# Application of machine learning techniques to predict fire development in an ISO 9705 room

**Lareb Nasar[1,2], Harriet Dixon[1,2], Georgios Drosopoulos[2], Eleni Asimakopoulou[2, *]**

[1] London Fire Brigade, 169 Union St, London SE1 0LL
[2] School of Engineering, University of Central Lancashire, Fylde Road, Preston, PR1 2HE, UK
[*] EAsimakopoulou@uclan.ac.uk

**Abstract.** Machine learning, a subset of artificial intelligence, shows potential for enhancing computational fire modelling compared to traditional methods such as computational fluid dynamics. This study explored using artificial neural networks to predict heptane fire development within a compartment, varying heat release rates from 100 to 3000 kW and ventilation areas from 0.16 to 4.8 m$^2$. Artificial neural networks (ANNs) were trained using computational data from an ISO 9705 room. Network optimisation involved adjusting training-to-validation ratios and fine-tuning hidden layer neuron counts. Results indicate optimised ANNs achieved less than 7% error for heat release rate predictions and 1.5% for ventilation size predictions, with a notable computational cost reduction exceeding $10^4$-fold. These findings suggest a promising future for integrating machine learning into fire engineering, significantly reducing analysis time therefore fostering safety improvements and innovation in the field.

## 1. Introduction

This work investigates a crucial part of the world of fire engineering, where understanding and predicting fire behaviour is crucial. Given the nature of the discipline, experimental data is often limited, resulting in alternative tools being commonly used to predict fire behaviour. Among these tools, computational fluid dynamics (CFD) stands out as the most widely used approach in both industry and academia, but it is not without limitations, namely high computational cost. Therefore, it is crucial to explore new tools and how they can be used to carry out modelling more effectively.

Machine learning, and in particular the use of artificial neural networks (ANNs), has been a fast-growing area of research to be applied to numerous fields. The ANN is inspired by the way the human brain works, which is by having neurons/nodes which are connected to each other by some constraints [1]. Both the number of neurons and the nature of the connections are vital in the overall set up of the network. The architecture of a neural network can consist of three main layers which are the input layer, the hidden layer, and the output layer (see Figure 1.1). The input layer is the feature being fed into the network, quite often this is the independent variable, and the output is what the network is being trained to predict (dependent variable). The hidden layer(s) is used to map the input layer to the output layer. The layers are made up of elements known as neurons. The number of neurons in the hidden layer changes the way the input maps to the output and can result in an underfitted or overfitted model, making it a key parameter to optimise during network training. Deep learning features networks with multiple hidden layers; whereas, shallow neural networks (SNN) have only one [1].

A study carried out by Hodges [2] focused on how machine learning and ANNs could be used to predict parameters when looking into wildland fires. The dataset used for the neural network included 10,000 wildland fire spread simulations. The network was created to produce an estimate for the standard heat flux, and it was found that the predictions had a 10% error compared to the simulated values for 95% of scenarios. Overall, it demonstrated a reduction in computational time by the order of

$10^2 - 10^5$. Further to this, other researchers [3] reviewed several machine learning techniques including deep learning with neural networks and found that predictions were 2-3 times faster than CFD simulations. The variation in the reduction in computational time is a result of how the model is built and what it's predicting. This review also discussed that further work would be required to improve techniques and increase accuracy. Research on machine learning applications to compartment fires to obtain fire prediction has been further conducted, using transpose convolutional neural networks (TCNN) [4]. A TCNN is a specialised type of neural network architecture that is predominantly used for tasks involving grid-like structures such as images or spatial data. They differ from ANNs in many ways, with TCNNs designed to be used for image generation, whereas ANNs can be used to produce data and text. Recently, a method for forecasting temperatures and velocities within an enclosed space by utilising a TCNN based on zone fire modelling has been presented [4]. The TCNN model was also trained and validated with an extensive set of results from 1,333 CFD simulations, each characterised by varying fire attributes, compartment configurations, and ventilation layouts. In two compartment scenarios, the TCNN demonstrated its predictive capability by achieving temperature and velocity estimates that closely aligned with CFD predictions, deviating by ±17.2% and ±0.30 m/s, respectively. The model's robustness extended to more complex multi-compartment situations, where discrepancies with CFD predictions were confined within a range of ±11% for temperatures and ±0.25 m/s for velocities. The need for experimental measurements and high-resolution CFD to produce a model which is more reflective of real circumstances and the complexities of actual fire behaviour has been highlighted.

