Predicting Low-Cost Gas Sensor Readings From Transients Using Long Short-Term Memory Neural Networks

Jelena Čulić Gambiroža, Toni Mastelić, Tonko Kovačević, and Mario Čagalj, *Associate Member, IEEE*

***Abstract*—With the everyday growth of the Internet of Things (IoT), the number of connected sensor devices increases as well, where each sensor consumes energy while being constantly online. During that time, they collect large amounts of data in short intervals leading to the collection of redundant and perhaps irrelevant data. Moreover, being commonly battery powered, sensor batteries need to be frequently replaced or recharged. The former requires smarter and less frequent data collection, while the latter being complementary to the former requires putting them to sleep while not being used in order to save energy. The focus of this article is low-cost gas sensors as they need to preheat for several minutes to reliably collect gas concentration. However, instead of waiting for a sensor to heat up, a transient, i.e., a data trend that the sensor collects while heating up is analyzed. It is shown that long short-term memory (LSTM) neural network can be used to learn and later predict the actual gas level from a part of the transient. This way, instead of being constantly online or fully preheating, the sensor needs to be turned on for only 20 s and then sleep for 120 s. With high accuracy, our approach decreases energy consumption by up to 85% compared to a system where sensors are constantly online, and more than 50% compared to a system where a sensor collects actual values instead of a part of the transient.**

***Index Terms*—Energy efficiency, gas sensor, Internet of Things (IoT), long short-term memory (LSTM).**

I. Introduction

YEAR after year, the Internet of Things (IoT) is growing, along with the number of sensors collecting the data and the amount of data itself. Ericsson forecasts that the number of devices connected by Massive IoT and other emerging cellular technologies will reach 4.1 billion by 2024 [1]. According to Cisco predictions, there will be 847 ZB of collected data, out of which 1.3 ZB will be stored in data centers by 2021 [2]. However, a large portion of that data is redundant and perhaps irrelevant, which costs money and creates unnecessary overheads and thus should be eliminated early in the process [3]. In order to eliminate such data in time, preprocessing on the edge of an IoT network is required, namely, on sensors and gateways. Eliminating "garbage" data before they are moved to cloud helps keeping data centers less loaded and future data analysis more efficient [4].

Processing data on the IoT edge can be achieved in several ways [5]. Data collected by sensors can be compressed, aggregated, and/or correlated on IoT gateways and only then transferred to the cloud. To move even closer to the edge, data can be filtered on sensors themselves in order to forward only relevant information to an IoT gateway [6]. However, although the amount of data is decreased, sensors are still constantly turned on and consume energy. To reduce sensors' online time, they can be put to sleep for a longer time period and turned on for a very short period just to collect data. The challenge with this approach is sensors that require preheating such as gas sensors, which cannot collect data at the exact moment they are turned on.

In the context of gas sensors, it is important to detect hazardous gases on time [7] using sensors with low power [8] and low cost [9], due to the rising scale of IoT. However, when a circuit is turned on and off, voltages and currents take time to stabilize to obtain readings of the actual gas concentration. A momentary variation in the current or the voltage during this preheating transition is called a transient, only after which an actual value can be read. However, transients in low-cost sensors can take minutes, which consumes a significant amount of energy from battery-powered sensors. Therefore, instead of waiting for the sensors to fully preheat, only parts of the transient can be collected. Jia *et al.* [10] showed in their research, energy can be saved if gas values are predicted from a part of the transient.

In this article, instead of a high-precision sensor as used in [10], low-cost MQ-2 [11], MQ-5 [12], and MQ-6 [13] gas sensors are used. The goal is to investigate if an actual gas concentration can be predicted from the transient of a low-cost sensor and thus achieve low cost but reliable IoT gas sensor network. On the one hand, the online period of the sensors must be long enough to provide a sufficient amount of data to distinguish one gas level from another. On the other hand, it should be short enough for the prediction process to be energy efficient. The resulting approach gives a minimized online period without significantly reduced accuracy.

Long short-term memory (LSTM) neural network (NN) is used to predict the actual value from the transient. An LSTM

Manuscript received November 14, 2019; revised February 19, 2020 and April 17, 2020; accepted April 22, 2020. Date of publication April 27, 2020; date of current version September 15, 2020. *(Corresponding author: Jelena Čulić Gambiroža.)*

Jelena Čulić Gambiroža and Toni Mastelić are with ETK Research, Ericsson Nikola Tesla, 21000 Split, Croatia (e-mail: jelena.culic.gambiroza@ericsson.com; toni.mastelic@ericsson.com).

Tonko Kovačević is with the University Department of Professional Studies, University of Split, 21000 Split, Croatia (e-mail: tonko.kovacevic@oss.unist.hr).

Mario Čagalj is with the Department of Electronics and Computing, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, 21000 Split, Croatia (e-mail: mario.cagalj@fesb.hr).

Digital Object Identifier 10.1109/JIOT.2020.2990526

network is suitable for predicting time-series data, as it can distinguish and eliminate irrelevant and redundant data from the relevant one. In order to prepare the LSTM algorithm that predicts actual values from the transients, data are collected with an MQ-2 gas sensor. The collected data are normalized in a range between 0 and 1. A threefold validation method is used to split the data and train LSTM NN. Finally, algorithm parameters and the entire approach prepared for the MQ-2 sensor are later evaluated on two other MQ sensors, namely, MQ-5 and MQ-6.

Results show that the MQ-2 gas sensor can predict values with a root mean-square error (RMSE) 0.05938. When using parameters that provide the best results for the MQ-2 gas sensor, RMSE of the MQ-5 sensor is as low as 0.07002, but it is slightly higher for MQ-6 with a value of 0.1293. With high accuracy, this solution can save up to 85% energy compared to a system where sensors are constantly online, and more than 50% compared to a system where sensors heat up to collect actual gas concentrations instead of a part of the transient.

The remainder of this article is organized as follows. The introduction is followed by the related work where LSTM NN applications are shown. Section III describes the preliminary analysis and methodology used in this article, while Section IV explains our implementation and results. Section V gives the evaluation of our approach on MQ-5 and MQ-6 sensors with discussion. Section VI concludes this article.

II. Related Work

Jia *et al.* [10] proposed a new, low-power, automatic, accurate, and wireless ammonia monitoring approach that uses metal–oxide sensors. This approach does not wait for equilibrium as this consumes a significant amount of energy, rather it tries to predict the resistance at equilibrium using the sensor's transient measurements in the short heating window (as short as 200 ms) to predict the actual value. A prediction model is built on LSTM NNs. The proposed model accurately predicts the equilibrium state resistance value with an average error rate of 0.12%. The final average estimation error for the ammonia concentration level is 9.38 ppm.

Salhi *et al.* [14] proposed implementation of a preventive system for gas leakage and fire incidences in a smart home environment to enhance safety using low-cost and low-energy consumption devices through M2M standard communication protocols. They applied supervised machine learning on several algorithms and predicted the level of risks for gas leakage and fire and alerted responsible person.

Chen *et al.* [15] used the LSTM network as a method to predict the mechanical state. The simulation results of the LSTM network are compared with the results obtained with a support vector regression machine (SVRM). A computation study is carried out to verify the algorithms. It is found that using the same window width, the MSE network test results of LSTM are smaller than the results of SVRM, making the LSTM model better than the SVRM model in the field of mechanical state monitoring and prediction.

Wang *et al.* [16] predicted water quality that is significant not only for the management of water resources but also for the prevention of water pollution. Since it is a time-series prediction problem, LSTM NN is used. Data set of water quality indicators in Taihu Lake measured monthly from 2000 to 2006 years is used for training the model. LSTM is compared with two methods, namely, backpropagation NN (BP NN) and online sequential extreme learning machine (OS-ELM). Several simulations and parameter selections are carried out in order to improve model accuracy. Results show that compared with BP NN and OS-ELM, the predictive accuracy of LSTM NN is higher and more generalized.

