Deep Science Publishing
https://doi.org/10.70593/978-93-49910-80-5_6

Chapter 6: Development of real-time monitoring solutions using embedded sensors and edge computing

6.1. Introduction

The rapid development of wireless technologies has enabled the creation of a wide range of low-cost embedded network sensors that allow easy access to information concerning air quality in urban areas. Through the capture, transfer, and subsequent processing of information, it is possible to understand the existing conditions in a given area. However, the overwhelming amount of data that the sensors are generating in real-time needs to be efficiently processed in a timely manner in order to detect and warn against possible fluctuations and emergencies in the environment. For that, real-time processing of embedded sensor data using edge computing solutions has been implemented to achieve a greater level of reactivity and lower network demands. The main purpose of this chapter is to present two real-time monitoring solutions employed for measuring indoor and outdoor air quality. The first case focuses on the real-time tracking of volatile organic compounds present in an air sample captured by a low-cost e-nose, which performs the identification of different types of polluted samples using machine learning techniques. The second case points to the integration of an indoor air quality monitoring system based on technology, which uses an adapted technique for data processing in a distributed way (Alrowaily & Lu, 2018; Gusev & Dustdar, 2018; Jeong et al., 2022).

The interest in real-time processing of embedded intelligent sensors for monitoring environmental pollution has gained a lot of attention over the last few years. The embedding and deployment of powerful sensor nodes in a given area allow the collection of information regarding the quality of the environment at a specific location in a spatiotemporal way. However, in certain scenarios, the amount of data that the sensors need to send may exceed the communication bandwidth of the network. Environmental monitoring applications typically generate large amounts of data since they use high sampling rates in order to capture atmospheric transients that can occur very quickly. Data transmission at high rates can lead to problems such as data overloading, increased latency, temporal data correlation, excessive energy consumption, and excessive costs in the network infrastructure (Yu et al., 2018; Liu et al., 2021).

6.1.1. Background and Significance

The increased number of sensor nodes continuously monitoring processes has generated large amounts of sensor data and stimulated the research into data processing at the sensor nodes or in their immediate proximity, utilizing small-size devices with limited energy, storage, and computational capabilities. The memory limitations imposed in these devices often do not allow storing large time-series for either on-line or off-line data analysis. The energy limitation of the devices usually determines the need for on-line data analysis to generate decisions eliminating uninformative, from distributed decisions' quality standpoints, data transmissions. The computational limitations necessitate rather low complexity and low power consumption algorithms. These constraints on embedded data processing will be reviewed below.

There are many ways to improve data processing at embedded devices. For example, on-line decision-making allows to reduce the data stored in sensor nodes' internal memory or the data networked to remote users and/or computation/energy required for delivery or remote analysis. Such embedded data processing should also provide good or satisfactory quality of decisions, considering the limited computation power of the units. Suboptimal and approximate algorithms should be designed reducing the energy and time needed for process state decision-making. For certain applications, it might be also both practically and economically more advantageous to process a limited amount of raw data and transmit these data to a remote user. In this case, raw data transmission is triggered by the embedded process on-line decision-making results. Embedded real-time monitoring process decision-making systems with the "small" sensor nodes will be focused on in this chapter.

One of the process characteristics that can be calculated from a small amount of data is the process mean value, its change being an indicator of a change in the embedded process state. This statistical characteristic is monitored both for engineering/computer processes as well as for medical processes and other applications. Embedded means' monitoring can be conducted without the introduced monitoring costs being introduced except for the costs of embedded sensor nodes.

6.2. Overview of Embedded Sensors

Embedded sensors are the crucial elements of any information gathering device. Today, the reduced form factor, reduced power consumption and enhanced performance characteristics of embedded sensors are enabling the deployment of highly distributed and mobile wireless sensor networks. Compared to traditional monitoring devices, embedded sensors on tiny motes and boards are optimized for use in real-time monitoring applications. In particular, embedded sensors are becoming available for gathering data in a variety of application domains, including monitoring the environment, health and well being of individuals and their activities indoors and outdoors, and observing characteristics of various structures. The tiny form factor and optimized design at low cost for such embedded sensors have facilitated their increasingly ubiquitous deployment. With the large numbers of such sensors predicting the future state of a monitored system and reporting such information to some monitor station for visualization and analysis, it is possible to initiate appropriate actions. The field monitoring played a vital role in solving the problem of negative impact in the environment due to the influence of disturbances.

Besides being on tiny motes or boards, embedded sensors are also available as single sensor ICs, packaged sensors, and bare die MEMS sensors. Applications include the use of MEMS sensors for motion detection in portable electronics, the use of packaged gas sensors in odor detection and other air quality applications, temperature sensors in automotive applications, and the use of bare die pressure sensors in mobile cellular.

6.2.1. Types of Embedded Sensors

Embedded sensors are a specific class of sensor that are small, self-contained, low-cost, and application-specific. Embedded sensors can consist of the sensor element and a small amount of embedded processing, storage, and connectivity modules. Embedded sensors have the advantages of miniaturization, low-cost and low-power consumption due to few external components, lower assembly and manufacturing costs, minimal field wiring, and resilience to the deployment environment through their compact packaging. The small size of embedded sensors enables direct integration of the sensor element with the embedded intelligent modules, resulting in a compact device that is small enough for a specific application. Evaluating the cost and power consumption tradeoffs associated with a small number of external support components can provide added flexibility in the design choices for embedded sensors.

