Proceedings of The Canadian Society for Mechanical Engineering International Congress 2018
CSME International Congress 2018
May 27-30, 2018, Toronto, On, Canada

Quantitative Natural Gas Discrimination For Pipeline Leak Detection Through Time-Series Analysis of an MOS Sensor Response

Matthew Barriault, Mahyar Mohaghegh Montazeri, Allen O'Brien, Homayoun Najjaran, Mina Hoorfar
School of Engineering
University of British Columbia
Kelowna, Canada

Abstract—In order to detect natural gas pipeline leaks, ethane in the natural gas must be discriminated from background methane emissions. Our gas detection apparatus is well-suited for this application due to its flexibility and low cost. We present a comparison of machine learning models for quantitative estimation of concentrations of both methane and ethane in a target gas sample, using a response over time from a single sensor in our apparatus. We also demonstrate that the use of synthetic data is very effective for training a model to discriminate between methane and ethane.

Keywords-machine learning; gas detection; microfluidic; diffusion simulation; pipeline leak detection

I. INTRODUCTION

Natural gas pipeline leak detection can be broadly categorized into "internal" and "external" methods, depending on whether the detector resides inside or outside the pipeline. Internal methods include acoustic measurement, pressure/flow monitoring, and statistical analysis. These methods also often make use of one of several mathematical modeling options to predict when a leak has occurred. External methods are more hardware-based, relying on, for example, acoustic, optical, soil monitoring, or vapor sampling sensors [1]. These hardware solutions can either be permanently installed in a fixed location, or used in conjunction with a mobile monitoring apparatus, such as a handheld detector or even a drone. We present here the application of a small, cost-effective, and highly flexible gas sensing apparatus that can be used in either permanent or mobile applications.

Leak detection requires differentiation between methane and ethane, since natural gas will typically contain ~5% ethane and must be detected in the presence of background methane emission from, for example, nearby agriculture. However, the difference between the sensor's response to methane and ethane may not be immediately clear. We employ machine learning techniques to discover patterns that will enable this discrimination. To offset the requirements of some algorithms that a large dataset be provided, we also test the performance of our estimation models using synthetic data. Using simulations, we will be able to generate predictions of the sensor's responses to wide ranges of concentration, temperature, pressure and humidity. All of this can be used to train the pattern recognition system to be able to take into account the effect of these parameters and give more accurate results.

II. SENSING APPARATUS AND DATA COLLECTION

The sensing apparatus consists of a Figaro 2610 metal oxide semiconducting gas sensor embedded in a 3D-printed microchannel, which is coated with chromium, gold, and parylene-C to increase selectivity. Full fabrication details are given in [4]. This apparatus is quite flexible, and target gases can be altered simply by changing which MOS sensor is included. This is made easier by the fact that Figaro manufactures sensors with similar dimensions, but for a variety of target gases. Parallel work in our lab has demonstrated the suitability of this apparatus for nuisance sewer gas detection [2], wine identification [3], and breath analysis. Its small size enables it to be used for both stationary and mobile applications, which is extremely beneficial in the context of pipeline leak detection, where both types of devices may be needed, depending on the individual situation. Figure 1 displays a model of the microchannel sensing apparatus.

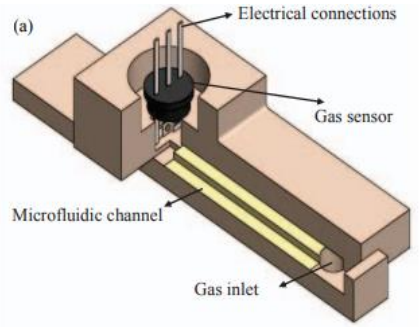

Figure 1. Gas sensing apparatus. Figure from [4]

Our dataset consists of time-series curves from our sensing apparatus in response to a variety of target gas concentrations. These targets are created through the use of a mass flow controller, and contain known concentrations of methane or ethane. We have not yet collected enough data with mixtures of both gases to make meaningful predictions, but this is the focus of ongoing work, primarily on using simulations to create synthetic mixture data to alleviate the time required to perform manual tests. Therefore, in the current work, the goal of the model is to not only determine which gas it has been exposed to, but also to estimate the concentration of that gas.

The time-series curves are generated by exposing the sensor to the target gas for 40 seconds, then allowing the sensor to recover by placing it in fresh air for 150 seconds. An example curve is shown in Figure 2. The exposure and recovery phases are clearly distinguishable. Some of the curves' features are also shown, which will be discussed in Section IV-A.