Yu *et al.* [17] used LSTM NN for spectrum prediction. The LSTM network is compared to the backpropagation (BP) network results. They also study the influence of different LSTM NN depth and width on prediction accuracy. The results show that LSTM has better performance than BP in case of the same number of hidden layers and neurons.

Liu *et al.* [18] used LSTM NNs to analyze and predict stock transaction data. The results show the accuracy of about 72% for the short period of data. Furthermore, Yao *et al.* [19] proposed an LSTM network combined with the fuzzy-rough set (FRS) theory for short-term wind speed prediction. The usage of FRS reduces input. The experimental results show that the FRS-LSTM model has about 40% higher prediction accuracy than the traditional BP NN.

Kim *et al.* [20] proposed a short-term electricity consumption prediction method. The LSTM network is used to predict month-ahead electricity consumption. The results on the real data show that the proposed method performs well with accuracy above 80%. They also state that the test accuracy can be improved with a longer period of training time and a deliberate hyperparameter setting.

Qian and Chen [21] conducted a stationary analysis of the stock's time-series data and then used the LSTM network to predict stock data under different stationary conditions. The results are compared with the ARIMA algorithm results. As shown on a large number of experimental results, the error rate of the LSTM algorithm is 66.78% lower than that of ARIMA. They also point out that the main disadvantage of the LSTM algorithm is that it takes a lot of time to train the model and requires a large sample of data.

Finally, in this article, the energy consumption of the gas sensor is considered, as well as the consumption for data transmission. While the focus is given on the gas sensor due to its higher consumption, there is also a large body of work focusing on energy consumption optimization in wireless sensor networks [22], [23] as a forerunner of IoT. There is also a number of researches focusing on energy consumption optimization in IoT systems addressing other power-hungry components such as radio [24], [25]. These approaches can be further used for decreasing the overall power consumption of such IoT systems.

III. Gas Sensor Characterization

In order to collect the actual value representing the gas concentration, a gas sensor needs to heat up. During this preheat time, a voltage needs to stabilize, where a set of unstable values during this period are referred to as a transient. Once sensor readings are stabilized, the sensor can periodically read

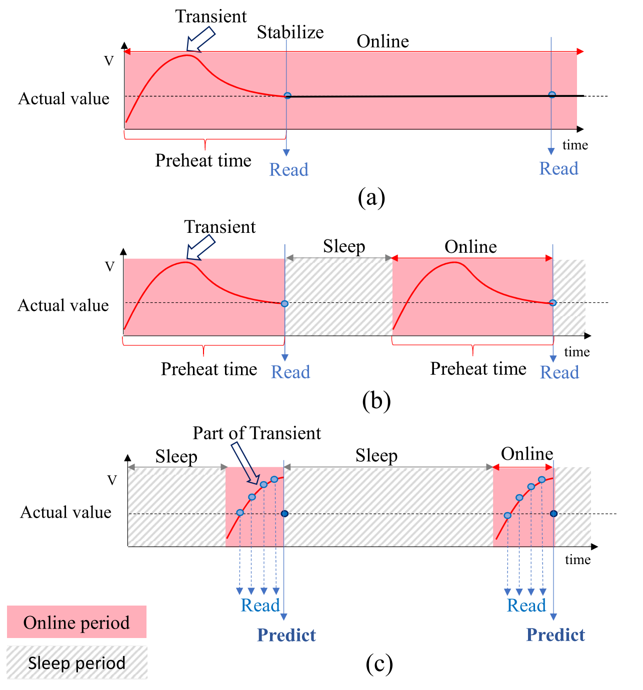


Fig. 1. Three scenarios for sensors setup, namely, (a) always online, (b) with full preheat and sleep period, and (c) with prediction based on a partial preheat and sleep period.

data while continuously being online as shown in Fig. 1(a). In case the sensor is put to sleep to save energy, it has to preheat after every sleep and then read the last value of the transient, i.e., the actual value of gas concentration. This scenario is shown in Fig. 1(b). Our approach is depicted in Fig. 1(c), where a sensor reads multiple values from the first part of the transient, while the actual value is estimated from that part using machine learning. This way, the sensor will have a longer sleep period, while more values are read in its short online period.

A. Environment Setup

We utilize low-cost MQ gas sensors, more specifically MQ-2 for defining our methodology. An MQ-2 sensor detects multiple hazardous gases, such as liquefied petroleum gas (LPG), methane (CH4), Alcohol, Smoke, and Propane, where we use LPG in this article. Since the goal is to predict the actual value from the transient and turn the sensor off, two sensors are used, namely, a test sensor for predicting the actual values and a control sensor for the ground truth.

The two sensors are connected to an Arduino Uno board that collects actual gas concentration values and then sends them via serial communications to the computer as shown in Fig. 2. The control sensor MQ-2(1) is always online without a sleep period. The test sensor MQ-2(2) has a predefined sleep period after which it wakes up upon receiving a signal from the real-time clock (RTC) DS3231 [26]. In order to turn off the test sensor completely during the sleep period, a power MOSFET IRFZ44N [27] is used. Finally, to reduce external influences the entire setup, depicted in Fig. 2, is placed in a sealed plastic container.

The data are collected with both sensors to find their correlation as they do not have the same nominal readings in

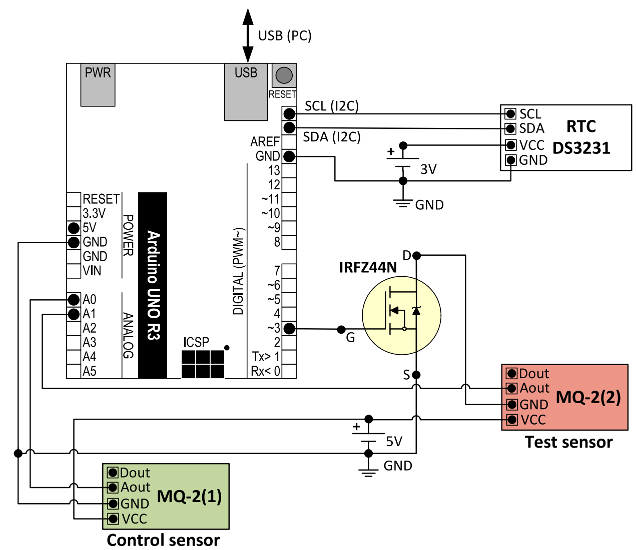


Fig. 2. Setup scheme with the test and control sensors, and RTC for turning on and off the test sensor.

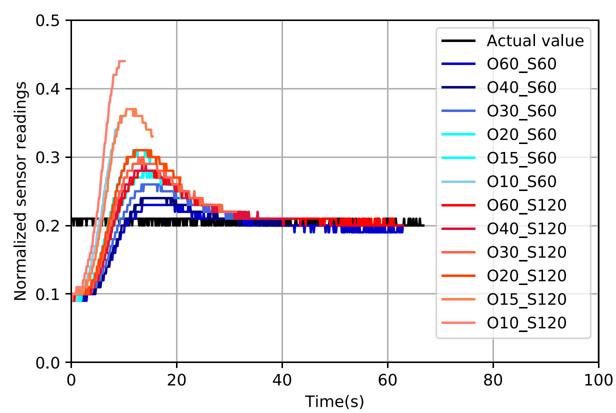


Fig. 3. Transient for different online (O)–sleep (S) ratios (without gas).

the same environment. Therefore, linear regression is used to correlate their readings. The control sensor is left to collect the data continuously, while the test sensor collects only transients. The expected values of the test sensor are calculated from the known correlation with the control sensor. Finally, since MQ-2 sensors collect the data in a range of 0–5 V, the data are normalized on a scale between 0 and 1. The highest value obtained for LPG gas during the experiments is 0.8 (4 V) due to the specifics of MQ-2 sensors [11].

