# A TinyML-based system for gas leakage detection

Anargyros Gkogkidis*, Vasileios Tsoukas†, Stefanos Papafotikas‡, Eleni Boumpa§ and Athanasios Kakarountas¶
Intelligent Systems Laboratory,
Department of Computer Science and Biomedical Informatics,
University of Thessaly, Lamia, Greece
Email: *agkogkidis@uth.gr, †vtsoukas@uth.gr, ‡spapafotikas@uth.gr, §eboumpa@uth.gr, ¶kakarountas@uth.gr

***Abstract*—Internet of Things devices are commonly utilized in smart homes to provide smart services such as lighting, entertainment, and easy access, but they are also employed to warn occupants in the event of an emergency. Due to the computationally intensive nature of existing Neural Network implementations, data must be transmitted to the cloud for analysis to produce intelligent machines. TinyML is a promising approach that the scientific community has proposed as a method of constructing autonomous and secure devices that can collect, analyze, and output data without requiring it to be transferred to remote entities. This work presents a TinyML-based system for detecting hazardous gas leaks. The system may be trained to detect irregularities and notify occupants using BLE technology via a message sent to their smartphones, as well as through an integrated screen. Two different test cases are presented and evaluated via experiments. For the smoke detection test case the system achieved an F1-Score of 0.77, whereas for the ammonia test case the evaluation metric of F1-Score is 0.70.**

***Index Terms*—TinyML, machine learning, gas detection, constrained hardware, microcontrollers, sensors**

## I. Introduction

Increased scientific interest is observed in monitoring Indoor Air Quality (IAQ) as an important procedure for people with health related issues such as respiratory diseases. Over the years, many IAQ monitoring systems have been implemented for various environments such as homes, schools, hospitals, and so on. The previously proposed IAQ systems can measure and monitor various indoor pollutants including nitric oxides ($NO_x$), $O_3$, carbon oxides ($CO_x$), sulfur dioxide ($SO_2$), and Volatile Organic Compounds (VOCs), that cause various health effects [1], [2].

The majority of implemented IAQ monitoring systems are utilizing Internet of Things (IoT) devices and are based on Wireless Sensor Networks (WSNs). These systems collect in real-time different types of sensory data and transmit them on the cloud, where are stored, analyzed, and processed. However, a major challenge that arises from the broad use of IoT monitoring systems is the security and the privacy of the people's homes collected data. Hence, the research community focused on this challenge and developed alternative and emerging approaches to decrease the risk of sensitive data leakage [3].

One alternative approach is enabling edge intelligence. The use of intelligence on edge devices can result in the local processing of data and therefore privacy-aware processing. Furthermore, despite the security advantages, data results can be extracted in real-time, and the developed devices could be small, low-cost, low-power and autonomous [4].

In this work, an IAQ based on Tiny Machine Learning (TinyML) for real-time gases leakage and smoke detection system is presented. The system collects sensors' data and processes it on the edge with the use of the TinyML technology. To the best of our knowledge this is the first implemented IAQ TinyML-based system.

The rest of this paper is organized as follows. Section II presents previous work regarding IAQ monitoring. Section III presents the technology of TinyML. Section IV introduces the TinyML-based system including both hardware and software details. Finally Section V concludes on the work's findings and discusses future plans.

## II. Previous Work

Gas detection and IAQ monitoring systems were proposed over time, utilizing different types of sensors, architectures, and methods of data processing. Spachos et al. [5] proposed a real-time $CO_2$ monitoring system consisting of sensor nodes, simple relay nodes, and a control room system. Each sensor node transmits the data packets through a path of available relay nodes to the control room, where they are stored. The data is stored, processed, and monitored using the MonArch, a scalable monitoring system that is capable of storing and processing monitoring values. The advantages of the proposed system are real-time data aggregation, robustness to interferences, drop-and-play units, and offline monitoring in a complex indoor environment.

The iAirCO2 [6] is an IAQ system that monitors the $CO_2$ by using an MHZ19CO2 sensor and stores the ambient data in a SQL server database. The data transmitted via Wi-Fi, with the use of an ESP8266 MCU and the .Net web services, can be accessed only from authorized users. Also, the messages are encrypted and signed, while the web services use SSL certification for the authentication. The system's user can access the data over a web browser or a smartphone application and set a threshold for real-time notifications, such as e-mail, SMS, or smartphone notifications. The proposed system focus on providing detection of poor air conditions in an environment, like a home, while the ambient data can help the clinical professionals to analyze the history of IAQ parameters of a patient's environment.

