

# Leakage Source Localisation Employing 3D-CFD Simulations and Gated Recurrent Units in Hydrogen Releases

André Z. Selvaggio\*, Sávio S.V. Vianna

University of Campinas, School of Chemical Engineering  
a116138@dac.unicamp.br

Gas leakages present significant explosion hazards in industrial settings, potentially leading to substantial human casualties, environmental damage, and economic losses in accident scenarios. Rapid detection and location of leak sources are crucial for timely corrective maintenance, mitigating the most hazardous situations. This study focuses on developing gated recurrent units (GRU) to identify hydrogen leaks in a chemical process module due to hydrogen release. Training and test databases were generated using 3D-CFD simulations for four leak scenarios and a non-leakage scenario. The inputs consisted of concentration profiles from eleven sensors, considering four leak sources, four wind speeds, and eight wind directions across various temporal lengths. The results demonstrated improved performance with higher input timesteps, achieving an accuracy of over 86.75% for unseen data. This indicates the strong generalisation of the model capabilities and potential for predicting leaks using easily obtainable inputs.

## 1. Introduction

Leaks of hazardous materials are frequent occurrences in most industrial facilities, and their containment failures can result in undesirable consequences. Continuous monitoring of the potential source of leakage enables prompt response in emergencies, mitigating the overall risk of fire and explosions. Consequently, numerous strategies have been investigated over the past few years to provide rapid and accurate predictions of the release location.

Over the past few years, artificial intelligence (AI) has been utilised to develop prediction models for locating the source of leaks. One specific AI technique that has been successfully employed to predict leakage sources is recurrent neural networks (RNN), a set of deep learning algorithms that feature loop connections to model sequential dependencies. This approach significantly enhances the performance of neural networks in problems that involve time dependence (Patterson, J. and Gibson, A., 2017). Among RNN algorithms, the long short-term memory (LSTM) networks have garnered significant attention for their ability to capture long-term dependencies in sequential data. Kim et al. (2019) employed LSTM models to identify the five most probable leak sources within a chemical site. The training and test data were acquired through two-dimensional computational fluid dynamics (CFD) simulations. The input parameters were the wind velocity, wind direction, and one-minute-long concentration profiles from eleven sensors. The output consisted of the leakage probability of forty potential sources and a non-leakage scenario. The findings demonstrated exceptional performance, with test results achieving an accuracy of over 97%. However, the authors employed a predetermined temporal duration and failed to investigate its impact. No regularisation techniques were employed, resulting in overfitting of the final models. Furthermore, the influence of geometry was disregarded in the 2D-CFD simulations.

Similarly, Selvaggio et al. (2022) developed LSTM models to identify gas leaks within a chemical module. The training and test datasets were generated through 3D-CFD simulations that incorporated varying wind velocities and directions, four relatively close leak sources, and a non-leak scenario. The authors employed regularisation techniques to mitigate the overfitting issue, and the input vectors were of variable length. The results demonstrated high generalisation capabilities for unseen data and achieved an accuracy of up to 96.30%.

One limitation of LSTM networks is the number of gates and the parameters that need to be optimised. In this context, gated recurrent units (GRU) have a similar structure, but their gates perform distinct functions, reducing

dimensionality compared to LSTM (Aggarwal, C.C., 2019). This algorithm has shown interesting results for multiple different classification problems such as malware identification (Athiwaratkun, B. and Stokes, J.W., 2017), spectral-spatial classification on hyperspectral images (Pan, E. et al., 2020), cardiac diagnosis using biometric electrocardiogram (Lynn, H.M. et al., 2019) and emotion recognition employing noisy speech (Rana, R., 2016).

The gated recurrent unit (GRU) is a recursive convolutional neural network proposed by Cho, K. et al. (2014) in an encoder-decoder architecture to address the variable-length sequence problem in machine translation. This approach recursively employs all network parameters in the input sequence to generate an output vector. While initially developed for natural language processing, GRU can effectively handle any sequential problem by leveraging recursive calculations to store long-term and short-term information, enhancing their performance.

The GRU networks comprised two pivotal structures, termed update and reset gates, responsible for regulating the flow of information within the model. The update gate determined the data that should be transmitted to the subsequent state, while the reset gate determined the information that should be discarded from the previous state. This structure could store both long-term and short-term information while selectively filtering extraneous information, mitigating the vanishing and exploding gradients challenges that were commonly encountered during the optimisation phase in basic recurrent neural networks (Salem, F.M., 2022).

