controlled farms. In the structure, these indicators are fed to the gateways-converters-SHP. The latter are necessary for converting and transmitting the captured milk quality indicators to the cloud environment. A server is involved in the cloud environment. The server contains databases, a solver, and a website.

2 The data structures of the removed milk quality indicators are presented for relational representation in the database for their further display on the site pages. The solver generates messages if the milk quality indicators go beyond the control limits. The principles of forming database tables for the main controlled milk quality indicators are considered. The descriptions of the fields of the database tables are presented.

3 The process of creating an Internet of Things network and adding devices to it is demonstrated. sensor data is generated by the Coral Environmental Sensor Board and transmitted to the IoT Core via the MQTT protocol. Sensor data is automatically published to Pub/Sub IoT Core. The data published in Pub/Sub automatically launches a cloud function that processes the data and stores it in BigQuery.

4 A database for milk quality control has been created. The developed database will allow storing and processing a large amount of information. Structuring information allows you to analyze and process it: make user queries, selections, sorting. The information that is stored in the database can be constantly updated. Its relevance depends on how often this is done. Information about milk indicators can also be changed and supplemented.

5 In the process of database development, diagrams were obtained: use cases, classes. Based on the data processing technology, the necessary software tools were installed on the IoT platform. At the end of the development process, the database was tested, the results were presented in test cases. The database works with sensors and the client application can be used to store, create and edit data on product quality indicators.

References

1 Visniakou, U. A. Organization of management and structure in local networks internet of things / U. A. Visniakou, A. H. Al-Masri, S. K. Al-Haji // SA&AI. – 2020. – № 2. – P. 11–16.

2 Visniakou, U. A. Model and structure of the network internet of things for monitoring milk quality / U. A. Visniakou, A. H. Al-Masri, S. K. Al-Haji // SA&AI. – 2021. – № 1. – P. 39–44.

3 Visniakou, U. A. Structure, network protocols of the internet of things for quality production control / U. A. Vishniakou, B. H. Shaya, A. H. Al-Masri, S. K. Al-Haji // Siberian Journal of Science and Technology. – 2020 – № 4. – P. 478–482.

4 Vishniakou, U. A. Technologies of intelligence multiagent information processing with blockchain for management systems / U. A. Visniakou, B. H. Shaya, A. H. Al-Masri, S. K. Al-Haji // Open Semantic Technologies for Intelligent Systems (OSTIS-2019) : proceedings of the Intern. Scient. and Techn. Conf., Minsk, February 21–23, 2019 / Belarusian State University of Informatics and Radioelectronics ; editor : V. V. Golenkov (Chief Editor) [et al.]. – Minsk, 2019. – P. 311–314.

5 Visniakou, U. A. Technology of adaptive intelligence multiagent processing for information defense system design / U. A. Vishniakou, A. H. Al-Masri, S. K. Al-Haji // Coding and Digital Signal Processing in Infocommunications: materials of the Intern. Scient. and Practical Conf., Minsk, April 4, 2019 / Belarusian State University of Informatics and Radioelectronics ; editors : V. K. Konopelko, V. Yu. Tsvetkov, L. A. Shichko. – Minsk, 2019. – P. 99–103.

6 Vishniakou, U. A. Specialists training in area of networks IoT / U. A. Vishniakou, H. A. Al-Masry, K. S. Al-Haji // Informatization of education and methods of e-learning: digital technologies in education : materials of the IV Intern. Scient. Conf., Krasnoyarsk, October 6–9, 2020 : at 2 parts / under the general editorship of M. V. Noskov. – Krasnoyarsk : Siberian Federal University, 2020. – Part 1. – P. 403–406.

7 Visniakou, U. A. Automation tools for the development of internet of things network / U. A. Vishniakou, B. H. Shaya, A. H. Al-Masri, S. K. Al-Haji // Telecommunications: Networks and Technologies, Algebraic Coding and Data Security : materials of Intern. Scient. and Techn. Seminar, Minsk, November – December 2020 / Belarusian State University of Informatics and Radioelectronics ; editorial board : M. N. Bobov [et al.]. – Minsk, 2020. – P. 13–16.

8 Visniakou, U. A. Working with big data and database in the Internet of Things / U. A. Vishnyakou, K. S. Al-Haji // Telecommunications: Networks and Technologies, Algebraic Coding and Data Security : materials of Intern. Scient. and

Techn. Seminar, Minsk, November – December 2020 / Belarusian State University of Informatics and Radioelectronics ; editorial board: M. N. Bobov [et al.]. – Minsk, 2020. – P. 73–78.

9 Visniakou, U. A. Modeling a network of things on the basis of a cloud platform / U. A. Vishniakou, A. H. Al-Masri, S. K. Al-Hajj // Coding and Digital Signal Processing in Infocommunications: materials of the Intern. Scient. and Practical Conf., Minsk, April 19, 2021 / Belarusian State University of Informatics and Radioelectronics ; editors : V. K. Konopelko, V. Yu. Tsvetkov, L. A. Shichko. – Minsk, 2021. – P. 65–68.

10 Visniakou, U. A. IoT components for production quality monitoring / U. A. Vishniakou, A. H. Al-Masri, S. K. Al-Haji // Telecommunications: Networks and Technologies, Algebraic Coding and Data Security : materials of the Intern. Scient. and Techn. Seminar, Minsk, November – December 2021 / Belarusian State University of Informatics and Radioelectronics ; editor : M. N. Bobov [et al.]. – Minsk, 2021. – P. 26–30.

11 Vishnyakou, U. A. Model, structure and components of the Internet Network for dairy farm management / U. A. Vishniakou // Problem of infocommunications. – 2021. – № 1. – P. 36–39.

12 Milk quality analyzers [Electronic resource]. – Access mode : https://www.csm.brest.by/analizatory-kachestva-moloka. – Date of access : 04.25.2022.

13 Milkotronik ltd-Bulgaria-high-strength enterprise tested milk [Electronic resource]. – Access mode : https://lactoscan.coм/?p=20&c=3&l=5. – Access date : 04.25.2022.

14 Milk analyzer EMOMILK [Electronic resource]. – Access mode : http://ekomilk.ru/product/view/22. – Date of access : 04.26.2022.

15 I turn on the COM port directly to the Cloud [Electronic resource]. – Access mode : https://moxa.RGO/News/New/podklyuchaem-som-Port-parguamiui-v-oblako. – Date of access : 04/26/2022.

16 MySQL e-book [Electronic resource]. – Access mode : https:// scienceforum.ru/2016/article/2016027811. – Data of access : 04.26.2022.

17 UML variant use diagram [Electronic resource]. – Access mode : https://coderlessons.coм /Tutorials/kompiuternoe-programmironie/uchebnik-uml/7-diagramma-variantov-ispolzovaniia-uml. – Data of access : 04.27.2022.

18 UML class diagram [Electronic resource]. – Access mode : https://coderlessons.coм/Tutorials/kompiuternoe-programmironie/uchebnik-uml/5-diagramma-klassov-uml. – Date of access : 04.27.2022.

19 Google Cloud Platform – introduction in BigQuery [Electronic resource]. – Access mode : https://translated.turbopages.org/proxy_u/en-ru.ru.b514e267-

62815e55-c4e58c60-74722d776562/https/www.geeksforgeeks.org/google-cloud-platform-introduction-to-bigquery. – Data of access : 19.04.2022.

20 Querry Guide to Goolge Analytics: arr + web [Electronic resource]. – Access mode: https://www.симоахава.som/Analytics/BigQuery-query-Guide-Google-Analytics-arr-web. – Data of access : 04.28.2022.

20 SQL to create [Electronic resource]. – Access mode : https://ruseller.com/lessons.php?id=557. – Date of access : 04.28.2022.

21 SQL. Database management systems [Electronic resource]. – Access mode : https://xsltdev.ru/php/tutorial/db/sql/. – Data of access: 04.29.2022.

22 PhpMyAdmin Documintation [Electronic resource]. – Access mode : https://www.phpMyAdmin.net/Docs. – Data of access : 30.04.2022.

23 Google Cloud IoT Soge documettatione [Electronic resource]. – Access mode : https://cloud.Google.com/IoT/Docs/. – Date of access : 01.05.2022.

24 SQL-documentation on the creation of the primary key [Electronic resource]. – Access mode : https://voencms.ru/sozdanie-pervichnogo-kljucha-sql-dlja-chego-nuzhno. – Date of access : 01.05.2022.

25 Software testing [Electronic resource]. – Access mode : https://www.bsuir.by/m/12_108786_1_98216.pdf. – Date of access: 05.05.2022.

26 Database testing [Electronic resource]. – Access mode : https://coderlessons.coм/tutorials/kachestvo-programnogo-obespecheniia/testirova nie-baz-dannykh/testirovanie-baz-dannykh. – Date of access : 05.05.2022.

27 Google Cloud Platform Console [Electronic resource]. – Access mode : https://support.Google.coм/Cloud/cnswer/6262505?hl=ru#zippy=. – Date of access : 05.05.2022.

4 MODEL, STRUCTURE, ALGORITHMS OF IoT SYSTEM FOR PROCESSING OF ENVIRONMENTAL SOUND INFORMATION

4.1 Basis of Distributed Multiagent System for Processing Sound Information of en Environment

Measuring, registering and analyzing the various levels of sounds and their effect on the surrounding areas is a very complex process. In fact, the amplitude and effect of sound waves vary considerably across the continuous spatio-temporal dimensions. For instance, the noise produced by a taking-off airplane is perceived by its neighborhood with varying amplitudes over time: it starts loud, then decreases gradually while it is flying away. On the other hand, industrial machines typically expose the workers to continuous or repetitive cycles of noise. Hence, modeling the various types of sounds and their spatio-temporal effects becomes crucial.

Using a model that embeds the representation of sounds properties on sources, and the rules that govern their propagation across the various surrounding mediums would help in both, tracking the historical variations and predicting the future changes of sounds properties along the spatio-temporal dimensions. In fact, such a model can represent levels of noise in a large urban space and help in studying noise pollution at various layers: inside a given building, in a specific public park or around the whole city. It shall also help in predicting how spatio-temporal changes may affect the levels of noise pollution at any of these layers, for instance when a new building complex or a compound community take place in the city.

The region of Metz in France made a study on the geographical distribution of noise in the region [1], in accordance with the decision of the French ministry of ecology amended in 2006, and which obliges all the French municipalities and regions to perform exhaustive studies on urban noise effects. The study showed, not only the effect of the detected high levels of noise pollution, but also the need for more elaborated models and tools to further understand the collected data and represent its time and space variations. Another study made in Dalian Municipality of Northeast China [2] showed the potential of the «Land Use Regression Method» in accurately studying noise effects at three special scales, yet the temporal aspects of the noise variations were not completely taken into account.

Consider the task of recognizing environment sounds for the understanding of a scene (or context) surrounding an audio sensor. By auditory scenes, refer to a location with different acoustic characteristics such as a coffee shop, park or quiet hallway. Consider, for example, applications in robotic navigation and obstacle detection, assistive robots, surveillance, and other mobile device based services. Many of these systems are dominantly vision-based. To understand unstructured environments, their robustness or utility will be lost if visual information is compromised or totally absent. Audio data could be easily acquired, in spite of challenging external conditions such as poor lighting or visual obstruction, and is relatively cheap to store and compute than visual signals. To enhance the system's context awareness, we need to incorporate and adequately utilize such audio information [1, 2].

Research in general audio environment recognition has received some interest in the last few years [3], but the activity is much less as compared to that for speech or music. Other applications include those in the domain of wearable's and context-aware applications [4]. Unstructured environment characterization is still in its infancy. Most research in environmental sounds has centered mostly on recognition of specific events or sounds [5]. To date, only a few systems have been proposed to model raw environment audio without pre-extracting specific events or sounds [6]. Similarly, our focus is not in analyzing and recognition of discrete sound events, but rather on characterizing the general acoustic environment types as a whole.

Despite the reduced emphasis in modern times on listening to sounds other than speech or music, humans retain the ability to identify a wide range of sounds, and can still make subtle discriminations among sounds that have particular importance. One major issue in building a recognition system for multimedia data is the choice of proper signal features that are likely to result in effective discrimination between different auditory environments. Environmental sounds are considered unstructured data, where the differences in the characteristics to each of these contexts are caused by random physical environment or activities from humans or nature.

Unlike music or speech, there exist neither predictable repetitions nor harmonic sounds. Because of the nature of unstructured data, it is very difficult to form a generalization to quantify them. In order to obtain insights into these data, we performed an analysis to evaluate the characteristics from a signal processing point of view. There are many features that can be used to describe audio signals. The choice of these features is crucial in building a pattern recognition system. Therefore, we examined a wide range of features in order to evaluate the effect of each feature and to select a suitable feature set to discriminate between the classes.

Acoustic features can be grouped into two categories according to the domain in which they are extracted from: frequency-domain (spectral features) and time-domain (temporal features). The temporal information is obtained by reading the amplitudes of the raw samples. Two common measures are the energy and zero-crossing rates [7]. Short-time energy:

$$E_n = \frac{1}{N}\sum_m [x(m)w(n-m)]^2 ,$$

where $x(m)$ – the discrete time audio signal; n – the time index of the short-time energy; $w(m)$ – the window of length N; E_n – energy provides a convenient representation of the amplitude variation over time.

Zero-crossings occur when successive samples have different signs. It is a simple measure of the frequency content of a signal. Short-time average zero-crossing rate (ZCR):

$$Z_n = \frac{1}{2}\sum_m |sgn\,[x(m) - sgn\,[x(m-1)]|w(n-m),$$

where

$$sgn[x(n)] = \begin{cases} 1\,, x(n) \geqslant 0, \\ -1, x(n) < 0. \end{cases}$$

Similarly, $w(m)$ is the window of length N. Since the energy level varies depending on the distance from the sound source, we use the range of the short-time energy as a measure and feature, instead of the average.

Classification of sound information from the environment. Three different classification methods were investigated: KNearest Neighbors (KNN) [8], Gaussian Mixture Models (GMM) [9], and Support Vector Machine (SVM) [10]. For KNN, we used the Euclidean distance as the distance measure and the 1-nearest neighbor queries to obtain the results. As for GMM, we set the number of mixtures for both training and testing to 5. For the SVM classifiers, we used a 2-degree polynomial as its kernel with regularization parameter C = 10 and the epsilon $\varepsilon = 1e^{-7}$, which controls the width of the e-insensitive zone, which used to fit the training data, affecting the number of support vectors used. Since SVM is a two-class classifier, we used the one-against-the-rest algorithm [10] for our multi-class classification in all of the experiments. We performed leave-one-out cross-validation on the data. The recognition accuracy using leave-one-out cross-validation was found from calculating:

$$\text{Accuracy} = \frac{\text{\# of correctly classified}}{\text{Tota\# of dataset}}.$$

There are many features that can be used to describe audio signals.

A multiagent system (MAS) is an extension of the agent technology where a group of loosely connected autonomous agents act in an environment to achieve a common goal. This is done either by cooperating or competing, sharing or not sharing knowledge with each other. Multiagent systems have been widely adopted in many application domains because of the beneficial advantages offered. Some of the benefits available by using MAS technology in large systems [8] are:

1) an increase in the speed and efficiency of the operation due to parallel computation and asynchronous operation;

2) a graceful degradation of the system when one or more of the agent fail. It thereby increases the reliability and robustness of the system;

3) scalability and flexibility – agents can be added as and when necessary;

4) reduced cost – this is because individual agents cost much less than a centralized architecture;

5) reusability – agents have a modular structure and they can be easily replaced in other systems or be upgraded more easily than a monolithic system.

Though multiagent systems have features that are more beneficial than single agent systems, they also present some critical challenges. Some of the challenges are highlighted in the following section.

Environment: In a multiagent system, the action of an agent not only modifies its own environment but also that of its neighbors. This necessitates that each agent must predict the action of the other agents in order to decide the optimal action that would be goal directed. This type of concurrent learning could result in non-stable behavior and can possibly cause chaos. The problem is further complicated, if the environment is dynamic. Then each agent needs to differentiate between the effects caused due to other agent actions and variations in environment itself.

Perception: In a distributed multiagent system, the agents are scattered all over the environment. Each agent has a limited sensing capability because of the limited range and coverage of the sensors connected to it. This limits the view available to each of the agents in the environment. Therefore decisions based on the partial observations made by each of the agents could be sub-optimal and achieving a global solution by this means becomes intractable.

In agent system, it is assumed that an agent knows its entire action space and mapping of the state space to action space could be done by experience. In MAS, every agent does not experience all of the states. To create a map, it must be able to learn from the experience of other agents with similar capabilities or decision making powers. In the case of cooperating agents with similar goals, this can be done easily by creating communication between the agents. In case of competing agents it is not possible to share the information as each of the agents tries to increase its own chance of winning. It is therefore essential to quantify how much of the local information and the capabilities of other agent must be known to create an improved modeling of the environment.

Conflict resolution: Conflicts stem from the lack of global view available to each of the agents. An action selected by an agent to modify a specific internal state may be bad for another agent. Under these circumstances, information on the constraints, action preferences and goal priorities of agents must be shared between to improve cooperation. A major problem is knowing when to communicate this information and to which of the agents.

Inference: A single agent system inference could be easily drawn by mapping the State Space to the Action Space based on trial and error methods. However in MAS, this is difficult as the environment is being modified by multiple agents that may or may not be interacting with each other. Further, the MAS might consist of heterogeneous agents, that is agents having different goals and capabilities. These may be not cooperating and competing with each other. Identifying a suitable inference mechanism in accordance of the capabilities of each agent is crucial in achieving global optimal solution.

It is not necessary to use multiagent systems for all applications. Some specific application domains which may require interaction with different people or organizations having conflicting or common goals can be able to utilize the advantages presented by MAS in its design.

The classification of MAS is a difficult task as it can be done based on several different attributes such as Architecture [12], Learning [13–15], Communication [12], Coordination [16]. Architecture based on the internal structures of the particular individual agents forming the multi-agent system, it may be classified as two types: homogeneous structure, heterogeneous structure.

Hierarchical Organization [17]: is one of the earliest organizational design in multiagent systems. Hierarchical architecture has been applied to a large number of distributed problems. In the hierarchical agent architecture, the agents are arranged in a typical tree like structure. The agents at different levels on the tree structure have different levels of autonomy. The data from the lower levels of hierarchy typically flow upwards to agents with a higher hierarchy. The control signal or supervisory signals flow from higher to a lower hierarchy [18]. The flow of control signals is from a higher to lower priority agents.

