Forecasting Short-Term Indoor Radon: A Machine Learning Approach Using LSTM Networks

Valdo Mpinga*, A. M. Rosado da Cruz*§, and Sérgio Ivan Lopes*†‡¥

* ADiT-Lab, Instituto Politécnico de Viana do Castelo, 4900-348 Viana do Castelo, Portugal

† CiTin - Centro de Interface Tecnológico Industrial, InovArcos, 4970-786 Arcos de Valdevez, Portugal

‡ IT - Instituto de Telecomunicações, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

§ Algoritmi Research Centre, Universidade do Minho, Guimarães, Portugal

¥ Corresponding Author: sil@estg.ipvc.pt

Abstract — **Indoor radon is a radioactive gas that can accumulate in homes and pose a health risk for humans. Forecasting indoor radon levels may be used as a tool for mitigating human exposure risk, and thus help to effectively manage indoor radon risk. Forecasting based on Machine Learning (ML) techniques involves predicting future levels of indoor radon gas based on past and current data, and thus help identify trends and patterns in the data over time. This work presents preliminary results regarding the implementation and evaluation of two LSTM-based approaches, for indoor radon forecasting, which can then be used as a tool to trigger preventive management procedures for Indoor Air Quality management. Preliminary results have shown that the normalized data using the Long Short-Term Memory (LSTM) algorithm proved to be the optimal approach for this application case, demonstrating superior accuracy across various forecasting time windows when compared to other approaches evaluated in this work.**

Keywords - LSTM, Bi-LSTM, Forecasting, IoT, Radon.

I. INTRODUCTION

Indoor radon exposure has a known relationship with lung cancer and other respiratory system problems. In [1], Nunes *et al.*, examined several studies that statistically justify the correlation between indoor radon exposure and the incidence of lung cancer in regions where concentrations are particularly high. Moreover, some studies also indicate that, even in situations where low indoor radon concentrations exist, a smaller tendency can also be identified, with respect to an increased incidence of lung cancer. The development of new technological solutions that lower indoor radon exposure, and thus contribute to the improved health of exposed populations, is an urgent matter.

Forecasting indoor radon levels may be used as a tool for mitigating human exposure risk, helping to effectively manage indoor radon risk. Forecasting based on Machine Learning (ML) techniques involves predicting future levels of indoor radon gas based on past and current data. This data is typically organized as time-stamped time-series data, and may include, in addition to radon gas concentration, other environ- mental parameters such as temperature, humidity, atmospheric pressure, etc. These ML-based methods can be used to make predictions of indoor radon levels based on historical data and can help identify trends and patterns in the data over time. However, it is important to evaluate the performance of these ML-based methods, validate their predictions, and ensure they are robust and reliable before making decisions based on the forecasted radon levels.

This work presents preliminary results regarding the implementation and evaluation of two LSTM-based approaches, for indoor radon forecasting, which can then be used as a tool to trigger preventive management procedures regarding Indoor Air Quality degradation. Preliminary results have shown that the normalized data using the Long Short-Term Memory (LSTM) algorithm proved to be the optimal approach for this application case, demonstrating superior accuracy across various forecasting time windows when compared to other approaches evaluated in this work.

This document is organized as follows. Section II presents an overview of related works. Section III introduces the adopted methodology. Section IV is devoted to the detailed description of the implementation. Section V presents the results and lastly, in Section VI, the main conclusions and future work directions are put forward.

II. RELATED WORKS

Over the years, several approaches have been implementing not only the LSTM (Long-Short Term Memory) algorithm, but also some algorithms derived from it that are mentioned in this section, in order to be able to make short/long term forecasts, solving the RNN's vanishing gradient problem [2], which is a problem encountered when training an artificial neural network with gradient-based learning methods and backpropagation. LSTMs differ from more conventional feedforward neural networks in that they feature feedback connections. With the use of this property, LSTMs can process whole data sequences (such as time series) without considering each data point separately. Instead, they can preserve pertinent information about earlier