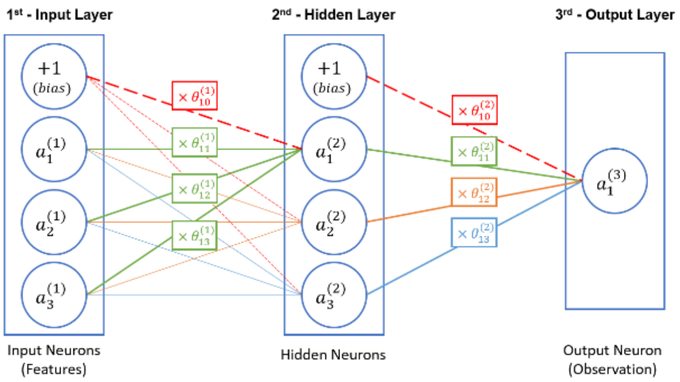


Figure 1.1: ANN's general architecture for a shallow neural network.

In summary, it is evident that machine learning has huge prospects to be beneficial and become a widely used tool within industry. There is a lot to gain from exploring ANN modelling and how this can be used to predict actual values as opposed to producing an image, specifically in the context of building/compartment fires. The suggested method is a step towards implementing digital twins, where for each physical system, its digital counterpart is built and can be used to predict the response of the physical system. This study focuses on building and using an ANN to create an algorithm that can predict temperature at a given point in a fire compartment by varying two parameters, namely heat release rate (HRR) and ventilation. To determine whether the ANN is a useful tool, predicted values were compared to that which were obtained through CFD modelling, as well as what was found in literature. This provided an insight into whether this is a technique that the industry would benefit from further exploring and it also allowed further understanding of potential applications.

## 2. Methodology

### *2.1. Numerical methodology*

Numerical analysis was performed using Fire Dynamics Simulator (FDS) version 6.8. The approach taken was to model an ISO 9705 room according to Hwang et al [5]. Internal dimensions of the room measured 2.4 m x 3.6 m x 2.4 m with a single ventilation, centred at the bottom of the front wall. The fire was modelled in the centre of the room with dimensions of 1m x 1m and heptane was used as the fuel. The ramp up time was set to 1 second, as this would reduce the time taken to reach steady state conditions which would in turn reduce the overall model run time. Thermocouples to record gas temperature were placed in the interior of the fire room.

*2.2. Grid independence study*

Before running any models, a grid independence study was carried out to determine the optimal grid resolution to produce accurate results, without it being too computationally expensive. A grid sensitivity study was performed for a fire with a heat release rate of 2070 kW where the ventilation area measured 200 cm in height and 80 cm in width. Figure 2.1 shows the numerical steady-state temperatures at the rear and front thermocouple, located according to Hwang et al [5] using three different grids, cell size of 0.20 m, 0.10 m and 0.05 m. Whilst a mesh size of 0.20 m produced lower values for temperature at the rear thermocouple and higher values at the front thermocouple once steady state was reached, there was no significant difference in the numerical results of a 0.10 m and 0.05 m grid cell size. Given that a larger mesh would be more computationally cost effective, it was determined that 0.10 m would be used for all model runs.

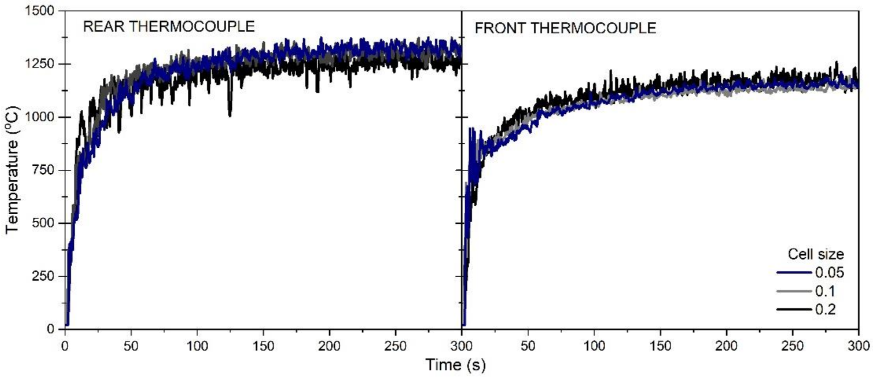


Figure 2.1: Temporal distribution at front thermocouple for different mesh sizes.