B. Online Period Characteristics

Prior to collecting a data set used for building the LSTM prediction model, a preliminary analysis is performed in order to characterize the behavior of the transient. During online periods, the sensor is warming up, while during sleep periods, it is cooling down. That said, transients for different online–sleep ratios in the atmosphere without gas are compared in Fig. 3, where online periods of 10, 15, 20, 30, 40, and 60 s are considered, in combination with 60 and 120 s of sleep period, respectively.

As depicted in Fig. 3, MQ-2 sensor transients in the environment without gas reach their maximum in the first 15 s and

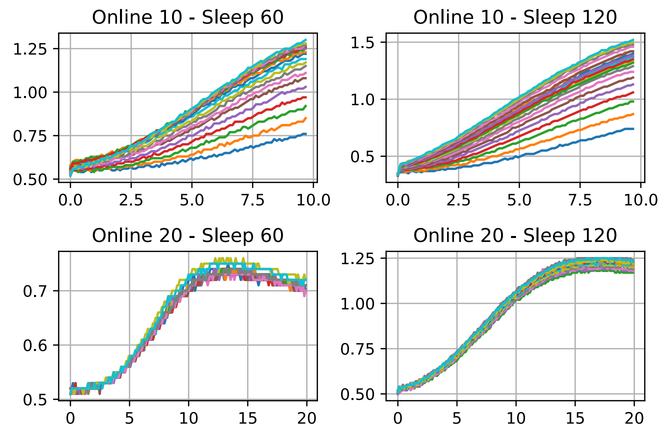


Fig. 4. Correlation of transients in case of different online/sleep ratios.

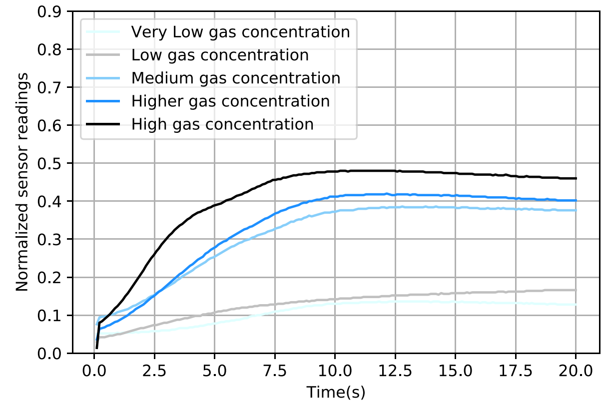


Fig. 5. Transients in different gas levels.

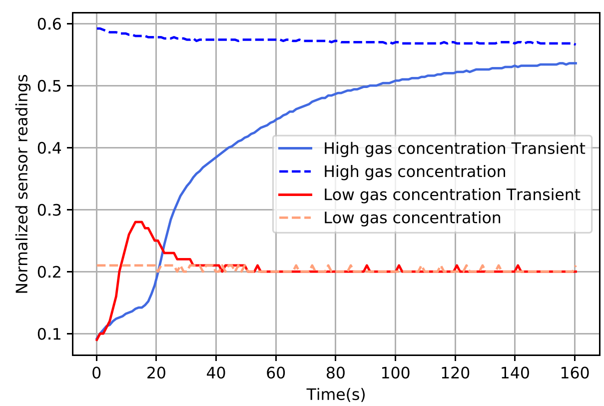


Fig. 6. Low versus high gas-level transients.

the actual gas concentration (ground truth) in approximately 40 s. Regardless of the sleep periods, the readings stabilize at the actual gas level within those 40 s, where both sleep and online periods affect only the maximum value, which is then regulated with its climbing and falling slopes to again reach their actual values after 40 s. Consequently, we select an online period of less than 40 s in order to achieve savings with LSTM predictions. However, further experiments show that periods shorter than 20 s have high variance. As seen from Fig. 4, in the case of 10-s period, consecutive readings result with inconsistent transients during longer runs, while the actual gas concentration remains the same. Thus, 20 s for the online period is selected, which shows stable readings during longer periods as depicted in Fig. 4 where multiple continuous readings show a correlation above 95%.

C. Sleep Period Characteristics

Further data collection is performed with 20-s online period and again 60- and 120-s sleep periods, as well as with different gas levels inside the sealed container. Fig. 5 shows transients for five different LPG gas levels. The higher the gas concentration, the higher the transient maximum. However, transients in the highest gas concentrations exhibit different patterns than the ones in lower ones, or without the gas at all. In the first 20 s, the value jumps to 0.2 (1 V), after which it starts to rapidly increase for the next 1.5 min, as seen in Fig. 6. However, the initial jump is still significant enough to be distinguishable from lower gas levels, thus the online period of 20 s remains valid.

As depicted in Fig. 6, the MQ-2 sensor requires over 120 s to reach the actual value in the environment with high gas concentrations. Consequently, any use of sleep functionality in such an environment would require a preheat time of more than 120 s in order to read correct values. Therefore, we select 120 s for our sleep period along with 20 s for the online period and thus get gas sensor readings approximately every 140 s.

D. Collection of Transients

In order to collect a data set comprising a set of transients, previously defined online–sleep period is used, along with different gas concentrations. For the 20-s online period, data are collected ten times in a second (i.e., every 100 ms) resulting in 186 values in a single transient (186 instead 200 since every reading takes 0.15 ms). The total data set contains 381 transients with their matching actual values, hence resulting in a total of 70.866 values. The entire data set is visualized in Fig. 7.

Furthermore, gas concentrations are classified in eight classes listed in Table I, where the entire data set C comprises all classes, namely, $C = c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7$. It is important to highlight that class c_5 contains the most elements since 0.5 value is recognized as a border between the environment with and without significant gas concentration. Therefore, the classification is used for even stratification of transients when applying the threefold validation approach during LSTM model training and testing, i.e., to even out the presence of all classes in training as well as in the test data sets.

IV. LSTM Model Training and Validation

In this section, we explain the configuration and usage of LSTM NN for predicting readings of the MQ-2 gas sensor based on its initial transient. The LSTM algorithm and the data set stratification and diversification are explained. Afterward, LSTM parameters are selected based on the best model accuracy and used for selecting the read frequency for input data as well as for testing model's performance with scarce data.

TABLE I
GAS CONCENTRATION CLASSES

Class name	Gas concentration
c_0	$0 - 0.1$
c_1	$0.1 - 0.2$
c_2	$0.2 - 0.3$
c_3	$0.3 - 0.4$
c_4	$0.4 - 0.5$
c_5	$0.5 - 0.6$
c_6	$0.6 - 0.7$
c_7	$0.7 - 0.8$

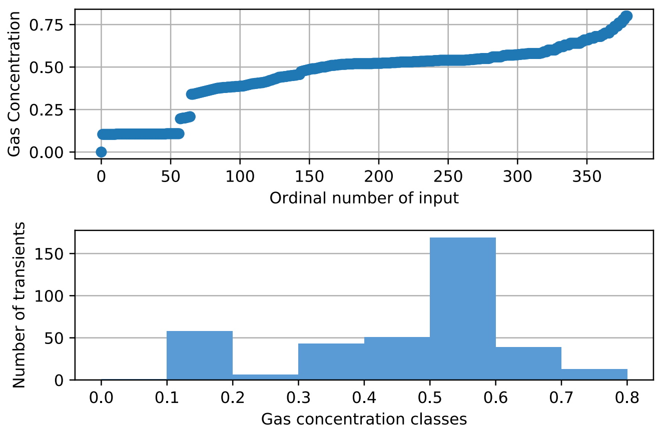


Fig. 7. Actual values for 381 transients (top figure) grouped in eight classes (bottom figure).