2023 18th Iberian Conference on Information Systems and Technologies (CISTI)
20 – 23 June 2023, Aveiro, Portugal
ISBN: 978-989-33-4792-8

data in the sequence to aid in the processing of subsequent data points. Because of this, LSTMs are very good at processing data sequences like text, audio, and general time-series. In the end, LSTM networks are RNN's extensions, which increase the memory. Building pieces for an RNN's layers. By giving data "weights" LSTMs enable RNNs to either accept new information, forget it, or give it enough weight to affect the result. In [3], Adil Mir *et al.*, proposed time-series models using several of algorithms, to predict soil radon concentration, as a precursor for an earthquake because several studies found correlations between the soil radon gas and earthquake occurrences. In [4], Qing Luo, *et al.*, proposed an air condition cooling forecasting based on bidirectional LSTM. The author made a comparison between RNN, GRU, LSTM and Bi-LSTM and the Bi-LSTM outperformed the other algorithms, being the only one to outperform LSTM. In [5], Liu Wenya, *et al.*, proposed a cooling, heating and electric load Forecasting for Integrated Energy Systems Based on CNN-LSTM. The author also compared the errors between three algorithms, CNN, LSTM and the CNN-LSTM itself, which was the one that outperformed the other two models which used CNN and LSTM. B Ravi Krishna *et al.* [6], proposed a LSTM model to predict short-term traffic flow. The authors claim to have chosen this model for the fact that it can memorize extensive history input data and find the best time lags automatically. In [7] Karim Moharm *et al.*, proposed LSTM and bidirectional LSTM using different gate and state activation functions to predict the provisional trend of wind speed in a wind farm in Egypt. Idris Bodur, *et al.* [8], addressed short-term load demand forecasting based on LSTM, made a comparison to RNN, and concluded that it could predict with more than 90% accuracy and was more consistent in making short-term forecasts. In [9], Tiantian Tang *et al.*, proposed medium and long-term precipitation forecasting method based on data augmentation and machine learning algorithms. The data augmentation algorithm was based on K-means clustering algorithm and SMOTE (Synthetic Minority Oversampling Technique) and the machine learning algorithms used were RNN and LSTM. In [10] Jixiang Lu, *et al.*, presented a hybrid LSTM-CNN model method to improve short-term load forecasting accuracy. The authors claim that in addition to examining the useful information and prospective resources present in the enormous input data of CNN, the method leverages a deep network to learn temporal information through the LSTM network. The authors also made a benchmark using different algorithms, which included LSTM, ARIMA, and Random Forest. While dealing with time series forecasting problems, although there are more options and algorithms besides LSTM or LSTM variants, LSTM-based approaches for indoor radon forecasting seemed very promising since these approaches are optimal to capture long-term and short-term dependencies in time series sequential data.

III. Materials and Methods

The adopted methodology is depicted in detail in Fig. 1. To feed the machine learning models, the data has been acquired from RnMonitor project's database [11] which consisted in IoT devices installed in several rooms of distinct buildings, which measured radon gas concentration, CO2, temperature, humidity and atmospheric pressure. The data was accessed using a Grafana API endpoint which returned measurements from all the sensors since May 2019, which was made once an hour.

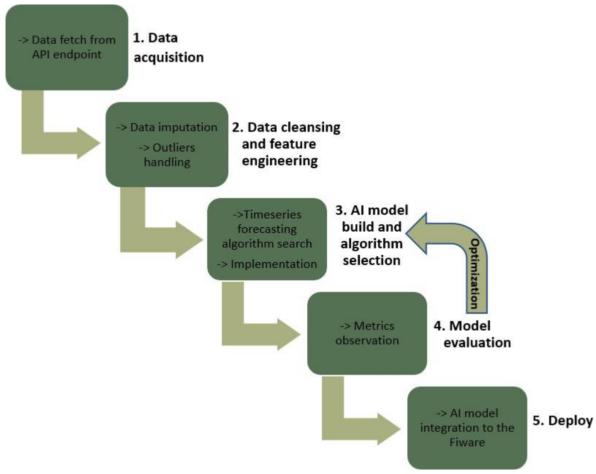

Fig. 1. Adopted methodology.