In a holonic multiagent system, an agent is a single entity and may be composed of many sub-agents bound together by commitments. The sub-agents are not bound by hard constraints or by pre-defined rule but through commitments. These refer to the relationships agreed to by all of the participating agents inside the holon. Each holon appoints or selects a Head Agent that can communicate with the environment or with other agents located in the environment. The selection of the head agent is usually based on the resource availability, communication capability and the internal architecture of each agent. In a homogeneous multiagent system, the selection can be random and a rotation policy could be employed similar to that used with distributed wireless sensor networks. In the heterogeneous architecture, the selection is based on the capability of the agent. The holons formed may group further in accordance to benefits foreseen in forming a coherent structure.

In coalition architecture, a group of agents come together for a short time to increase the utility or performance of the individual agents in a group. The coalition ceases to exist when the performance goal is achieved. The agents forming the coalition may have either a flat or a hierarchical architecture. Even when using a flat architecture, it is possible to have a leading agent to act as a representative of the coalition group. The overlap of agents among coalition groups is allowed as this increases the common knowledge within the coalition group. It helps to build a belief model.

Team MAS architecture [19] is similar to coalition architecture in design except that the agents in a team work together to increase the overall performance of the group. Rather than each working as individual agents. The interactions of the agents within a team can be quite arbitrary and the goals or the roles assigned to each of the agents can vary with time based on improvements resulting from the team performance.

Reference [20] deals with a team based multiagent architecture having a partially observable environment. In other words, teams that cannot communicate with each other has been proposed for the Arthur's bar problem. Each team decides on whether to attend a bar by means of predictions based on the previous behavioral pattern and the crowd level experienced which is the reward or the utility received associated with the specific period of time. Based on the observations made in [17], it can be concluded that a large team size is not beneficial under all conditions. Consequently some compromise must be made between the amount of information, number of agents in the team and the learning capabilities of the agents. Large teams offer a better visibility of the environment and larger amount of relevant information.

The author propose is the structure of multiagent system of sound information (MASSI) monitoring of surrounding environment. MASSI structure has many agents such as sound transformation, analysis of the information received from sound agents, decision-making. It implements the functions to e-sure the required class of protection of people (working or living) and allows to implement an environmental safety system [21]. MASSI can handle noise levels in urban space and help with learning noise pollution of various areas: inside the building, in a public park or around the entire area, increasing the protection of the space to the required level.

1 Modeling different types of sounds and their spatiotemporal effects becomes important for assessing the sound situation both in working spaces and in recreation areas. Two estimations of a sound situation are given: on the basis of short-term

energy and average speed of change. Three different methods of classification are investigated: KNearest neighbors, Gaussian mixture model and support vector machine, the latter being preferred.

2 The multiagent system is considered as an extension of agent technology, in which a group of related autonomous agents acts in an environment for information processing. Its characteristics are given, the classification of MAC, trends in the use of multi-agent intelligent technologies for information processing are presented.

3 It is proposed to use a multiagent system for monitoring sound information in the environment, which includes a variety of agents for the transformation of sound information, its analysis and decision-making on environmental assessment. MASSI can adjust noise levels in the urban environment and help in the study of its pollution from various noise.

4.2 Model and Structure of Multiagent System for Sound Monitoring

The most important classification methods for information processing use Hidden Markov Models (HMM), Gaussian Mixture Models and Support Vector Machines, which are discussed in below, although there are other useful methods that are summarized as follows [22].

K-Nearest Neighbours (k-NN): a simple algorithm that, given a testing pattern, uses the majority vote of the *k* nearest training patterns to assign a class label. It is often described as a lazy algorithm, as all computation is deferred to testing, and hence can have a slow performance for a large number of training samples. For the case of the 1-NN, the method has a 100 % recall performance, which is unique.

Dynamic Time Warping (DTW): this algorithm can find the similarity between two sequences, which may vary in time or speed. This works well with the *bag-of-frames* approach, as it can decode the same word spoken at different speeds. However, it has largely been superseded by HMM for Automatic Speech Recognition (ASR).

Artificial Neural Networks (ANN): this method, also referred to as a Multi-Layer Perceptron (MLP), is a computational model inspired by neurons in the brain. Given a sufficient number of hidden neurons, it is known to be a universal approximator, but is often criticized for being a black-box, as the function of each neuron in the network is hard to interpret. It also suffers from difficulty in training, as the most common method of back propagation is likely to get stuck in local minima.

There are many features that can be used to describe audio signals. The feature vector for the experiments consisted of features summarized in [8, 23].

Consider the authors' approach to the symbolic description of a multiagent model for collecting and processing sound information (CPSI) from the environment, based on the concept of object algebra [25]. Let's present this model as a set of agents [24, 26, 27]:

$$MAS_{CPSI} = (A_{IP}, A_{FC}, A_{IDB}, A_{CM}, A_{KB}, A_{DS}, A_{UI}),$$

where A_{IP} – agent of input and preprocessing of sound information; A_{FC} – filtering and classification agent; A_{IDB} – agent working with data base of sound information; A_{CM} – conceptual modeling agent; A_{KB} – agent working with knowledge base of sound information classification; A_{SD} – agent for of decision support (DS) making about sound situation; A_{UI} – agent of user interface.

In general this model works including several algorithms. Describe the main of them. The first algorithm starts from work of many agents A_{IP} which input and collect local sound information in some region and send it to the A_{FC} agent which will also get information from the A_{CM} agent and then send it to the agent A_{DS}. It works with agents A_{IDB}, A_{KB}, A_{UI} (data and knowledge bases, DS and user interface). The second algorithm have the same cycle as the first one, but in this case if there is some sound information missing so the A_{DS} agent will get back to the A_{IP} agent to get the needed information, and then the cycle will complete as the first one to get a better result.

The third algorithm is that: before the A_{CM} agent give information to the A_{FC} agent it will communicate with the A_{IP} agent and get the required information.

The structure of multiagent system for Collection & Processing Sound Information (CPSI) is composed of several different components that works together to collect sounds from the environment to get a required output (Figure 4.1). These components can be classified in two parts. The first part is many detecting and collecting sounds processors from the environment (A_{IP} agents of model), while the other part is software-hardware realization on Server (A_{FC}, A_{IDB}, A_{CM}, A_{KB}, A_{DS}, A_{UI} agents of model) [27].

The detecting or collecting processor is composed of detectors or sound sensors so that there number of detectors will take in to consideration the zone or place that will be covered so that we will insure that we collect all the sound waves from a specified environment or zone.

These sensors (SDi) will be connected to the server by the usage of wireless connection not wired, so that we will have the flexibility to relocate the sensors depending on the zone that it necessary to cover, and at the same time to insure that these detectors can be dynamic detectors not static, and also be flexible with different types of environment.

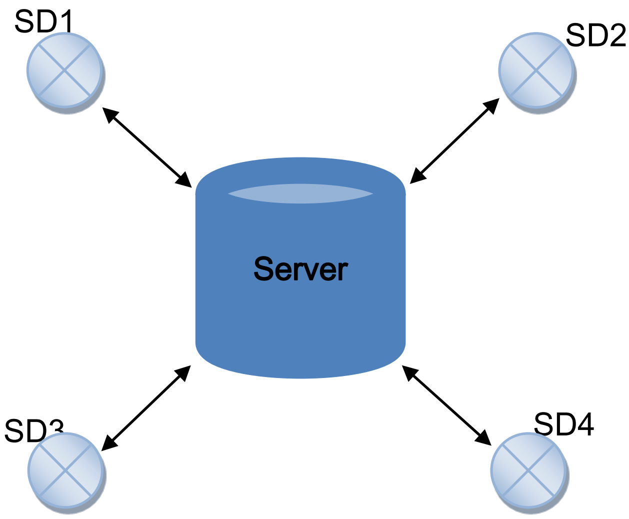


Figure 4.1 – MAS for Collection and Processing Sound Information

These components of MAS CPSI realizes the model special algorithms given above. Common algorithm of MAS will allow specialist to select a specific sound length or sound range. If user willing to collect not only this but also avoid duplication in sound collecting so that if two or more detectors (SD) collect the same sound source from the environment and send it to the Server, it will drop all the common sound waves to decrease the data that will be studied and select the most clear data that was collected with the sensors by the help of special methods that are provided by the second part of these agents as shown in (see Figure 4.1).

4.3 Optimization of the Choice of IoT Network Protocol for Monitoring Audio Information

Currently, there are several standards for data transmission in IoT networks. Let's consider some of them in more detail [28].

1 The Wireless USB standard is a wireless data transmission standard developed by Wireless USB Promoter Group. During the development, much attention was paid to improving energy efficiency. Devices manufactured in accordance with Specification 1.1 consume less power in idle mode. Wireless USB 1.1 supports NFC technology, which simplifies the setup and operation of wireless USB devices.

The Wireless USB standard is designed to be a replacement for traditional USB drives. Typical devices include a keyboard, mouse, camera, printer, external drives, etc. Wireless USB can also be used for easy sharing of printers that do not have a standard network interface or are not connected to a print server [29].

The transmission parameters correspond to standard USB 2.0, but the bandwidth depends on the distance between the devices. At a distance of up to

3 meters, the data transfer rate can theoretically reach 480 Mbit/s. At a distance of 10 meters – only up to 110 Mbit/s (under optimal conditions). Wireless USB is designed to operate in the frequency range from 3.1 to 10.6 GHz. Data transmission is encrypted using AES-128/CCM [30].

2 The Narrowband Internet of Things (NB-IoT) standard is a mobile communication standard for telemetry devices with low data exchange volumes. It was developed by the 3GPP consortium as part of the work on new generation mobile network standards. The first working version of the specification was presented in June 2016. It was intended to connect a wide range of autonomous devices to the digital communication network, for example, medical sensors, resource consumption meters, smart home devices, etc. NB-IoT is one of three IoT standards developed by 3GPP for mobile communications: eMTC (enhanced machine-type communication), NB-IoT and EC-GSM-IoT [28]. The standard eMTC has the highest bandwidth and is built on the basis of LTE standard equipment. The NB-IoT standard can be used both on LTE mobile devices and separately, including GSM. The EC-GSM-IoT standard provides the lowest bandwidth and goes beyond the GSM network. Among the advantages of NB-IoT, the following can be distinguished [30]:

– flexible device power management (up to 10 years online from a 5WH battery);

– huge network capacity (hundreds of thousands of connected IoT devices per base station);

– low cost of IoT devices;

– optimized to increase the sensitivity of the signal modulation.

Hierarchy analysis method for protocol selection. Currently, several methods of expert evaluation and selection of the IoT protocol are known [31], in particular, the hierarchy analysis method (HAM), the Delphi method. According to various studies, we will consider the use of the following important (basic) criteria when choosing the IoT protocol [32]: downlink speed, latency, device bandwidth, device transmission power.

We use the hierarchy analysis method to optimize the choice of a cloud platform. The HAM algorithm includes steps [33]:

1 Identification of the problem and formulation of the goal.

2 Definition of the main criteria and alternatives.

3 Building a hierarchy: from goals through criteria to alternatives.

4 Construction of a matrix of pairwise comparisons of criteria in order to select alternatives by criteria.

5 Application of the method of analysis of the obtained matrices by calculating local vectors and priority vectors of quality indicators.

6 Determining the weights of alternatives in accordance with the values of the components of the vector of global priorities.

7 The choice of the maximum of them.

The importance of different system options (at level 3) and different quality indicators (at level 2) are compared in pairs. The results of paired comparisons of elements are reduced to a matrix form

$$A = \| a_{ij} \| , \tag{4.1}$$

where $a_{ij} = w_i/w_j$ – estimates of paired comparisons of w_i and w_j selection elements.

The diagonal of this matrix is filled with single values, and the elements of the matrix lying below the diagonal are filled with inverse values, for example, for the value 2 will be 1/2, for the value 3 – 1/3, etc.

The components of the main eigenvector P_j (local vectors) of the matrix of paired comparisons (1) of quality indicators are calculated as the root of the nth degree of the geometric mean in the row of the matrix of paired comparisons V_j, divided by the sum of the geometric mean S and the priority vector of quality indicators P_j according to formulas:

$$V_j = n\sqrt{\prod_{i=1}^{n} a_{ij}}, \; j = \overline{1, n} \;, \tag{4.2}$$

$$P_j = \frac{V_j}{S}, \; j = \overline{1, n}, \tag{4.3}$$

$$S = \sum_{j=1}^{n} V_j \,. \tag{4.4}$$

Similarly, the estimates of the matrices of paired comparisons for the variants of the system at level 3 are found separately with respect to each indicator of the quality of the system. Based on these matrices, the components of the corresponding local eigenvectors of the priorities of quality indicators Q_{ij} in relation to individual quality indicators of systems are calculated.

Using the obtained data, the values of the components of the global priority vector C_i are calculated according to the formula:

$$C_i = \sum_{j=1}^{n} P_j Q_{ij}, i = \overline{1, N} \;, \tag{4.5}$$

where P_j – local priorities of alternatives; Q_{ij} – local estimates of the importance of indicators.

According to the maximum value of the components of the vector of global priorities (5), the corresponding preferred variant of the system is selected.

Analysis of protocols for the network and sound monitoring. For a comparative analysis, we will select four protocols for the network and sound monitoring. The names of the data transmission protocols with the selected characteristics for the network and sound monitoring are given in Table 4.1 [34].

Table 4.1 – Comparative analysis of LTE Cat 0, eMTC, NB-IoT, EC-GSM-IOTs data transmission protocols

Characteristics / Protocols	LTE Cat 0	eMTC	NB-IoT	EC-GSM-IoT
Downlink speed	1 Mbit/s	1 Mbit/s	250 kbit/s	474 kbit/s or 2 Mbit/s
Delay	not deployed	10–15 ms	1.6–10 s	700 ms – 2 s
Device bandwidth	1.4–20 MHz	1.4 MHz	180 kHz	200 kHz
Device transmission power	23 dBm	20/23 dBm	20/23 dBm	23/33 dBm

Table 4.2 provides a matrix of paired comparisons of protocol parameter indicators based on expert assessments. In the last two columns, the calculation data of the components of the main eigenvector V_j, the priority vector of quality indicators transmission protocols and P_j are given according to formulas (4.2)–(4.4).

Table 4.2 – Matrix of expert assessments corresponding to the importance of 4 indicators of network protocols

The goal is to select the appropriate network protocol link speed	Down link speed	Delay	Device band-width	Device Trans-missi mission power	Compo-nents of the main eigen vector V_j,	The priority vector of quality indicators P_j
Down link speed	1	2	1/3	1/2	0,759	0,167
Delay	1/2	1	1/2	1/3	0,536	0,118
Device bandwidth	3	2	1	3	2,059	0,453
Device transmission power	2	3	1/3	1	1,189	0,261

Tables 4.3–4.6 show matrices of paired comparisons of 4 protocols in relation to quality indicators: downlink speed (N_1); latency (N_2); device bandwidth (N_3); device transmission power (N_4). The calculated components of the corresponding local eigenvectors and priority vectors are given according to formulas (4.2)–(4.4).

Table 4.3 – Matrix of paired comparisons of transmission standards variants (N_1– N_4) with respect to the speed indicator

Quality indicators	N_1	N_2	N_3	N_4	V_1	Q_{1j}
N_1	1	1	1/3	2	0,904	0,274
N_2	1	1	1/3	2	0,904	0,274
N_3	3	3	1	1/5	1,15	0,454
N_4	1/2	1/2	5	1/3	0,343	0,104

Table 4.4 – Matrix of paired comparisons of transmission standards options with respect to the delay indicator

Quality indicators	N_1	N_2	N_3	N_4	V_2	Q_{2j}
N_1	1	2	1/3	1/5	0,601	0,122
N_2	1/2	1	1/5	1/7	0,346	0,07
N_3	3	5	1	2	2,340	0,473
N_4	5	7	1/2	1	1,655	0,335

Table 4.5 – Matrix of paired comparisons of transmission standards options in relation to the throughput indicator

Quality indicators	N_1	N_2	N_3	N_4	V_3	Q_{3j}
N_1	1	1/5	1/8	1/7	0,244	0,042
N_2	5	1	1/5	1/4	0,707	0,122
N_3	8	5	1	1/2	2,115	0,365
N_4	7	4	2	1	2,736	0,472

Table 4.6 – Matrix of paired comparisons of transmission standards options in relation to the power indicator

Quality indicators	N_1	N_2	N_3	N_4	V_4	Q_{4j}
N_1	1	1	1	2	1,189	0,286
N_2	1	1	1	2	1,189	0,286
N_3	1	1	1	2	1,189	0,286
N_4	1/2	1/2	1/2	1	0,595	0,143

Table 4.7 shows the results of calculating the main vector of global priorities of protocol options according to the formula (4.5).

Table 4.7 – The results of calculating the final vector of global priorities

Protocol variants / Vectors of global priorities	Q_{1j}	Q_{2j}	Q_{3j}	Q_{4j}	C_i
N_1	0,274	0,122	0,042	0,286	0,151
N_2	0,274	0,07	0,122	0,286	0,133
N_3	0,454	0,473	0,365	0,286	0,367
N_4	0,104	0,335	0,472	0,143	0,311
P_j	0,167	0,118	0,453	0,261	–

Analyzing the results, we conclude that the preferred protocol option for organizing communications in the network and sound monitoring is option N_3 with a maximum value of C_i 0.367. This is the NB-IoT protocol.

4.4 Optimization of IoT Cloud Platform Choosing of IoT Network for Monitoring Audio Information

Let's look at some of the most well-known IoT platforms by a larger market share in accordance with the main functions and capabilities in order to choose the optimal option based on the required conditions [35].

Amazon Web Services (AWS) platform. It was launched in 2006 and is the leading platform with a 33 % market share in 2018 [36]. This IoT platform will make it easier for developers to connect sensors for various applications – from cars to turbines and smart home light bulbs [37]. The main functions of the AWS IoT platform are a registry for device recognition; a software development kit for devices; device shadows; a secure device gateway; a mechanism for evaluating incoming messages [38].

IBM Watson IoT platform. It held a market share of 18 % in 2018 and provides connectivity, analysis, device management and information management [36]. On this platform, you can try to run sample applications to get an idea of how it works. You can store your data for a certain period of time in order to receive time-expanded information from connected devices. IBM Watson offers some security features based on machine learning and data science [39]. IBM Watson users get access to device management; secure communication; real-time data exchange; data storage; data sensor and meteorological data service [37].

Microsoft Azure platform. It was launched in 2010 and in 2018 held a market share of 24 % [36]. Microsoft representatives have cloud storage, machine learning, IoT services and have even developed their own operating system for IoT devices [39]. The main functions of the Azure IoT platform are device shadowing; rules mechanism; identity registry; information monitoring [38].