Embedded sensors can be classified based on several factors, including the quantity and type of data being processed, the number of devices performing the processing, and the spatial-temporal scales. Embedded sensors can perform addition, multiplication, filtering, and higher-level processing functions where sensor data from one or more devices embedded in a smart object are processed locally before wireless communication, thereby conserving communication bandwidth and energy. Embedded sensors operate in a spatial-temporal profile that depends on the scale of the application, with hundreds to thousands of short bursts of data generated in response to a single thermal or mechanical event, or longer-term monitoring of an environmental state with long periods between individual data samples.

This is done by embedding non-invasive sensors within patches, clothes, and various accessories such as watches, glasses, belts, and bands. These mobile units communicate wirelessly and without interrupting the mobility of the individual with implanted physiological logger devices to monitor and manage the health of the individual.

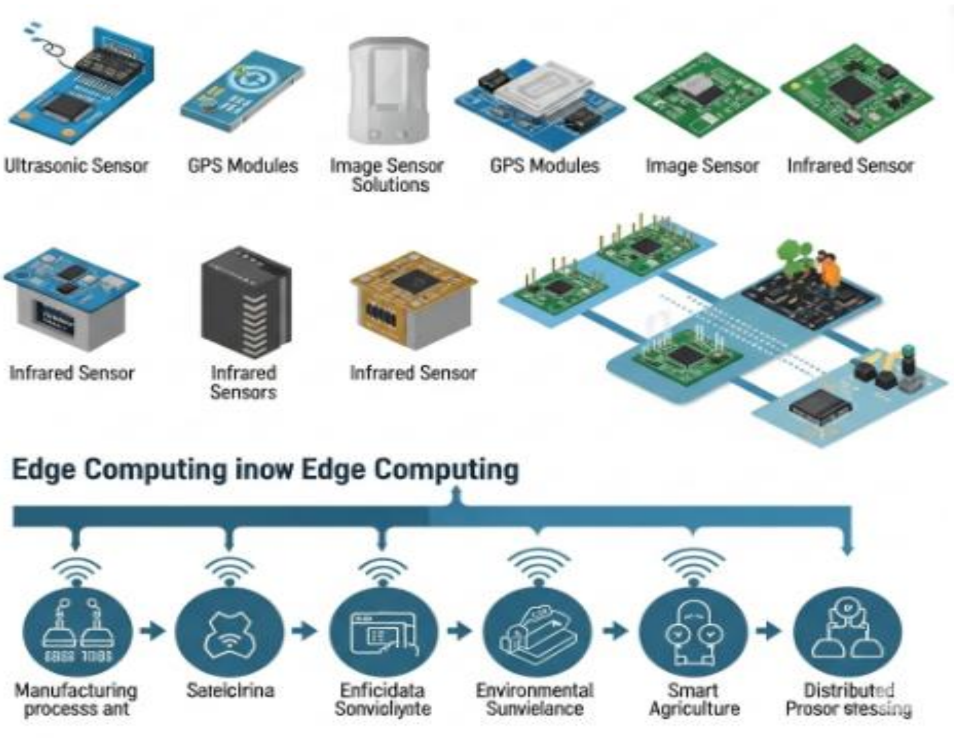

Fig 6.1: Types of Embedded Sensors.

6.2.2. Applications of Embedded Sensors

Embedded sensors are gaining popularity due to their importance in our everyday lives and ease of use. Embedded sensors measure physical or biochemical changes in ambient and transmit the output signal to a microcontroller or higher processing unit. There is a wide variety of sensors available, which can measure such parameters as temperature, humidity, light, proximity, acceleration, pressure, vibration, moisture, UV, IR, smoke, gas, and radioactive particles, as well as analyze ECG, EEG, and blood and insulin glucose levels, to name a few. Available sensors and their use cases are listed at the end of this section. Wireless Sensor Networks (WSNs) consist of small-scale sensors that have the ability to collect information about their environment, manipulate data, communicate over a wireless network, and interface with the outside world via signal processing. Available sensor solutions for WSNs include various platforms. With the commercialization of WSNs, it is possible to find consumer applications based on WSNs. These applications span over different domains such as precision agriculture, home automation, industrial automation, utilization monitoring, wildlife monitoring and tracking, healthcare, intelligent and smart buildings, military and instrumentation, security, road and traffic monitoring, hazardous area monitoring, as well as which people are adjudicated necessities. Continuous human monitoring of patients with chronic diseases and identification of the changes in their dynamic behavior.

6.3. Edge Computing Fundamentals

Cloud computing delivers computing power by aggregating abundant and cheap resources in large facilities dispersed in the world, creating virtual computing nodes. An edge computing infrastructure makes a better use of existing resources in support of time-sensitive services. Edge computing collects, analyzes, and shares data generated by devices in proximity to the user. It employs a small but powerful micro-data center that can optimize a given application response time. Edge and fog computing both overcome cloud computing latency. However, fog and edge puppets need response execution in the real world, and edge computing cooperates with the cloud when required. Edge computing builds a weak but close virtual time node that algorithms can use for overcoming a problem unique to the service. Reducing the data traveling time associated with an edge puppet response execution can save not only waiting but also traveling times.