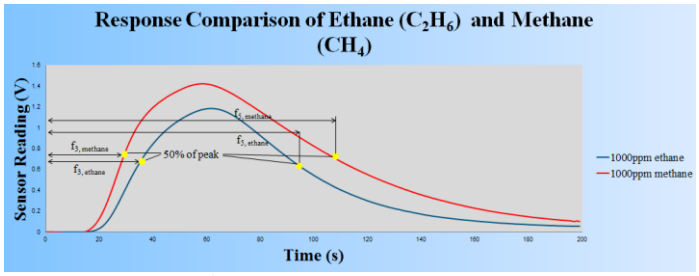

Figure 2. Comparison of responses to 1000ppm methane and 1000ppm ethane, with features #3 and #5 shown

III. SIMULATION METHODOLOGY

One of the main challenges of developing microfluidic-based gas detectors is calibration of the sensor, based on wide ranges of different parameters such as changes in the mixture concentration, temperature, pressure and humidity. This requires a vast number of experiments to generate enough data to be able to take into account the effect of each parameter accurately. Simulation of the gas sensor can help solve this problem as it reduces the number of experiments needed for calibrating the sensor, saves time, reduces human and instrument errors, and removes many limitations.

In the current study diffusion of a target gas inside the sensor's 3D printed micro-fluidic channel is studied. The simulation is done in a three dimensional model and the effect of gas adsorption to the channel walls is also applied to the simulation results.

This methodology does not take into account the individual differences between each sensor. To use the simulation data to train a model used to estimate real data, the simulations need to be tailored to account for the fact that the real sensor does not exactly match the theoretical model. However, we present here a proof of concept that synthetic data is well-suited to training a discriminative model such as those discussed in Section IV-B and IV-C. Such a calibration procedure that would allow the model to be trained on synthetic data and tested on real data is the focus of ongoing work.

The synthetic dataset was generated by simulating a target gas with concentrations from 100ppm to 1000ppm in increments of 100ppm, with five repeats for each. To make our predictions more robust to day-to-day variations in the sensor, and to make our data more realistic, we introduce some randomness in the simulations. The actual simulated target concentrations were sampled from Gaussian distributions with means equal to the ideal target concentrations, and standard deviations of 10ppm. Since this randomness is unknown to us in a real situation, the estimation targets for these tests are kept as the ideal targets i.e. 100ppm, 200ppm, etc. The simulated curves for different concentrations of methane are shown in Figure 3.

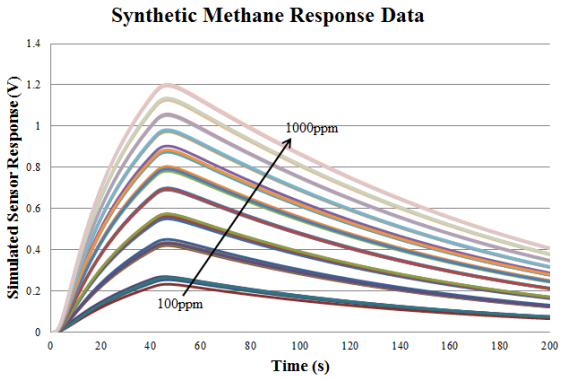

Figure 3. Simulation results for methane, with actual concentrations sampled from a Gaussian centered on the ideal concentration

A. Diffusion model

As in this study, diffusion of a mixture of gas into another is the governing transport phenomena. For this, the Maxwell-Stefan equation, which is an accurate model for multicomponent diffusion for low density gases, is chosen:

$$\nabla x_a = -\sum_{\beta=1}^{n} \frac{1}{CD_{a\beta}}\left(x_\beta N_a - x_a N_\beta\right), \qquad \alpha = 1,2,3,\dots,n \tag{1}$$

where x_i is the mole fraction, N_i the flux, C the total concentration and D_{ij} the diffusion coefficient of component i in component j. It worth mentioning that, in a binary system, the well-known Fick's law can also be used to simplify the model. [5]

B. Surface adsorption model

As the gas diffuses inside the micro-fluidic channel, some of the molecules adsorb to or desorb from the channel walls which affects the transport phenomena rate. In this simulation the adsorption is taken into account using the Langmuir adsorption model, which considers the phenomena an equilibrium reaction and provides the adsorption and desorption rate as:

$$r_{ad} = k_{ad} p_A [s] \tag{2}$$

$$r_d = k_d [A_{ad}] \tag{3}$$

Where k_{ad} and k_d are the adsorption (forward) and desorption (backward) reaction rates, p_A partial pressure of A, $[S]$ empty sites concentration and $[A_{ad}]$ is the concentration of compound A molecules adsorbed on the surface. [6]

C. Model assumptions and boundary conditions

In this model we assume that there is no flow, and the diffusion is the governing transport phenomena. Also, there is no diffusion of gas molecules to the bulk of channel walls and the adsorption is occurring only on surface. The simulation model is shown in Figure 4.