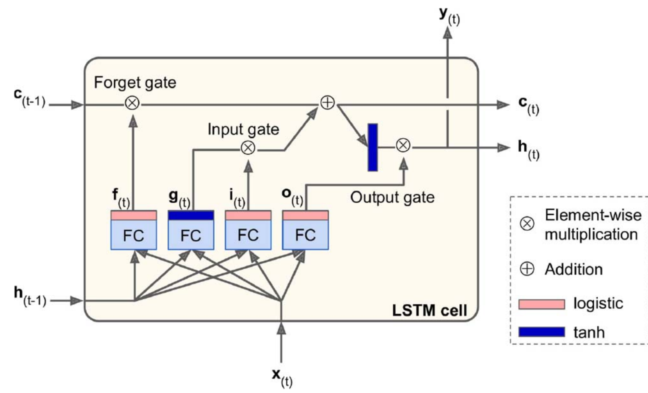


Fig. 8. LSTM cell [29].

A. LSTM Algorithm

LSTM NNs [28], [29] are used to predict actual values from transients. LSTM is a type of recurrent NN (RNN) that has a feedback connection. Along with single data points, it can process data sequences as well (e.g., gas transient values). A common LSTM cell contains an input gate, an output gate, and a forget gate, as depicted in Fig. 8. The cell remembers the values over a time period and the three gates (input, output, and forget gates) regulate the data flow in and out of the cell. The forget gate decides which information will be removed from the cell state. The input gate decides which states will be updated and the output gate decides which part of the cell states will be outputted. Therefore, LSTM has the ability to remove or add information to the cell state, instead of a mechanism that completely overrides cell states taken by classical RNN [15].

LSTM cell state is split in two vectors: 1) $h_{(t)}$ and 2) $c_{(t)}$; $h_{(t)}$ stands for short-term state while $c_{(t)}$ stands for the long-term state. On the one hand, an input vector $x_{(t)}$ and a previous short-term state $h_{(t-1)}$ are used as inputs to four different fully connected layers, namely, $f_{(t)}$, $g_{(t)}$, $i_{(t)}$, and $o_{(t)}$ depicted in Fig. 8. The main layer that analyzes $x_{(t)}$ and $h_{(t-1)}$ outputs $g_{(t)}$, which is used for calculating $h_{(t)}$ and the output vector $y_{(t)}$ expressed with

$$g_{(t)} = \tanh\left(W_{xg}{}^{T} * x_{(t)} + W_{hg}{}^{T} * h_{(t-1)} + b_g\right). \quad (1)$$

On the other hand, a previous long-term state $c_{(t-1)}$ enters the cell and goes through the forget gate, which defines parts of the long-term state that should be forgotten, and is controlled by the output of the $f_{(t)}$ layer expressed with

$$f_{(t)} = \sigma\left(W_{xf}{}^{T} * x_{(t)} + W_{hf}{}^{T} * h_{(t-1)} + b_f\right) \quad (2)$$

where b_g and b_f are the bias terms, W_{xg} and W_{xf} are the weight matrices of the input vector $x_{(t)}$, and W_{hg} and W_{hf} are the weight matrices of the previous short-term state $h_{(t-1)}$.

Other layers also give logistic output, namely, 0 or 1, which controls the closing and opening of the gates, respectively. Layer $i_{(t)}$ controls the input gates, while $o_{(t)}$ controls the output gates, and both can be expressed with

$$i_{(t)} = \sigma\left(W_{xi}{}^{T} * x_{(t)} + W_{hi}{}^{T} * h_{(t-1)} + b_i\right) \quad (3)$$
$$o_{(t)} = \sigma\left(W_{xo}{}^{T} * x_{(t)} + W_{ho}{}^{T} * h_{(t-1)} + b_o\right) \quad (4)$$

where b_i and b_o are again the bias terms, while pairs of W_{xg} and W_{xf}, as well as W_{hg} and W_{hf} are the weight matrices of the input vector $x_{(t)}$ and the previous short-term state $h_{(t-1)}$, respectively.

When some long-term memories are removed, output from the forget gate is combined with $g_{(t)}$, which gives a long-term state

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)}. \quad (5)$$

Additionally, the output from $g_{(t)}$ goes through the input gate, which defines what part of it should be combined with a duplicate of the long-term state $c_{(t)}$. Such, they enter the tanh function and go through the output gate, which defines what part of it should be outputted as both $h_{(t)}$ and $y_{(t)}$, expressed with

$$y_{(t)} = h_{(t)} = o_{(t)} \otimes \tanh\left(c_{(t)}\right). \quad (6)$$

Output $y_{(t)}$, as well as the short-term state $h_{(t)}$ and the long-term state $c_{(t)}$ are given for each input $x_{(t)}$, which is represented in a form of time-series data.

That said, LSTM is suitable for our application as the transient represents time-series data. While LSTM NN handles time-series data very well, it is still important to carefully select parameters, as well as a data set. In order to find the best parameters, different combinations of a number of neurons, learning rates, and epochs are tested. More neurons can help in case of underfitting, reducing the number of epochs helps in case of overfitting, while a smaller learning rate helps to eliminate exploding gradients. The LSTM network used in

TABLE II
PARAMETER COMBINATIONS FOR BUILDING LSTM

Parameter	Values
Number of neurons	20, 50, 80
Number of Epochs	250, 500, 750, 1000
Learning Rate	0.001, 0.0001, 0.00001

this article contains one LSTM layer, Relu activation function, Adam optimizer, and MSE loss calculation with the fixed seed (= 123). The fixed seed is used in order to be able to reproduce the results. Different combinations of neurons, learning rates, and epochs listed in Table II are used in order to obtain the best results.

B. Data Set Stratification and Diversification

To train the LSTM algorithm for an MQ-2 gas sensor, 304 transients out of 381 from the data set shown in the previous section are used. This leaves 77 transients for the test set, which is around 20% of the whole data set. The training set is further split using the stratified threefold algorithm, which selects one third of transients from each class to validate the LSTM algorithm.

In order to diversify the training set and reduce data transmission, other combinations are considered as well besides 186 values long transient.

1) Analyzing all 186 values collected during 20 s ($\sim$ 9 readings per second).
2) Analyzing 20 values collected during 20 s (reading every second).
3) Analyzing only 10 values collected during 20 s (the sensor still has to be online for 20 s, otherwise it will not heat up enough). Here, we include three combinations:
    a) data collected every 2 s;
    b) data from the first 10 s (every second);
    c) data from the first second (collected every 100 ms).

Furthermore, to test how the best of these solutions deal with scarce data, i.e., missing entire classes of readings, the algorithm is executed on the training set without data from every class (c_0–c_7), as defined in Table III. Finally, the results are discussed and compared using RMSE [30]. Since RMSE reflects the distance between real and predicted values, it is used to evaluate the performance of predictions, i.e., smaller RMSE implies higher prediction accuracy. Equation (7) defines a formula for RMSE, where e is the error between the predicted and the actual value

$$\mathrm{RMSE} = \sqrt{1/n \sum_{i=1}^{n} e^2}. \tag{7}$$

C. LSTM Parameters Selection

As described in the previous section, LSTM is created based on the LPG gas concentration collected by the MQ-2 sensor. Table IV shows the results of different combinations of cell numbers, epochs, and learning rates with their respective RMSE. Some large updates to weights during training cause a numerical overflow or underflow often referred to as exploding gradients, which results with NaN values for some combination of parameters, especially in case of a larger learning rate. Nevertheless, 50 cells, 500 epochs, and the learning rate of 0.0001 give the best performance during training. This set of parameters also outperforms others when applied on the test set by giving even better results, hence confirming that it does not either overfit or underfit.