Edge computing focuses on moving the service as close as possible to the user. Its main advantage is the associated latency reduction. Latency involves the user waiting for the result of an action requested to the edge puppet instance. Other systems affected by latency involve remote developers involved in the system design that provides assistance for problem solving, service setup, internet connection recovery, and other dynamic support. Data transmission requires communication between the edge puppet and other system elements. Latency can be powerful for avoiding the service request and response wait. Furthermore, latency can be negative for data transmission, when it reduces the volume of data traveling along edges where the communication speed is high. Data traveling time also requires being taken into account to support the edge puppet task.

6.3.1. Definition and Importance

Before the development of the Internet of Things (IoT) with its many devices and massive processing needs, there was a clearly established hierarchical model of thought for the development of computer systems. IoT accelerates the evolution of the most diverse sectors of society – industrial, agricultural, health, services, security, etc. – requiring real and immediate solutions to the challenges that arise. As people migrate to urban areas, cities will be under increasing pressure in terms of quality of life, such as waste disposal, environmental monitoring, smart public transport systems, health service management, building energy consumption, basic infrastructure and city logistics. Edge Computing arises to overcome the challenges of high latency in the delivery of data from the sensors to the datacenter to be processed and the delivery of the result back to the place where the decision is to be made. The longer the latency of a system, the less utility it will have. These are the characteristics that determine that in many situations it is impossible to make the decision in the datacenter.

Edge computing systems are essentially decentralized computing systems. These are generally more difficult to implement than traditional hierarchical computer systems because the distribution of computational capabilities throughout the hierarchy must closely follow a defined criterion such as a spatial or logical divide and distribute criterion with the appropriate computational capabilities according to the need, including data collection, actuator control, and real-time processing. On the other hand, with the explosion of the need for processing power from the millions or billions of Internet connected devices, it is increasingly necessary to create a solution that uses the available devices to process the data generated by other devices, thereby decreasing the burden on the datacenters where most of the processing is still carried out.

6.3.2. Architecture of Edge Computing

Edge computing promotes the emergence of localized processing by creating a virtual repository of computing and storage resources among end-user devices and edge servers located close to the sensing devices. The architecture connected to the edge of the network is typically made up of edge devices that perform operational data pre-processing and filtering prior to transfer to a data center for longer-term storage, as well as high-level analytics and algorithms for knowledge generation. This architecture has the potential to reduce operational costs associated with bottlenecks arising from high volumes of data transfer across long distances over the internet. Connectivity costs can increase relative to the volume of data transferred through telecommunications networks, particularly for high volume and continuous transmission of data originating from sensor nodes that are deployed as part of industrial, environmental, or safety monitoring solutions. Bandwidth may also be a limitation, with a fixed capacity of line speeds that determines how quickly packets can be transmitted and received, and latency delays are increased for communications that must traverse greater distances throughout the path from the original source of information to destination. Furthermore, continuous transmission of raw data sans processing may be excessive and often unnecessary because end-user requirements for high-level event detection may be satisfied using techniques that employ local intelligence to detect such events. With early event detection and alarm escalation using pre-conditioned informational templates, there is less need for satellite-based systems to be operational on a 24/7 basis. Additionally, connectivity is not guaranteed and can be less reliable during times of natural disasters; therefore, intermittent local data storage is critical. More specifically, local storage is important for the short-term collection of sensor data during times of interrupted connectivity, coupled with continuous local processing and filtering to pre-condition data for subsequent upload.

6.4. Integration of Embedded Sensors with Edge Computing

These embedded sensors, however, come with intrinsic constraints in terms of limited computing and storage resources, battery life, and network bandwidth and latency. Whenever possible, they resort to data pre-processing and low-level data analysis at the edge nodes, such as filtering, aggregation, and running lightweight algorithms. This way, they decrease data flow and extend battery life, thus reducing deployment costs and enhancing the overall monitoring solution efficiency. Moreover, in several applications, data processing at the edge provides results at sub-second latency, which is paramount for the responsiveness of mission-critical real-time applications, such as patient lifeline monitoring. In this section, we first address the choice of communication protocols that enable the connection of heterogeneous embedded sensors to the edge computing infrastructure. Thus, the discussion inputs the more general question of how to effectively integrate embedded sensors with edge computing infrastructure within monitoring solutions. The proper integration addresses not only the communication and data transport between the devices but also security and power management, configuration, and precious data transfer delay, according to the application needs. Then, we summarize edge data processing via function and data storage implementation. Despite being a recent concept, edge node pre-processing and analysis of raw data being gathered by embedded sensors is important to address issues in the implementation of solutions. We cover state-of-the-art work that already addressed this issue, introducing an important degree of homogeneity to the data processing solutions contributing to the design of such solutions. The integration of edge nodes into the solution is also an important question related to the power that they present, which will influence the implementation of applications with different levels of complexity.