As the experimental test consists of two steps, exposure and recovery, the simulation is also designed in two different steps which have different initial and boundary conditions. In the exposure phase, there is no target gas inside the channel and it is filled with air. At time=0s the sample concentration is introduced as the boundary condition at the opening of the channel and during this step the diffusion happens for 40s, at which point the recovery phase starts. The initial concentration is derived from the last time point of the previous step and the boundary condition will be set to zero concentration of the target gas.

It is worth mentioning that many other model assumptions are inherited from the diffusion and the surface adsorption models which can be found in the previously mentioned references.

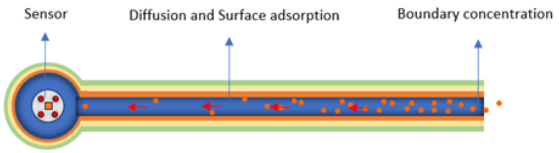

Figure 4. Diffusion simulation model

IV. CONCENTRATION ESTIMATION

In this work, we evaluate the performance of both feature-based and feature-less classifiers. In the case of a feature-based model, the input to the classifier is a set of pre-designed features that are extracted from each of the time-series curves. In other words, the input is a set of training vectors $x_i \in \mathbb{R}^{\boldsymbol{d}}, i = 1 \dots n$, where n is the number of samples in the training dataset, and d is the number of features extracted from each sample. For the feature-less models, the input is simply the raw data, so the training vectors are $x_i \in \mathbb{R}^{\boldsymbol{t}}, i = 1 \dots n$, where t is the length of each time-series sample. In both cases, the targets y are the known concentrations of both methane and ethane: $y_i \in \mathbb{R}^{\boldsymbol{2}}, i = 1 \dots n$. Feature-based methods have found extensive use with electronic noses, and a considerable effort has been made in past decades to design features that will produce good classification results [1]. However, it is desirable to use a classifier that does not require the hand-designing of such features. For this reason, we investigate the performance of a recurrent neural network, which can take the raw sensor data as input, without the need for any feature design.

Preliminary hyperparameter selection for each model (including model depth, width, and regularization parameters) was done using a common held-out test set consisting of 5% of the total data. Once hyperparameters were identified that maximized performance on the test set, the final performance of the model was evaluated by predicting the methane and ethane concentrations for each sample, using a leave-one-out method. Leave-one-out can be considered a special case of k-fold validation with $k = n$, where n is the total number of samples. This means that n models were trained, with one sample excluded from each, giving the best possible prediction for each sample. The leave-one-out method becomes impractical for even moderately-sized datasets, for which standard k-fold validation should be used instead, with k chosen such that the held-out test data in each case would be about 5% of the total dataset.

A. *Feature extraction*

In order to get an idea of which features will discriminate well between methane and ethane, we examine the comparison of the sensor's response to 1000ppm of each gas in Figure 2. The significant difference in speed of response, especially between 20 and 50 seconds, suggests that features such as the time at which the signal reaches 50% of its peak value might be useful. Along with good discrimination between gases, it is also important for the magnitudes of predicted concentrations to be accurate. For this reason, features such as the peak value and the area under the curve will also be useful because these features relate directly to the magnitude of the target gas concentration. Table 1 provides a full description of the features used.

Once the features have been extracted, they must be processed to ensure that the models can learn properly from them. This processing is to make the distributions of each of the features have a mean of zero, and a standard deviation of one. This is done so that one feature with a much larger magnitude than the others does not completely dwarf the contributions of the rest.

Table 1. Extracted features used for feature-based classifiers

Feature Number	Feature Description
1	Peak value
2	Area under the curve
3	Time to 50% of peak value (exposure phase)
4	Time to 75% of peak value (exposure phase)
5	Time to 50% of peak value (recovery phase)
6	Time to 75% of peak value (recovery phase)

B. *Feature-based models*

Many machine learning models perform poorly with time-series data if the entire time-series is naively given to the model to use as training/testing data, due to their difficulty in learning temporal relationships. Multi-layer perceptrons (MLPs) and support vector machines (SVMs) are two such models. Therefore, we evaluate the performance of these model using extracted features.