Google Cloud IoT (GCP), was launched in 2008 with a market share of 12 % in 2018 [36]. GCP uses cloud and cloud computing. It offers data analysis and machine learning using Google maps to track asset locations.

A multi-criteria approach to choosing an IoT platform. Evaluation and selection of a rational IoT platform is a rather complicated process for many reasons, including: multifactorial evaluation when choosing a platform; the complexity of preliminary consideration of all possible stages of decision-making; insufficient awareness of the features of the development of modern information technologies and the IoT services market; insufficient technical and material base, etc.

In most cases, the choice of an IoT platform for the development of IoT systems is reduced to a comparative analysis of their capabilities and taking into account the pricing policy for services provided by developers of their own IoT platforms. In addition, IoT developers often prefer well-known IoT platforms, without taking into account criteria (factors) that in the future may affect the development, maintenance, updating, reliability, security and scaling of the IoT systems being developed [38].

In [37], it was noted that when choosing an IoT platform, the following features and features of platforms should be taken into account: orientation to a hybrid application environment; the ability to receive data and prepare them for analysis; the statement of the owner of the cloud infrastructure; data reliability and security; peripheral data processing and control. One of the approaches to choosing an Internet of Things platform is based on determining the reference architecture of the platform, which includes the advantages and capabilities of existing modern IoT platforms [38]. Then a comparative analysis of the selected platforms with the reference one is carried out and the best IoT platform is determined.

Currently, several methods of expert evaluation and selection of IoT platforms are known [40], in particular, the HAM, the Delphi method and decision-making methods based on fuzzy sets and fuzzy logic [41]. According to various studies, we will consider the use of the following important (main) criteria when choosing an IoT platform [42]: the level of security and reliability (Q_1); data analytics (Q_2); data collection protocols (Q_3); visualization tools (Q_4). We use the hierarchy analysis method to optimize the choice of a cloud platform.

For a comparative analysis, let's take the four IoT platforms described above [35, 36]: Amazon Web Services (AWS), IBM Watson (IBMW), Microsoft Azure (MSA), Google Cloud IoT (GCP).

The quality indicators of cloud platforms are competitive. To select the preferred platform option, it is necessary to apply multi-criteria optimization methods. Table 4.8 shows the initial values of quality indicators for 4 different quality indicators of platforms: Q_1 (reliability and security), Q_2 (data analytics); Q_3 (data collection protocols), Q_4 (visualization tools), using research in [43].

Table 4.8 – Values of quality indicators for types of IoT cloud platforms

Quality indicators / types of platforms	AWS	MSA	IBMW	GCP
1	2	3	4	5
Q_1 reliability and safety	TLS, Sig V4, X.50	SSL/TSL	SSL/TSL	TSL, SSO
Q_2 data analitics	Real Time analytics (Rule engine, Kinesis, AWS Lambda)	Real Time analytics	Real Time analytics (IBM IoT Real time insights)	Real Time analytics(Cloud IoT Core)
Q_3 protocols	MQTT, HTTPS	MQTT, HTTPS, AMQP	MQTT, HTTPS	MQTT

1	2	3	4	5
Q_4 visualization tools	AWS IoT dash board, web portal	Web portal	Web portal	Google data stu dio (Dashboard)

Table 4.9 provides a matrix of paired comparisons of platform quality indicators based on expert assessments [33, 43]. Next, estimates of the components of the main eigenvector and local vectors of priorities of quality indicators are calculated. As a result of processing, main eigen vectors V_j and priority vectors P_j are calculated according to formulas (4.2)–(4.4).

Table 4.9 – Matrix of paired comparisons of quality indicators, estimates of the components of the main eigenvector and the priority vector of quality indicators

Quality indicators	Q_1	Q_2	Q_3	Q_4	V_j	P_j
Q_1	1	1/2	2	1/2	0,841	0,188
Q_2	2	1	3	1	1,57	0,351
Q_3	1/2	1/3	1	1/3	0,487	0,109
Q_4	2	1	3	1	1,57	0,351

Tables 4.10–4.13 show matrices of paired comparisons of 4 platforms in relation to quality indicators: reliability and security, data analytics, data collection protocols, visualization tools. The calculated components of the corresponding local eigenvectors and priority vectors are given according to the formulas (4.2)–(4.4).

Table 4.10 – Matrix of paired comparisons of platform variants ($N_1 – N_4$) with respect to the reliability and safety indicator

Quality indicators	N_1	N_2	N_3	N_4	V_1	Q_{1j}
N_1	1	3	3	2	2,06	0,488
N_2	1/3	1	1	2	0,905	0,214
N_3	1/3	1	1	2	0,905	0,214
N_4	1/2	1/2	1/2	1	0,354	0,084

Table 4.11 – Matrix of paired comparisons of platform variants in relation to the data analytics indicator

Quality indicators	N_1	N_2	N_3	N_4	V_2	Q_{2j}
N_1	1	4	2	3	2,213	0,459
N_2	1/4	1	1/3	1/3	0,453	0,094
N_3	1/2	3	1	2	1,316	0,273
N_4	1/3	3	1/2	1	0,841	0,19

Table 4.12 – Matrix of paired comparisons of platform variants with respect to the indicator transmission protocols

Quality indicators	N_1	N_2	N_3	N_4	V_3	Q_{3j}
N_1	1	1/2	1	4	1,189	0,26
N_2	2	1	2	3	1,861	0,408
N_3	1	1/2	1	3	1,107	0,242
N_4	1/4	1/3	1/3	1	0,409	0,09

Table 4.13 – Matrix of paired comparisons of platform options in relation to the indicator of the means of visualization

Quality indicators	N_1	N_2	N_3	N_4	V_4	Q_{4j}
N_1	1	3	3	2	2,06	0,455
N_2	1/3	1	1	1/2	0,64	0,141
N_3	1/3	1	1	1/2	0,64	0,141
N_4	1/2	2	2	1	1,189	0,263

Table 4.14 shows the results of calculating the main vector of global priorities of platform options according to the formula (4.5). The maximum value of the global vector is 0.572, for platform N_1, platform N_3 is in second place, its global vector is 0,211.

Table 4.14 – Results of calculating the vector of global priorities

IoT platform variants / Vectors of global priorities	Q_{1j}	Q_{2j}	Q_{3j}	Q_{4j}	C_i
N_1	0,488	0,459	0,26	0,455	0,572
N_2	0,214	0,094	0,408	0,141	0,166
N_3	0,214	0,273	0,242	0,141	0,211
N_4	0,084	0,19	0,09	0,263	0,185
P_j	0,188	0,351	0,109	0,351	–

Analyzing the results, we conclude that the preferred option of the cloud platform is option N_1. This is the AWS platform, which occupies 33 % of the global market for cloud platforms for building and modeling networks and B. These results can be used to model the IoT network, discussed below.

4.5 Structure, Components of IoT Network Emulation on the Amazon Platform

To simulate the IoT network, using IoT platforms, stops the absence of an IoT device that would be compatible with protocols and access methods. An ordinary smartphone can be used as a device, then the implementation (emulation) of a working chain of the IoT network takes a little time [44]. Let's take a smartphone that will emulate an IoT device with 3 variants of sound level and send readings to the Amazon IoT platform. On the platform, we will formulate a rule that, when data is received from our device, will call the notification service, which in turn will send an e-mail with the received data (Figure 4.2).

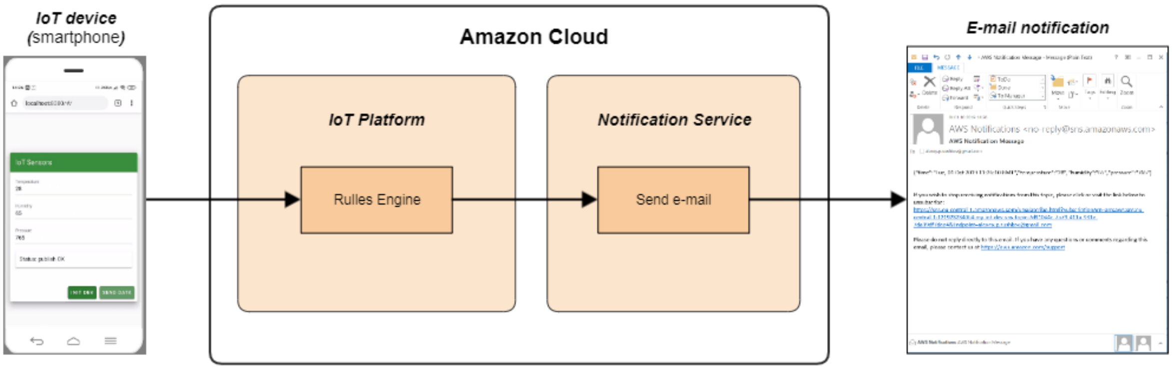


Figure 4.2 – The structure of the emulation of the IoT network on the Amazon platform

Amazon's IoT platforms create services taking into account global trends and as a result, «universal» systems are obtained, the basic principles of which are used by all manufacturers. The cloud platform has a bigger plus – it is the ability to independently simulate the system in a short time without involving the corporate IT service and security [45].

With careful consideration, the smartphone emulates the IoT device well. It has Linux, on which you can run applications; there is a mobile connection to the Internet; with the help of software, you can emulate sensor readings.

Therefore, working with a real IoT device will be no different from working with a smartphone, except using a specific SDK (System Development Kit) to obtain sensor readings. All other communications will be similar [46].

The composition of the AWS IoT platform. Amazon presents a fairly clear structure of its platform (Figure 4.3).

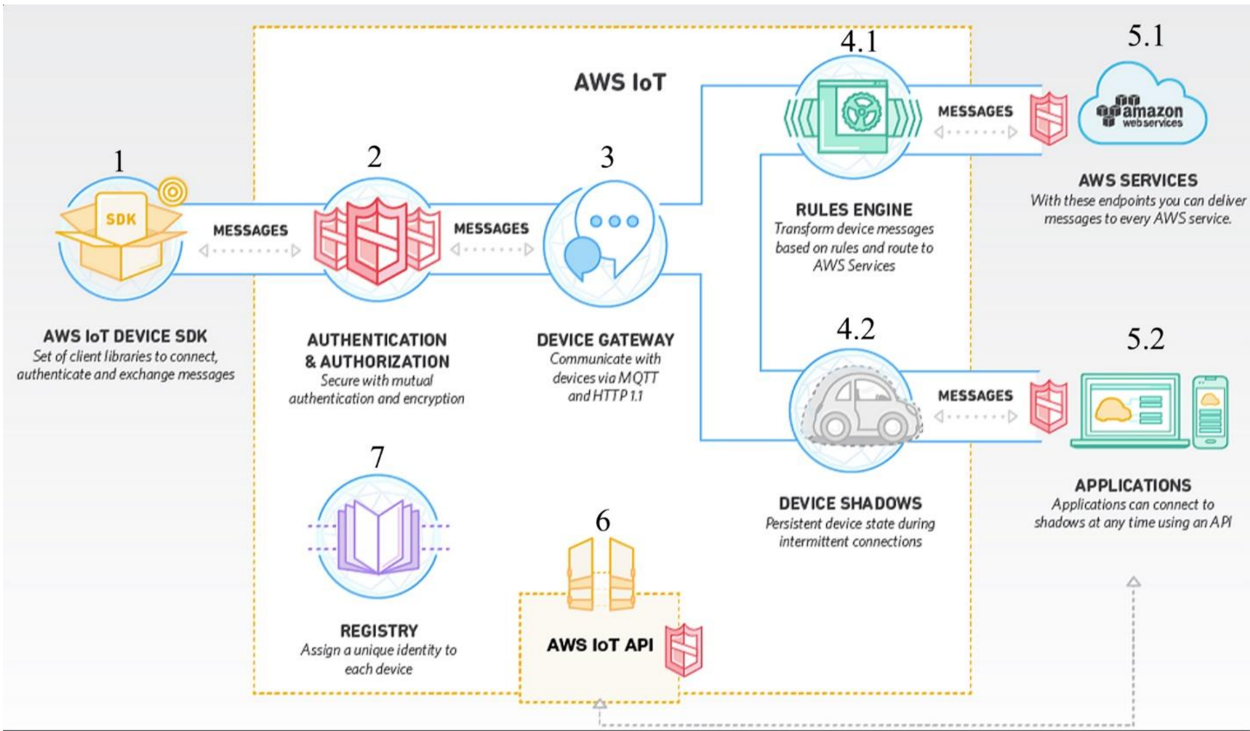


Figure 4.3 – Amazon Platform Structure

Module 1 and Module 2 describe how to set up your environment (or use Greengrass device setup to run these modules):

– configure your core device for Greengrass;

– run the dependency checker script;

– create a Greengrass group and Greengrass core;

– download and install the latest AWS IoT Greengrass Core software from a tar.gz file;

– start the Greengrass daemon process on the core.

AWS IoT Greengrass also provides other options for installing the AWS IoT Greengrass Core software, including apt installations on supported Debian platforms. For more information, see Install the AWS IoT Greengrass Core software.

Module 3 describes how to use local Lambda functions (or use Greengrass device setup to run Module 3):

– create Hello World Lambda functions in AWS Lambda;

– add Lambda functions to your Greengrass group;

– create subscriptions that allow MQTT communication between the Lambda functions and AWS IoT;

– configure local logging for Greengrass system components and Lambda functions;

– deploy a Greengrass group that contains your Lambda functions and subscriptions;

– send messages from local Lambda functions to AWS IoT;

– invoke local Lambda functions from AWS IoT;

– test on-demand and long-lived functions.

Module 4 shows how devices connect to the core and communicate with each other.

Module 5 shows how devices can use shadows to control state:

– register and provision AWS IoT devices (represented by command-line terminals):

– install the AWS IoT Device SDK for Python. This is used by devices to discover the Greengrass core;

– add the devices to your Greengrass group;

– create subscriptions that allow MQTT communication;

– deploy a Greengrass group that contains your devices;

– test device-to-device commumication;

– test shadow state updates.

Module 6 shows you how Lambda functions can access the AWS Cloud:

– create a Greengrass group role that allows access to Amazon DynamoDB resources;

– add a Lambda function to your Greengrass group. This function uses the AWS SDK for Python to interact with DynamoDB;

– create subscriptions that allow MQTT communication;

– test the interaction with DynamoDB.

Module 7 shows you how to configure a simulated hardware security module (HSM) for use with a Greengrass core.

The general algorithm for working with the platform is as follows:

1 Sensors measure the parameters of processes (devices) that interact with the IoT platform using the SDK.

2 Devices send messages that are verified by the authentication and authorization service (Certificate).

3 Messages arrive at the Device Gateway block using different network protocols and then get into the rules handler (Rules 4.1) and are copied to the device shadows block (Device Shadows 4.2).

4.1 Device Shadows are digital twins that store the current states of devices, they are always available to applications. On the other hand, in the absence of communication with the device Device Shadow executes the control commands from applications and when you restore connectivity synchronizer the current status with the device.

4.2 The rules engine depending on the received data and performs predetermined actions (5.1), for example, stores the data in the database, send SMS or e-mail notification, HTTP API calls, sends data to the system analysts, etc.

5.2 Applications use this data to monitor and manage devices using the AWS API (6).

Information about all devices is stored on the AWS IoT platform (7). We begin to detail our structure, the scheme of which is becoming more complicated (Figure 4.4).

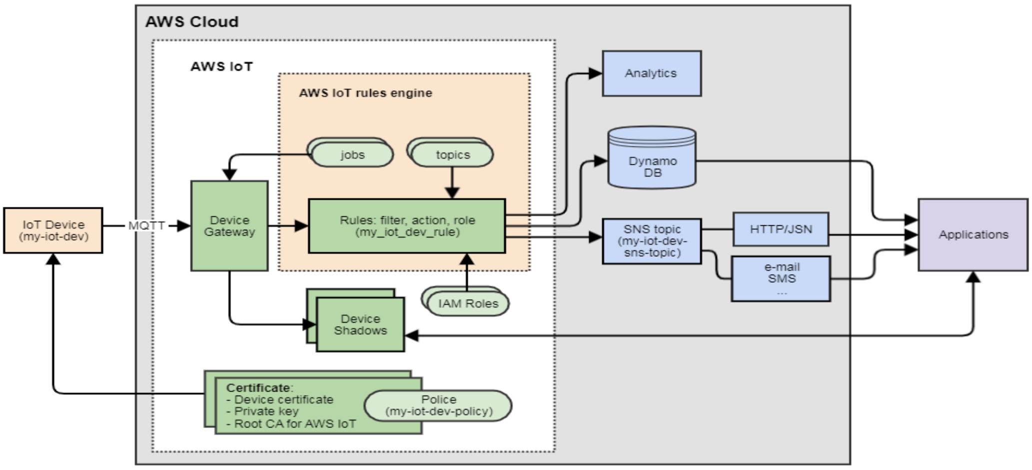


Figure 4.4 – Detailed structure of the Amazon platform

The rules consist of the following components:

– Filter – filter messages for processing. It is set as an SQL query;

– Action – the action to be performed;

– Role – one or more IAM roles.

Certificate – they are uploaded to an IoT device, with their help, devices are authenticated on the AWS platform. Consist of blocks:

– the X.509 device certificate (includes private key, AWS platform root certificate);

– police – policies are attached to certificates that determine what actions can be performed by the device. With the help of policies, devices are authorized. AWS services that receive information from the IoT platform are detailed: Analytics, DB, SNS notification service.