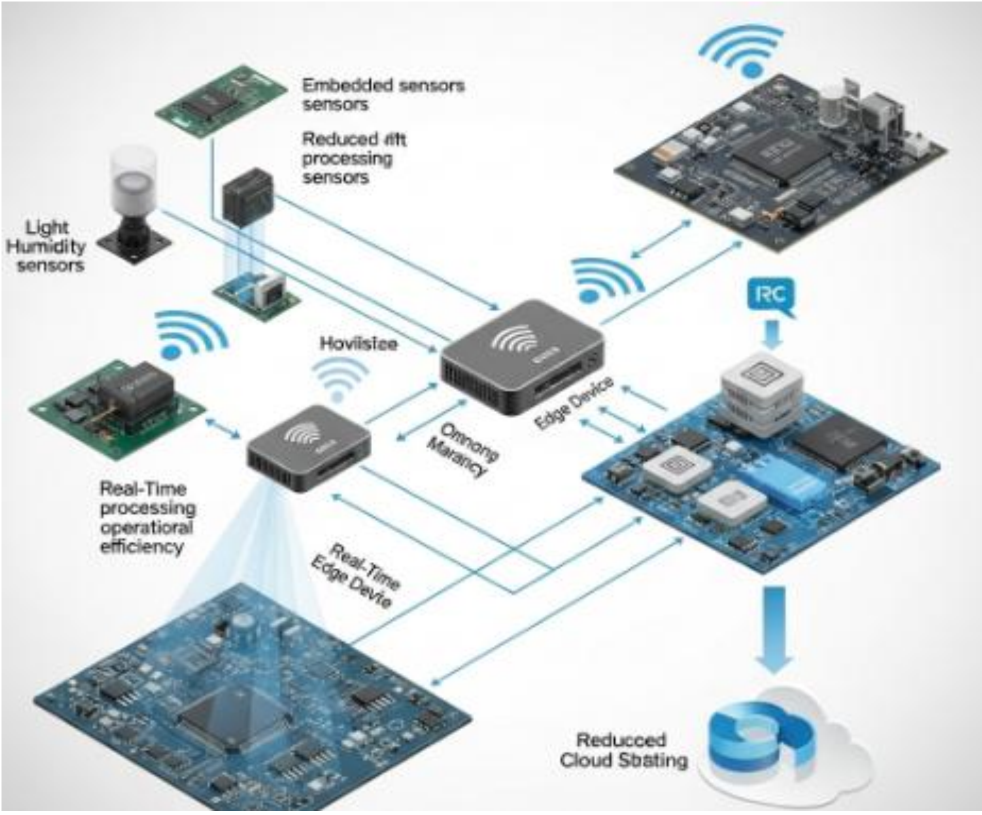

Fig 6.2: Integration of Embedded Sensors with Edge Computing.

6.4.1. Communication Protocols

Embedded sensors are mainly integrated with edge computing devices for preprocessing, filtration, and data fusion before the transmission of critical information to the cloud for remote monitoring and control. Custom integration of embedded sensors and edge computing platforms requires coordination and synchronizations between a variety of heterogeneous components including sensors, computing modules, batteries, antennas (both transmission and reception), and sometimes even visual and user interface devices. Customization of such systems leads to application specific needs, resource limitations, and cost budget. The data transmission is often realized over radio protocols that may be configurable, while some other devices require wiring for custom communication modes.

The standard communication protocols ensure compatibility, interoperability, and collaboration between various apparatus, but customizing them according to application needs may ensure better performance. Embedded systems require reliable communication that ensures embedded sensors' real-time operation, perform data synchronization with edge computing devices, and if possible, suit the energy and bandwidth constraints. The communication path may be uni- or bi-directional depending on the pattern of the embedded system's operational use cases. It could also be periodic, at certain intervals, or continuous, depending on the installed sensors or data computational and processing loads. Real-time embedded sensors and edge computing integration solutions also require timely sensor control information to ensure reliable operation under various scenarios.

Industrial solutions comply with certain protocols such as I2C, SPI, RS485, RS232, UART, CANBus, LINBus, OneWire, etc. However, the lack of common standards and many proprietary communication protocols for specific energy-efficient networks leads to benefit analysis for both hardware and software custom design.

6.4.2. Data Processing Techniques

The focus of this section is to provide an overview of data processing techniques suitable for edge computing. The three important questions in processing real-time data from sensors are: 1) How to represent data so that the processing is optimal? 2) What operations to perform to extract the desired information as quickly as possible? 3) How to implement these operations for efficient execution on low-cost microcontrollers? These questions are discussed in detail in the following subsections.

6.4.2.1. Data Representations

A central issue in the development of data processing techniques is finding an optimal representation of the data. To support many applications in a natural manner, it is necessary to represent data with different spatial and/or temporal resolutions. The combination of multiple resolutions increases the complexity of implementation for data processing, storage, and transmission. Codecs encode data with different resolutions for different time scales and are implemented to optimize data representation based on features derived from data. The principles of these codecs can be used to represent sensor data in a form that is compact and efficient for transmission and is easily amenable to data processing. Moreover, they can be designed to represent data continuously over the desired time interval and to focus on parts of data that are more important.

6.4.2.2. Data Processing Operations

Various operations are performed on the data obtained from a sensor. Filtering has been the most common operation that reduces a continuous stream of noisy data from sensors to that of the desired signal. Other operations include detection of specific features or events, such as one on activities for surveillance application. These operations can also act in tandem, where sensors are designed to perform minimal processing to coarsen data from continuous monitoring in situ and transmit only when needed.