Fig. 9(b) shows a histogram of errors for 186 values long transients for the MQ-2 sensor on the test set. Most errors are below 5% although RMSE is 0.06829. There are few errors above 30%, however, as visible from Fig. 9(a), those errors

TABLE III
SETS OF DATA SETS WITH INCLUDED/EXCLUDED CLASSES

Dataset	Included/excluded classes
$C \setminus \{c_0\}$	[ ][c_1][c_2][c_3][c_4][c_5][c_6][c_7]
$C \setminus \{c_1\}$	[c_0][ ][c_2][c_3][c_4][c_5][c_6][c_7]
$C \setminus \{c_2\}$	[c_0][c_1][ ][c_3][c_4][c_5][c_6][c_7]
$C \setminus \{c_3\}$	[c_0][c_1][c_2][ ][c_4][c_5][c_6][c_7]
$C \setminus \{c_4\}$	[c_0][c_1][c_2][c_3][ ][c_5][c_6][c_7]
$C \setminus \{c_5\}$	[c_0][c_1][c_2][c_3][c_4][ ][c_6][c_7]
$C \setminus \{c_6\}$	[c_0][c_1][c_2][c_3][c_4][c_5][ ][c_7]
$C \setminus \{c_7\}$	[c_0][c_1][c_2][c_3][c_4][c_5][c_6][ ]
$C \setminus \{c_0, c_2, c_3, c_4, c_6\}$	[ ][c_1][ ][ ][ ][c_5][ ][c_7]
$C \setminus \{c_0, c_2, c_3, c_4, c_5\}$	[ ][c_1][ ][ ][ ][ ][c_6][c_7]

TABLE IV
SELECTING OPTIMAL PARAMETERS FOR LSTM NN

Cells	Epochs	Learning rate	Train set RMSE	Test set RMSE
20	250	0.001	0.1748	0.1557
20	500	0.001	NaN	NaN
20	750	0.001	0.2103	0.2505
20	1000	0.001	NaN	NaN
20	250	0.0001	0.1687	0.1599
20	500	0.0001	0.1116	0.0872
20	750	0.0001	0.1083	0.0919
20	1000	0.0001	0.1060	0.0881
20	250	0.00001	0.4042	0.4007
20	500	0.00001	0.2889	0.2890
20	750	0.00001	0.2162	0.2239
20	1000	0.00001	0.1945	0.1885
50	250	0.001	NaN	NaN
50	500	0.001	NaN	NaN
50	750	0.001	NaN	NaN
50	1000	0.001	NaN	NaN
50	250	0.0001	0.1217	0.0992
50	**500**	**0.0001**	**0.0926**	**0.0683**
50	750	0.0001	NaN	NaN
50	1000	0.0001	0.1496	0.1610
50	250	0.00001	0.2906	0.2923
50	500	0.00001	0.2011	0.2028
50	750	0.00001	0.1682	0.1678
50	1000	0.00001	0.1421	0.1280
80	250	0.001	NaN	NaN
80	500	0.001	NaN	NaN
80	750	0.001	NaN	NaN
80	1000	0.001	NaN	NaN
80	250	0.0001	0.1126	0.0807
80	500	0.0001	0.1175	0.0991
80	750	0.0001	NaN	NaN
80	1000	0.0001	NaN	NaN
80	250	0.00001	0.2347	0.2407
80	500	0.00001	0.1615	0.1577
80	750	0.00001	0.1517	0.1401
80	1000	0.00001	0.1226	0.1025

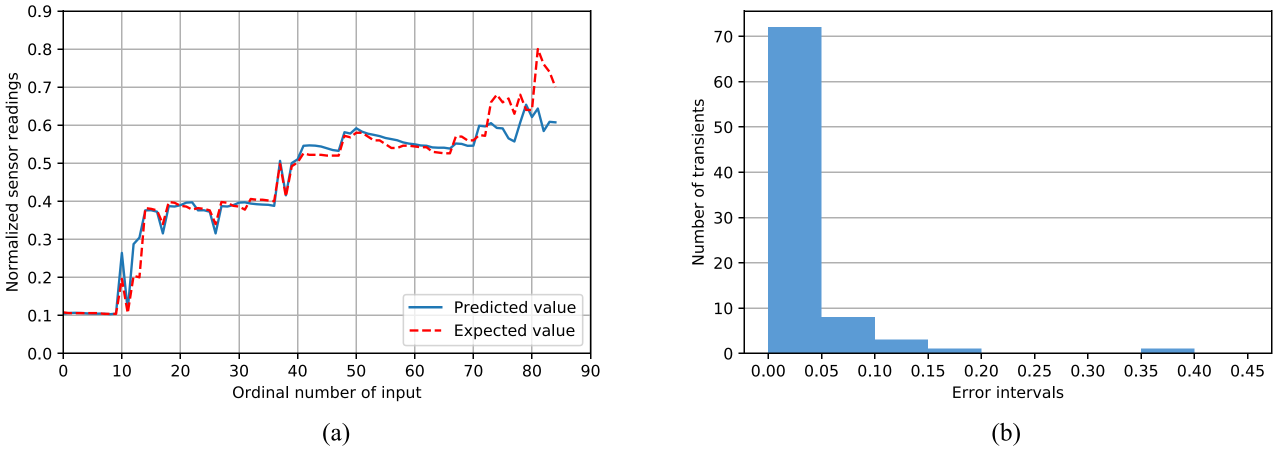


Fig. 9. Comparison of predicted and expected values for MQ-2 gas sensor. (a) Absolute values (MQ-2). (b) Errors histogram (MQ-2).

TABLE V
MQ-2 RMSE FOR DIFFERENT TRANSIENT SIZES

Test	**Size**	**Train set RMSE**	**Test set RMSE**
1)	186 values – every 100ms	0.0926	0.0683
2)	**20 values – every second**	**0.0615**	**0.0594**
3a)	10 values – every 2s	0.0940	0.0871
3b)	10 values – first 10 seconds	0.1226	0.1286
3c)	10 values – first second	0.1606	0.1673

TABLE VI
RMSE FOR DIFFERENT CLASSES (TRANSIENT SIZE 186)

Dataset	**Train set RMSE**	**Test set RMSE**
$C \setminus \{c_0\}$	0.0794	0.0859
$C \setminus \{c_1\}$	0.0546	0.2074
$C \setminus \{c_2\}$	0.1171	0.0985
$C \setminus \{c_3\}$	0.1311	0.1102
$C \setminus \{c_4\}$	0.1104	0.1140
$C \setminus \{c_5\}$	0.1992	0.1580
$C \setminus \{c_6\}$	0.1044	0.1204
$C \setminus \{c_7\}$	0.0862	0.1013
$C \setminus \{c_0, c_2, c_3, c_4, c_6\}$	0.2052	0.1662
$C \setminus \{c_0, c_2, c_3, c_4, c_5\}$	0.1662	0.2279
C	*0.0926*	*0.0683*

TABLE VII
RMSE FOR DIFFERENT CLASSES (TRANSIENT SIZE 20)

Dataset	**Train set RMSE**	**Test set RMSE**
$C \setminus \{c_0\}$	0.0715	0.0634
$C \setminus \{c_1\}$	0.0409	0.1731
$C \setminus \{c_2\}$	0.0751	0.0668
$C \setminus \{c_3\}$	0.0564	0.0715
$C \setminus \{c_4\}$	0.0648	0.0685
$C \setminus \{c_5\}$	0.1916	0.1435
$C \setminus \{c_6\}$	0.0611	0.0608
$C \setminus \{c_7\}$	0.0664	0.0634
$C \setminus \{c_0, c_2, c_3, c_4, c_6\}$	0.0819	0.1166
$C \setminus \{c_0, c_2, c_3, c_4, c_5\}$	0.2210	0.1789
C	*0.0615*	*0.0593*

occur for high gas concentrations. Since predicted values are above 0.5 (2.5 V), they are still classified as the dangerous gas level.