To be able to operate on the fetched data, the following steps were taken:

1) **Data grouping and pre-handling.** After the fetch, it was noticed that the endpoint returned all data from all sensors at once, which represented a small problem. Some sensors had irrelevant data that needed to be handled. In a universe of thousands of records, it would be much more feasible to have the data grouped by a sensor on a database for better visualization. A Python script had been written to make that grouping on a local Mongo database. This was a critical step because it allowed us to see which sensors had relevant data and which had irrelevant or few data due to some kind of malfunction, and because that data was useless, it was discarded.
2) **Data cleansing.** Unfortunately, in real world scenarios, data doesn't come always clean. Most of the time we get empty values, which are not good for machine learning. In this step, it has been verified if those empty values existed, and they did. Deleting all this data would not be the wisest decision, so to solve this issue data imputation comes to play, which is the substitution of estimated values for missing or inconsistent data items. There are several approaches to deal with data imputation. A simple approach is by using interpolation. For this we used the Pandas library, an open-source data analysis and manipulation library for Python. This operation consists in inserting intermediate values between two other values.

2023 18th Iberian Conference on Information Systems and Technologies (CISTI)
20 – 23 June 2023, Aveiro, Portugal
ISBN: 978-989-33-4792-8

IV. Implementation

Dealing with a time-series forecasting problem, two approaches were used in this project to proceed with the model training and evaluation, which were LSTM and Bidirectional LSTM. These approaches were implemented on data based on two different sensors, which had the healthiest data. The data was divided into 3 parts:

1) **Training data -** These took 80% of the data.
2) **Validation data -** These took 3% of the data.
3) **Test data -** These took 17% of the data.

A. Architecture

Figure 2 represents the overall forecasting architecture, and the additional building blocks of the indoor radon mitigation system. In this work, the ML algorithms under evaluation will be the LSTM (Long Short-Term Memory) and Bi-LSTM (Bidirectional LSTM), which are both recurrent neural net- works (RNNs) implementations commonly used in time series forecasting. Such implementations can handle short-term and long-term dependencies in time series data, and in this work will be applied for indoor radon forecasting.

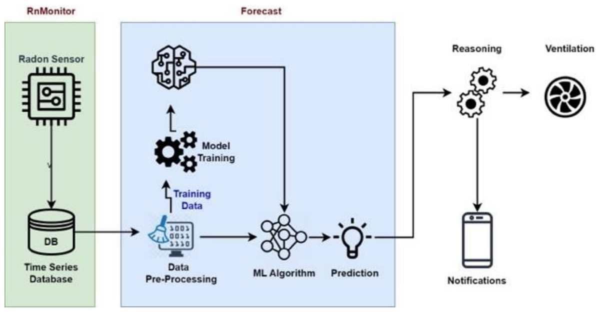

Fig. 2. Forecasting architecture.

B. Model definition and Training

The main goals on the model creation were to enable predicting radon levels until 6 hours ahead. The evaluation metric used was Mean Absolute Error (MAE) [12].

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

For the model training, using LSTM and Bi-LSTM, it was necessary to define a lag size, which is the number of samples the models will use to forecast future values. e.g., on the dataset, there is data (temperature, radon, pressure, CO2, and humidity) for every hour since May 2019. If we decide to train a model using a lag size of 6, the X_{train} data would include data from, for example, 1 PM, 2 PM, 3 PM, 4 PM, and 5 PM, and the y_{train} would be the data at 6 PM. This process is repeated iteratively, storing the data in this specific format in an array to train the model. So, in this example, we would be making a model to predict data 1 hour ahead. If we wanted to predict 2 hours ahead, instead of passing 6 PM on the y_{train} data, the data passed into it would be the data at 7 PM. This is the approach that was used for training the models. A total of 7 different lags were used to train these models, starting from lag 6 to 12, to compare which one would have the best error, with the lowest error being the best. These models were trained on data from two different sensors that had healthier data, meaning that there were no more than 30-40% missing data. It's important to note that the impact of missing data on time-series forecasting can be more severe compared to other machine learning tasks, as time-series data is inherently temporal and missing values can disrupt the continuity of the series. For each of these sensors, data were applied both LSTM and Bi-LSTM algorithms, including the 7 lags for each algorithm and in total. In machine learning, normalization is a technique used to adjust the range of feature values in a dataset. This process facilitates machine learning algorithms in learning and generalizing patterns effectively. The primary objective of normalization is to bring all the features to a similar scale and eliminate any bias caused by different units or ranges. In this project, the temperature variable can go up to 36°C or higher in the summer, while radon levels can reach up to 500 Bq/m³. The scales of the variables are disproportionate, which can hinder the model's ability to generalize. To address this issue, normalization was applied to all the variables in the dataset. Normalized data is generally easier to train, requiring fewer epochs to converge and achieve optimal results. In the unnormalized data, 100 epochs were required for convergence, but with the normalized data, only 50 epochs were needed, resulting in less time to train. Further details of the comparison will be presented ahead.