In study [7] an artificial intelligent-based multiple hazard gas detector system, mounted on a motor vehicle-based robot that can be remotely controlled, was proposed. The system uses an array of sensors for the classification of three hazardous gases in residential buildings; cigarette smoke, inflammable ethanol, and off-flavor from spoiled food. Also, it uses three different ML algorithms; k-Nearest Neighbors (kNN), Support Vector Machine (SVM), and Softmax regression, in which the input vector is the feature set generated from the sensor array. The system's training is executed automatically with the use of MatLab.

Taheri et al. [8] implemented a dynamic indoor model to predict $CO_2$ concentrations with the use of various Machine Learning (ML) algorithms, like SVM, AdaBoost, Random Forest, Gradient Boosting, Logistic Regression, and Multilayer Perceptron. The model's accurate results are used to modulate in real-time the ventilation rate of a campus classroom, where the data was collected, while the energy consumption of the heating, ventilation, and air conditioning fan can reduce by up to 51,4% from the standard's levels.


Fig. 1. **The fully assembled system**

## III. TinyML

The majority of IoT devices are not intelligent, and those using ML models collect information and transfer it to the cloud for analysis. The rationale for transmitting data to a remote entity is dependent on the type of process through which data must pass. The enormous, complex nature of algorithms and ML models necessitates more processing power and resources than a tiny IoT device can deliver. This leads also to a massive volume of data that the device is incapable of storing. IoT devices are programmed to communicate with other intelligent devices via wireless technologies. The data being transmitted is frequently not safeguarded, and the devices are being considered to lack adequate security features.

TinyML is a relatively new emerging technology that is garnering increased interest from researchers [19]. The technology combines careful hardware and software design and enables the deployment of ML models and Deep Learning (DL) algorithms on small, reasonably priced, and power-efficient devices. With the utilization of this new field, new services, and technologies that do not require high-end systems and address IoT device challenges such as latency and bandwidth constraints could be developed [20]. The IoT devices will be used to collect, examine, and extract information locally. This information is not shared with other entities, resulting in more secure and private devices. Furthermore, the hardware required to perform the operations, namely the microcontroller (MCU), is considered to be ultra-low-powered and extremely efficient. It typically consumes less than one milliwatt of energy and can provide intelligence in a small time-frame. TinyML may be the field that fundamentally changes the way developers approach innovative and security applications for home use today. Alerting a user for a possible gas leak or increased risk is crucial, and there is no room for latency issues or interruptions between communications. These devices will perform real-time analysis and alert home residents without the need for data transfer, giving rise to a new era of autonomous devices in the size of a coin or credit card and the only requirement they have is the provision of power from a battery.

## IV. TinyML-based System

The proposed system is comprised of a development board, two gas detection sensors, an LCD monitor that alerts the user through text messages and a buzzer to be used on the high volumes of hazardous gas detection. The primary goal was to create a tiny, self-contained, low-cost, and efficient gadget capable of detecting gas leaks and notifying the user in real time. The device can be installed in a household setting to alert occupants to the presence of gas or smoke. Another application may be to alert the owner of a car leaking Liquefied Petroleum Gas (LPG), when the system is positioned in a garage. The system is TinyML-based, resulting in an autonomous system that does not require an internet connection, communication with other devices, or access to the cloud for data processing and alerting. As a consequence, a system capable of continuous monitoring and real-time notifications is created that is not constrained by bandwidth or latency restrictions. Additionally, the device is not a conventional IoT device that operates on basic logical actions based on predefined criteria, such as thresholds. The device may be educated in the home to provide customized results and alarms. For instance, the system will not notify residents if a trace of smoke is detected in a residence occupied by smokers. The gadget will be trained and learn to differentiate between cases such as when someone smokes and when there is a genuine threat. All of this may be achieved simply by training the model on data collected throughout a typical day. If the system detects outliers in this data, it will recognize an emergency and will alert the residents.

### A. Hardware

The development board of our choice was the Nano 33 BLE Sense from Arduino, a well-known and extensively used hardware for developing TinyML applications. The board is based on the Nordic Semiconductor nRF52840, which features a 64-MHz 32-bit ARM®CortexTM-M4 CPU, 256KB SRAM, and 1MB of flash memory. It works at 3.3V and has a size factor of 45x18mm, resulting in one of the tiniest boards available in the market. Additional information is available on the board's official datasheet [9].

Furthermore, the board is compatible with the Edge Impulse web framework utilized to build the ML models, which will be discussed in more detail in the following section. Three different sensors were initially tested, the MQ-2, MQ-5, and MQ-135.