6.5. Real-Time Data Acquisition

The availability of real-time data is crucial for the design of quality monitoring or decision-support systems within the framework of the Internet-of-Things, where devices are used for collecting, analyzing and transmitting monitoring data. Devices need to sample at sufficiently high rates and with sufficient resolution to capture relatively fast phenomena that may indicate a change in the observed phenomenon. Furthermore, a sufficiently low latency is also required from the sampling, processing and transmission cycle. Power consumption considerations play an important role as well, usually requiring work in a low-power or sleep-consumption mode. On the receiving side, as the number of monitoring devices can be large, the channel capacity required for each node to upload full-resolution data will likely exceed the maximum possible capacity of the channel, indicating that further measures are required to minimize the amount of bandwidth used.

A variety of sensors and systems are available for online monitoring including high-throughput, non-invasive, real-time systems, such as hyperspectral imaging, infrared thermography, thermal imaging, and X-ray imaging. The continuous inspection at a high throughput requires fusion methods for the different multi-sensor systems for getting information about the quality attributes being monitored. Such platform systems either fuse different signals in a multi-sensor system or use sensors that can detect different attributes using the same platform. The choice of sensors and their mode of operation must be guided by the requirements for the necessary resolution and throughput.

6.5.1. Methods of Data Collection

Real-time Data Acquisition aims at continuously acquiring task-relevant data, as the task is being accomplished, without any gap. It enables researchers to closely observe the procedure being followed by the subjects and see how they accomplish the task. For example, a psychologist can use this approach to observe facial expressions to observe a person under an interview condition; however, it may be difficult if the person is asked to recall a video and report on the facial reactions in detail. Psychologists, anthropologists, and behavioral scientists appreciate such a methodology to understand any immature processes, which cannot or may not be restructured simply by retrospective methods. In an experimental design, the time lag may also become significant in cognitive task experiments where the dependent measures are time-critical reaction times, because of the lag between the physical stimulus and its reception by the subject. Psychologists use a variety of practical tools to collect response latency and other physiological or subjective data.

The instruments which they use significantly depend on the task demands, subject age, and experimental or methodological design. For example, the communication process between a child and a caregiver is generally recorded using a video or audio recording device. In addition, there are a variety of sensors, including physiological sensors, eye-tracking equipment, transcranial doppler ultrasound, functional magnetic resonance imaging, Eeg, and NIRS systems that allow scientists to collect real-time data. Such studies may have the objective of discovering the timing and interaction of the physiological component of task performance in relation to behavioral activity, or in general, a relationship between the physiological and psychological variables of human beings.

6.5.2. Challenges in Real-Time Monitoring

The use of embedded sensors removes physical barriers to device placement, enabling real-time monitoring of assets or processes at scale. However, the distributed nature of data acquisition presents challenges that impact data quality, volume, cost, and latency. This section outlines the unique challenges of real-time embedded and edge monitoring, and selects solutions to those challenges.

There are several challenges with real-time embedded and edge monitoring that must be taken into consideration when designing a solution that will impact the quality and cost of the solution. First, the collection of monitored data by embedded sensors and edge computing systems drastically increases the size of data related to monitored assets or processes, increasing the need for long-term storage capabilities. Second, the risks of damaging potentially expensive or hazardous assets or processes reduces the available options for sensor placement. Third, devices are often deployed without full integration into existing operations, meaning that some processes may go unmonitored. Fourth, the duration of monitored phenomena are often poorly understood or highly variable meaning that data are often collected for long periods without signal of interest being present, limiting the utility of such large data volumes. Fifth, signal processing is necessary during data collection on embedded sensors at the point of data collection to reduce the data volume, add value to existing data, and to extract signals of interest from background noise. Lastly, localized conditions can vary widely and rapidly impacting the data quality, and therefore the need for adaptive data quality monitoring capability at a localized level is key to a successful embedded sensor and edge computing solution.

6.6. Data Analytics for Real-Time Monitoring

The emergence of the Internet of Things (IoT) and the availability of numerous smart devices allows for the implementation of enhanced real-time monitoring methods in various fields and industries, such as smart home, healthcare, smart cities, precision agriculture, and intelligent transport systems. With the availability of embedded sensors and networks of devices, new proposed monitoring methods intelligently combine data from different devices and places in order to increase user awareness. Data analytics and other tools are used to approve, cleanse, validate, and analyze data coming from various sensors and send timely updated alerts to end users. Data analytics tools could be cloud-based, while for very constrained or real-time applications, and when it is required to avoid cloud latency, they could also be running locally on networks of edge computing devices. In this chapter, we analyze different techniques of the data analytics stage and on how they could impact the performance of the proposed algorithms.