C. Forecasting

Tables I, II, III, and IV, show the correlation between the lags and the forecasting window. Depending on the forecasting window size, it influences the final estimation error, i.e., the higher the window is, the higher the error will be. It also shows that the bigger the lag size, doesn't mean that the Mean Absolute Error is going to be lower, which would be better. The Mean Absolute Error is actually simple to understand, if we look at table 1, on the 1H forecast result, on the lag 6 results, the Error is 37.59, this means that when evaluating the test data, which in this case had about 8000 data points for each of those hours and lag sizes, the model was at least either 37.59 wrong above or 37.59 wrong below the true value of the forecast.

V. Results

Having the model trained and ready for use, a web application has been created, to enable a final user to make an indoor forecast where a sensor is present, cf., Figures 3, 4, 5 and 6. In this process, the user would get into the website and click on the forecast data, and if the sensor data was from the last 12 hours, the system would retrieve that data to a server responsible to make the predictions for the different hours and selecting the model which has the best error according to the lag and then retrieve the forecasts along with the errors and finally that data is presented to the user on the website, along with the last 5 hours radon levels just to situate the user how the radon level has been going and where it might be going.

ISBN: 978-989-33-4792-8

TABLE I
NON-NORMALIZED LSTM MEAN ABSOLUTE ERROR RESULTS

Forecasting Window	Lag 6	Lag 7	Lag 8	Lag 9	Lag 10	Lag 11	Lag 12	Training Duration
1 Hour	34 Bq/m3	94 Bq/m3	38 Bq/m3	49 Bq/m3	36 Bq/m3	37 Bq/m3	37 Bq/m3	2h 15min 16s
2 Hour	56 Bq/m3	57 Bq/m3	65 Bq/m3	56 Bq/m3	70 Bq/m3	57 Bq/m3	56 Bq/m3	2h 14min 48s
3 Hour	68 Bq/m3	74 Bq/m3	73 Bq/m3	70 Bq/m3	69 Bq/m3	73 Bq/m3	79 Bq/m3	2h 16min 56s
4 Hour	85 Bq/m3	87 Bq/m3	87 Bq/m3	85 Bq/m3	86 Bq/m3	91 Bq/m3	88 Bq/m3	2h 17min 19s
5 Hour	98 Bq/m3	97 Bq/m3	109 Bq/m3	105 Bq/m3	106 Bq/m3	102 Bq/m3	100 Bq/m3	2h 41min 32s
6 Hour	107 Bq/m3	110 Bq/m3	111 Bq/m3	106 Bq/m3	109 Bq/m3	114 Bq/m3	108 Bq/m3	2h 10min 5s