4.6 Connecting Devices (Sensors) to the IoT Platform

The full instructions for connecting an IoT device to the Amazon platform are given in [45, 46]. To understand this task, we list the steps of the algorithm that need to be done to make the scheme work:

1 Creating a my-iot-dev device on the platform:

1.2 We get the X.509 device certificate, private and public keys.

1.3 We get the root certificate of the AWS platform.

2 Create the my-iot-dev-policy. For our demo version, we allow all actions of IoT (Figure 4.5).

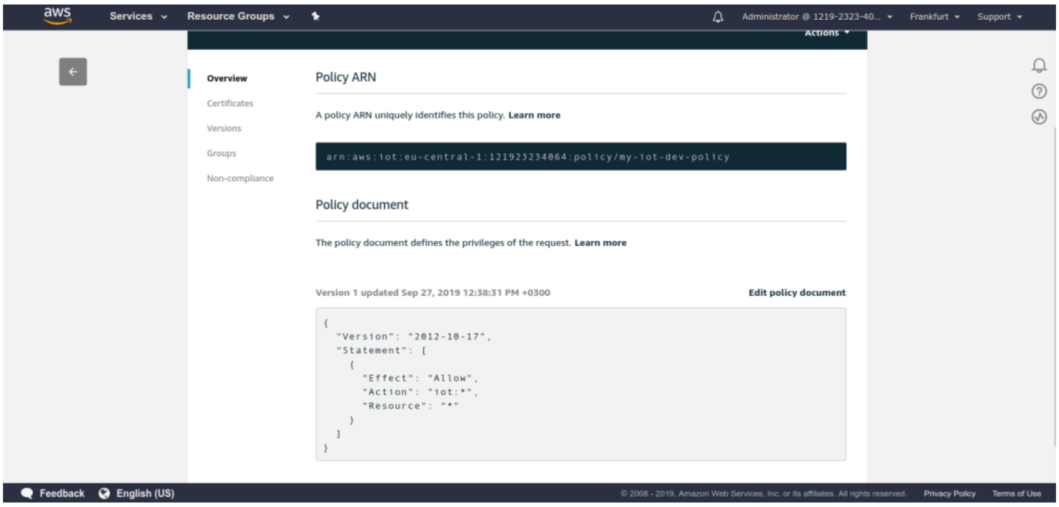


Figure 4.5 – All actions IoT

3 To create a topic in AWS SNS (my-ios-dev-sns-topic Figure 4.6).

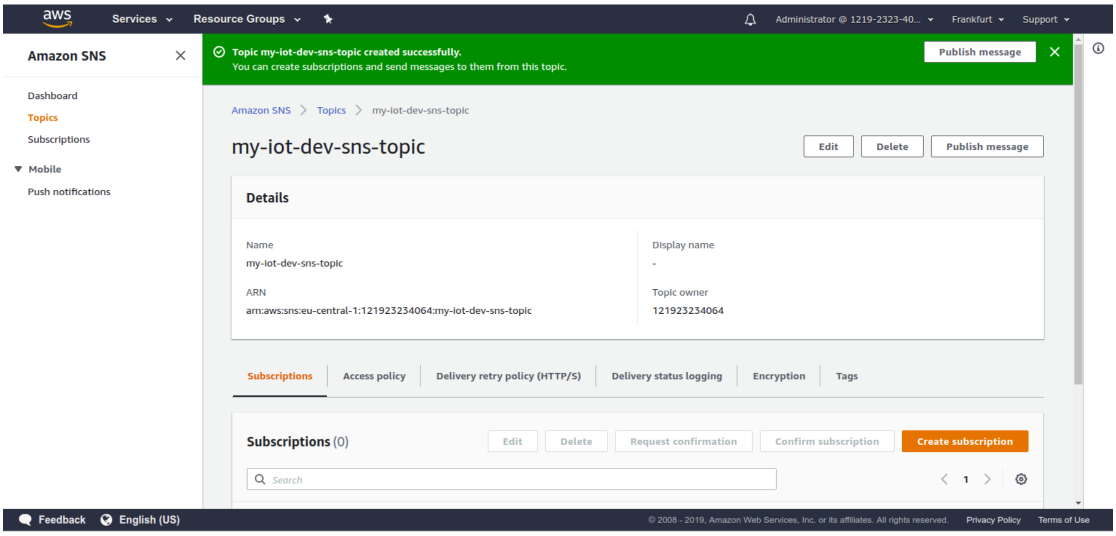


Figure 4.6 – Getting the my-iot-dev-sns-topic

4 Configure what exactly this topic will do when receiving data. To do this, create a subscription to the topic (Subscribe to the Amazon SNS topic), enter the target e-mail address, wait for the verification letter, confirm the e-mail (Figure 4.7).

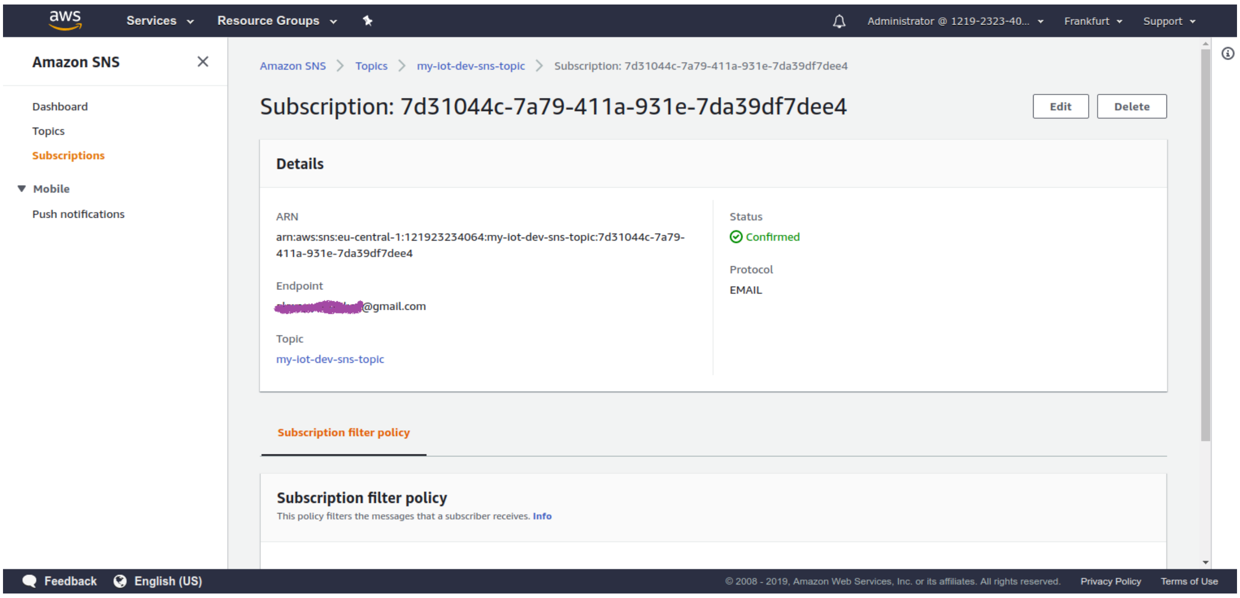


Figure 4.7 – Creating a subscription to a topic

5 Create the rule itself (my_iot_dev_rule), which will call the created topic (Figure 4.8):

– Filter: SELECT * FROM 'my/dev-topic’ – the filter is triggered when any message is received in the topic named 'my/dev-topic’;

– Action: sending a message to a pre-created SNS topic «arn:aws:sns:eu-central-1:1219xxx34064:my-iot-dev-sns-topic»;

– IAM role: creating the my-dev-role role with access to SNS topics.

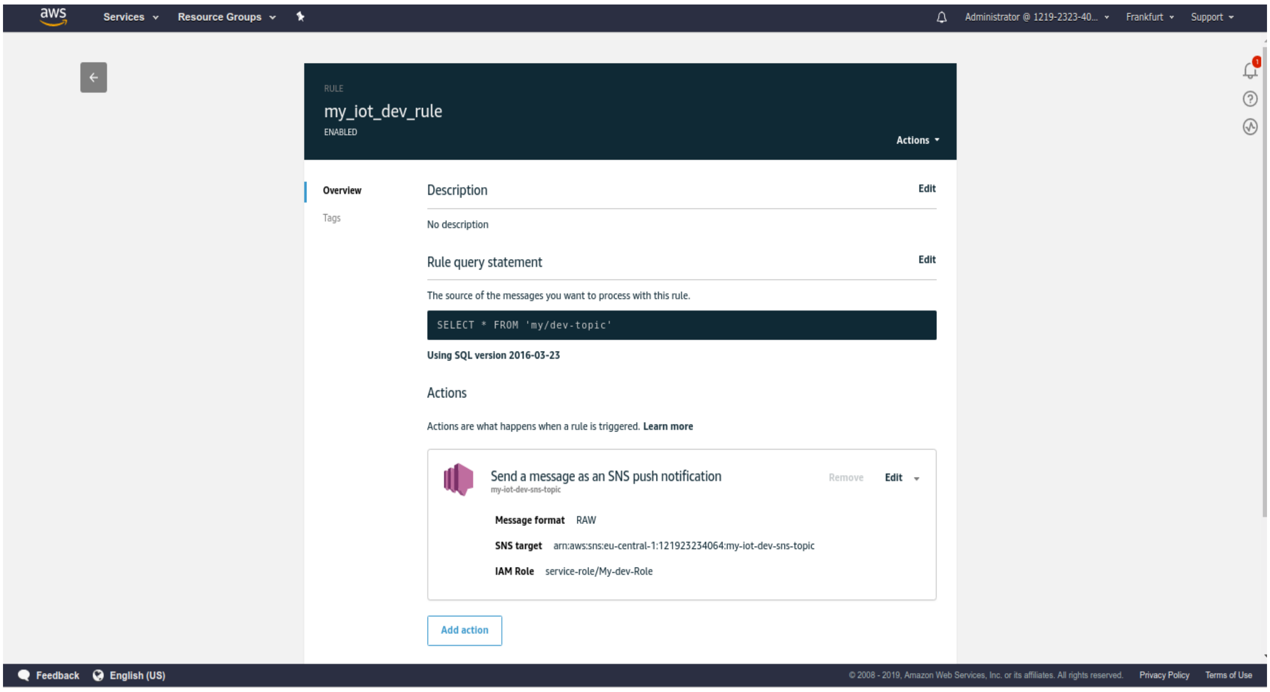


Figure 4.8 – Creating a rule (my_iot_dev_rule)

6 All logical entities have been created for our device. Now you can test that theoretically the scheme works. To do this, AWS has a test tool that allows you to send and receive messages similarly to real devices. We launch it, subscribe to topic (my/dev-topic) and send «Hello World!» message (Figure 4.9).

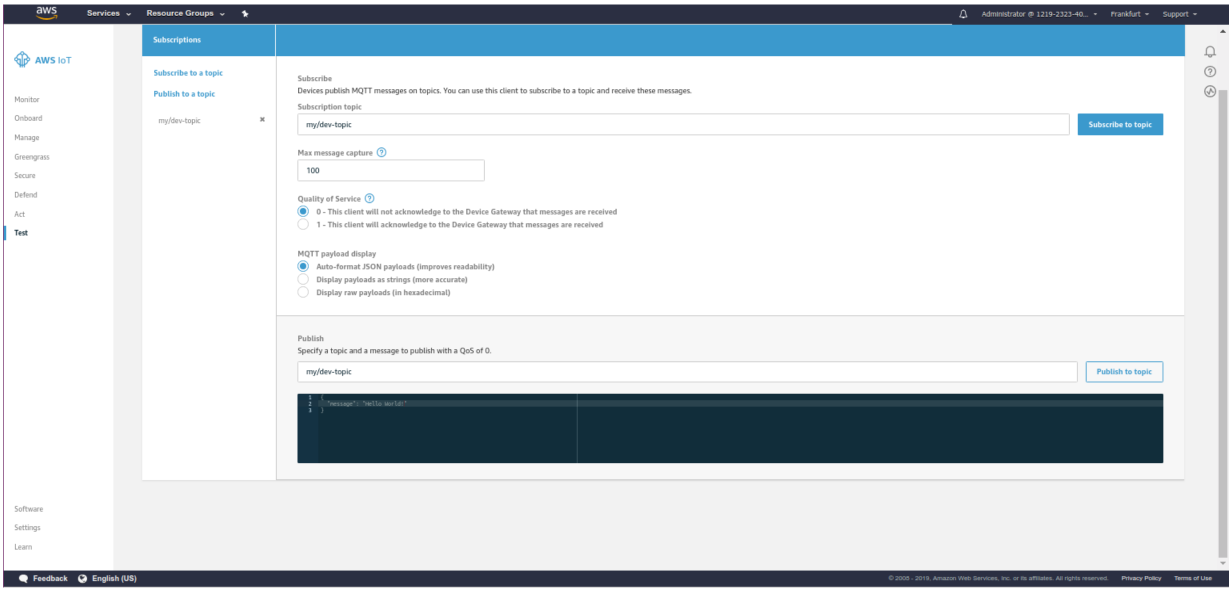


Figure 4.9 – Network operation testing

7 We check that an e-mail has arrived with the message «Hello World!» and we conclude that the scheme works.

4.7 Emulating a Device on a Smartphone

Next, we need to configure IoT devices, which will be in our smartphone [44]. To do this, we use the work [46]. To turn a smartphone into an IoT device, you need to: copy to the device: private key, X.509 and «Root CA for AWS» certificates. Than:

– install Node.js and npm package manager;

– install the AWS SDK and run the test program.

In our case, everything will be easier, because the certificates, AWS SDK and test program are recorded on GitHub and we clone the IoT-Sensors repository [47]. If anyone wants to use this test program, then they will need to write their certificates to the /IoT/certs directory and register the relevant certificates for the device in the /server/src/services/IoT-AOI-server fileRest API Endpoint:

```
device = deviceModule({
…
        host: 'a2lqo1xxx4zydi-ats.iot.eu-central-1.amazonaws.com',
…
    })
```

Rest API Endpoint it is taken from the device settings (Figure 4.10).

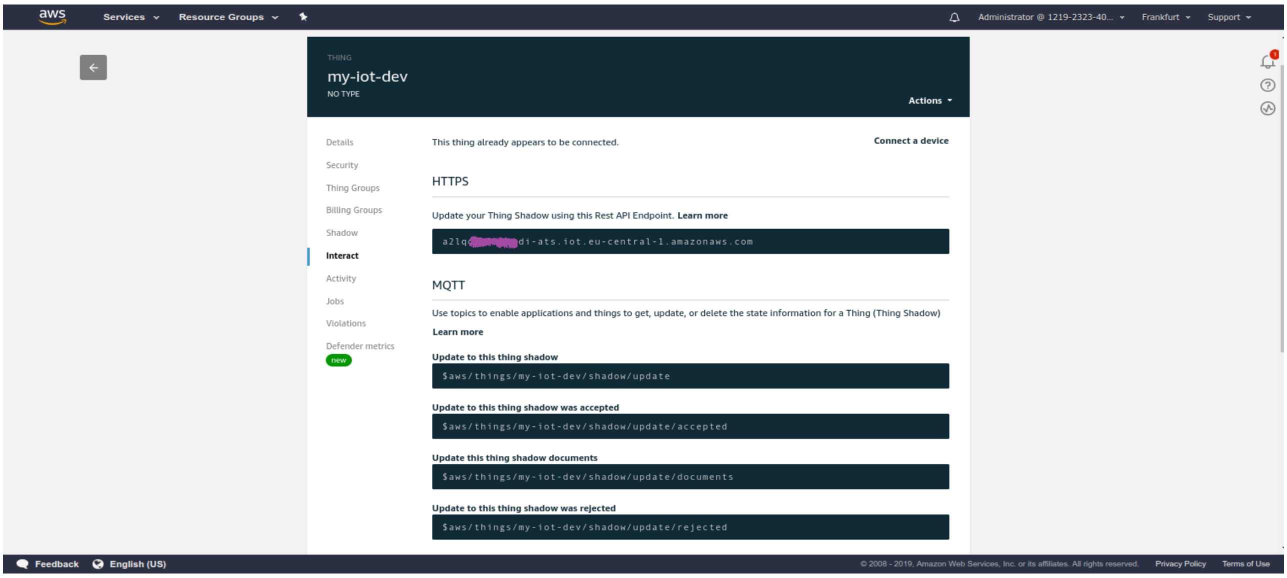


Figure 4.10 – Interface configuration data

If you want to try something standard, you can use test programs from the AWS SDK.

Android – it is based on Linux OS, but with its own limitations, so to run JS applications, we install a special terminal, for example, Termux [48].

For the initial development of Termax, there are a number of articles, for example [49]. But after installing Thermex, it need to run a few commands:

```
git clone https://github.com/AlexeySushkov/IoT-Sensors.git
```

Installing the server

```
cd ~/IoT-Sensors/server
```

```
npm install
```

```
npm start
```

If everything was successful, the line will appear in the terminal: Server started on port 8081. It looks like this (Figure 4.11).

Figure 4.11 – Server installation commands

Next, enter the commands to install the client:

```
cd ~/IoT-Sensors/client
```

```
npm install
```

```
npm run serve
```

If everything was successful, the line will appear in the terminal: App running at port: 8080.

Next, in the smartphone browser, enter: http://localhost:8080 a text application will appear on the screen (Figure 4.12).

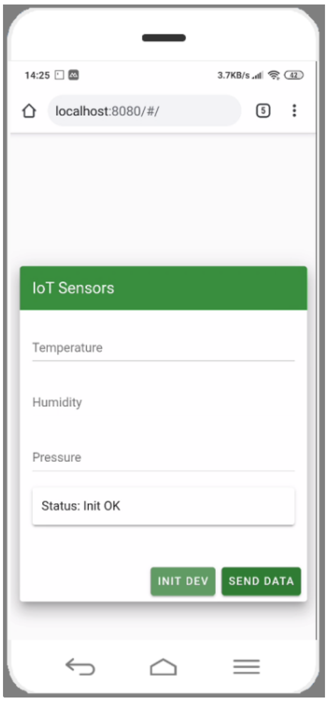


Figure 4.12 – Device initialization on a smartphone

4.8 Simulation of Sensors and Network Operation

Click the «INIT DEV» button. At the same time, the Ios device is authenticated and authorized on the AWS Ios platform. On success, the status becomes «INIT OK». Next, enter the values of the temperature, humidity and pressure sensors, for example: Sound 1: 23; Humidity: 65; Pressure: 787, press the «SEND DATA» button (Figure 4.13).

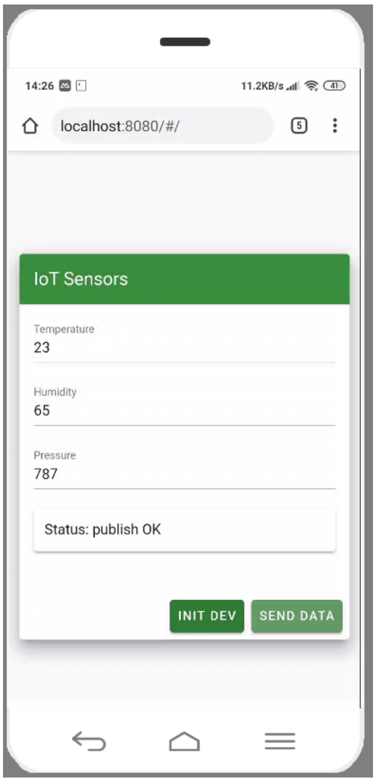


Figure 4.13 – Initiating sensor readings on a smartphone

After that, the application adds a timestamp and sends the data in the form of an MQTT message to the topic «my/dev-topic». The IT platform receives a message and activates a rule that sends a message to the AWS SNS notification service, which sends an e-mail with the received data in JSON format [50]:

{«time»:»Mon, 30 Sep 2021 13:54:52 GMT», «sound»:»23»,
«humidity»:»65», «pressure»:»787»}

If the message is sent successfully, the status changes to: «publish OK» and an e-mail arrives (Figure 4.14).