D. Frequency Selection for Input Data

In further analysis, we use the algorithm configuration that gives the best results for 186 values long transient and test it on a smaller number of transient values. The results in Table V show that more data are not necessarily better than less data. Despite expectations that a larger transient will result in better predictions, the best predictions are in the case when values are evenly sampled at 1 Hz. If the algorithm learns from the data that contains a large number of features, it often overfits the noise and does not work well on real-world data.

E. Performance Testing With Scarce Data

To test how the models deal with scarce data, the algorithm is executed without transients from every class in the training set and then tested on the train test that contains excluded classes. The results given in Table VI show that some classes are smaller in both train and test sets, hence they do not affect the results significantly, while the largest c_5 class affects results significantly. The algorithm also tries to learn behavior from the first and two last classes, however, results are not promising. When the biggest c_5 class is included in the training, the test set RMSE is 0.16624, which is a satisfying result in terms of machine learning.

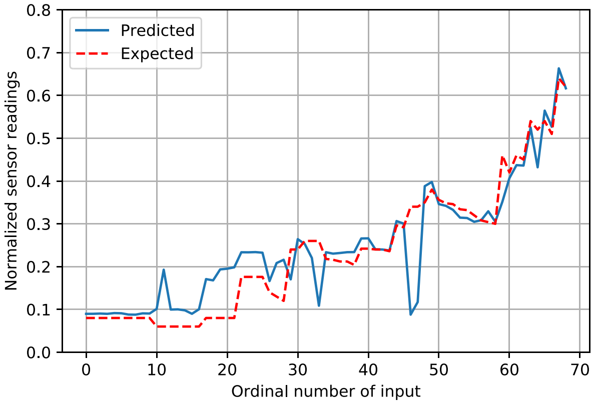
Since 20 values long transients show better results than 186 values long one, similar tests are performed with such setup as well. The results in Table VII show that it is possible to train data only with the first, last, and biggest class c_5 with RMSE of 0.116627. In general, using 20 values long transient outperforms 186 long one.

V. EVALUATION AND DISCUSSION

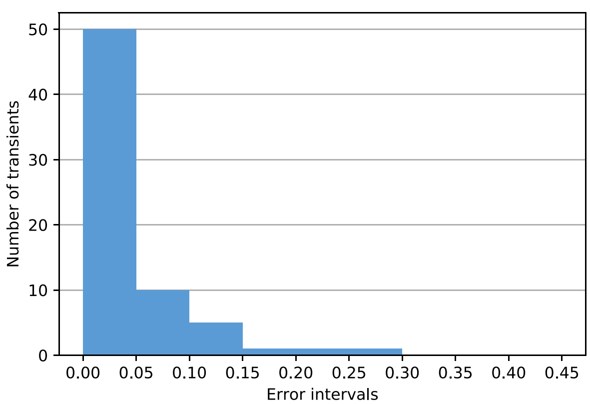
In order to evaluate our methodology, the approach is tested on the MQ-5 and MQ-6 gas sensors to verify its applicability on different low-cost sensors. The same online–sleep ratio of 20 s for an online and 120 s for a sleep period is used for testing MQ-5 and MQ-6 sensors.

TABLE VIII
MQ-5 RMSE

Size	Train set RMSE	Test set RMSE
186	0.1104	0.1080
20	**0.0702**	**0.0700**

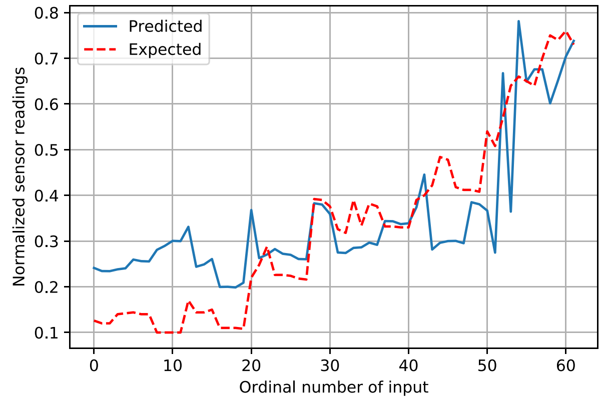


(a)

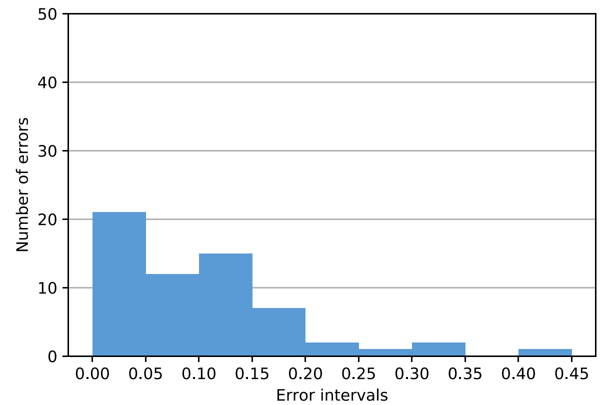


(b)

Fig. 10. Comparison of predicted and expected values for MQ-5 gas sensor. (a) Absolute values (MQ-5). (b) Errors histogram (MQ-5).



(a)



(b)

Fig. 11. Comparison of predicted and expected values for MQ-6 gas sensor. (a) Absolute values (MQ-6). (b) Errors histogram (MQ-6).

TABLE IX
MQ-6 RMSE

Size	Train set RMSE	Test set RMSE
186	0.1597	0.1405
20	**0.1238**	**0.1238**

A. MQ-5 Sensor Results

After data are collected with an MQ-5 gas sensor following the same approach described in Section III, NN with parameters that provide the best results for the MQ-2 gas sensor are used. The collected data set contains 223 transients in the train set and 70 transients in the test set (∼24%). In the case of 186 values long transient, RMSE on the test set is 0.10801, while in the case of 20 values long transient that has better results for MQ-2, the test set RMSE is better as well with the value of 0.07002, as shown in Table VIII. As it is visible from Fig. 10(a) and (b), the most errors are less than 10%, with two predictions from class 0.3–0.4 with the error above 20%.

B. MQ-6 Sensor Results

The same procedure is used for the MQ-6 gas sensor. However, in this case, the results are only slightly worse than the ones for MQ-2 and MQ-5. The used data set contains 216 transients in the train set and 59 transients in the test set (∼27%). Again, the test set results are acquired for both 186 and 20 values long transients. As with MQ-2 and MQ-5 gas sensors, MQ-6 also gives better results with 20 values long transients having RMSE of 0.1293, while 186 values long transients give RMSE of 0.1405, as shown in Table IX. Fig. 11(a) and (b) depicts that more than 85% of predicted values have an error lower than 15% with the maximum error of 25%.

C. Discussion on Energy Efficiency

The results presented in this article show that the LSTM prediction model can be used with low-cost gas sensors, such as MQ-2, MQ-5, and MQ-6 with sufficient accuracy. Predictions for MQ-2 give the best accuracy measured with RMSE of 0.05938067 as the model is calibrated for the specific sensor. In the case of the MQ-5 sensor, RMSE is 0.07002, which is almost as good as in the case of the MQ-2 sensor,

TABLE X
ENERGY CONSUMPTION

Scenario	P_{Tx}	P_{MQ}	P_{MA}	P_{MS}	T_{OA}	T_A	T_S	E
a) Always online	92.4mW	900mW	615mW	131.5mW	0.206s	140s	0s	212.25J
b-1) With full preheat and sleep period (the best scenario)					0.206s	40s	100s	73.89J
b-2) With full preheat and sleep period (the worst scenario)					0.206s	120s	20s	184.57J
c) With prediction based on a partial preheat and sleep period					**0.370s**	**20s**	**120s**	**46.22J**

while the results for MQ-6 are slightly worse with the RMSE value of 0.1293 that is still acceptable in terms of machine learning. Note that parameters that give the best results for the MQ-2 sensor are used *as is* without the calibration for MQ-5 and MQ-6, which implies that the model could be further tweaked to obtain better results.