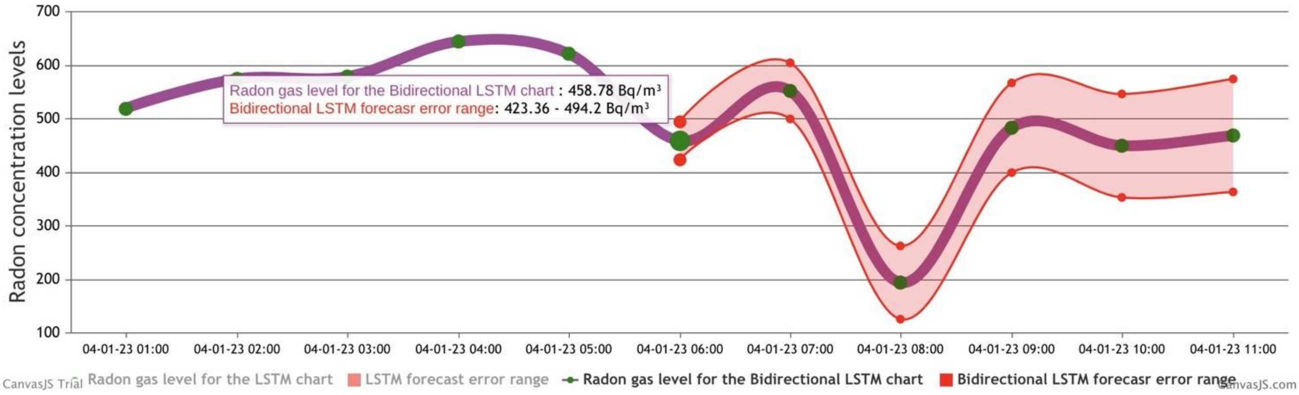

Fig. 3. Web App GUI. Forecasting example using the LSTM method for the next 6 hours, with mean absolute error identified in red.

TABLE II
NON-NORMALIZED BI-LSTM MEAN ABSOLUTE ERROR RESULTS

Forecasting Window	Lag 6	Lag 7	Lag 8	Lag 9	Lag 10	Lag 11	Lag 12	Training Duration
1 Hour	38 Bq/m3	36 Bq/m3	36 Bq/m3	36 Bq/m3	37 Bq/m3	35 Bq/m3	38 Bq/m3	6h 21min 24s
2 Hour	56 Bq/m3	54 Bq/m3	52 Bq/m3	57 Bq/m3	60 Bq/m3	72 Bq/m3	57 Bq/m3	6h 56min 1s
3 Hour	69 Bq/m3	68 Bq/m3	72 Bq/m3	71 Bq/m3	72 Bq/m3	73 Bq/m3	74 Bq/m3	5h 32min 10s
4 Hour	84 Bq/m3	85 Bq/m3	83 Bq/m3	85 Bq/m3	88 Bq/m3	90 Bq/m3	87 Bq/m3	6h 35min 35s
5 Hour	97 Bq/m3	99 Bq/m3	97 Bq/m3	98 Bq/m3	106 Bq/m3	100 Bq/m3	99 Bq/m3	4h 00min 15s
6 Hour	105 Bq/m3	107 Bq/m3	105 Bq/m3	110 Bq/m3	112 Bq/m3	111 Bq/m3	110 Bq/m3	4h 19min 15s

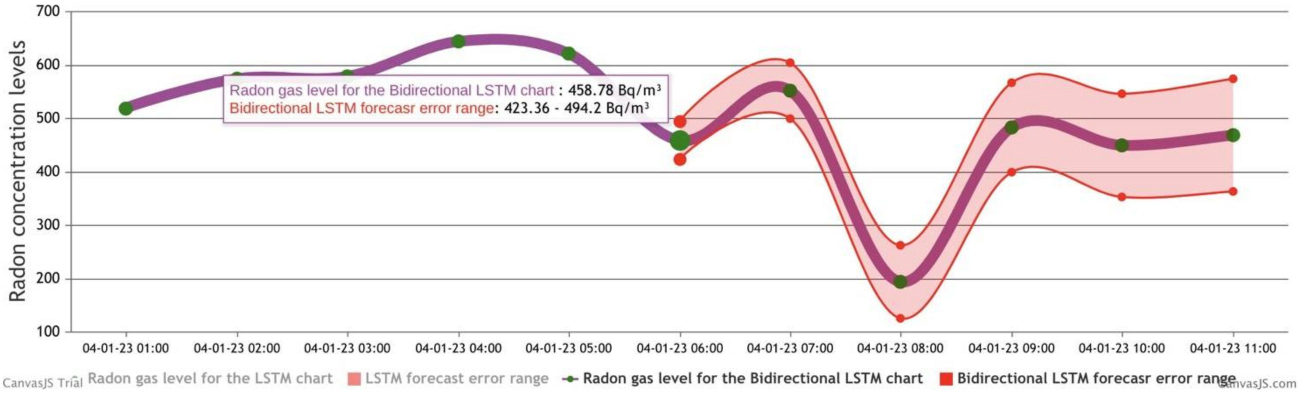

Fig. 4. Web App GUI. Forecasting example using the Bi-LSTM method for the next 6 hours, with mean absolute error identified in red.