A multilayer perceptron is a type of feed-forward neural network consisting of at least three layers, all but the first of which apply a nonlinear transform to a weighted sum of the previous layer's activations. The first layer is called the input layer, and it is where the features are input. The last layer is called the output layer, and it is where the network's predictions appear. Any layers in between these two are called

hidden layers. Since a nonlinear transform is applied at every layer, the addition of more hidden layers means that the network can learn more complex functions [7]. The activation *z* of layer *j* is given by:

$$z_j = \sigma(W_{ij} z_i + b_j) \tag{4}$$

where z_j is the activation vector of the previous layer, b_j is the bias vector for layer *j*, W_{ij} is the weight matrix between layers *i* and *j*, and σ is a nonlinear transform function, usually either the logistic sigmoid or the hyperbolic tangent function.

MLPs are supervised models, meaning that for the network to predict the correct output values, it must be allowed to learn on a training dataset for which the correct outputs are already known. The goal of this learning is for the network's predictions to be as close to the true outputs as possible. This is accomplished by altering the network's weights and biases (W_{ij} and b_j for each layer) through a process known as backpropagation. It involves updating the weights and biases along a gradient that maximally decreases the error. The weights and biases are updated according to:

$$W_{ij,new} = W_{ij} - \delta \frac{\partial \varepsilon}{\partial W_{ij}} \tag{5}$$

$$b_{j,new} = b_j - \delta \frac{\partial \varepsilon}{\partial b_j} \tag{6}$$

where δ is a constant called the learning rate, and ε is the error function, such as the mean-squared loss function. The learning rate controls how large the weight and bias updates are. If it is too small, the network will train slowly, but if it is too large, the algorithm may not converge.

In our case, the network must have five input units, and two output units, since we are using five features to predict two concentrations. Preliminary testing indicates that three hidden layers with 50 units each performed best on our data.

A support vector machine is a binary classification model that can construct a very complex classification surface through the use of a kernel function. They are based around the idea of achieving the maximum margin separation between classes. They achieve non-linear classification by mapping the inputs into a higher-dimensional space, where linear classification may be able to be accomplished. This idea can also be extended to regression analysis by fitting a regression hyperplane to the training cases. This allows for a highly non-linear regression surface. In our case, we actually have to fit two SVMs, one for each concentration target, since the standard SVM is applicable only to single targets [11].

C. Feature-less models

Recurrent neural networks (RNNs) differ from traditional neural networks in the fact that they incorporate memory. Each new data point in a time-series that is given to an RNN will produce not only an output, but also an update to the network's internal memory state. At each time step, the network's hidden units see not only the input data, but also the memory state. Using the training data, the network will learn how best to use this memory state throughout the duration of a single time series data vector [8]. A simple recurrent neural network is shown in Figure 5.

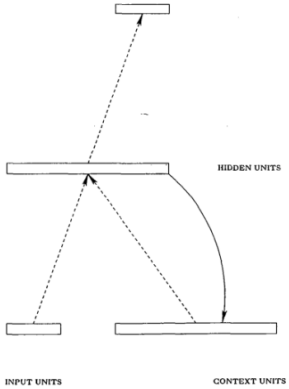

Figure 5. Simple recurrent neural network, with connections from the hidden units back to themselves. Figure from [8]

The hidden layer outputs for an RNN are similar to those of an MLP, except that the outputs are now indexed by time due to the fact that the output changes for every time point in each series. Also, there is an extra term in the expression, since the hidden layer's output at the previous time point influences the output at the current point:

$$z_{j,t} = \sigma(W_{ij} z_{i,t} + W_h z_{j,t-1} + b_j). \tag{7}$$

In the above equation, W_h represents the weight matrix from the previous hidden activation to the current hidden activation.

V. RESULTS

A. Real data prediction

As mentioned in Section IV, we are using a leave-one-out validation method for determining the quality of our models. Our quality metric is the mean-squared prediction error over our entire dataset. The error ε is given by:

$$\varepsilon = \sum_{i=0}^{n} \sum_{j=0}^{d} (y_{ij} - y_{pred,ij})^2 \tag{8}$$

where y_{ij} refers to the j^{th} concentration target for the i^{th} data sample, $y_{pred,ij}$ is the network's prediction of y_{ij}, *n* is the number of samples, and *d* is the number of concentration targets per sample. In our case, *d* is equal to two, since we are predicting methane and ethane. The error rates for the real data are presented in Table 2.