Putting sensors to sleep only makes sense when trying to achieve energy savings. Energy consumption is described with the following expression:

$$E = P_{Tx} \cdot T_{\mathrm{OA}} + P_{\mathrm{MQ}} \cdot T_A + P_{\mathrm{MA}} \cdot (T_{\mathrm{OA}} + T_A) + P_{\mathrm{MS}} \cdot T_S \tag{8}$$

where:

- P_{Tx} is the power consumed by a LoRaWAN module used for data transmission [31];
- P_{MQ} is the power consumed by an MQ-2 sensor [11];
- P_{MA} is the power consumed by a microcontroller during online period [32];
- P_{MS} is the power consumed by a microcontroller during sleep period [32];
- T_{OA} is time on air calculated based on calculations shown in [33] using LoRaWAN with code rate = 1, spreading factor = 10, and bandwidth = 125 kHz;
- T_A is the online period;
- T_S is the sleep period.

Energy consumption is calculated for three sensor setups described in Section III-A (Fig. 1) and results are shown in Table X. Since the transmission interval is less than a second, when combined with transmission power, it results in insignificant energy consumption compared to large power and intervals of online and sleep periods. The most significant energy consumer is a gas sensor which requires up to 900 mW (Fig. 12). Note that a microcontroller consumes energy during both periods; online and sleep. However, energy consumed during the sleep period is ~ 4.5 times lower compared to the power consumed during the online period. Energy saving can be even bigger when using other microcontrollers instead of Arduino Uno. That said, energy consumption highly depends on the online period; the longer the online period, the higher the power consumption. Due to specifics of MQ gas sensors explained in Section III-C, the shortest online period that is suitable for such predictions is 20 s, followed by 120-s long sleep period.

As it is shown in Fig. 12 and Table X, our solution uses ~80% less energy compared to Scenario a) when the sensor is always online, while the online period is decreased by ~85%. Since the MQ-2 sensor needs at least 40 s to read the actual value in the atmosphere without gas, and around 120 s in the conditions with high gas concentration, our approach can be

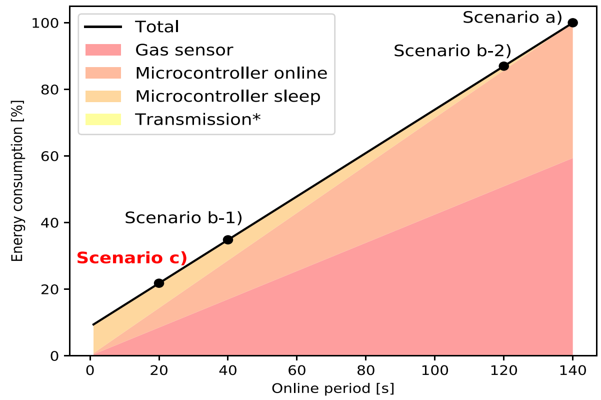


Fig. 12. Energy consumption compared to the sensor that is always online (*Transmission energy is insignificant compared to others, thus not visible in this figure).

compared to the best and the worst case. As it is visible from Fig. 12 and Table X, comparing our solution to Scenario b-1) with full preheat and sleep period—the best scenario (40 s online–100 s sleep); our solution consumes ~40% less energy while collecting ~50% less data. Note that Scenario b-1) will miss large gas concentrations. In the worst case, the sensor is online for 120 s and sleeps 20 s, which is the exact opposite of our approach, meaning that the sensor will reduce the data amount for ~15% with ~13% energy savings.

To sum up, the longer sleep period results with higher energy savings. In our case, the sensor is online for 20 s and then asleep for 120 s, resulting with approximately 500–520 s (~8.5 min) of online time per hour, while sleeping for 3000–3120 s (~51 min). Therefore, it is being online ~3.5 h a day; 1 day a week; 4.5 days a month; or 52.15 days in a year. That said, our solution significantly reduces the amount of data with a duty cycle of (~85%) while increasing energy efficiency for ~80%.

Comparing our solution to a scenario with the full preheat and sleep period [Fig. 1(b)], one tradeoff has to be noted. Our solution sends 20 values toward a gateway and the prediction is calculated on the gateway side, while the sensor in the scenario in Fig. 1(b) sends only a single value. As it is visible from Fig. 12 and Table X, transmission energy in case of 1 and 20 B is almost the same and it can be ignored compared to the energy consumed by the gas sensor and Arduino Uno. However, TensorFlow light for microcontrollers (including the Arduino platform) [34] is now available and machine learning models can be deployed on the edges of an IoT system, i.e., a gateway. Using this approach, another potential tradeoff is the amount of energy consumed when the machine learning algorithm is deployed on a battery-powered sensor as the energy consumption can be higher than the energy savings gained with 120-s long sleep period. Although this is out of our research scope in this article, it is important to be aware of the potential challenges and tradeoffs.

VI. Conclusion

In this article, we used LSTM NN to predict the actual values of gas concentration from the transient of the MQ-2 gas sensor acquired during its preheat period. Furthermore, we evaluated our approach on two different gas sensors, namely, MQ-5 and MQ-6. In the case of MQ-2 gas sensors, the obtained RMSE is 0.05938067. The optimal LSTM parameters used for MQ-2 were applied on MQ-5 and MQ-6 as well. Predictions for the MQ-5 sensor give an RMSE value of 0.07002, while only slightly higher for MQ-6 with a value of 0.1293.

The main contribution of this article is the investigation of the MQ-2 sensor behavior, as well as the methodology for building a prediction model based on LSTM that we applied on two additional low-cost gas sensors. The results show that in order to extend battery life, a sensor should wake up periodically to collect data and sleep in the meanwhile. In the case of sensors that require heating up before collecting the actual value, only the beginning of the transient can be collected and the actual value predicted on the gateway side using LSTM NN. This way, instead of being continuously online, the sensor collects data for 20 s and sleeps for 120 s. Compared to the solution when the sensor is online until the actual value is reached (up to 2 min) and then being sent to sleep, our approach collects half as much data in the same 140-s period.

Our future work includes investigating the energy consumption in terms of data transmission. We also plan to implement a machine learning algorithm on the Arduino board and investigate its energy consumption compared to the energy savings achieved with 120-s sleep period.

References

[1] "Cellular IoT evolution for industry digitalization," Ericsson, Stockholm, Sweden, Rep. GFMC-19:000017 UEN, 2019. [Online]. Available: https://www.ericsson.com/en/white-papers/cellular-iot-evolution-for-industry-digitalization

[2] "Cisco global cloud index: Forecast and methodology," Cisco, San Jose, CA, USA, Rep. 2016-2021, 2016, [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html

[3] S. Kumar and V. K. Chaurasiya, "A strategy for elimination of data redundancy in Internet of Things (IoT) based wireless sensor network (WSN)," *IEEE Syst. J.*, vol. 13, no. 2, pp. 1650–1657, Jun. 2019.

[4] R. Radha and M. Muralidhara, "Removal of redundant and irrelevant data from training datasets using speedy feature selection method," *Int. J. Comput. Sci. Mobile Comput.*, vol. 5, pp. 359–364, Jul. 2016.

[5] J. Č. Gambiroža and T. Mastelic, "Big data challenges and trade-offs in energy efficient Internet of Things systems," in *Proc. 26th Int. Conf. Softw. Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2018, pp. 1–6.

[6] T. Shu, J. Chen, V. K. Bhargava, and C. W. de Silva, "An energy-efficient dual prediction scheme using LMS filter and LSTM in wireless sensor networks for environment monitoring," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6736–6747, Aug. 2019.

[7] C. Baratto, F. Rigoni, G. Faglia, E. Comini, D. Zappa, and G. Sberveglieri, "ZnO and SnO_2 one-dimensional sensors for detection of hazardous gases," in *Proc. IEEE SENSORS*, Oct. 2017, pp. 1–3.