2023 18th Iberian Conference on Information Systems and Technologies (CISTI)
20 – 23 June 2023, Aveiro, Portugal
ISBN: 978-989-33-4792-8

TABLE III
NORMALIZED LSTM MEAN ABSOLUTE ERROR RESULTS

Forecasting window	Lag 6	Lag 7	Lag 8	Lag 9	Lag 10	Lag 11	Lag 12	Training duration
1 Hour	29 Bq/m3	29 Bq/m3	33 Bq/m3	29 Bq/m3	29 Bq/m3	32 Bq/m3	28 Bq/m3	2h 13min 6s
2 Hour	44 Bq/m3	44 Bq/m3	47 Bq/m3	45 Bq/m3	44 Bq/m3	46 Bq/m3	43 Bq/m3	1h 5min 42s
3 Hour	58 Bq/m3	59 Bq/m3	60 Bq/m3	60 Bq/m3	59 Bq/m3	61 Bq/m3	59 Bq/m3	29min 6s
4 Hour	72 Bq/m3	73 Bq/m3	74 Bq/m3	74 Bq/m3	73 Bq/m3	74 Bq/m3	73 Bq/m3	31min 16s
5 Hour	85 Bq/m3	86 Bq/m3	86 Bq/m3	87 Bq/m3	84 Bq/m3	86 Bq/m3	85 Bq/m3	29min 1s
6 Hour	96 Bq/m3	97 Bq/m3	97	98 Bq/m3	94 Bq/m3	96 Bq/m3	95 Bq/m3	27min 9s

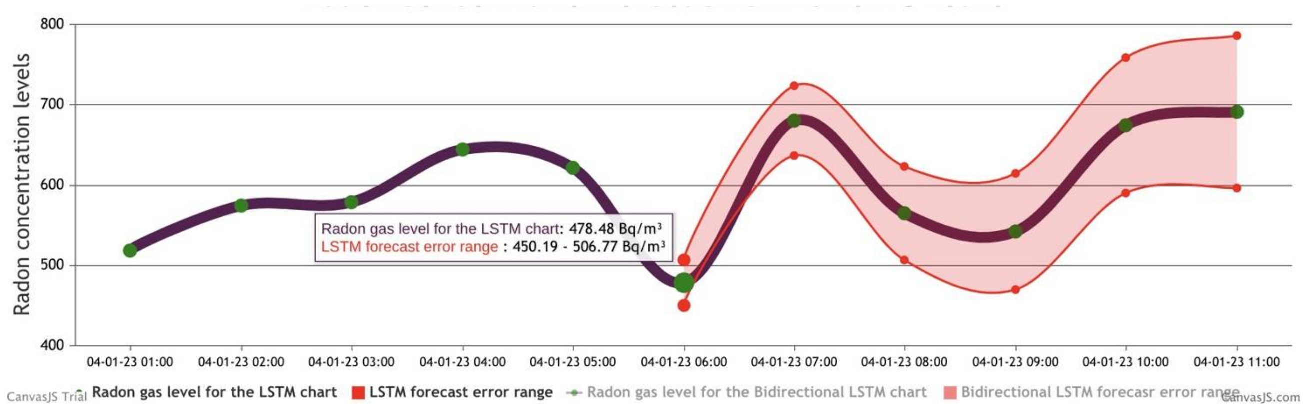

Fig. 5. Web App GUI. Forecasting example using the LSTM method for the next 6 hours, with mean absolute error identified in red.

TABLE IV
NORMALIZED BI-LSTM MEAN ABSOLUTE ERROR RESULTS

Forecasting window	Lag 6	Lag 7	Lag 8	Lag 9	Lag 10	Lag 11	Lag 12	Training duration
1 Hour	34 Bq/m3	36 Bq/m3	41 Bq/m3	32 Bq/m3	40 Bq/m3	32 Bq/m3	33 Bq/m3	2h 8min 33s
2 Hour	49 Bq/m3	51 Bq/m3	55 Bq/m3	48 Bq/m3	54 Bq/m3	47 Bq/m3	48 Bq/m3	49min 10s
3 Hour	64 Bq/m3	65 Bq/m3	69 Bq/m3	63 Bq/m3	68 Bq/m3	62 Bq/m3	62 Bq/m3	45min 16s
4 Hour	78 Bq/m3	79 Bq/m3	83 Bq/m3	77 Bq/m3	81 Bq/m3	76 Bq/m3	75 Bq/m3	38min 45s
5 Hour	90 Bq/m3	91 Bq/m3	96 Bq/m3	89 Bq/m3	92 Bq/m3	88 Bq/m3	86 Bq/m3	54min 44s
6 Hour	101 Bq/m3	102 Bq/m3	107 Bq/m3	99 Bq/m3	102 Bq/m3	99 Bq/m3	96 Bq/m3	38min 53s