The data analytics techniques rely both on expert-constructed heuristics, or rules, and on artificial intelligence techniques, mostly collected from supervised and unsupervised learning techniques. Due to the very high volume of data generated from the sensors of different nodes, the employment of statistical and AI techniques is essential. However, most of the available commercial tools for AI techniques are still cloud-run and not installed locally on the devices themselves. Although this is in contrast with the requirement for the real-time performance of the proposed algorithms, some attempts are made for implementing them on local networks of low compute power edge devices. Also, cloud-based tools for data analytics are used for off-line training or fine-tuning of AI algorithms, where an expert supervises the output as well as adopts appropriate parameters of the employed techniques. Due to their unique ability to discover hidden patterns in very high-dimensional datasets, deep learning techniques have been very popular in the process of modifying raw sensor data for achieving enhanced performance.

6.6.1. Machine Learning Techniques

The Data Analytics phase takes the processed data from the Edge Computing Engines and machines Cloud Intermediary Systems. Several machines share the same Data Analytics Engines in the Cloud. The data-generated verification or action instructions come from Dashboards related to high-level decisions. Consequently, the Data Analytics Engines comprise an important part of the capacity of the system. Therefore, a wide set of Machine Learning Techniques is used in this project.

MLP is a type of ANN that consists of an input layer, at least one hidden layer, and an output layer. Each layer is composed of nodes that are connected through directional, weighted connections. Each node contains an activation function that processes the input from the previous layer. MLP uses a backpropagation training algorithm, which minimizes the error in the output of the nodes. The training process requires a dataset labeled with the correct outputs. MLP are the simplest form of neural networks.

RF are supervised learning methods. They are applied in classification and regression tasks but are more commonly utilized in classification problems. In the RF, decision trees are used as the base classifiers, which are combined in an ensemble to describe the concept better than any single classifier. They build a forest using a random sample of the training dataset and create a decision tree for each sample. As the redundancy is finished among the training datasets and makes the forest diversify, the RF takes the average of all the predictions made by the different decision trees.

KNN is used in classification problems. This method classifies the sample data into the class that contains most of its K neighbors. The KNN is a simple algorithm that stores all available cases and categorizes unknown cases based on a similarity measure. The KNN ignores the internal structure of the data. Thus, this algorithm is a memory-based system and requires a larger amount of storage by default. This method is suitable for bigger datasets with enough storage since it achieves maximum accuracy.

6.6.2. Data Visualization Tools

Data Visualization enables better interpretation of large amounts of data, from the real time sensor data obtained with the above ML algorithms. Environments such as NodeRed include easy to use dashboards. Node-RED is a flow-based development tool for visual programming for wiring together hardware devices, APIs and online services in new and interesting ways. Node-RED consists of a visual editor that allows the user to wire together devices and services using a web browser. In addition to the built-in nodes, Node-RED has a large library of 3rd party nodes available for easy plug and play extension. Built on Node.js it can easily be deployed in the cloud or at the edge.

This enables end users to program flow and create their own dashboards, even without programming skills. Everything is done as it flows. It enables rapid development of applications that interface with various devices, sensors, actuators, and services through the various built-in nodes. Node-RED offers an extensive library of contrib-nodes created by the community for many other services and devices. This makes it flexible and adaptable by any developer and rapidly deployable. Node-RED is also a community-driven initiative and intends to ease the use of the Internet of Things. Node-RED allows devices to implement the protocol of their choice and for the service level implementation to include the service of their choice. Flow-based programming consists of the definition of flows of execution between various services. The flows are created on an editor and the compiled version is executed at the local machine or on the cloud.

Node-RED application is a graphical web-based flow editor allowing you to define the application by assembling the nodes in a flow. Node-RED is an application in which the user can start a menu in the command line. After that, if the user is running on a desktop or video, they can type node-red. After that, they can use the editor added to the browser. Node-RED can sit behind a corporate firewall. A dashboard node added to Node-RED gives you easy node-RED-based dashboarding features without the cumbersome steps needed to install and configure the Node-RED Dashboard folder. The dashboarding is a fast way to build a Node-Red based monitoring and control dashboards in minutes.

6.7. Use Cases of Real-Time Monitoring Solutions

Development of low-cost intelligent sensor-based systems has enabled Internet of Things applications in many domains including but not limited to Industrial Automation, Smart Cities, and Healthcare, with the promising use of real-time data streaming and analysis features. Real-time sensor monitoring can help discover hidden patterns, manage abnormal operation, and reduce system downtime. The application domains have different real-time sensor monitoring features, with important variations in terms of sensor networks deployed, measurement methods, sensor data transmission techniques, analysis and processing requirements. This section discusses some prominent real-time monitoring applications in the aforementioned domains, as a precursor to the proposed real-time monitoring techniques.

Healthcare Applications

Real-time health monitoring applications use wearable sensors, such as motion detectors, temperature sensors, pulse oximeters, ECG sensors, heart beat sensors, and EEG sensors, on a person to monitor body vitals for emergency assistance. However, developing these uses effectively is challenging because of power and resource constraints inherent in wearable devices, and concurrency. Furthermore, people using wearables work in public places such as offices, restaurants, cafes, schools, hospitals, so it is difficult to exploit temporal locality or known activity patterns of each person. 3C smart sensors have their original computing capabilities, have moderate and relatively stable battery lifespan, and are less restricted by the aforementioned challenges. Therefore 3C sensors can be deployed to enable subsequent actions for assisting monitored users including motion reminder, fall/become unconscious alert, tiredness detection, person detection, distance maintaining alert, and sleep monitoring.