MQ-2 is a Metal Oxide Semiconductor (MOS) type gas sensor, often referred to as chemiresistors, since it detects changes in the resistance of the sensing material when the gas comes into contact with it. Gas concentrations may be sensed using a simple voltage divider network. MQ-2 gas sensor operates at 5V DC and consumes around 800mW. It is capable of detecting values of LPG, smoke, alcohol, propane ($C_3H_8$), hydrogen ($H_2$), methane ($CH_4$), and CO ranging from 200 to 10000 ppm [10].

The gas sensor MQ-5 is essential for detecting gas leaks in home and industry. It can detect $H_2$, LPG, $CH_4$, CO, and alcohol. Due to the instrument's great sensitivity and rapid reaction time, measurements may be obtained immediately. Whenever the gas concentration increases, the gas sensor's output voltage also increases [11].

The MQ-135 gas sensor is capable of detecting dangerous gases and smoke such as ammonia ($NH_3$), sulfur (S), benzene ($C_6H_6$), and $CO_2$. As with other MQ series gas sensors, this one also features both a digital and analog output pin. When the concentration of these gases in the air exceeds a preset limit, the digital pin swings high. The onboard potentiometer can be used to adjust this threshold value. The analog output pin generates an analog signal that may be used to estimate the atmospheric concentration of certain gases. The sensor module works at 5V and draws around 150 mA [12].

The LCD 1602 [13], with a display format of 16 Characters x 2 Lines, was utilized regarding the messages for alerting the user. Additionally, the communication chip on the Arduino supports wireless protocols for information transmission. When an abnormality is discovered, the onboard BLE module is utilized to alert the user through a mobile phone. Finally, a case to store the system was 3D-printed [18]. Figure 1 depicts the fully assembled system.

### B. Datasets

Several datasets were created in a period of two weeks. All monitoring was carried out using the Arduino Nano 33 BLE Sense and the sensors listed above. The datasets were created by taking advantage of Edge Impulse data forwarder, a tool which transfers the data in real time to the web platform. The Laboratory's cloud infrastructure was used to store and evaluate the results. Experiments revealed the redundancy of utilizing all three sensors due to the high resemblance of the MQ-2 and MQ-5 sensors. We decided to implement the MQ-2 and the MQ-135 sensors to reduce the ML model complexity and increase the system's power efficiency.

### C. Model Training and Inference

We chose to use Edge Impulse for the model training and inference on the device. Edge Impulse is a framework for integrating ML models into MCUs. One of the framework's most notable features is its ability to handle a diverse range of devices, including the Arduino Nano 33 BLE Sense, the Portenta H7 [14]. And the Sony Spresense [15]. It lets users collect data directly from their devices, classify it, and transmit it to the cloud in the form of a dataset. Figure 2 depicts the steps followed for the model training.

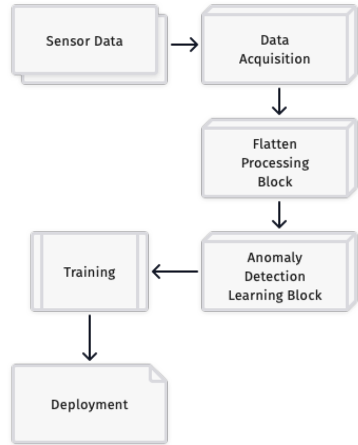


Fig. 2. **Model Training Procedure**

Edge Impulse contains pre-trained ML blocks that may be fully adjusted to match the project's requirements. Additionally, users may acquire live categorization data while evaluating the model's performance on devices, ensuring that the model performs as planned while it performs real-time monitoring. A new impulse was designed in the aforementioned framework, consisting of a time series data block for the data fed from the MQ-2 and MQ-135 sensors.

Next, the flatten processing block was used due to its high adaptability on sensor data such as temperature and other metrics similar to our use case. Figure 3 shows the Multidimensional data using the average distance from each cluster. As depicted, the orange data are real time classified by the TinyML inference on device. Left side classifies gas related to MQ-2 and right side gas related to MQ-135. For the learning block, we used the anomaly detection block, which utilized the K-Means algorithm. The model was evaluated using Edge Impulse's model testing tab. Additionally, the impulse converted into optimized source code ready to be deployed to Arduino Nano 33 Sense. Finally, more tests were conducted using the Arduino IDE and Laboratory's cloud infrastructure to find outliers in new data.