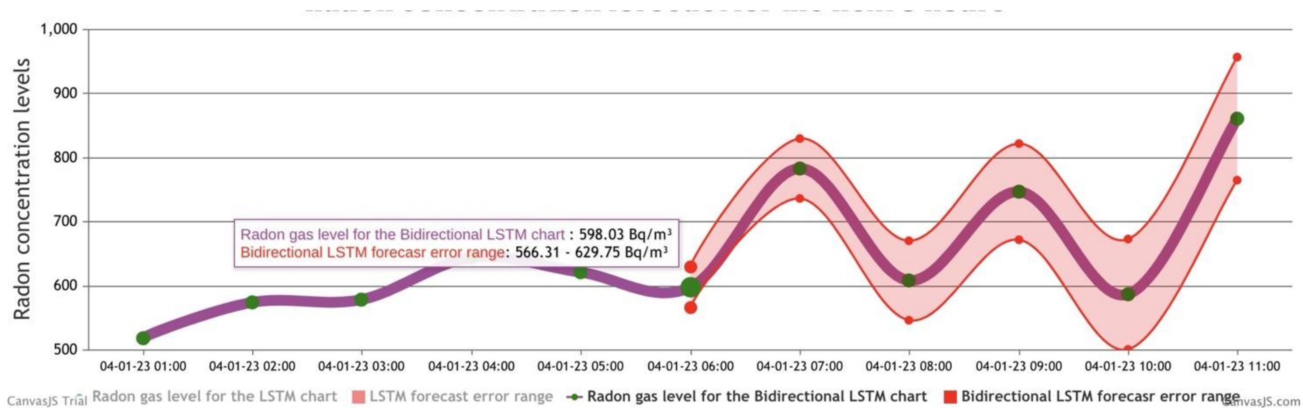

Fig. 6. Web App GUI. Forecasting example using the Bi-LSTM method for the next 6 hours, with mean absolute error identified in red.

ISBN: 978-989-33-4792-8

TABLE V
FORECAST PERFORMANCE COMPARISON TABLE

Date	LSTM	Bi-LSTM	LSTM	Bi-LSTM	Real Values
1 hour	27% [23%]	34% [29%]	29% [25%]	7% [7%]	643 Bq/m3
2 hour	43% [35%]	22% [12%]	1% [1%]	13% [13%]	687 Bq/m3
3 hour	38% [25%]	242% [71%]	19% [15%]	10% [9%]	672 Bq/m3
4 hour	38% [29%]	33% [24%]	21% [15%]	11% [17%]	639 Bq/m3
5 hour	38% [32%]	35% [37%]	6% [5%]	20% [19%]	714 Bq/m3
6 hour	110% [52%]	40% [33%]	2% [2%]	23% [22%]	703 Bq/m3

On the concept, the user may also switch between the algorithms, LSTM and Bi-LSTM to see the forecasts. Bi-LSTM is stronger and better than a LSTM because it can consider both past and future inputs when making predictions, while a LSTM can only consider past inputs, however, as shown on the tables containing the Mean Absolute Error, the LSTM slightly outperformed Bi-LSTM using normalized and non-normalized data. The results showed that normalizing the data can reduce errors, though they may not always decrease significantly. However, training these normalized models takes much longer than training non- normalized ones, so using normalized data is preferable. The only drawback of this approach is that it takes a little longer to normalize the data in order to make unitary predictions and normalize the prediction in order to obtain the normal scale. Table V presents the difference in percentage between the predicted values and the real radon concentration values, including the error margin variation represented in percentage within square brackets.