Industrial Automation

The 4th industrial revolution aims to create next generation intelligent factories, by integrating advanced technologies like Cyber Physical System and the Internet of Things. CPS connects low-cost real-time sensors and actuators to the internet in order to bring intelligent devices into industrial deployments. Real-time monitoring applications in factories are used for ensuring safety, improving efficiency, and detecting abnormal work patterns. 3C device-based industrial applications can solve some of the real-time monitoring problems in people-centric factories without requiring additional sensors.

6.7.1. Healthcare Applications

The development of real-time monitoring solutions using embedded sensors and edge computing services has been focusing on biomedical applications such as vital sign monitoring, food quality monitoring, and wheelchair user emotion analysis. The biomedical area is one of the critical subjects linking medical research and industrial applications; the attractive topics are related to how we can design advanced wearable/positionable inertial sensor-based solutions that can be used to measure relevant biomedical signals. To cite a few examples, global navigation satellite systems have been mostly used for outdoor purposes; however, the combination of highly accurate smartphones and new processing methods has made this approach so popular for indoor applications, such as gait analysis. The estimation of functional signs such as the number of steps, walking speed, or triggered events, such as seat vomiting or stumbling, are real-time applications that fall into the context of telemedicine provided that the results are sent over the Internet to a medical doctor for subsequent analysis.

Medical expert system development as eHealth is becoming more and more popular. Mobile health is regarded as a separate part of the eHealth systems offering new analytical approaches for those patients who prefer home treatment instead of hospitalization. The development of new smart wearable devices based on inertial sensors can provide a large source of information about the dynamic behavior of the user and, consequently, their health state, which may help to prevent serious health problems. However, the need for a high sensitivity of the inertial sensors as well as an energy-efficient long-term recorder design becomes an important problem for the designers of such solutions.

6.7.2. Industrial Automation

Real-time monitoring is an essential process in industrial automation since it is a key component that improves processes and avoids machinery failures. For many industries, everything is automated, and real-time monitoring becomes even more critical for protecting the automated systems. Today, continuous monitoring of several industrial parameters is usually done with a wired SCADA that is expensive, complicated, offers low performance, and cannot be relocated between machines. As an improvement, the introduction of wireless sensor networks into SCADA systems enables more complex monitoring systems for industry 4.0. Nevertheless, WSN consumes an elevated amount of power and cannot be used for continuous operation. Another alternative for these systems is smart cameras. However, these solutions are very costly and complex, and they also suffer from performance issues, as the cameras cannot be placed close to every element that needs monitoring. Therefore, there are several applications that need real-time monitoring, such as control of heavy mechanical systems, machine tool monitoring, or the production of specific components and the quality of thermal processes. These services become mandatory and need to be continuously monitored by a dedicated system.

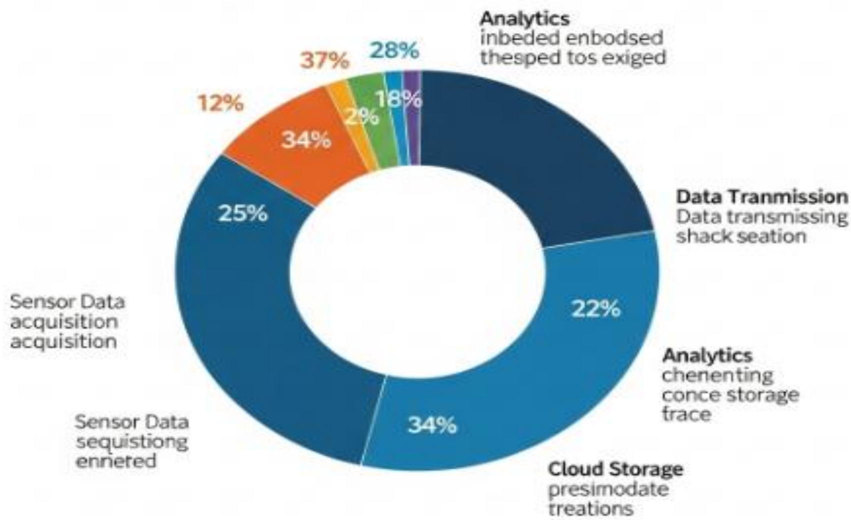

Fig : Development of Real-Time Monitoring Solutions Using Embedded Sensors and Edge Computing.

As presented above, several industrial parameters require continuous monitoring to ensure proper processes and machine health. In this case, we present a complete integrated IoT solution for implementing a low-cost and low-power monitoring system utilizing remote sensors placed close to each machine that need monitoring. Additionally, we offer a more precise approach able not only to register the signal of interest in the cloud but also capable of real-time signal processing at the edge. For this, we are introducing a DSP as a real-time processor that should be placed physically close to the machine, merging the traditional WSN structure and an edge processing layer.