The AWS Ios platform has a monitoring system that shows the number of connections and messages from iOS devices, statistics on protocols, message types, etc. (Figure 4.15).

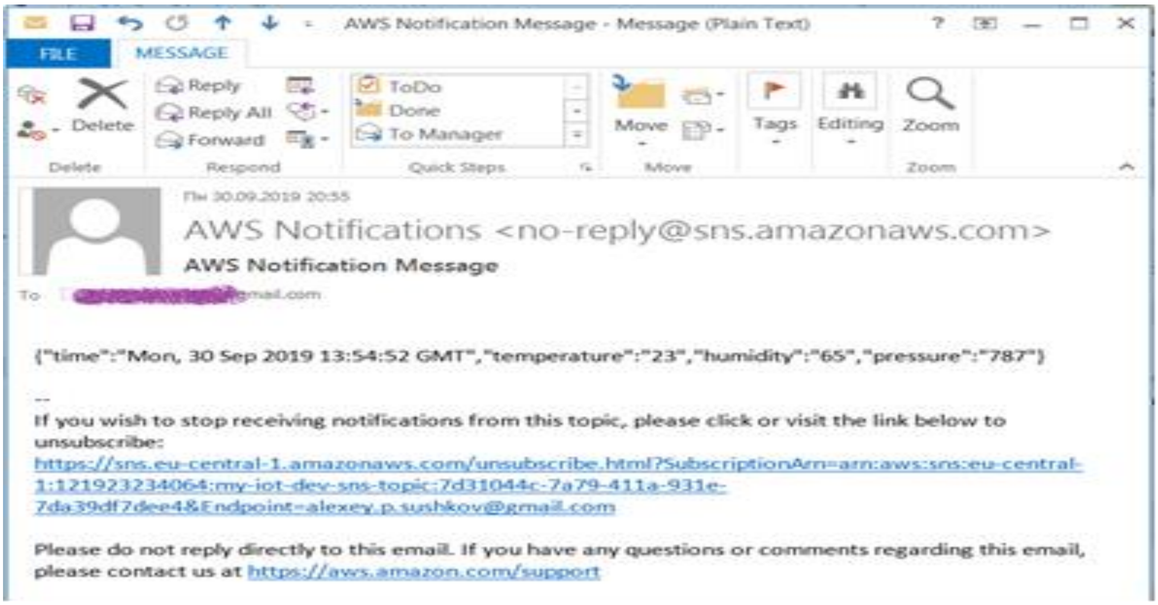


Figure 4.14 – E-mail notification of sensor readings

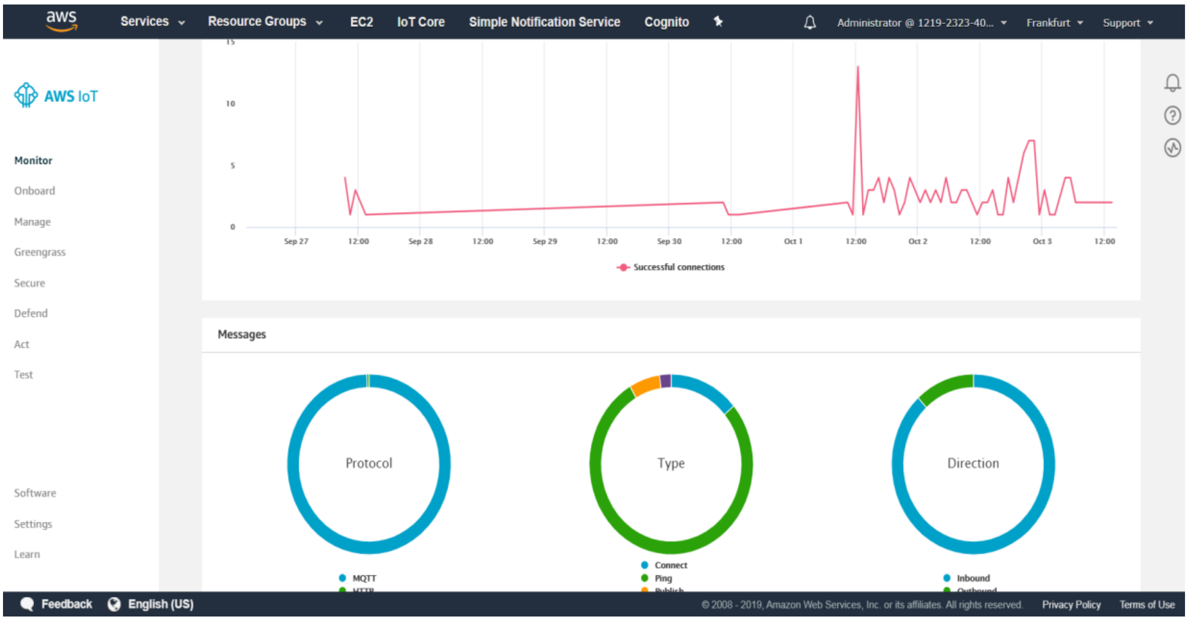


Figure 4.15 – Statistics on protocols, message types, etc.

4.9 Multiagent System for Monitoring Sound Information Using IoT

The multiagent system for monitoring sound information using multiagent IoT network (MAIOT) is composed of two different agents that works together with the supervision of IoT [51]. The process of MAIOT has several algorithms that can cover various needs at the same time we can modify this concept so it can be used is several domains and needs.

This system consists of a Raspherry Pi 3 tiniest computer and incredibly small microcomputer packed onto a single board (Figure 4.16). For all that, the Raspherry Pi 3 is packed with enough power to handle demanding computer projects [52–54].

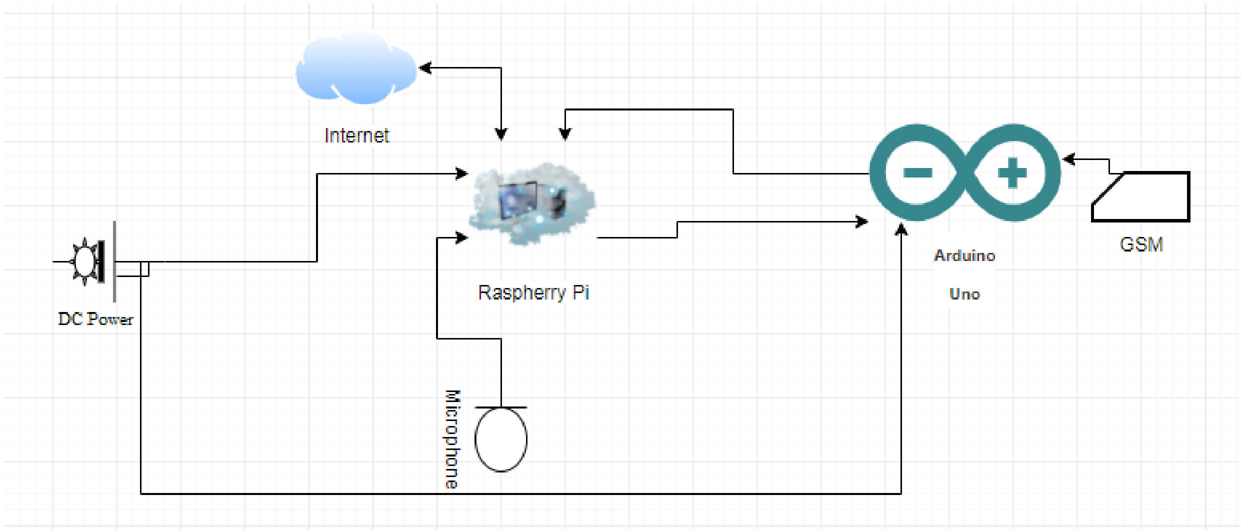


Figure 4.16 – Multiagent system for monitoring sound information using MAIOT

It packs enough technology to run the full version of Raspberry Pi 3 is a rewarding device that's ideal for creating Internet of Things, wearable, and embedded projects, to keep the size down, the Raspherry Pi 3 features a smaller-than-normal mini HDMI socket, and it offers a full computer experience. Raspherry Pi 3 will be connected to a solar power rechargeable battery that provide it power 24/7, and on the other hand it will be connected to wireless to the internet using the Wi-Fi, and a microphone is connected also to Raspherry Pi 3 to detect the sound from the environment, and also Raspherry Pi 3 is connected to an Arduino Uno that is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits (Figure 4.17) [55–57].

From the side of the Arduino will be connected to the solar power rechargeable battery and a Global System for Mobile Communications (GSM) [51].

Algorithm Multiagent System for Monitoring Sound Information using IoT.

The algorithm of MAIOT will start by detecting the sound by the help of the sound detector that is connected to the Raspherry, so that this sound will be send to the sound analyses system that is on the internet by the help of Raspherry Pi, this system will reply to the Raspherry Pi 4 types of messages [58, 59].

1 First type of message is to decline this message because it doesn't have any value that we need.

2 Second type of message that action should be taken:

a) Call Police;

b) Call Firefighters;

c) Alarm On.

After receiving the respond from the internet to the Raspherry Pi, the Raspherry Pi will send a notification to Arduino to complete the action that should be taken, so if this action need to turn the alarm on, the Arduino will activate this task, and if the action is to call the police or Firefighters, the Arduino will send the action to the GSM module that will directly send the message to the police or firefighters to notify them with location [60]. More over this system can be upgraded with a cam so that when the GSM will send the massage the picture will be also attached to the notification that the GSM will send.

The first type that tell the Raspherry Pi to dismiss this record because it doesn't have any content that we need and the second type of message is that from the side of the Arduino will be connected to the solar power rechargeable battery and a Global System for Mobile Communications (GSM).

The sound sensor module provides an easy way to detect sound and is generally used for detecting sound intensity. This module can be used for security, switch, and monitoring applications. Its accuracy can be easily adjusted for the convenience of usage. It uses a microphone which supplies the input to an amplifier, peak detector and buffer. When the sensor detects a sound, it processes an output signal voltage which is sent to a microcontroller then performs necessary processing (Figure 4.17).

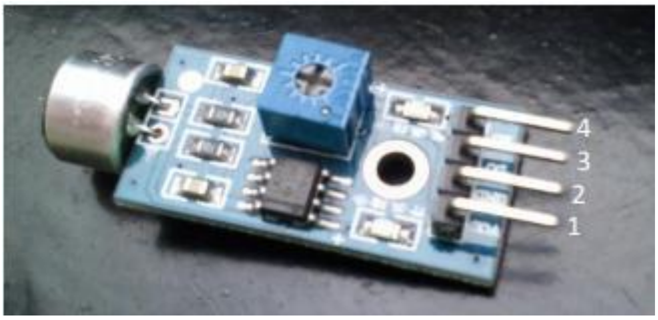


Figure 4.17 – Sound sensor

Specification of Sound Sensor: Operating voltage 3.3–5 V; Output model: digital switch outputs (0, 1); Voltage Gain 26 dB; Microphone Impedance 2.2 KΏ; Microphone Frequency 16.20 kHz. Install Debian Linux OS and connect the peripherals. As the controller is applicable computer «Raspberry Pi 3 Model B» is based on the Debian Linux OS version 4.19.59-v7. In Raspberry Pi 3 Model B built-in Wi-Fi, Bluetooth 4.1, and have access to the GPIO ports (interface I/O) for direct connection of devices (e. g., sensors or actuators) [59].

As a storage for storing data and programs, we use a Kingston 16 GB microSD memory card (Class 10). It should be noted that the microSD card must be formatted as FAT32. The official operating system for all Raspberry Pi is Raspbian (Debian Linux distribution).

Let's install Raspbian using the NOOBS installer of the operating system for Raspberry Pi 3, which contains Raspbian. To do this, you need to download the NOOBS archive file from the site to your PC. Then unpack the contents of the NOOBS archive into a separate directory on the computer and copy the contents to the root directory of the microSD card connected via a card reader to the PC. After that, you need to remove the microSD from the card reader and insert it into the connector on the Raspberry Pi 3 Model B panel. After connecting a monitor, a mouse manipulator, a keyboard and a power supply to a Raspberry Pi computer, NOOBS should boot.

From the screen saver with the list, select «Raspbian» and click «Install». The Raspberry Pi computer boots with a graphical user interface, but you can disable it and configure the computer so that it boots with the «LXTerminal» terminal (with a command line interface). In addition, on a Raspberry Pi with a graphical user interface, you can open LXTerminal from the taskbar by clicking on the LXTerminal icon (Figure 4.18).

Figure 4.18 – Workplace based on Raspberry Pi 3 controller

Let's install the Node platform.js on Raspberry Pi. To implement a webserver and WoT applications for Smart Devices, we will use the Node software platform.js (v10.16.0) for working with JavaScript and the module Express.js (4.17.1) or a framework for Node.js. Node.js, which is suitable for implementing various servers (web, CoAP, WebSocket, MQTT, etc.) and for building scalable Web applications.

Before installing the Node platform.js on the controller, it is necessary to determine the version of the processor core architecture on the Raspberry Pi. To check the processor architecture version, you can use the commands:

```
uname -m or cat / proc / cpuinfo,
```

which must be entered in LXTerminal. The result of checking the processor model version on «Raspberry Pi 3 Model B» are: ARMv7 Processor rev 4 (v7l). On the website (nodejs.org/en/download) for the «ARMv7» version, the recommended version of the node platform is v10.16.0.

Node.js can be installed from repositories nodesource.com or nodejs.org containing the latest versions Node.js. For the Debian Linux version 4.19.59 operating system installed on the Raspberry Pi 3 Model B, you can download data from the command line for the ARMv7 version:

```
curl-sL https://deb.nodesource.com/setup_10.x|sudo-E bash-install Node.js:
sudo apt-get install-y nodejs and check the version:
node–v v10.16.0.
```

It should be noted that to download files, you can use the wget console utility, for example, from NodeSource repositories NodeSource (https://deb.nodesource.com/setup_10.x):

```
wget-qO-https://deb.nodesource.com/setup_10.x|bash-or from storage nodejs.org:
wget https://nodejs.org/dist/v10.16.0/node-v10.16.0-linux- armv7l.tar.gz
with the subsequent unpacking of the file and installation in the file system.
```

Installing IoT Network Software. Implement the directory for the WoT application and the application framework. Let's create a WoT application (Web Thing) with an interface and a Web Thing REST API programming interface for accessing and managing both from the browser and from third-party WoT applications. The WoT application being created is designed for Smart Objects, that is, for the drive (actuators). As an actuator, an LED is applicable (simulates a drive).

It should be noted that access to the resources of WoT applications can be carried out through the protocols: HTTP, CoAP, WebSocket, REST/HTTP, MQTT. The HTTP protocol can be used to access through the user interface using browsers in WoT applications hosted on HTTP servers. The CoAP, WebSocket, and REST/HTTP protocols can be used to access other WoT applications hosted on CoAP, WebSocket, and HTTP servers, respectively, via the API.

To establish access to WoT applications (Figure 4.19) using the CoAP and WebSocket protocols, you can use the Firefox browser with the Copper plugin (Copper (cu) CoAP Firefox plugin) and with the Wang Fenjin extension, respectively. MQTT protocol access to WoT applications with MQTT clients installed on them is carried out through broker servers (for example, mqtt.org) using client applications (MQTT clients) installed on PCs or smartphones.

We will begin development by creating a WoT application for a drive with an interface that is located on an HTTP server. To do this, using the file manager in the «pi» directory, we will create a wot-led directory and a WoT-application framework consisting of two directories: public, in which all static files will be located; views, in which templates will be stored, and two files: package.json – for storing all information about the program; server.js -for various servers and a server WoT application.

Using the pi @ raspberrypi terminal ~ $ let's move from the root directory «pi» to the created directory wot-led [51]:

cd wot-led.

Next, in the directory pi@raspberrypi~/wot-led$, we will execute the command to generate the package file.json:

npm init.

This command will launch a command-line questionnaire, which will end with the creation or modification of the package file.json, to do this, you can fill in the suggested options or not fill in, but press Enter.

Then install the express framework by running the command:

npm install express-save.

As a result, the express library will be installed (in our case, version v.4.17.1), the name of which will be saved in the package file.json, and the contents of the library will be installed in the created node_modules directory in the wot-led folder.

Then we will install a library for working with GPIO on Raspberry Pi, a PATH library for specifying directories of executed programs and a template engine that will transfer data to the template.

Choose a template with the .ejs extension, in this template you can output the html code of the smart devices website with embedded scripts, styles and images (Figure 4.19):

– npm install rpi-gpio-save;
– npm install path-save;
– npm install ejs-save.

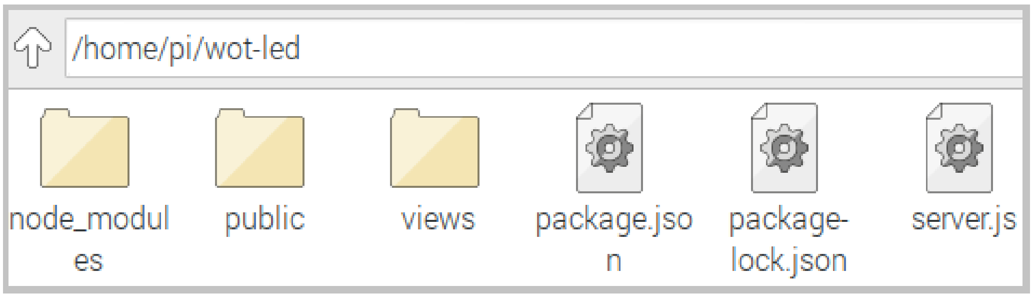


Figure 4.19 – WoT application structure

Implementation of the server part of the WoT application (server.js) using the express framework. Development of the server WoT application server file. Let's start with js or express applications by downloading the express framework using the require() function. This WoT application is placed on an HTTP server wrapped in the structure of the express library. Next, create an express application object named app. For LED control (read/write mode) download the «rpi-gpio» library and the path library to specify the absolute path to the directory to provide files.

Next, we configure contact 7 to the DIR_OUT recording mode. Then we define the template engine type with the .ejs extension. Calling the temporary work function (middleware or middleware) express.static for the default route. Defining handlers for routes to GET and POST HTPP requests. We launch the HTTP server, and assign the port for the WoT application: 8585.