VI. CONCLUSIONS AND FUTURE WORK

Forecasting indoor radon levels may be used as a tool for mitigating human exposure risk, and thus help to effectively manage indoor radon risk. This work presents preliminary results regarding the implementation and evaluation of two LSTM-based Machine Learning (ML) approaches for indoor radon forecasting, which can then be used as a tool to trigger preventive management procedures for Indoor Air Quality improvement. LSTM captures long-term dependencies in sequential data, even when the dependencies are separated by many time steps. On the other hand, Bi-LSTM (which is a variation of LSTM) has two separate hidden states, one that processes the data in the forward direction (from the first sample to the last), and one that processes the data in the backward direction (from the last sample to the first). This allows Bi- LSTM to capture both forward and backward dependencies in the data, which can be useful in some applications where the order dependency of the data matters. However, Bi-LSTMs can be more powerful than regular LSTMs for tasks that require understanding of both forward and backward dependencies in the data, but at the cost of an increased computational increase and longer training times, when compared to LSTMs. The choice between LSTM and Bi-LSTM depends on the specificities of the task and the data at hand and should be guided by experimental evaluation and analysis.

ACKNOWLEDGMENT

This research is a result of the project TECH–Technology, Environment, Creativity and Health, Norte-01-0145-FEDER-000043, supported by Norte Portugal Regional Operational Program (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF).

REFERENCES

[1] L. J. R. Nunes, A. Curado, L. C. C. d. Graça, S. Soares, and S. I. Lopes, "Impacts of indoor radon on health: A comprehensive review on causes, assessment and remediation strategies," *Int'l Journal of Environmental Research and Public Health*, vol. 19, no. 7, 2022.
[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.
[3] A. A. Mir, F. V. C¸ Çelebi, H. Alsolai, S. A. Qureshi, M. Rafique, J. S. Alzahrani, H. Mahgoub, and M. A. Hamza, "Anomalies prediction in radon time series for earthquake likelihood using machine learning-based ensemble model," *IEEE Access*, vol.10, 2022.
[4] Q. Luo, Y. Chen, C. Gong, Y. Lu, Y. Cai, Y. Ying, and G. Liu, "Research on short-term air conditioning cooling load forecasting based on bidirectional LSTM," in *2022 4th Int'l Conf. on Intelligent Control, Measurement and Signal Processing*, pp. 507–511, 2022.
[5] L. Wenya, "Cooling, heating and electric load forecasting for integrated energy systems based on CNN-LSTM," in *2021 6th International Conference on Power and Renewable Energy (ICPRE)*, pp. 808–812, 2021.
[6] B. R. Krishna, M. H. Reddy, P. S. Vaishnavi, and S. V. Reddy, "Traffic flow forecast using time series analysis based on machine learning," in *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 943–947, 2022.
[7] K. Moharm, M. Eltahan, and E. Elsaadany, "Wind speed forecast using LSTM and Bi-LSTM algorithms Gabal El-Zayt Wind Farm" in *2020 International Conference on Smart Grids and Energy Systems (SGES)*, pp. 922–927, 2020.
[8] I. Bodur, E. Celik, and N. Ozturk, "A short-term load demand forecasting based on the method of LSTM," in *2021 10th International Conference on Renewable Energy Research and Application (ICRERA)*, pp. 171–174, 2021.
[9] T. Tang, D. Jiao, T. Chen, and G. Gui, "Medium- and long-term precipitation forecasting method based on data augmentation and machine learning algorithms," *IEEE Journal Selected Topics in Applied Earth Obs & Remote Sensing*, vol.15, pp.1000–1011, 2022.
[10] J. Lu, Q. Zhang, Z. Yang, and M. Tu, "A hybrid model based on convolutional neural network and long short-term memory for short-term load forecasting," in *2019 IEEE Power Energy Society General Meeting (PESGM)*, pp. 1–5, 2019.
[11] S. I. Lopes, P. M. Moreira, A. M. Cruz, P. Martins, F. Pereira, and A. Curado, "RnMonitor: a WebGIS-based Platform for Expedite in situ deployment of IoT Edge Devices and Effective Radon Risk Management," in *IEEE Int'l Smart Cities Conf.*, pp.451–457, 2019.
[12] L. Mendo, "Estimation of a probability with guaranteed normalized mean absolute error," *IEEE Communications Letters*, vol. 13, no. 11, pp. 817–819, 2009.

ISBN: 978-989-33-4792-8