### D. Model Evaluation

To assess our models for smoke and ammonia, we looked at the number of genuine anomalies discovered by our systems, the proportion of anomalies detected and were indeed actual anomalies, and lastly the overall performance of the models. Precision, Recall, and F1-Score were utilized as evaluation cri-

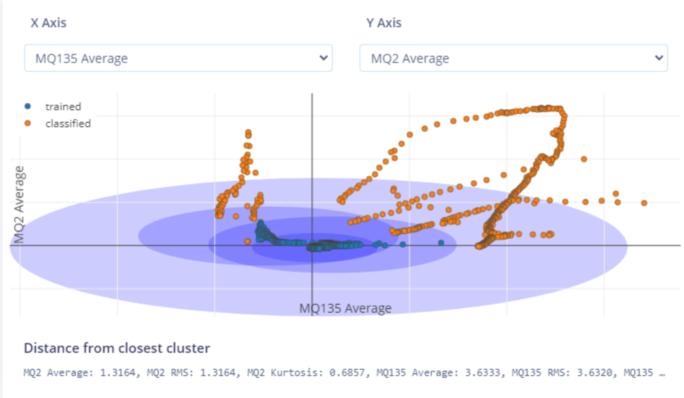


Fig. 3. Data visualization related to MQ-2 and MQ-135 sensors

teria to accomplish this. Precision was 0.73, Recall was 0.84, and the F1-Score was 0.77 for the smoke use case. The False Negative Rate was 12% and the False Positive Rate 36%. The ammonia test case experiments were relatively comparable, with the largest variation being the score of 0.60 for Recall. Precision and F1-Score metrics values were 0.85 and 0.70, respectively. In this test case the experiments revealed a False Negative Rate of 35% and a False Positive Rate of 20%. Table I shows the evaluation metrics revealed from the experiments regarding the two test cases, smoke and ammonia. The findings are encouraging, but there is potential for improvement, which may be accomplished through additional training and better hardware exploitation.

TABLE I
MODEL EVALUATION

| Test Cases | Evaluation Metrics | | |
|---|---|---|---|
| | ***Precision*** | ***Recall*** | ***F1-Score*** |
| Smoke | 0.73 | 0.84 | 0.77 |
| Ammonia | 0.60 | 0.85 | 0.70 |

## V. CONCLUSIONS AND FUTURE PLANNING

With recent technological advancements, households may become even safer by warning occupants in the event of an emergency. While IoT devices appear to be ideal for the aforementioned use case, they are forced to transfer essential data to the cloud for further processing to deliver intelligence and personalized solutions using ML and DL technologies.

TinyML is a revolutionary technology that appears to be the answer by delivering autonomous devices that do not require internet or data transfer owing to their ability to execute ML and DL models on-device successfully.

The present study demonstrates a TinyML-based system for detecting hazardous gas leakage. The system may be trained to detect and warn inhabitants of possible gas leakage or smoke detection.

Our future agenda includes three major improvements that will help into making the device more user friendly, safer and overall improve the system into offering more tailored and better results.

Our first plan is to offer the user the ability to offline train or retrain the system with the press of a button. Offline training is one of challenges encountered with the TinyML technology. Recent works related to offline training [16], [17] show encouraging results and this the sector where our efforts are going to be focused.

The following step is to ensure the security of the system and data transmission to the mobile device. In terms of the system, we will investigate methods for ensuring firmware integrity, and for data transmission, a well-established security protocol will be employed.

Finally, based on our experiments we believe that using more specialized sensors for each element to be detected instead of utilizing sensors that detect several gases will improve the alerts and the values shown to the users.

## ACKNOWLEDGMENT

We acknowledge support of this work by the project “Par-ICT_CENG: Enhancing ICT research infrastructure in Central Greece to enable processing of Big data from sensor stream, multimedia content, and complex mathematical modeling and simulations” (MIS 5047244) which is implemented under the Action “Reinforcement of the Research and Innovation Infrastructure”, funded by the Operational Programme ”Competitiveness, Entrepreneurship and Innovation” (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

## REFERENCES

[1] J. Saini, M. Dutta, and G. Marques, “A comprehensive review on indoor air quality monitoring systems for enhanced public health,” Sustainable Environment Research, vol. 30, no. 1, p. 6, Jan. 2020, doi: 10.1186/s42834-020-0047-y.

[2] O. US EPA, ‘Volatile Organic Compounds’ Impact on Indoor Air Quality’, Aug. 18, 2014. https://www.epa.gov/indoor-air-quality-iaq/volatile-organic-compounds-impact-indoor-air-quality (accessed Jan. 20, 2022).