*2.2.1. Validation study.* Numerical data collected from CFD was compared with experimental data. A heat release rate (HRR) of 2070 kW was used as an input to replicate the model in the work of Hwang et al [5]. The model was run until steady state was achieved (for 800 seconds) and then an average temperature was calculated. The average temperature during this period for the front (FT) and rear thermocouple (RT) was found to be 1195°C and 1364°C respectively. Table 2.1 shows that percentage differences between experimental and numerical values to be less than 5% for both thermocouples.

Table 2.1: Average temperatures of CFD modelling and experiments (for validation).

| | RT Temperature (°C) | FT Temperature (°C) |
|---|---|---|
| Experiments (Hwang et al, 2010) | 1310 | 1160 |
| CFD | 1364 | 1195 |
| Percentage difference | 4.12% | 3.02% |

*2.3. Machine learning – ANN*

*2.3.1. Data collection.* Two variables were investigated, heat release rate and ventilation area, so two datasets were required for each ANN. Each dataset was used to train and build an ANN. The CFD model, section 2.2, was used to simulate 30 scenarios with HRR ranging from 100 kW - 3000 kW (where ventilation area was 1.6m$^2$) and 60 scenarios with ventilation areas ranging from 0.16 – 4.8 m$^2$ (where HRR was 2070 kW). Each of these models were run until steady state conditions were achieved. The average temperature over the steady state period was then calculated. Results that demonstrated temperatures had not reached a steady state were either omitted from the dataset or reran with extended simulation time.

*2.3.2. Building, training and validating ANNs.* The ANN was built using MATLAB. It was important to determine the number of input and output parameters that the network would have. Due to the small dataset, it was decided that two simple networks would be created. Therefore, both networks had no more than two inputs (HRR or width and height of ventilation area) and one output (temperature at front thermocouple at steady state). Each dataset (see 2.3.1) was split into the training set and validation set which were used to train and validate the networks. The training set is used to train the model and optimise parameters such as the values of the bias and the weights of the connections of the neurons. Using the optimal values for these parameters, the validation data could be used to evaluate and set the internal parameters of the model (e.g. the number of neurons in the hidden layer). The optimisation of the network was done primarily in two ways. One being the ratio of the training data to the validation data in the dataset and the other being the number of neurons in the hidden layer. Each time the ratio was changed, the optimal number of neurons were found based on the root mean square error (RMSE). This would indicate how well the network was performing relative to the actual values in the dataset. Once optimisation was complete for the training and validation data, the ANNs could be assessed against new data points. Input points were fed into the network to obtain the predicted values for temperature. Comparing these to results found in research papers, experimental data and numerical data obtained through CFD modelling can show whether the ANN model is able to produce accurate predictions.

## 3. Results & Discussion

### *3.1. Effect of HRR*

*3.1.1. Training and optimisation.* The dataset was made of 30 points of varying heat release rate (ranging from 100 – 3000 kW). As described in section 2, the network was created with varying ratios and number of neurons. For each ratio, the optimal number of neurons was determined. In the case of a 70:30 ratio of training data to validation data, Figure 3.1, it can be seen the optimal number would be 6 or 8 neurons. This was repeated for the different ratios and each optimised ANN was run 5 times before an average was taken. This was a crucial step as the ANN re-trains each time it is run. For each run, the results were analysed based on the RMSE and correlation coefficient (R value). The results for the optimised networks can be seen in the tables below.

Table 3.1: Average R value from the 5 runs/iterations of the ANN.

| Training: Validation data ratio | Number of neurons in the hidden layer | Average R value (4 s. f.) | | |
|---|---|---|---|---|
| | | Training | Validation | All |
| 50:50 | 4 | 0.9997 | 0.9966 | 0.9990 |
| 50:50 | 5 | 0.9999 | 0.9978 | 0.9986 |
| 60:40 | 4 | **1.0000** | 0.9996 | 0.9997 |
| 60:40 | 6 | 0.9999 | 0.9958 | 0.9976 |
| 70:30 | 6 | 0.9999 | 0.9997 | **0.9998** |
| 70:30 | 8 | 0.9844 | 0.9955 | 0.9841 |
| 80:20 | 5 | 0.9998 | **0.9998** | 0.9998 |
| 80:20 | 8 | 0.9973 | 0.9998 | 0.9980 |