This work aimed to design GRU capable of accurately detecting gas leaks utilising readily accessible variables within chemical facilities. To achieve this, neural networks were trained using a database generated through 3D-CFD simulations. The module geometry and wind velocity field were incorporated into the model.

## 2. Methodology

This study employs a three-stage approach. First, computational fluid dynamics (CFD) modelling is used to select the dispersion cases for numerical experimentation. Second, the data acquisition process is detailed, including post-processing techniques. Third, an in-depth analysis of the GRU-ANN network employed in this research is provided.

### 2.1 Computational Fluid Dynamics (CFD)

The computational simulations employed FLACS (Flame ACceleration Simulator), a CFD solver developed by Gexcon, to generate the database. The module's geometry comprised two 36x12 m plate decks with a height of 26 m. Eight wind directions, four wind velocities, and four leakage sources were utilised to simulate gas dispersion within the module. Eleven locations were identified through optimisation using the Set Covering Problem to monitor the profile concentration. For a comprehensive description of the CFD simulations and their results, readers are referred to the corresponding CFD sections in Selvaggio et al. (2022).

The geometry, depicted in Figure 1, was designed based on a conventional Floating Production Storage and Offloading (FPSO) module. The mesh generation adhered to the most stringent guidelines for CFD solvers, employing the porosity-distributed resistance (PDR) approach for parameterisation. A grid line was fixed at each deck level. Within the module, the minimum grid cell size was determined by the leak area. The mesh was expanded by a 20% factor up to 0.5 meters and maintained at a constant resolution until the geometry's boundary.

The FLACS-CFD solver employs the finite volume method (FVM) to address transport equations on a structured Cartesian grid (Gexcon A.S., 2020). To solve the Reynolds-averaged Navier-Stokes (RANS) equations, the standard k-epsilon model for turbulence is utilised (Lauder, B.E. and Spalding, D.B., 1983). The FLACS solver exhibits accuracy of second order in space and first/second order in time, employing the SIMPLE pressure correction scheme (Patankar, S.V., 2018), which is further augmented with source terms for compression work in the enthalpy equation for compressible flows (Gexcon A.S., 2020). Due to the significance of geometrical details in gas dispersion, particularly within offshore environments, CFD codes that utilise PDR solvers are considered the more suitable option.

The gas dispersion scenarios were evaluated for four leak locations (Figure 1a and Figure 1c), eight wind directions and four wind velocities (2, 4, 6 and 8 m/s). The leak rate was set to 25 kg/s. In the transient simulations, the initial 15 seconds were designated for wind build-up, followed by a 160-second gas leak and an additional 40 seconds for diluting the concentrated plume. To facilitate the training of the GRU-RNN model, 32 CFD simulations were conducted with no leak, considering only ventilation (eight wind directions and four wind speeds). In total, the dataset included 160 CFD simulations. Each of the 11 monitor points (MPs), as depicted in Figures 1b and 1d, reads the dynamic methane concentration. The transient gas concentration, wind speed, and wind direction serve as input for the GRU model.