Creating a client part of a WoT application or a graphical user interface file index.ejs based on the html5 markup language, the CSS3 stylesheet language, the css framework for Bootstrap Buttons. Bootstrap provides various button styles that are applied in the application. To use Bootstrap Buttons, it is necessary to register the connection of the necessary files from the Bootstrap cdn storage between the <head> </head> tags in the index.ejs web page. In addition, for the LED-on button, we apply class = «btn btn- danger», and for the LED-OFF button, we apply class = «btn btn-primary». The index.ejs file will be placed in the views directory, and the style.css and less.png files will be placed in the public directory.

4.10 Testing of IoT Network Operation

Testing the work of the WoT application. To test the server, you need to run the WoT application through the terminal. To do this, go to the wot-led directory where the WoT application is located and start the server.js by running the command: node server.js. [63]. In this case, the terminal will display the entry «WOT-application listens to port: 8585». The terminal image is shown in Figure 4.20.

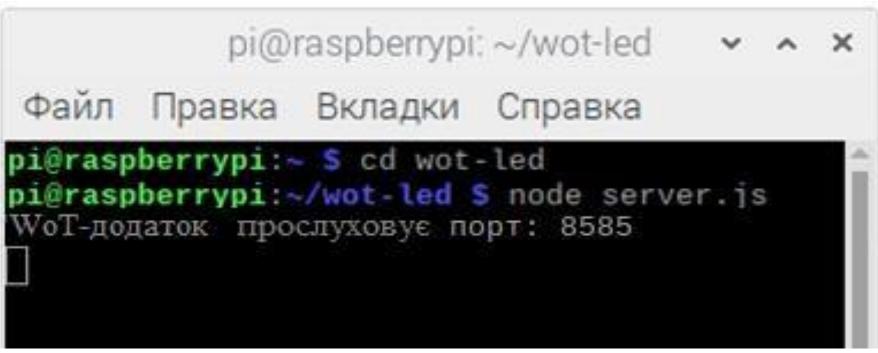


Figure 4.20 – Screenshot of the graphical user interface

After launching the program through the terminal, you can access resources in the browser at: localhost: 85; raspberrypi: 8585 or at the IP address 192.168.1.129:8585, which is intended for the wireless router (for example, LinksysWRT160N) of the local network. Access to the LED by IP address can be performed both from the Raspbian OS browser on the monitor connected to the Raspberry pi 3, and from desktop PC browsers, laptop, smartphone, etc.

Photo of a graphical user interface or a web page index.html The buttons shown in Figure 4.21, 4.22 contain buttons for turning on/off the LED-On/LED-Off, as well as the Position string of LEDs, in which entries can be displayed: «Press button LED-On!», «On» and «Off».

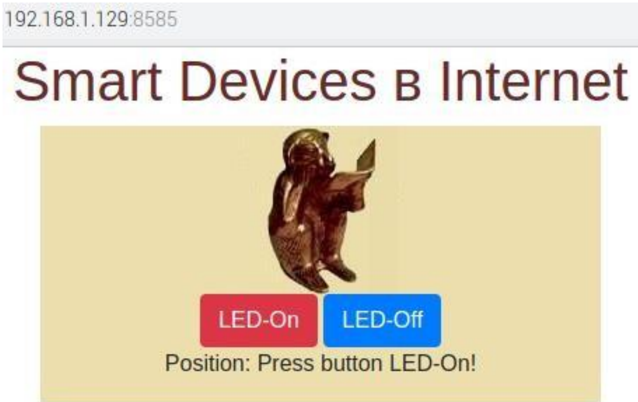


Figure 4.21 – Screenshot of the graphical user interface

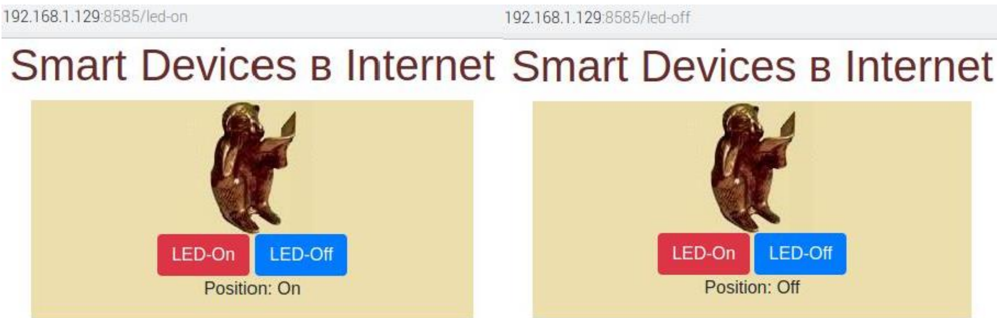


Figure 4.22 – Photo of the graphical user interface

When running in the server terminal.js the browser displays «Press button LED-On!», while the LED is off. When you press the LED-On button in the browser, the LED turns on and the value «On» is displayed, and the entry «pin 7: true» appears in the command line. When you press the LED-Off button in the browser, the LED turns off and the value «Off» is displayed, and the entry «pin 7: false» appears in the command line, etc. (Figure 4.23).

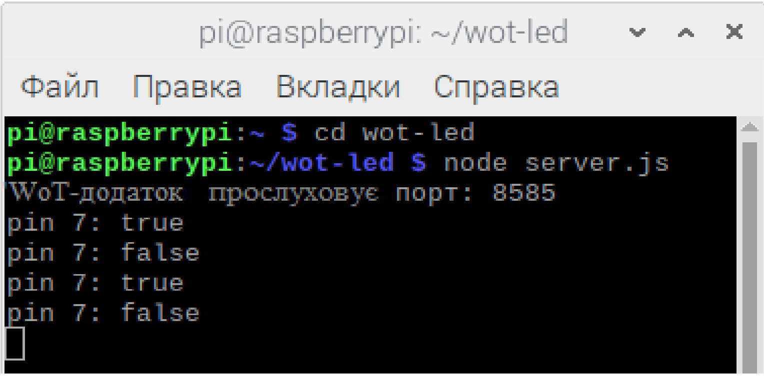


Figure 4.23 – Screenshot of the terminal with WoT application

4.11 Multiagent System for Automatic Sound Detecting Based on Raspberry PI and Arduino

Conside the algorithm of MAS for automatic sound detecting. Raspberry Pi Single board computer and Arduino were used to create the sound detecting module, which is mostly software-based. All of the hardware components are ready-to-use peripherals that are connected straight to the Raspberry Pi and Arduino and customized for the project's needs [51, 61].

There are various models of Raspberry Pi but project works on model IoT. It has eight general-purpose input/outputs, two USB ports; a high definition media interface (HDMI) output, and other non-relevant features for this project. Additionally, Raspberry Pi needs an operating system which is stored on a secure digital (SD) card. Operating system selection was not easy, because of the good quantity of them; the more relevant one are Raspbian, Pidora, and RISC OS.

Raspbian was selected for current project for the wide diversity of tutorials and info available online. Raspbian is a free-license Linux operating system based on Debian and optimized for its usage with Raspberry Pi hardware [52].

The project is classified as an Internet of Things (IoT) initiative in general. It can only work if you have access to the internet. Google voice and speech APIs are used in the apps described here. The sound command is picked up by the microphone. The Google voice API is then used to transform this to text. The text is then compared to the other commands in the commands configuration file that have already been defined. The bash command linked with it will be performed if it matches any of them. It may also utilize the Raspberry Pi as an interactive response system by sending message to Arduino, and depending on this message the Arduino will communicate with the GSM connected to it so that the GSM will send message to the needed destination. Here's a block diagram showing the basic working of the sound detecting software for Raspberry Pi and Arduino (Figure 4.24).

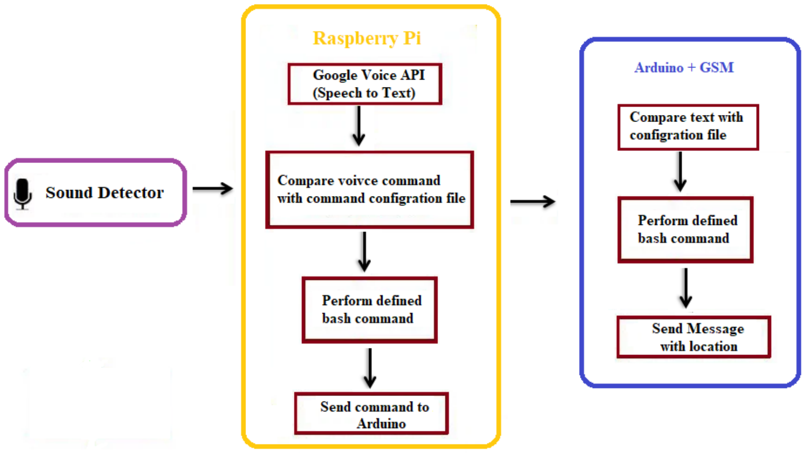
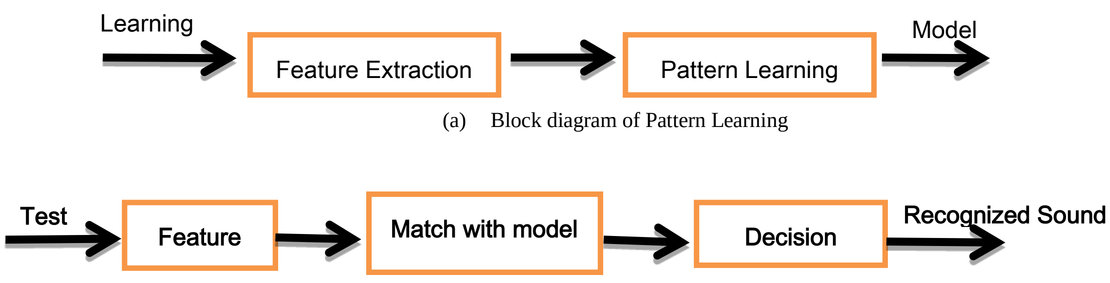


Figure 4.24 – Block Diagram for basic working of the automatic sound detecting

The intended system's prototype was created. Experiments were carried out to test the suggested system and determine its correctness. The sound recognizer was trained using a variety of audio samples (in.wav format). Following that, the voice of the same individual who had been enrolled was used in various situations to assess the system's accuracy.

Speech recognition is the process of voice identification based on the spoken word by performing a conversion of a signal [64], which is captured by the audio device (voice input device). Speech Recognition is also a system used to recognize the word commands of the human voice and then translate into data that can be acted upon by a computer. Sound is something that can be heard and has certain signal characteristics, while speech is a sound consisting of spoken words. Voice recognition or speech is one of the efforts required to make the sound recognizable or identifiable so that it can be utilized. Voice recognition can be divided into three approaches, namely the acoustic-phonetic oncoming, an artificial intelligence oncoming, and a pattern recognition approach [65]. The pattern recognition approach for speech can be explained with block diagram (Figure 4.25) [66].

Machine Learning is part of the Google Cloud Platform in building applications that can hear, see, and understand the world around it. In pre-trained Machine Learning Models.



(a) Block diagram of Pattern Learning

(b) Block diagram of voice recognition Learning

Figure 4.25 – Speech recognition block diagram

The Google Cloud Speech API enables the developers to turn audio into text by applying neural network models easily using the API. The API can recognize more than 110 languages and variants, to support a global user base. It is also possible to write user text by dictating using the application microphone, to enable voice command-and-control, or to write audio files, among many other usage cases by recognizing the uploaded audio on demand, and integrate with the audio storage in the Google Cloud Storage [27].

Evaluation Sound Detecting Using Automatic Sound Detecting. The intended system's prototype was created. Experiments were carried out to test the suggested system and determine its correctness. The sound recognizer was trained using a variety of audio samples (in.wav format). Following that, the voice of the same individual who had been enrolled was used in various situations to assess the system's accuracy.

The sound detecting system was put to the test under a variety of scenarios, including:

1) in a busy environment with a lot of background noise;

2) quiet environment with minimal background noise;

3) when the voice is not too loud;

4) when the voice is loud.

Figure 4.26 represents the results of how well the module could perform on 150 test samples of the sound voice, after the sound voice has been pre-enrolled into the database of the module. Table 4.15 shows these statistics.

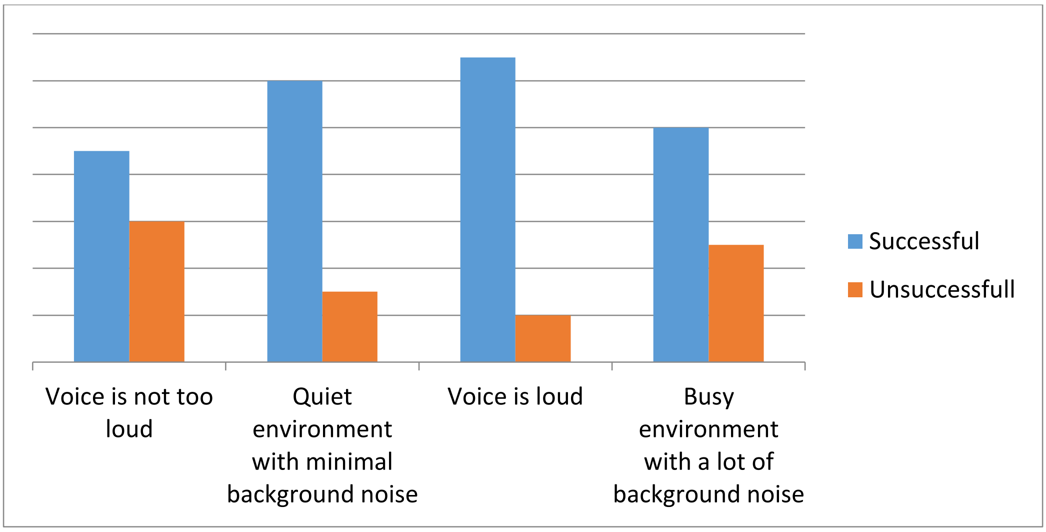


Figure 4.26 – Sound detection in various environments results

Table 4.15 – Success rate of automatic sound detecting system

№	Voice data statistics		
	Test Condition	Number of Accuracy	Percentage (%)
1	Voice is not too loud	90/60	60.00
2	Quiet environment with minimal background noise	120/30	80.00
3	Voice is loud	130/20	86.67
4	Busy environment with a lot of background noise	100/50	66.67

Conclusion on Chapter 4

1 It is present the model and structure of MAS for CPSI. Creating a hierarchy for conceptual representation and a complete set of pictograms to cover all noise types of sound. In the transformation agent will focusing on creation of decision tree in order to swap between scales. Regarding the specific types of sound different types of detectors will propose the suggestion for the related sound detectors. The last will be used as GUI graphical user interface that create automatic report for the end user.

2 To select the best protocol in the sound monitoring network, the hierarchy analysis method was used. As characteristics of the protocols, the following are considered: downlink speed, latency, device bandwidth, device transmission power. A matrix of paired comparisons of protocol parameter indicators based on expert assessments has been formed. Four matrices of paired comparisons of transmission standards variants in relation to the indicator of speed, delay, bandwidth of devices, transmission power of devices are constructed. The matrix of vectors of global priorities is calculated. The preferred protocol option for organizing communications in the network and sound monitoring is option N_3 with a maximum value of C_i – 0.367. This is the NB-IoT protocol.

3 Four cloud platforms for creating and modeling IB networks, occupying 87 % of the global market, are described. The analysis of individual optimization methods for the creation of IB networks is presented. The algorithm of the hierarchy analysis method related to expert methods is given. An example of choosing an IoT cloud platform using MAI is given, which recommends the AWS IoT platform, which occupies 33 % of the global cloud platform market.

4 To simulate the IoT network for sound processing, using platfors, virtual IoT device that would be compatible with protocols and access methods. An ordinary smartphone can be used as the virtual device.

5 The description of the platform modules is presented. A general algorithm for the operation of sound sensors (smartphone) with the platform has been developed. An

algorithm for connecting devices (sensors) to the platform is presented, including the creation of a device on the platform; device and platform certificates, representation of action policies, attach the policy to the certificate and the certificate to the device, create a rule that will call the AWS SNS notification service to send e-mail, use a test tool that allows you to send and receive messages similarly to real devices.

6 During the simulation, the Ios device is authenticated and authorized on the AWS Ios platform. The application adds a timestamp and sends data in the form of an MQTT message to the topic «my/dev-topic». The iOS platform receives a message and activates a rule that sends a message to the AWS SNS notification service, the latter sends an e-mail with the received data to the user. A model of a working IoT network has been created using the Amazon platform.

7 The multiagent system for monitoring sound information using internet of things (MAIOT) is proposed, which consisting of two different agents that works together with the supervision of IoT. Algorithm Multiagent System for Monitoring Sound Information using Internet of Things (MAIOT) is presented and including 2 stages. Installing of the Node platform.js on Raspberry Pi is shoun. To implement a web server and WoT applications for Smart Devices, we will use the Node software platform.js (v10.16.0) for working with JavaScript and the module Express.js (4.17.1) or a framework for Node.js.

8 IoT and multi agent system for detection of audio data for safe environment is discussed. Developed the IoT system with the capability of sensing some types of domestic violence and recognizing with the help of IoT. Hardware and software installation of IoT are shown. This system can be used to detect and report the specific incidents with sounds and use in laboratory of Belarusian State University of Informatics and Radioelectronics for study course «Base of IoT».

9 Sound detecting has been successfully implemented on the Raspberry Pi Single Board Computer and Arduino. This system can be further upgraded and used to reduce crimes, and help people in their daily life. From the tests and the results carried out, it can be deduced that the sound detection module performs best when the voice is loud and when there is silence in the place where the module is performing computation.

References

1 Droin, L. Elaboraiton d'une Cartographie Stratégique du Bruit dans l'Agglomération Messine / L. Droin, A. David, C. Boutin, J. Gouleme. – Metz Metropole, 2009. – 99 p.

2 Xie, D.Y. Mapping Urban Environmental Noise: A Land Use Regression Method / D. Y. Xie, L. J. Chen // Environmental Science and Technology, 2011. 45 (17). – P. 7358–7364.

3 Chu, S. Unstructured Audio Classification for Environment Recognition / S. Chu // Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. – 2008. – P. 1845–1846.

4 Mcheick, H. Modeling Context Aware Features for Pervasive Computing / H. Mcheick // Procedia Computer Science. – 2014. – Vol. 37. – P. 135–142.

5 Chachada, S. Environmental sound recognition: a survey / S. Chachada, C.-C. J. Kuo // SIP (2014). – 2014. – Vol. 3. – P. 115.