[3] ‘Edge Intelligence for Connected In-home Healthcare: Challenges and Visions - IEEE Internet of Things’. https://iot.ieee.org/newsletter/march-2021/edge-intelligence-for-connected-in-home-healthcare-challenges-and-visions (accessed Apr. 27, 2022).

[4] R. David et al., “TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems,” arXiv:2010.08678 [cs], Mar. 2021, Accessed: Apr. 27, 2022. [Online]. Available: http://arxiv.org/abs/2010.08678

[5] P. Spachos, J. Lin, H. Bannazadeh, and A. Leon-Garcia, “Smart room monitoring through wireless sensor networks in software defined infrastructures,” in 2015 IEEE 4th International Conference on Cloud Networking (CloudNet), Oct. 2015, pp. 216–218. doi: 10.1109/CloudNet.2015.7335310.

[6] G. Marques, C. R. Ferreira, and R. Pitarma, “Indoor Air Quality Assessment Using a CO2 Monitoring System Based on Internet of Things,” J Med Syst, vol. 43, no. 3, p. 67, Feb. 2019, doi: 10.1007/s10916-019-1184-x.

[7] Y. Wu, T. Liu, S. H. Ling, J. Szymanski, W. Zhang, and S. W. Su, “Air Quality Monitoring for Vulnerable Groups in Residential Environments Using a Multiple Hazard Gas Detector,” Sensors, vol. 19, no. 2, Art. no. 2, Jan. 2019, doi: 10.3390/s19020362.

[8] S. Taheri and A. Razban, “Learning-based CO2 concentration prediction: Application to indoor air quality control using demand-controlled ventilation,” Building and Environment, vol. 205, p. 108164, Nov. 2021, doi: 10.1016/j.buildenv.2021.108164.

[9] Arduino, “Arduino Nano 33 Datasheet”, https://docs.arduino.cc/resources/datasheets/ABX00031- datasheet.pdf (accessed Feb. 22, 2022).

[10] Last Minute Engineers, “How MQ2 Gas/Smoke Sensor Works? & Interface it with Arduino”, https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial (accessed Feb. 22, 2022).

[11] Seeed, “Grove - Gas Sensor(MQ5)”, https://docs.arduino.cc/resources/datasheets/ABX00031- datasheet.pdf (accessed Feb. 22, 2022).

[12] Quartz Components, “MQ-135 Air Quality Gas Sensor Module”, https://quartzcomponents.com/products/mq-135-air-quality-gas-sensor-module (accessed Feb. 22, 2022).

[13] Open Hacks, “Specification for LCD Module 1602A-1”, https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf (accessed Feb. 22, 2022).

[14] Arduino, “Portenta H7”, https://www.arduino.cc/pro/hardware/product/portenta-h7 (accessed Feb. 22, 2022).

[15] Sony, “Spresense 6-core microcontroller board with ultra-low power consumption”, https://developer.sony.com/develop/spresense/ (accessed Feb. 22, 2022).

[16] L. Ravaglia, M. Rusci, D. Nadalini, A. Capotondi, F. Conti, and L. Benini, “A TinyML Platform for On-Device Continual Learning with Quantized Latent Replays,” arXiv:2110.10486 [cs], Oct. 2021, Accessed: Feb. 27, 2022. [Online]. Available: http://arxiv.org/abs/2110.10486

[17] M. M. Grau, R. P. Centelles, and F. Freitag, “On-Device Training of Machine Learning Models on Microcontrollers With a Look at Federated Learning,” in Proceedings of the Conference on Information Technology for Social Good, New York, NY, USA, Sep. 2021, pp. 198–203. doi: 10.1145/3462203.3475896.

[18] W. Beckmann, ”CO2-Sensor - Arduino/ESP/Display/Wifi”, https://www.thingiverse.com/thing:4650900/files, (accessed Feb. 22, 2022).

[19] V. Tsoukas, E. Boumpa, G. Giannakas, and A. Kakarountas, “A Review of Machine Learning and TinyML in Healthcare,” in 25th Pan-Hellenic Conference on Informatics, New York, NY, USA, Nov. 2021, pp. 69–73. doi: 10.1145/3503823.3503836.

[20] V. Tsoukas, A. Gkogkidis, A. Kampa, G. Spathoulas, and A. Kakarountas, “Enhancing Food Supply Chain Security through the Use of Blockchain and TinyML,” Information, vol. 13, no. 5, Art. no. 5, May 2022, doi: 10.3390/info13050213.