Table 2. Mean-squared error comparison for the models studied

Model	Mean-squared Error	Mean Error (ppm)
MLP	1,845	43.0
SVM	3,079	55.5
RNN (1 layer)	22,781	150.9
RNN (4 layers)	8,276	91.0

While the RNN requires less effort in feature engineering, it did not perform as well as the more traditional feature-based models. This can be attributed to the fact that the RNN's performance increases significantly with a deep model (four layers vs one), and deep models tend to require much more training data than shallow models [9]. This makes the possibility of using synthetic data particularly attractive.

B. *Synthetic data prediction*

For the tests with synthetic data, there was no additional model parameter tuning; the same values were used on both the real and synthetic datasets. However, since we have 100 simulated examples, k-fold validation with k=20 was used instead of the leave-one-out method. This significantly reduced the training time. The error rates for the synthetic dataset are presented in Table 3.

Table 3. Mean-squared error rates for the sythetic dataset

Model	**Mean-squared Error**	**Mean Error (ppm)**
MLP	74	8.6
SVM	288	17.0
RNN (1 layer)	4,120	64.2
RNN (4 layers)	6,025	77.6

As predicted, all of the models saw an improvement when using the synthetic dataset instead of the real data. This is likely due to the fact that the synthetic data is cleaner and contains less variance than the real data. In future tests, the randomness added to the synthetic data should be increased to match the distribution of the real data.

The deep RNN did not see as much of an improvement as the other models, which might mean that it requires additional training time since more data samples were used.

VI. CONCLUSION

Even though the RNNs failed to outperform the feature-based methods, the overall results show that the quantification of methane and ethane with a single MOS sensor is very feasible, even when the identity of the target gas is unknown. Once a large volume of simulation data has been accumulated, the application of the models described in this work, along with the integration of our sensing apparatus with a stationary or mobile platform will be viable, low cost, and non-invasive method of detecting natural gas pipeline leaks.

VII. FUTURE WORK

The estimation results presented here have been limited to the discrimination between two different gases, and concentration estimation of the target gas. Future work will include extension to estimating the concentration of both methane and ethane in an arbitrary mixture. To accomplish this, we anticipate the need for more information than a single time-series. For example, multiple different sensors can be used with one microchannel, and ideally one of them would be more sensitive to either methane or ethane than the other. Electronic noses typically do contain more than one type of sensor, but since methane and ethane are so similar, the approach of adding different types of sensors will not likely make a big improvement. The more promising approach is to use multiple identical sensors, each with a microchannel of a different length. This will accentuate the differences in diffusion between the two gases, rather than the difference in how they affect the sensor.

Future work will also include the extension of the algorithms described here to other projects in our lab, as described above [2][3]. In addition, we will be investigating a variant of the RNN called the long short-term memory (LSTM) network, which tends to learn longer-term dependencies better than regular RNNs [10].

REFERENCES

[1] J. Yan et al., "Electronic nose feature extraction methods: a review," *Sensors*, vol. 15, pp. 27804-27831, 2015.

[2] M. M. Montazeri et al., "A sensor for nuisance sewer gas monitoring," in *IEEE Sensors*, Glasgow, 2017, pp. 1-3.

[3] M. Paknahad, A. Ahmadi, J. Rousseau, H. R. Nejad, and M. Hoorfar, "On-Chip Electronic Nose For Wine Tasting: A Digital Microfluidic Approach," *IEEE Sensors*, vol. 17, no. 14, pp. 4322-4329, July 2017.

[4] M. Paknahad, J. S. Bachhal, A Ahmadi, and M. Hoorfar, "Characterization of channel coating and dimensions of microfluidic-based gas detectors," *Sensors and Actuators B: Chemical*, vol. 241, pp. 55-64, 2017.

[5] R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*.: John Wiley & Sons, 2007.

[6] I. Langmuir, "The adsorption of gases on plane surfaces of glass, mica and platinum," *Journal of the American Chemical society*, vol. 40, no. 9, pp. 1361-1403, 1918.

[7] D. W Ruck, S. K Rogers, and M. Kabrinsky, "Feature Selection Using a Multilayer Perceptron," *Journal of Neural Network Computing*, vol. 2, no. 2, pp. 40-48, 1990.

[8] J. L. Elman, "Finding structure in time," *Cognitive Science*, pp. 179-211, 1990.

[9] Z. Zhou and J Feng, "Deep Forest: Towards an Alternative to Deep Neural Networks," in *International Joint Conference on Artificial Intelligence*, Melbourne, 2017.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735-1780, 1997.

[11] S. R. Gunn, "Support vector machines for classification and regression," University of Southampton, Technical 1998.