6 Chu, S. Environmental Sound Recognition with Time-Frequency Audio Features / S.Chu, S. Narayanan, C.-C. J. Kuo // IEEE Transactions on Audio, Speech, and Language Processing. – August 2009. – Vol. 17. – № 6. – P. 1142–1158.

7 Mitchell, H. A. A Soft K-nearest neighbor voting scheme / H. A. Mitchell, P. A. Schaefer / Int. Journal of Intelligent Systems. – April 2001. – P. 459–468.

8 Moore, A. W. [2004]. Clustering with gaussian mixtures [Electronic Resource]. – Code of Access : http://www.autonlab.org/tutorials/gmm.html. – Data of Access : 02.02.2019.

9 Boser, B. E. A training algorithm for optimal margin classifiers / B. E. Boser, I. M. Guyon, V. N. Vapnik // In D. Haussler, editor, 5th Annual ACM Workshop on COLT. – Pittsburgh: PA, 1992. – P. 144–152.

10 Vlassis, N. A Concise introduction to multiagent systems and distributed artificial intelligence / N. Vlassis. – Synthesis Lectures On Artificial Intelligence and Machine Learning. – 1st edition. – 2007. – 84 p.

11 Stone, P. Multiagent systems: A survey from the machine learning perspective / P. Stone, M. Veloso // Autonomous Robotics. – July 2000. Vol. 8. – № 3. – P. 1–56.

12 Ren, Z. Learning in multi-agent systems: a case study of construction claim negotiation / Z. Ren, C. J. Anumba // Advanced Engineering Informatics. – 2002. – Vol. 16. – № 4. – P. 265–275.

13 Goldman, C. V. Learning in multiagent systems / C. V. Goldman // In Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference. – 1996. – Vol. 2. – P. 1363–1364.

14 Learning in multiagent systems / E. A. Lonso [et al.] // The Knowledge Engineering Review. – 2001. – P. 277–284.

15 Bergenti, A. Three approaches to the coordination of multiagent systems / A. Bergenti, F. Ricci // In Proceedings of the 2002 ACM Symposium on Applied Computing, 2002. – P. 367–373.

16 Damba, A. Hierarchical control in a multiagent system / A. Damba, S. Watanabe // International Journal of innovative computing, Information & Control. – 2008. Vol. 4, № 2. – P. 3091–3100.

17 Choy, M. C. Neural Networks for Continuous Online Learning and Control / M. C. Choy, D. Srinivasan, R. L. Cheu // IEEE Transactions on Neural Networks. – 2006. – Vol. 17, № 6. – P. 1511–1531.

18 Horling, B. A survey of multiagent organizational paradigms / B. Horling, V. Lesser // Knowledge Engineering Review. – 2004. – Vol. 19, № 4. – P. 281–316.

19. Agogino, A. K. Team formation in partially observable multiagent systems / A. K. Agogino, K. Tumer. – NASA Ames Research Center, NTIS, 2004. – 54 p.

20 Vishniakou, U. A. Approach to distributed multiagent system for processing sound information of the environment / U. A. Vishniakou, B. H. Shaya. – System analysis and applied computer science. – 2019. – № 3. – P. 47–53.

21 Parent, C. Semantic modeling for geographic information systems / C. Parent, S. Spaccapietra, E. Zimányi // Encyclopedia of Database Systems. – 2009. – P. 2571–2576.

22 Literature review of spatio-temporal database models / N. Pelekis [et al.] // Knowledge Engineering Review. – 2004. – № 19(3). – P. 235–274.

23 Vishniakou, U. A. Multiagent system for control of sound information / U. A. Vishniakou, B. H. Shaya // Sciense among us (Tomsk). – 2019. – № 10 (26). – P. 1–10.

24 Vishniakou, U. A. Information management and security: methods, models, software and hardware solutions. Monograph. / U. A. Vishniakou. – Minsk : MMU, 2014. – 215 p.

25 Vishniakou U. A. Model and structure of multiagent system for collection and processing sound information / U. A. Vishniakou, B. H. Shaya // Telecommunications: Networks and Technologies, Algebraic Coding and Data Security : materials of the Intern. Scient. and Technical Seminar, Minsk, November – December 2019 / Belarusian State University of Informatics and Radioelectronics; ed. Bobov M. N. [et al.]. – Minsk : BSUIR, 2019. – P. 18–21.

26 Vishniakou, U. A. Structure of internet of things network for sound control in university study process / U. A. Vishniakou, B. H. Shaya // Informatization of education and methods of e-learning: digital technologies in education : materials of the V Intern. Scient. Conf. Krasnoyarsk, September 22–24, 2021: at 2 parts, part 2 / under the general editorship of N. V. Noskov. – Krasnoyarsk : Siberian Federal University, 2021. – P. 391–394.

27 Vishniakou, U. A. Information security in corporate systems, e-Commerce and cloud computing: methods, models, software and hardware solutions. Monograph / U. A. Vishniakou. – Minsk : Bestprint, 2016. – 276 p.

28 Technologies of Intelligence Multiagent Information Processing with Blockchain for Management / U. A. Vishnyakou [et al.] // Open Semantic Technologies for Intelligent Systems (OSTIS-2019) : proceedings of the Intern.

Scient. and Technical Conf., Minsk, February 21–23, 2019 / Belarusian State University of Informatics and Radioelectronics ; editorial Board: V. V. Golenkov (Editor-in-chief) [et al.]. – Minsk, 2019. – P. 311–314.

29 Vishniakou, U. A. Structure and elements of the internet of things network for sound control / U. A.Vishnyakou, B. H. Shaja // Modern means of communication : materials of the XXVI Inter. Scient. Conf., Minsk, October 21–22, 2021 / Belarusian State Academy of Communications ; editor : A.O. Zenevich [et al.]. – Minsk, 2021. – P. 48–49.

30 Ogai, H. Experiments of Wireless Transfer Technology for Communication / H. Ogai, B. Bhattacharya // Pipe Inspection Robots for Structural Health and Condition Monitoring. Intelligent Systems, Control and Automation: Science and Engineering. – 2017. Vol. 89. – P. 61–78.

31 From the desk to the field: Recent trends in deploying Wireless Sensor Networks for monitoring civil structures / L. Mottola [et al.] // IEEE SENSORS Proceedings. – October 2011. – P. 62–65.

32 Massink, M. Model checking dependability attributes of wireless group communication / M. Massink, D. Latella, J. Katoen // International Conference on Dependable Systems and Networks. – June 2004. – P. 711–720.

33 Kecskemeti, G. Modelling and Simulation Challenges in Internet of Things / G. Kecskemeti, G. Casale, D. Jha // IEEE Cloud Computing. – 2017. – Vol. 4, № 1. – P. 62–69.

34 Hussain, F. Internet of Things: Building Blocks and Business Models / F. Hussein. – Cham : Springer, 2017. – 73 p.

35 Saati, T. Decision-making. Method of hierarchy analysis / T. Saati. – M. : Radio and Communications, 1993. – 320 p.

36 Bezruk, V. M. Multicriteria analysis and choice of telecommunication means / V. M. Bezruk, D. V. Chebotarev, Yu. V. Skorik. – Kharkiv : KhNUU, 2017. – 268 p.

37 Vishniakou, U. A. Platform choozinf for IoT network / U. A. Vishniakou // Problem of infocommunications. – 2022. – № 1. – P. 58–64.

38 Cloud Growth Rate Increased Again in Q1; Amazon Maintains Market Share Dominance – Synergy Research Group 2018 [Electronic Resource]. – Code of access : https://www.srgresearch.com/articles/cloud-growth-rate-increased-again-q1-amazon-maintains-marketshare-dominance. – Data of access : 21.01.2022.

39 Keramidas, G. Components and Services for IoT Platforms / G. Keramidas, N. Voros, M. Hubner. – Cham : Springer, 2017. – 383 p.

40 Kondratenko, Y. Multi-criteria Decision Making and Soft Computing for the Selection of Specialized IoT Platform / Y. Kondratenko, G. Kondratenko, I. Sidenko // Recent Developments in Data Science and Intelligent Analysis of Information. ICDSIAI 2018. Advances in Intelligent Systems and Computing. – 2020. – Vol. 836. – P. 71–80.

41 Comparison of IoT platform architectures: A field study based on a reference architecture / J. Guth [et al.] // Cloudification of the Internet of Things (CIoT). – November 2016. – P. 72–77.

42 Kecskemeti, G. Modelling and Simulation Challenges in Internet of Things / G. Kecskemeti [et al.] // IEEE Cloud Computing. – 2017. – Vol. 4. – № 1. – P. 62–69.

43 Piegat, A. Fuzzy Modeling and Control / A. Piegat. – Heidelberg : Springer, 2001. – 728 p.

44 Silva, R. Symbiotic resources sharing IoT platform in the smart cities context / R. Silva, J. Sa. Silva, F. Boavida // IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). – April, 2015. – P. 192–197.

45 Ullah, M. Highlighting the Key Factors of an IoT Platform / M. Ullah, K. Smolander // 42nd Intern. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). – 2019. – P. 901–906.

46 Vishniakou, U. A. Modeling the internet of things network for monitoring audio information on IoT Amazon platform / U. A. Vishniakou, B. H. Shaya. – System analysis and applied informatics. – 2021. – № 2. – P. 28–33.

47 Getting Started with AWS IoT [Electronic resource]. – Access mode : https://docs.aws.amazon.com/greengrass/latest/developerguide/gg-gs.html. – Date of access : 09.03.2021.

48 AWS SDK JavaScript instruction [Electronic resource]. – Access mode : https://aws.amazon.com/ru/sdk-for-javascript. – Date of access : 15.03.2021.

49 Design and Implementation of Cloud-Centric Configuration Repository for DIY IoT Applications [Electronic resource]. – Access mode : https://www.mdpi.com/1424-8220/18/2/474. – Date of access : 19.03.2021.

50 Friendly terminal termux [Electronic resource]. – Access mode : https://techrocks.ru/2019/07/01/termux-android-friendly-terminal. – Date of access : 25.03.2021.

51 Launching the Node JS application on Android [Electronic resource]. – Access mode https://habr.com/ru/post/301442. – Access date : 29.03.2021.

52 Ready-made It solutions for your business together with PSS.Platform [Electronic resource]. – Access mode : http://itpss.ru/iot-platform. – Date of access : 05.04.2021.

53 Vishniakou, U. A. Implementation of the internet of things network for monitoring audio information on a microprocessor and controller / U. A. Vishniakou, B. H. Shaya // System analysis and applied informatics. – 2022. – № 2. – P. 39–44.

54 Kasim, M. Home Automation using Raspberry Pi-3 / M. Kasim, F. Khan // Institute of Technology, Dehradun. 2017. – P. 9–14.

55 McManus, S. Raspberry Pi for dummies / S. McManus, M. Cook. – John Wiley & Sons. 2014. – 192 p.

56 Arduino. Arduino open-source prototyping platform [Electronic resource]. – Access mode : http://www.arduino.cc. – Data of access : 22.5.2022.

57 Lane, N. D. Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learning / N. D. Lane, P. Georgiev, L. Qendro // Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. – ACM, 2015. – P. 283–294.

58 Anggraini, N. Speech Recognition Application for the Speech Impaired using the Android-based Google Cloud Speech API / N. Anggraini [et al.] // Telecommunication Computer Electronics and Control. – 2018. – Vol. 16, № 6. – P. 2733–2739.

59 Endah, S. Comparison of Feature Extraction Mel Frequency Cepstral Coefficients and Linear Predictive Coding in Automatic Speech Recognition for Indonesian / S. Endah, S. Adhy, S. Sutikno // Telecommunication Computer Electronics and Control. – 2017. – Vol. 15, № 1. – P. 292–298.

60 Purwanto, H. Ortopedagogik Umum / H. Purwanto. – Yogyakarta : IKIP Yogyakarta, 1998. – 32 p.

61 Stenman, M. Automatic speech recognition an evaluation of Google Speech / M. Stenman. – UMEA University, 2015. – 29 p.

62 Structure, network protocols of the internet of things for quality production control / U. A. Vishniakou [et al.] // Siberian Journal of Science and Technology. – 2020. – № 4. – P. 478–482.

63 Automation tools for the development of internet of things network / U. A. Vishniakou [et al.] // Telecommunications: Networks and Technologies, Algebraic Coding and Data Security : materials of the Intern. Scient. and Technical Seminar, Minsk, November – December 2020 / Belarusian State University of Informatics and Radioelectronics ; ed. M. N. Bobov [et al.]. – Minsk : BSUIR 2020. – P. 13–16.

64 Modeling a network of things on the basis of a cloud platform / U. A. Vishniakou [et al.] // Coding and digital processing of signals in infocommunications: Materials of the Intern. Scient. study. conf. (Republic of Belarus, Minsk, April 19, 2021) / Redaction : V. C. Konopelko, V. Y. Tsykkov, L. A. Shicko. – Minsk : BSUIR, 2021. – P. 65–68.

65 Ahmed, M. C. Turkish Speech recognition using Mel-frequency cepstral coefficients (MFCC) and Hidden Markov Model (HMM) / M. C. Ahmed, H. E. Coger // Veri Bilim Derg. – 2019. – № 2(2). – P. 39–44.

5 VOICE DETECTION USING CONVOLUTIONAL NEURAL NETWORK

5.1 Task Statement

The integration of information technology with healthcare domain was a point of interest, a well sensitive domain that people lives is involved. However, technology gave a great advance in detecting, diagnosing, and treating diseases, as well as speeding and facilitating the huge work of people working in this domain. Patient monitoring tools, scanning tools (X-ray, MRI, etc.), electronic tools, are all examples of the advantages for this interference. But what seems interesting, and according to a study would experience, is automating the healthcare domain, by making early detection, accurate diagnosis, choosing the medicine, or sharing in surgeries [1].

Dr. Deborah Weatherspoon, a university nursing instructor, has published a study on the deadliest diseases in July 2019, and lower respiratory infections come in the third rank, which affect lungs and airways related infection [2]. Lower respiratory infections are considered contagious such as influenza, pneumonia, bronchitis, and tuberculosis in addition to the new virus COVID-19 which affected the whole world. Early detection of such diseases can prevent the spread in a society and help in controlling the infections besides treating them. All the aforementioned diseases share coughing as a common symptom, yet, cough sound is unique for each one of these diseases and diagnosis of the infection can be done from listening to the cough sound.

COVID-19 was a motivation for researchers to involve machine learning, deep learning, and artificial intelligence in detecting infections to stop its high speed spreading around all regions, inspired by others, cough detection was also a point of interest for many researchers before CoVId-19 pandemic. Lqudaihi et al. [3] stated that 93 papers related to cough detection and classification were found between 2012 and 2021 which indicates the huge interest of researchers in this domain and the benefits that comes from the detection of cough sound or the type of cough. Their survey also focused on the method that are used to classify cough sounds and the obtained results of different approaches. Artificial neural networks, convolutional neural networks, k-Nearest Neighbor, and Deep neural network are an example of the methods used in the approaches either to detect coughing or to diagnose the cough.

Respiratory diseases are known as a common cause for death around the world. «The Global Impact of Respiratory Disease», a report published by Forum of

International Respiratory Societies, shows a very concerning numbers and statistics of deaths and infections caused by respiratory and lung diseases. Around 3 million people die each year from only chronic obstructive pulmonary disease (COPD), and asthma affects more than 334 million people especially children, adding to them acute lower respiratory tract infections that takes the third rank of death causalities, not to mention tuberculosis that 1.4 million died from it and lung cancer the cause of death for more than 1.6 million annually [4]. All the aforementioned respiratory diseases share more than one symptom; cough is one of the mutual symptoms between and an illness can be diagnosed by the cough sound.

Cough is divided into two types, acute that happens suddenly because of flu or cold, and chronic which is more serious and lasts for months. Medics depend on their diagnosis on the sound of a cough when it is related to respiratory infections, because cough sound is unique for every disease, therefore detecting the sound can prevent complications and the spread of the illness.

According to what mentioned above from the serious danger of respiratory diseases and the first diagnosis step it is important to find a solution that can detect cough sound among other sounds to control the infections especially in crowded places as train stations, airports, libraries, universities, etc. Thus taking the advantage of artificial intelligence seems to be important to detect cough sounds in an accurate and fast manner by using machine learning technologies.

Manual detection of cough sound seems to be easy when it happens in a clinic, however when infected people don't realize their diseases or they don't care about other's health, they could travel in planes, trains or share restaurants and libraries with others. So the need raises for automated cough detection in a crowd of people.

The main objective of this work is to develop a well-trained machine learning model that will be able to classify cough sound from many other environmental sounds, in the sake of slowing or preventing the spread of the respiratory diseases in public environments.

The first step for implementing a machine learning model is selecting the dataset in order to select the suitable method accordingly. The used dataset was «Environmental Sound Classification» a public dataset that is labeled according to the sound type. This dataset includes cough sounds in addition to many others in the form of WAV type. Besides, neural network was the used ML technique to build the model according to the given dataset [5].

5.2 Machine Leaning and Neural Network

Machine learning (ML) uses a data set, which is a table containing rows and columns, information about the classified object. The columns represent a characteristic of the object of interest, and each row represents one object. Predictions and pattern recognition using machine learning can be performed using several types [6]:

1 Supervised: input of functions into the algorithm in addition to the result, which is called a class or label. Supervised machine learning is the most commonly used type. Along with this form, there are two main processes: the learning process, in which the algorithm receives big data and allows it to study the output data or result, and the testing phase, in which the remaining data is transmitted to the algorithm to obtain predictions and compare the result.

2 Unsupervised: unlike supervised learning, unsupervised learning is seen as an advanced way of machine learning. An unlabeled data set is passed to an algorithm that will perform its calculations and make predictions without any control.

3 Semi-supervised: a combination of both supervised and unsupervised learning, when algorithms can be assigned a labeled dataset, and training will take place. After that, according to their training, the algorithm continues for the remaining unlabeled datasets.

4 Reinforcement: learning from mistakes, machines can receive a set of instructions and templates that must act according to what is given in their path until the algorithm learns the optimal solution.

With neural networks, both supervised and unsupervised machine learning can be used. Training of a neural network with a labeled dataset can be performed by correcting the error and updating the weights using the back propagation method by moving backwards from the output layer through the hidden layers until the difference between the errors becomes zero. Unsupervised learning does not have a labeled dataset and depends on the categorization and ordering of the data.

Machine learning algorithms depend on some optimal parameters or hyper-parameters to train the model [7]. Hyper-parameters of neural network (NN) are defined:

1 Learning rate: a variable that determines the speed of the learning process and parameter updates. The lower the learning rate, the slower the learning process.