6.8 Smart Cities

Real-time sensor monitoring can also realize the global Smart Cities Initiative whose objectives are to increase the connectivity of city infrastructures, increase the productivity of urban functions as resources, services, and people and facilitate their integration, enable the monitor and predictive control of urban subsystems and processes to improve their performance, support better management of urban-related phenomena such as sewage vapors, climate assessment or air quality, reduce the carbon footprint of cities, promote energy efficiency in consumption and generation, enable alternative energy systems based on renewable sources or alternative technologies, promote the increase of urban greenery and urban biodiversity, enable real-time data access that give citizens the ability to see and evaluate how city leaders manage and design urban systems, infrastructure and operations on their behalf, support the development of self-governing, trusted, user-centric, secure, smart city service, support the development of sustainable cities that put as their main driver of resource use and management the economic, environmental, financial and social awareness of their citizens, offer citizens a diverse data access to empower them to service the needs and demands of those citizens, including through the utilization of crowdsourcing or mobile technology, coordinate the flows of automobiles, buses, trucks, bicycles, and pedestrians in real time, and smooth traffic in a way that minimizes congestion reduces the time spent by citizens in traffic.

Some of the aforementioned objectives are some of the SDGs proposed by the United Nations that aim at pursuing quality development in promoting the well-being of present and future generations, ensuring every individual a dignified life, in harmony with the ecosystem of which he is a part. Real-time monitoring solutions are useful to analyze natural and anthropogenic phenomena, detect their impact on city systems both in terms of efficiency and safety, and enable their predictive management. If well integrated, these solutions can also bring information to citizens as well as allow an easier exploitation of city functions, such as waste collection and disposal, and enable service and action for the safeguarding urban systems.

6.9. Conclusion

The fast increase in quality and type of distributed Captors has opened new perspectives in solving problems of environmental and social importance. If we used to dispose of few environmental Captors, capable of deeply characterizing local phenomena in a very limited number of points, and furthermore with long time gaps between contiguous measures, in the next years we could take advantage of a huge number of low cost Captors, placed into every corner of the world, and providing real time data on the local environment. This opens new perspectives both on environment control and on phenomenon modelization. Controlling and keeping track of what happens in the environment around us is becoming increasingly important, because of the risks it could bring to society as well as its influence on social habits. Detecting fire or smoke, monitoring the crowd in specific places, detecting a possible threat, monitoring weather conditions and their effects, all of these capabilities are nowadays essential to monitor the environment for specific needs. These needs should be met by pervasive systems able to provide data and services continuously and in almost real-time. This can be possible by the use of a distributed sensing strategy based on the concept of smart dust.

Smart dust comprises so-called motes, that is, small, low value, low weight, low-power sensor nodes capable of monitoring the environment around them. By the term motes we may indicate a wide variety of integrated sensor systems characterized by different features, costs, functionalities and obtainable performances. They can achieve the monitoring of several physical variables, providing a measure of their local value, equipped with local pre-processing features, and able to send the acquired data to a management or control entity. Then, several research activities are needed to tune the motes and make them capable of adaptive smart sensing in order to obtain the information surrounding us, in a very limited time and of a very high quality in contrast to what happens when we use actual automated single Captors.

6.8.1. Future Trends

A promising area of future research is the extension of embedded sensors and edge computing operations into mobile devices like drones, surface vehicles, submarines, and handheld devices. This opens up the ability to flexibly deploy a vast range of sensors in space and time. Mobile devices often come with very limited energy supply and low bearer bandwidth, which necessitate further development of low-power and energy-efficient solutions based on more efficient machine learning and on-device computing technology. Currently, deployments still require humans to be present to deploy water sensors or drones both large and small. Tomorrow a drone or a small robot on the ground will immediately deploy a sensor that could last for months, collecting acoustics and other sensor data and doing near real-time indexing and pattern detection and characterization. Tomorrow, we might use mobile sensors and mobile computing to monitor our environment, not only to use the information to notify and alert, but also to control and react. Movement from one location to the other is very typical in mobile devices, which gives rise to a new issue of how to effectively leverage the underlying mobility in performing SS. Although there have been a number of mobility aware models proposed that leverage the mobility of wireless devices, they simply adapt existing SS models to apply to such models.

The three new techniques address the key issues involved in mobile SS. The current growing interest in heterogeneous embedded sensor networks is driven by existing low submit bandwidth on traditional networks for data going back to online data repositories. Big Data has come of age, and people are starting to think of smart cities and the control and reaction aspect of big data and our lives, as well as the monitoring and making decisions and predictions aspect, such as decisions and users about where we travel, which cities are travel hotspots, and so on.

References

Jeong, Y., Oh, H. W., Kim, S., & Lee, S. E. (2022). An edge AI device-based intelligent transportation system. Journal of Information and Communication Convergence Engineering, 20(2), 166–173.MDPI

Gusev, M., & Dustdar, S. (2018). Going back to the roots—the evolution of edge computing, an IoT perspective. IEEE Internet Computing, 22(5), 5–15.MDPI

Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., Yang, X. (2018). A survey on edge computing for the Internet of Things. IEEE Access, 6, 6900–6919.MDPI

Alrowaily, M., & Lu, Z. (2018). Secure edge computing in IoT systems: Review and case studies. In Proceedings of the 2018 IEEE/ACM Symposium on Edge Computing (SEC) (pp. 440–

Liu, G., Shi, H., Kiani, A., Khreishah, A., Lee, J. Y., Ansari, N., Liu, C., & Yousef, M. (2021). Smart traffic monitoring system using computer vision and edge computing. arXiv Preprints. arXi