It's important to know what the R value represents to gain a comprehensive understanding of the results being produced by the network. Regression analysis enables an understanding of how the changes in the independent variable(s) are associated with the changes in the dependent variables. Typically, higher values of R suggest that the model is better fitted to the data. Therefore, the best fitted model based on the R value can be seen to be the 70:30 ratio with 6 neurons in the hidden layer, as its R value for 'All' is the highest (R=0.9998, as per Table 3.1). Other scenarios have a higher R value for training and validation – 60:40 with 4 neurons in the hidden layer and 80:20 with 5 neurons in the hidden layer. The R value for 'All' accounts for both the training and validation data, so it is the scenario where the ANN is well fitted to both. The results of this analysis were also in line with the RMSE calculated which was lowest for the test data when the network was run with a 70:30 ratio and 6 neurons were in the hidden layer, which indicates this to be the optimal ANN.

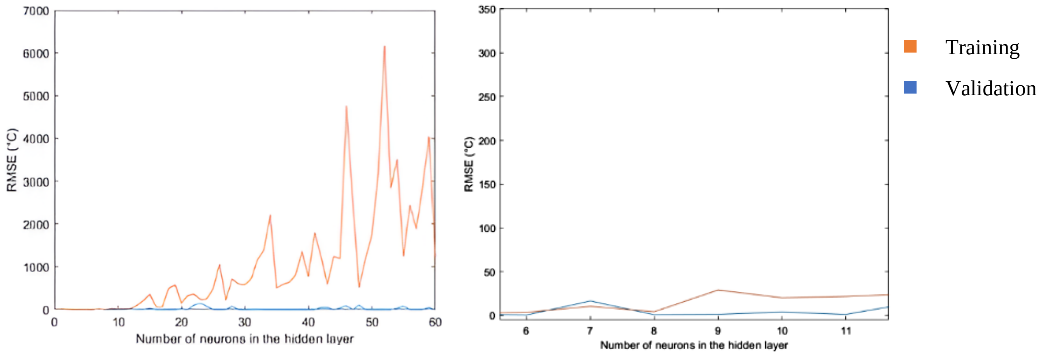


Figure 3.1: RMSE value for training and validation data (zoomed in snapshot on the right).

*3.1.2. Testing the network.* Test data was fed into the ANN, consisting of three data points (630 kW, 1080 kW and 2070 kW). The temperature at steady state for 630 kW and 2070 kW was obtained through running the FDS model. The temperature when the HRR was 1080 kW, however, was extracted from Hwang's study [5]. Table 3.2 shows percentage error calculated for each network. The best results were produced when there was a 70:30 training to validation data ratio with 6 neurons in the hidden layer, with an overall average error of 2.36%.

Table 3.2: Average percentage error for test data for different scenarios.

| Training: Val data ratio | Number of neurons in the hidden layer | Average percentage error (%) | | | Overall average error (%) |
|---|---|---|---|---|---|
| | | 630 kW | 1080 kW | 2070 kW | |
| 50:50 | 4 | 1.19 | 9.25 | 0.34 | 3.59 |
| 50:50 | 5 | 1.88 | 6.61 | 1.09 | 3.19 |
| 60:40 | 4 | 1.00 | 7.64 | 0.32 | 2.99 |
| 60:40 | 6 | 2.80 | 6.67 | **0.26** | 3.24 |
| 70:30 | 6 | 0.59 | **6.21** | 0.28 | **2.36** |
| 70:30 | 8 | 2.97 | 7.70 | 0.65 | 3.77 |
| 80:20 | 5 | **0.45** | 7.81 | 0.63 | 2.96 |
| 80:20 | 8 | 2.84 | 8.39 | 1.09 | 4.10 |

*3.2. Effect of ventilation*

The optimised network in this case was obtained in the same way as above, except a larger dataset of 60 was used (Section 2.3.1). This was due to the independent variable being the ventilation area, therefore both the height and width of the opening were changed. For this network, the 80:20 ratio appeared to be the best performing network based on the R value. However, when the models were tested, it was noted that the network with a 60:40 ratio and 5 neurons in the hidden layer provided the best predictions. High regression values do not always guarantee the best test predictions due to the risk of overfitting. Whilst the model may excel when it comes to the dataset, it may fail to capture the broader patterns present in real-world data which are required to produce accurate predictions for test data. The network was again tested with three new data points (0.8m x 2m, 1m x 2m and 2m x 1.6m), all of which were obtained through CFD simulations. The optimal network produced extremely accurate results, with a percentage error of less than 1.5% for all test points.