[8] L. Scholz, A. O. Perez, B. Bierer, P. Eaksen, J. Wöllenstein, and S. Palzer, "Carbon dioxide sensor for mobile devices: A novel approach for low-power consuming, highly sensitive NDIR sensors," in *Proc. IEEE SENSORS*, Oct. 2016, pp. 1–3.

[9] S. Rademacher, K. Schmitt, M. Mengers, and J. Wöllenstein, "Sensor network with energy efficient and low-cost gas sensor nodes for the detection of hazardous substances in the event of a disaster," in *Proc. IEEE Topical Conf. Wireless Sensors Sensor Netw. (WiSNet)*, Jan. 2015, pp. 59–61.

[10] Z. Jia *et al.*, "Continuous low-power ammonia monitoring using long short-term memory neural networks," in *Proc. 16th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2018, pp. 1–6.

[11] MQ-2 Gas Sensor. (2019). *MQ-2 Semiconductor Sensor for Combustible Gas*. [Online]. Available: https://www.pololu.com/file/0J309/MQ2.pdf

[12] MQ-5 Gas Sensor. (2019). *Technical Data MQ-5 Gas Sensor*. [Online]. Available: https://www.parallax.com/sites/default/files/downloads/605-00009-MQ-5-Datasheet.pdf

[13] MQ-6 Gas Sensor. (2019). *MQ-6 Semiconductor Sensor for LPG*. [Online]. Available: https://www.pololu.com/file/download/MQ6.pdf?file_id=0J312

[14] L. Salhi, T. Silverston, T. Yamazaki, and T. Miyoshi, "Early detection system for gas leakage and fire in smart home using machine learning," in *Proc. IEEE Int. Conf. Consumer Electron. (ICCE)*, Jan. 2019, pp. 1–6.

[15] Z. Chen, Y. Liu, and S. Liu, "Mechanical state prediction based on LSTM neural network," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 3876–3881.

[16] Y. Wang, J. Zhou, K.-J. Chen, Y. Wang, and L. Liu, "Water quality prediction method based on LSTM neural network," in *Proc. 12th Int. Conf. Intell. Syst. Knowl. Eng. (ISKE)*, Nov. 2017, pp. 1–5.

[17] L. Yu, J. Chen, and G. Ding, "Spectrum prediction via long short term memory," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 643–647.

[18] S. Liu, G. Liao, and Y. Ding, "Stock transaction prediction modeling and analysis based on LSTM," in *Proc. 13th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, May 2018, pp. 2787–2790.

[19] W. Yao, P. Huang, and Z. Jia, "Multidimensional LSTM networks to predict wind speed," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 7493–7497.

[20] N. Kim, M. Kim, and J. K. Choi, "LSTM based short-term electricity consumption forecast with daily load profile sequences," in *Proc. IEEE 7th Global Conf. Consum. Electron. (GCCE)*, Oct. 2018, pp. 136–137.

[21] F. Qian and X. Chen, "Stock prediction based on LSTM under different stability," in *Proc. IEEE 4th Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2019, pp. 483–486.

[22] I. Boulanouar, S. Lohier, A. Rachedi, and G. Roussel, "DTA: Deployment and tracking algorithm in wireless multimedia sensor networks," *Ad Hoc Sensor Wireless Netw.*, vol. 28, nos. 1–2, pp. 115–135, Jan. 2015.

[23] I. Boulanouar, A. Rachedi, S. Lohier, and G. Roussel, "Energy-aware object tracking algorithm using heterogeneous wireless sensor networks," in *Proc. IFIP Wireless Days (WD)*, 2011, pp. 1–6.

[24] A. A. Khan, M. H. Rehmani, and A. Rachedi, "Cognitive-radio-based Internet of Things: Applications, architectures, spectrum related functionalities, and future research directions," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 17–25, Jun. 2017.

[25] B. Liu, J. Wang, S. Ma, F. Zhou, Y. Ma, and G. Lu, "Energy-efficient cooperation in mobile edge computing-enabled cognitive radio networks," *IEEE Access*, vol. 7, pp. 45382–45394, 2019.

[26] RTC DS3231. (2020). *RTC DS3231 Datasheet*. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/DS3231.pdf

[27] MOSFET IRFZ44N. (2020). *MOSFET IRFZ44N Datasheet*. [Online]. Available: https://datasheet.octopart.com/IRFZ44N-Inchange-Semiconductor-datasheet-15981338.pdf

[28] N. McClure, *TensorFlow Machine Learning Cookbook*, 1st ed. Birmingham, U.K.: Packt, 2017.

[29] A. Geron, *Hands-On Machine Learning With Scikit-Learn and TensorFlow*, 1st ed. Sebastopol, CA, USA: O'Reilly, 2017.

[30] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY, USA: Springer, 2013.

[31] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, "Energy consumption model for sensor nodes based on LoRa and LoRaWAN," *Sensors*, vol. 18, no. 7, p. 2104, Jun. 2018.

[32] M. Al-Shorman, M. Al-Kofahi, and O. Al-Kofahi, "A practical microwatt-meter for electrical energy measurement in programmable devices," *Meas. Control*, vol. 51, nos. 9–10, pp. 383–395, Aug. 2018.

[33] D. Singh, O. G. Aliu, and M. Kretschmer, "LoRa wanevaluation for IoT communications," in *Proc. Int. Conf. Adv. Comput. Commun. Informat. (ICACCI)*, Sep. 2018, pp. 163–171.

[34] TensorFlow. (2019). *TensorFlow Lite for Microcontrollers*. [Online]. Available: https://www.tensorflow.org/lite/microcontrollers/overview

Jelena Čulić Gambiroža received the bachelor's and master's degrees in computer science from the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Split, Croatia, in 2013 and 2015, respectively, where she is currently pursuing the Ph.D. degree.

She is currently employed as a Researcher with Ericsson Nikola Tesla, Split. Her research interests include big data challenges in Internet of Things as well as data analytics, including statistical methods and machine learning.

Toni Mastelić received the bachelor's and master's degrees in computer science from the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Split, Croatia, in 2009 and 2011, respectively, and the Ph.D. degree from Vienna University of Technology, Vienna, Austria, in 2015.

He is currently employed as a Researcher with the ETK Research Department, Ericsson Nikola Tesla, Split. He worked as a Project Assistant with the Vienna University of Technology, where he worked as a University Assistant with the Institute of Software Technology and Interactive Systems. His research interests include cloud and edge computing, data analytic in Internet of Things, and artificial intelligence.

Tonko Kovačević received the Dipl.Ing. degree in computer science and electrical engineering from the University of Split, Split, Croatia, in 1993, the postgraduation degree from the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture in Split, University of Split, in 2016, under the supervision of Prof. M. Vrdoljak, and the Ph.D. degree in computer science from the University of Split in 2016.

He is currently employed as a Senior Lecturer with the University of Split, University Department of Professional Studies, Assistant to the Head of the Department, University of Split. His main research interests are usability, design, and analysis of security protocols for wireless networks.

Mario Čagalj (Associate Member, IEEE) received the Dipl.Ing. degree in computer science and electrical engineering from the University of Split, Split, Croatia, in 1998, and the Ph.D. degree in communication systems from the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in February 2006.

He is currently employed as a Full Professor with the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split. From 2000 to 2001, he completed the Predoctoral School in Communication Systems, EPFL, where he was a Research Assistant with the Laboratory for Computer Communications and Applications from 2001 to 2006. In September 2006, he joined the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, where he was an Assistant Professor with the Department of Electronics, from January 2007 to September 2010. In 2010 and 2016, he was promoted to an Associate Professor and a Professor, respectively. His research interests include the design and analysis of security protocols for wireless networks, applied cryptography, applications of game theory to wireless (and wired) networks, and the design of energy-efficient communication protocols for wireless networks.