2 Momentum: the direction of what will happen next is known by the momentum, which also helps to prevent fluctuations.

3 Batch size: each training iteration uses a certain number of samples, called the batch size.

4 Number of epochs: the number of iterations in which all input data is transmitted back and forth over the network during the learning process. The weights change during each epoch. The number of epochs should be increased until the model is well fitted.

5 The number of hidden layers: between the input and output layers determines the complexity of the network. The optimal number of hidden layers or blocks can be detected when the error remains constant.

In this article we consider the methodology and means of cough recognition using machine learning, neural networks, Python language tools.

Convolutional Neural Network (CNN) has shown high performance in classification and detection in many areas, especially related to pattern recognition, such as image classification, video tagging and sound classification, etc. The term convolution was first mentioned by Lekun et al. [8], since then CNN has become more popular and attracted serious attention of researchers and programmers [8, 9]. CNN consists of input and output layers and three main hidden layers:

1 Convolutional layer: convolution is a mathematical operation defined as «an integral that expresses the amount of overlap of one function g when it is shifted by another function f» [8], used in a convolutional layer for unification.

2 Pulling layer: dimension reduction is performed inside this layer, this is also called downsampling. There are two types of this layer: maximum and average union.

3 Fully connected layer: before reaching the output layer, each neuron in the output layer connects to each node in this layer.

5.3 Sound Processing and Cough Detection Methods

Encoding of audio signals can be represented in two main types: analog, and digital. Second is more widely used, in which the sound wave is sampled at a certain speed, includes MP3 and WAV types [10]. Audio objects are extracted according to audio data standards. They include the abstraction level (high, mid, low), the time domain (instantaneous, segment level, global), the signal domain (time domain, frequency, time-frequency). The article [11] discusses two types of sound functions in ML:

1 Traditional ML: The input data for the ML model are functions, including both time and frequency domains. An example of the most widely used functions is the Zero crossing rate (ZCR).

2 Deep learning: the approach is based on CNN and RNN methods for extracting signal features, using feature extraction methods known as spectrograms, Mel-spectrograms, Mel-frequency cepstrum coefficients (MFCC), perception linear prediction (PLP) and linear prediction coding (LPC).

PLP: Feroze et al. proposed a CNN classifier for detecting sound events in a polyphonic environment, where the PLP feature extraction method was used for transmission to the network [12]. The authors in [13] presented a comparative cough detection system. They encoded audio signals to visualize them as images and transmit them to CNN using 5 different PLP techniques. The RASTA-PLP spectral method showed best performance among others with an average accuracy of 0.99.

LPC: is considered a function of the time domain. Chowdhury et al. [14] proposed an approach to speaker recognition using MFCC and LPC functions, which are considered to be frequently used functions. These functions are combined with the use of 1D triplet CNN to improve speaker recognition performance.

MFCC: since audio signals are considered non-stationary, so that the values change rapidly over time, most feature extraction methods depend on short-term processing. MFCC are based on the perception of human hearing, since the perception of hearing takes into account the magnitude of the frequency components. Compared to perception of frequencies and magnitudes by human hearing, MFCC uses a nonlinear representation of magnitudes, frequency scaling using filter banks.

A deep CNN with two convolutional layers and two connected layers was trained and tested on the basis of MFCC with a low frequency for three different sets of environmental data in the work [15]. The results of the evaluation after cross-validation training using 10-folds and 5-folds validation showed accuracy for each data set of 64.5 %, 73.1 % and 73.7 %.

Bales et al. [16] presented a respiratory infection monitoring tool using CNN to detect cough sounds in order to diagnose bronchitis, bronchiolitis and pertussis. The environmental sound classification (ESC) dataset was used to feed to CNN, which was pre-processed, the lines were labeled as coughing and non-coughing with 993 samples for each. The raw audio signals from the dataset were converted into a Mel spectrogram, resulting in 432×288×3 images, which, in turn, were converted to grayscale. The CNN structure proposed to reduce its complexity consisted of 5 layers: a 2×2 max pooling layer, two convolutional layers, a ReLU activation function, and a 2×2 max pooling layer, in addition to the mel spectrogram image as an input and output layer. The results obtained after dividing the data set by 70-15-15 % in the quality of training, testing and verification showed an F1 score of 89.35 %.

The authors in [17] evaluated CNN for detecting cough sounds, where the sound sounds were segmented, and STFT was performed to obtain 64×16 spectrograms for a convolutional neural network, and labeled as coughing and non-coughing. CNN consists of five layers, 2 convolutional, 2 fully connected and a Softmax classification layer. The final data set contained 627 samples of cough sounds, also sampled up to 44.1 kHz and 16 kHz. Five experiments were conducted to evaluate the models. Comparative tables showed that the model was able to detect cough with a high accuracy of 82.5 %.

A diagnostic approach to pneumonia was proposed in [18] using logistic regression as a machine learning classifier. Mel Cepstral coefficients and non-Gaussianity index characteristics were added together with the extracted wavelet characteristics from the collected sounds. With the help of a microphone, a data set was collected from 91 patients suffering even from pneumonia, asthma or bronchitis. Sensitivity of 94 % and specificity of 63 % were achieved using wavelet functions. However, after combining other features, the specificity increased to 88 %.

Using CNN, the authors in [19] proposed a cough detection model to prove the possibility of obtaining high performance using sounds recorded by mobile phones. 43 volunteers were asked to cough 20 times at two different distances using 5 different smart phones. The sampling rate was 44.1 kHz, and the data transfer rate was 16 bits. The final data set consisted of 6737 samples labeled as cough and 8854 labeled as control sounds. The results of the two implemented scenarios were evaluated and compared with the previous work, which showed an improvement in accuracy that reached 91.7 % using the CNN and Mel-spectrogram in the first scenario. They also recorded an accuracy of 86.7 % in the second scenario using the same algorithm.

Using Tensorflow, a deep learning library from Google, Khomsei et al. [20] presented a cough detection method using a deep learning network. Principal component analysis was implemented as a feature extraction method. The data set was collected with the help of eight volunteers to record cough sounds using a microphone connected to a raspberry pi, which has a sampling rate of 44.1 kHz. The final data set contained a total of 810 sounds, 229 of them are productive sounds, 74 are unproductive and finally 507 were sounds unrelated to coughing. The division of the training test was divided by 70/30 % in such a way that two types of experiments were used to create the model: the first is to evaluate the dataset using DLN only, and the second is using PCA with DLN.

A machine learning approach using ESC-50 dataset was proposed in [21]. MFCC was first used as an object extraction method to extract 5000×36 objects, then PCA was applied to reduce the dimension to 1D vector 107. Random frog, UVE algorithm and VIP algorithm were used to reduce complexity and select informative features. After applying the feature selection methods, the SVM linear classifier was applied to classify sounds into coughing and non-coughing sounds. UVE with 20 functions showed the best performance results with an F1 score of 95 %.

ESC 50 in addition to two other datasets FSDKaggle2018 and Coughvid were evaluated in [22] in order to eliminate symptoms associated with breathing.

CoughNet, the proposed model, takes input data in the form of a Mel-spectrogram and imports it into CNN. The model is applied to a CNN-LSTM processor that accepts records as input. The prototype is developed using Verilog HDL on the Xilinx Kintex-7 160t FPGA, which has a lower power consumption.

To classify the COVID-19 cough, Bansal et al. [23] implemented a cough classifier using CNN as a machine learning model. Audioset and ESC-50 were the two selected datasets to train their model, which are labeled COVID and non-COVID in addition to MFCC and spectrogram as feature extraction methods for the proposed model. The dataset contained 871 YouTube videos and 40 audio files, which were reduced to 501 audio files separated for training and testing. The CNN architecture consists of three convolutional layers and three fully connected layers. The overall accuracy of the resulting model was 70 %.

5.4 The Proposed Cough Detection System Based on CNN

The proposed system is designed to classify and detect cough sounds. After selecting a sound classification dataset, there are four main steps. The first step is to extract features from audio files, such as MFCCs, chromagram, Mel-spectrogram, spectral contrast and tonal centroid features. The second stage is the labeling stage, so we classify the sound samples into coughing and non-coughing, then we enter the input data into CNN. Then we proceed to the training stage and record the results until we reach the optimal parameters in accordance with the best indicators (change in the number of epochs, learning rate, etc.). At the final stage, after creating the model, several tests were applied to the recorded sounds from the volunteers (Figure 5.1).

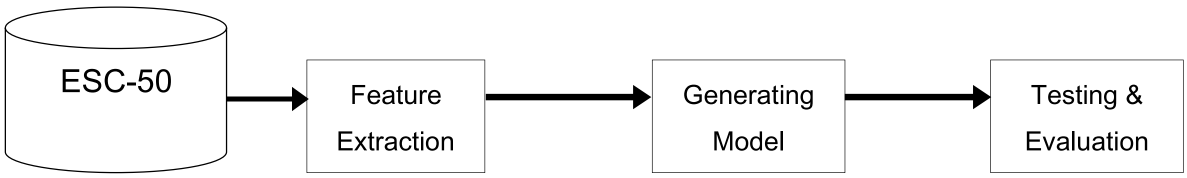


Figure 5.1 – Stages of the proposed system

The ESC dataset [24] is a set of 2,000 sound files representing 50 types of environmental sounds. ESC-50 was published in 2015 and has been used in many publications and systems. ESC-50 was used for research and downloaded from Kaggle's website for data scientists, which contains over 50,000 datasets in addition to over 400,000 publicly available notebooks.

More than 60 % of developers use Python because of its simplicity and a wide range of useful libraries, such as Pandas, Sci-Kit learn, Matplotlib, Seaborn and others. Tensorflow [25], an open source ML platform developed by Google for deep machine learning. Google Colab, designed for Python programmers. Colab specializes in data science and ML, users can import datasets from Kaggle. For our project, the following Python libraries and modules were used, which are related to deep machine learning and data science:

1 NumPy is a library used for structuring arrays, manipulating shapes, sorting. Several functions from NumPy were used, such as the «absolute» function, which was used with the Librosa library to extract short-term fourier transform (STFT) features from audio files.

2 Librosa is an indispensable library used for audio, speech and music analysis. Using Librosa, six methods of feature extraction were applied, starting with STFT, then MFCC and Mel-spectrogram, the remaining three methods were spectral contrast, chroma and tonnetz.

3 Matplotlib: for plotting and visualizing arrays, data and statistics. We used it to construct the matrix of the system and related graphs.

4 Hickle is used to save functions extracted from audio files in HDF5 type format.

5 Sci-Kit Learn: used to implement machine learning algorithms, including supervised and unsupervised. SK-learn was used to obtain model results and metrics like accuracy and recall.

6 Keras: to implement a deep learning model. The training, testing and evaluation of our system is based on Keras and its built-in functions.

Since our dataset is audio data, we used the Librosa library to extract functions, characteristics (STFT, MFCC, Mel-spectrogram, chromati, spectral contrast and tonnetz) and analyze them. 193 features as total were extracted using Librosa from all the above mentioned methods and resulted in a data-frame that contained 1977 rows and 193 columns from all the input data The train function accepts the following parameters: objects, labels, type, number of classes, epochs, and optimizer [26].

The neural network adjusts its weights according to the results obtained after each iteration using various algorithms known as optimizers, which implies calculating the difference between the results. The implemented network uses the stochastic gradient decent as an optimizer for its simplicity, which calculates the gradient of the loss function in the network. The initial learning rate for this system was 0.1 and momentum – 0.9. The functions are those that were saved using Hickle, as well as labels. The number of epochs usually needs to be changed to get the best results, we have increased the number of epochs from 300 to 500, 750, 1200, 1500.

To predict, some sounds were tested and classified using the «predict» function in addition to some recorded cough sounds of volunteers. The prediction method was implemented using a loop to iterate through several audio files and output the accuracy of the detected result for the top three classes that received the highest results. Using the Librosa and Matplotlib libraries, the extracted objects were presented as images showing the different forms of each extraction method.

Conclusion on Chapter 5

1 This chapter introduces the state of art of the most relevant cough detection systems and discuss their obtained results. As well, a well-designed cough detection system was build using the public dataset ESC-50, that was fed a convolutional neural network. The layers of the network as well as the training methods were stated in details [26].

2 After generating the model, another sound set was tested in order to evaluate the model. Thus, several volunteers recorded their voices while coughing using their smart phones and it was assured to record their voices in a public environment to introduce noise to the sounds, in addition to some audio files that were downloaded online [26].

References

1 Shakel, N. V. Medical worker and patient: interaction in the context of e-health / N. V. Shakel, M. S. Ablameyko. – Minsk : Ekoperspektiva, 2020. – 196 p.

2 Weatherspoon, D. The Top 10 Deadliest Diseases. Overview / D. Weatherspoon. – Health line. – July 2019. – P. 1–10.

3 Cough sound detection and diagnosis using artificial intelligence techniques: challenges and opportunities / S. K. Lqudaihi [et al] // IEEE Public Health Emergency Collection. – 2021. – № 9. – P. 27–44.

4 The Global Impact of Respiratory Disease. European Respiratory Society. Sheffield, 2017. – 43 p.

5 Amoh, J. Deep Neural Networks for Identifying Cough Sounds / J. Amoh, K. Odame // IEEE Transactions on Biomedical Circuits and Systems. – 2016. – № 10 (5). – P. 1003–1011.

6 Radhakrishnan, P. Towards Data Science [Electronic resource]. – Code of accsses : https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a. – Data of accsses : 02.08.2022.

7 Backpropagation Applied to Handwritten Zip Code / J. LeCun [et al.] // Neural Computation. – 1989. – Vol. 1, № 4. – P. 541–551

8 Dhillon, A. Convolutional neural network: A review of models, methodologies and applications to object detection / A. Dhillon, G. K. Verma // Progress in Artificial Intelligence. – 2019. – Vol. 9, № 2. – P. 85–112.

9 Kriszhevsky, G. K. ImageNet classification with deep convolutional neural networks / G. K. Kriszhevsky, G. Sutskever, G. E. Hinton // Proceedings of the 25th International Conference on Neural. – 2012. – Vol. 60, № 6. – P. 1097–1105.

10 Chu, S. Unstructured Audio Classification for Environment Recognition // Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. – 2008. – P. 1845–1846.

11 Banchhor, K. S. Musical Instrument Recognition using Zero Crossing Rate and Short-time Energy / K. S. Banchhor, A. Khan // International Journal of Applied Information Systems. – 2012. – Vol. 1, № 3. – P. 85–97.

12 Feroze, K. Sound Event Detection in Real Life Audio using Perceptual Linear Predictive Feature with Neural Network / K. Feroze, A. R. Maud // Proceeding of 15th International Bhurban Conferance on Applied Sceinces & Technology. – Islamabad, Pakistan, 2018. – P. 1–6.

13 Audio Signals Encoding for Cough Classification Using Convolutional Neural Networks: A Comparative Study / H.-H. Wang [et al.] // IEEE International Conference on Bioinformatics and Biomedicine. – Shanghai. – 2015. – P. 1–4.

14 Chowdhury, A. Fusing MFCC and LPC Features using 1D Triplet CNN for Speaker Recognition in Severely Degraded Audio Signals / A. Chowdhury, A. Ross // IEEE Transactions on Information Forensics and Security. – 2019. – Vol. 15. – P. 1616–1629.

15 Piczak, K. J. Environmental sound classification with convolutional neural networks / K. J. Piczak // IEEE International workshop on machine learning for signal processing. – Boston, 2015. – P. 1–7.

16 Can Machine Learning Be Used to Recognize and Diagnose Coughs? / C. Bales [et al.] // The 8th IEEE International Conference on E-Health and Bioengineering. – Romania, 2020. – P. 1–10.

17 Amoh, A. Deep Neural Networks for Identifying Cough Sounds / A. Amoh, K. Odame // IEEE Transactions on Biomedical Circuits and Systems. – 2016. – Vol. 10, №5. – P. 1003–1011.

18 Wavelet Augmented Cough Analysis for Rapid Childhood Pneumonia Diagnosis / K. Kosasih [et al.] // IEEE Transactions on biomedical engineering. – 2015. – Vol. 62, № 4. – P. 1185–1194.

19 Towards Device-Agnostic Mobile Cough Detection with Convolutional Neural Networks / F. Barata [et al.] // IEEE International Conference on Healthcare Informatics. – Xi'an, 2019. – P. 1–12.

20 Khomsay, S. Cough detection using PCA and Deep Learning / S. Khomsay, R. Vanijjirattikhan, J. Suwatthikul // Intern. Conf. on Information and Communication Technology Convergence (ICTC). – Jeju, 2019. – P. 1–6.

21 Chen, X. Cough Detection Using Selected Informative Features from Audio Signals / X. Chen, X. Hu, G. Zhai // 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). – Cirnell University, August, 2021. – P. 1–7.

22 Rashid, H.-A. CoughNet: A Flexible Low Power CNN-LSTM Processor for Cough Sound Detection / H.-A. Rashid // IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS), Washington DC, 2021. – P. 123–126.

23 Bansal, V. Cough Classification for COVID-19 based on audio mfcc features using Convolutional Neural Networks / V. Bansal, G. Pahwa, N. Kannan // IEEE International Conference on Computing, Power and Communication Technologies (GUCON). – Greater Noida, 2020. – P. 604–608.

24 Piczak, K. J. ESC: Dataset for Environmental Sound Classification / K. J. Piczak // Proceedings of the 23rd Annual ACM Conference on Multimedia. Brisbane, 2015. – 4 p.

25 TensorFlow: A System for Large-Scale Machine Learning / M. Abadi [et al.] // Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16). – November, 2016 Savannah, GA, USA. – P. 265–283.

26 Vishniakou, U. A. Proposed system architecture for cough detection / U. A. Vishniakou, B. H. Shaya // Modern means of communication: materials of the 27th Intern. Scient. Conf., Minsk, October 27–28, 2022 / BGAS ; editorial board : A. O. Zenevich [et al.]. – Minsk, 2022. – P. 29.

Св. план 2022, резерв

Научное издание

Вишняков Владимир Анатольевич

СПЕЦИАЛИЗИРОВАННЫЕ IoT-СЕТИ: МОДЕЛИ, СТРУКТУРЫ, АЛГОРИТМЫ, ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА

SPECIALISED IoT SYSTEMS: MODELS, STRUCTURES, ALGORITHMS, HARDWARE, SOFTWARE TOOLS

Редактор *Е. С. Юрец*
Компьютерная правка, оригинал-макет *А. А. Лущикова*

Подписано в печать 22.03.2023. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 10,81. Уч.-изд. л. 13,8. Тираж 50 экз. Заказ 227.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
Ул. П. Бровки, 6, 220013, г. Минск