*3.3. Comparison of run time*

The trained and optimised ANNs produced accurate results. The predictions were obtained significantly faster than running a CFD model. It is important to note that the comparison does not include set up or optimisation time as this can depend on the ability of the user. The CFD run time can also be lowered through the use of a better computer processor. With this said, the results show that using the ANN is considerably more computationally cost effective, as seen in Table 3.3. The results show that using the artificial neural network is considerably more computationally cost effective. This is in line with the results in some literature which found computational time was reduced by an order of $10^2$ – $10^5$ [2], however proved to be more effective than other research into machine learning where run time was only reduced 2-3-fold [3]. The accuracy of the CFD model can also be compared to the accuracy of the ANN. When the results of the CFD model were validated against those results found in literature (see section 2.1.2), the percentage error was under 5%. The results from the ANN when compared to those obtained from CFD modelling were also less than 5%. This implies that a well built and optimised ANN can produce results with a similar margin of error as produced from CFD modelling.

Table 3.3: model run time for the test data, including training time for MATLAB

| HRR (kW) | Run Time (seconds) | |
|---|---|---|
| | FDS | MATLAB |
| 630 | 11400 | 1 |
| 1080 | 16560 | |
| 2070 | 19080 | |
| Total | 47040 | 1 |

| Ventilation area ($m^2$) | Run Time (seconds) | |
|---|---|---|
| | FDS | MATLAB |
| 0.8 x 2 | 8013 | 1 |
| 1 x 2 | 7462 | |
| 2 x 1.6 | 9650 | |
| Total | 25125 | 1 |

## 4. Conclusion & Recommendations

The machine learning model proved to be effective in reducing computational time, holding significant implications for real-time decision-making in critical fire engineering scenarios. The ANNs displayed remarkable accuracy in predicting thermal responses, with percentage errors consistently below 7% for HRR and below 1.5% for ventilation area. The study highlights the potential of machine learning and ANNs in fire engineering but emphasises the need for further work. The model accurately predicts values within the dataset range but needs testing outside this range for broader applicability. Broadening the model's scope by including more parameters and exploring time as a variable could enhance its predictive power, although this would increase complexity. This would also establish how accurately an ANN performs in response to real-life changes that could occur. The study underscores the importance of high-quality datasets and suggests exploring models built solely from experimental data or using a hybrid approach. Additionally, comparing different types of ANN models in fire engineering could provide insights into the most effective approach. Overall, whilst the study considers a simple scenario, it demonstrates a good foundation for utilising ANNs in fire engineering, with opportunities for enhancing complexity, exploring diverse datasets, and investigating alternate architectures to improve safety and efficiency in fire engineering practices.

## References

[1] Šoljić, F., (2019). Sandro Skansi: Introduction to Deep Learning. From Logical Calculus to Artificial Intelligence. Synthesis Philosophica, 34(2), 477–479.
[2] Hodges, J. L., (2018) Predicting Large Domain Multi-Physics Fire Behaviour Using Artificial Neural Networks. *Virginia Polytechnic Institute and State University.*
[3] Lattimer, B. Y., Hodges, J. L., & Lattimer, A. M. (2020). Using machine learning in physics-based simulation of fire. *Fire Safety Journal, 114,* 102991. https://doi.org/10.1016/j.firesaf.2020.102991
[4] Hodges, J. L., Lattimer, B. Y., & Luxbacher, K. D. (2019). Compartment fire predictions using transpose convolutional neural networks. *Fire Safety Journal, 108,* 102854. https://doi.org/10.1016/j.firesaf.2019.102854
[5] Hwang, C.-H., Lock, A., Bundy, M., Johnsson, E., & Gwon Hyun Ko. (2010). Studies on Fire Characteristics in Over- and Underventilated Full-scale Compartments. *Journal of Fire Sciences,* 28(5), 459–486. https://doi.org/10.1177/0734904110363106