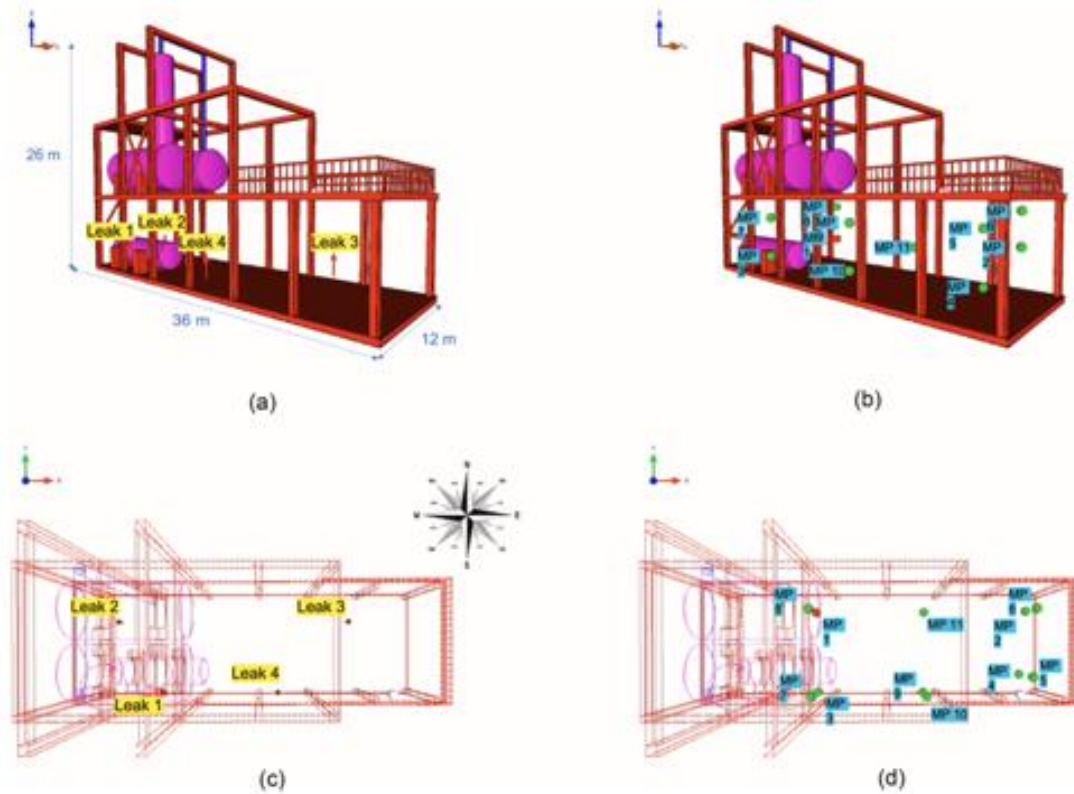


Figure 1: Perspective view for (a) Leak locations and (b) Monitoring Points (MP) at module geometry. x-y view for (c) Leak Locations with the wind rose for reference and (d) for Monitoring Points.

The leak was defined as Jet, which determines the area and subsonic velocity aftershock of a jet emerging from a high-pressure reservoir (stagnation point) (Gexcon A.S., 2020). Velocity was specified at the wind boundary condition, while outlet boundaries were used to model the nozzles. Regarding the numerical approach, the Courant-Friedrich-Levy number based on sound velocity (CFLC) and fluid flow velocity (CFLV) are set to 40 and 4, respectively, to optimise stability and grid refinement for the dispersion and ventilation simulations.

## 2.2 Data pre-processing

3D-CFD simulations were conducted to generate 160 leak scenarios encompassing five possible outcomes: four different leakage sources and a no-leakage condition. Each scenario involved a three-minute simulation with data collected every five seconds. The dataset included concentration profiles from eleven sensors, wind direction, wind velocity, and leakage source labels. Before training the GRU model, the variables were categorized into numerical (concentration and wind velocity) and categorical (wind direction and leakage source label) types and pre-processed accordingly. To prevent model bias due to varying magnitudes among input variables, the numerical variables were normalised to a range between 0 and 1. The categorical variables were transformed into binary vectors, resulting in eight new inputs from wind direction and five from leakage source labels. The number of time steps used determined the dimensionality of the input datasets. The scenarios were randomly split into training and testing subsets, allocating 80% to training and 20% to testing.

## 2.3 Optimisation of GRU models

The GRU were developed using Python version 3.7.7, TensorFlow 1.14.0, and Keras 2.3.1. The Adam optimisation algorithm was employed to train the model, utilising the default parameter settings specified in the Keras documentation. The categorical cross-entropy loss was used as the cost function for minimization. EarlyStopping was employed to avoid overfitting during the training stage. Validation data comprised 20% of the training dataset, and training was set to stop after thirty consecutive epochs without a minimum improvement of 0.01. In addition, ReduceLRonPlateau was used to reduce the learning rate by a factor of 0.5 when the validation loss plateaued for five epochs, with a minimum learning rate of  $10^{-6}$  (Prechelt, L., 1998). Consequently, the relatively small value of patience selected is satisfactory. Both callbacks (ReduceLRonPlateau and EarlyStopping) play a significant role in optimisation, helping to handle model training and convergence to a better solution efficiently.

Hyperparameter tuning was conducted using the Keras Tuner with RandomSearch, testing a wide range of configurations, including batch size, learning rate, dropout rate, and the number of neurons in the GRU layer. The hyperparameter search involved 50 trials, and the tuning process was aimed at finding the optimal architecture for maximising validation accuracy. The values and ranges for each hyperparameter are listed in Table 1.

*Table 1: Hyperparameters for GRU model*

Hyperparameter	Value/Description
Look Back	11 (Number of timesteps used for prediction)
Input variables	wind speed, wind direction, and H <sub>2</sub> concentration data from the 11 sensors.
Number of output variables	leakage source label
Fraction of dataset used in the training stage	0.8
Number of epochs	100
Batch size	64
Validation split percentage	0.2
Patience	15 (Number of epochs for early stopping)
SpatialDropout1D	0.3
GRU units	30 to 50 (Selected using Keras Tuner)
Kernel Initialiser	he_normal, glorot_normal, he_uniform, glorot_uniform (Chosen by Keras Tuner)
Kernel Regularisation	L1: 1e-5 to 1e-3 (Log-sampled) L2: 1e-3 to 1e-2 (Log-sampled)
Dropout Rate	0.2

The neural network was trained on a system with dual-socket Intel Xeon Silver 4214R CPUs (48 cores in total), 125 GiB of RAM, and running Ubuntu 22.04.1, utilising a x86\_64 architecture with VT-x virtualisation support and benefits from multiple levels of CPU cache (L1, L2, and L3) and NUMA optimisation to efficiently handle the demands of deep learning training. The hyperparameter search took 3505 seconds.

### 3. Results and discussions

The results indicated that GRU exhibited better performance with larger timestep values, possibly due to the more complex sequences of processed data during the optimisation stage. The recurrent nature of data processing in GRU enhances their learning capability, enabling the construction of models with higher levels of abstraction. Although the input set increases with the timestep value, the number of parameters—specifically synaptic weights and biases—remains unchanged for GRU with the same number of neurons in the output layers of their gates. The parameters for the optimal GRU model are presented in Table 2.

*Table 2: Best model parameters and results*

Metric	Value
Number of Neurons	30
L1 Regularisation	0.000569
L2 Regularisation	0.001796
Initialisation Method	he uniform
Dropout Rate	0.2
Training Log Loss	0.254
Training Accuracy	94.77%
Test Log Loss	0.401
Test Accuracy	86.75%
Number of Parameters	4955

The model was trained for 61 epochs before triggering early stopping and ceasing the training. The optimised parameters correspond to those obtained at the local minimum found at epoch 30 with a validation loss valued at approximately 0.42. Early stopping effectively prevented overfitting, ensuring that the GRU maintained good

generalisation properties while also reducing processing time. The relatively rapid convergence can be attributed to both the structure of the GRU networks and the relatively small dataset size.

Figure 2a illustrates the confusion matrix using the training dataset, while Figure 2b illustrates the results for the test dataset. Although the output of the GRU is a vector representing the probability of each leak source and the no-leakage case, the highest value was considered as the predicted class to generate the confusion matrix. The confusion matrices for both training and test datasets highlight the model's ability to correctly identify leakage sources, with minor misclassifications primarily occurring between Source 2 and Source 3.

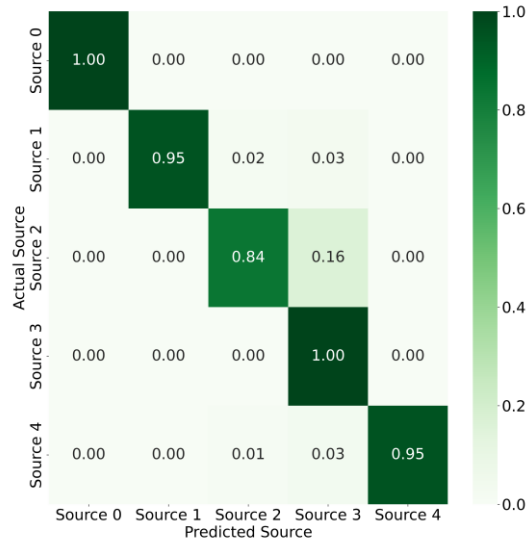


Figure 2a: Confusion matrix (training dataset).

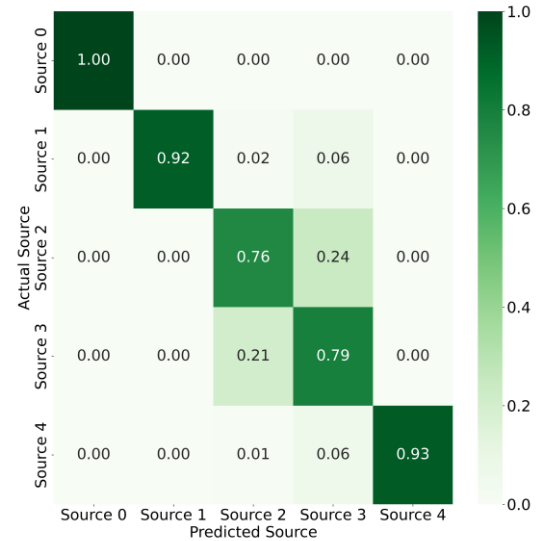


Figure 2b: Confusion matrix (test dataset).

The results of the training process are depicted in Figure 3, showing a steady decrease in both training and validation loss, indicating a successful optimization without signs of overfitting.



Figure 3: Best model training and validation losses.

The findings indicated precise prediction for leak sources 1, 3, and the no-leakage scenario for both datasets. The no-leakage scenario exhibited perfect classification, as expected, due to the regular input set characterised primarily by zero concentrations with only velocity and direction variables being non-zero. Misclassifications were observed for sources 2 and 4, with errors of 16% and 7% respectively in the training set, and 24% and 21% in the test set. These misclassifications are likely caused by similar concentration profiles between leak sources, especially given the proximity of leak points in the small simulated model. Similar results were observed by Kim et al. (2019), who noted that most misclassifications occurred for sources within a 10 m radius.

The performance of the GRU models remained comparable to that of the LSTM networks developed in the previous study by Selvaggio et al. (2022). Specifically, the best GRU model achieved an accuracy of

approximately 86.75%, whereas the best LSTM model from prior work had an accuracy of 96.30%. However, the GRU model had significantly fewer parameters—approximately 15 times fewer than the LSTM with comparable accuracy. This represented a considerable reduction in processing time and computational resource requirements, making the GRU a more efficient alternative in practical applications.

#### 4. Conclusions

GRU were trained to identify potential gas source leakages within a chemical processing module. The selected input variables for the location model are commonly available in chemical plants, including wind speed, wind direction, and hydrogen concentration readings from gas detectors positioned at fixed locations.

The training database was generated using three-dimensional computational fluid dynamics simulations, employing a single leak rate across four different leak locations subjected to four wind speeds and eight wind directions. Hydrogen gas sensors installed within the module tracked the concentration profiles, revealing that the flammable cloud varied considerably depending on the leakage location and wind conditions. To account for non-leakage scenarios, null concentration profiles were also included in the training dataset.

The accuracy of the best GRU model was 94.77% for the training dataset, indicating good weight optimisation, and early stopping avoided overfitting. The model's performance for the test dataset showed an accuracy of 86.75%, implying satisfactory generalisation ability. These findings indicate that GRU can serve as a low-cost auxiliary tool for rapid leak localisation using readily available data, facilitating prompt maintenance and enhancing safety measures.

#### Acknowledgments

The authors would like to thank the Coordination for the Improvement of Higher Education Personnel (CAPES) grant number 88887.674394/2022-00. Thanks are also due to Shell Brasil.

#### References

- Aggarwal, C.C., 2019. *Neural Networks and Deep Learning*. Springer Nature, Cham, Switzerland
- Athiwaratkun, B., Stokes, J.W., 2017. Malware classification with LSTM and GRU language models and a character-level CNN, In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE.10.1109/icassp.2017.7952603.
- Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches.10.48550/ARXIV.1409.1259.
- Gexcon A.S., 2020. FLACS–CFD v20.1 user's manual.
- Kim, H., Park, M., Kim, C.W., Shin, D., 2019. Source localization for hazardous material release in an outdoor chemical plant via a combination of LSTM-RNN and CFD simulation. *Comput. Chem. Eng.* 125, 476–489. <https://doi.org/10.1016/j.compchemeng.2019.03.012>.
- Lauder, B.E., Spalding, D.B., 1983. *The numerical computation of turbulent flows*.
- Lynn, H.M., Pan, S.B., Kim, P., 2019. A deep bidirectional GRU network model for biometric electrocardiogram classification based on recurrent neural networks. *IEEE Access* 7, 145395–145405. <https://doi.org/10.1109/access.2019.2939947>.
- Pan, E., Mei, X., Wang, Q., Ma, Y., Ma, J., 2020. Spectral-spatial classification for hyperspectral image based on a single GRU. *Neurocomputing* 387, 150–160. <https://doi.org/10.1016/j.neucom.2020.01.029>.
- Patankar, S.V., 2018. *Numerical Heat Transfer and Fluid Flow*. CRC Press.
- Patterson, J., Gibson, A., 2017. *Deep Learning*. O'Reilly Media, Sebastopol, CA.
- Prechelt, L., 1998. Early stopping-but when? *Neural Networks: Tricks of the Trade*. Springer, pp. 55–69.
- Rana, R., 2016. Gated recurrent unit (GRU) for emotion classification from noisy speech.10.48550/ARXIV.1612.07778.
- Salem, F.M., 2022. *Recurrent Neural Networks*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-89929-5>.
- Selvaggio, A.Z., Sousa, F.M.M., da Silva, F.V., Vianna, S.S., 2022. Application of long short-term memory recurrent neural networks for localisation of leak source using 3d computational fluid dynamics. *Process Saf. Environ. Prot.* 159, 757–767. <https://doi.org/10.1016/j.psep.2022.01.021>.