Generating Synthetic Sensor Data to Facilitate Machine Learning Paradigm for Prediction of Building Fire Hazard

Wai Cheong Tam[1], Thomas Cleary[1], Eugene Yujun Fu[2]
[1]National Institute of Standards and Technology, Gaithersburg, MD, USA
[2]Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Abstract

This paper presents a learning-by-synthesis approach to facilitate the utilization of machine learning paradigm to enhance situational awareness for fire fighting in buildings. An automated Fire Data Generator (FD-Gen) is developed. The overview of FD-Gen and its capabilities are highlighted. Using CFAST as the simulation engine, time series for building sensors including heat detectors, smoke detectors, and other targets at any arbitrary locations in multi-room compartments with different geometric configurations can be obtained. An example case is provided. Synthetic data generated for a wide range of fire scenarios from the example is utilized in a supervised machine learning technique. Preliminary results demonstrate that the proposed models can help to predict building fire hazards in real-time.

Keywords: machine learning, simulated data, fire fighting

Introduction

The rapidly expanding availability and diversity of Internet of Things (IoT) technologies revolutionize how observations and data measurements can be made in buildings with little to no human intervention. Interconnected terminal equipment and facilities, such as smart sensors, embedded actuators, mobile devices, and industrial systems, enable automated collection of specific data associated with the real-time conditions of both occupants and building spaces. Data can be systematically gathered and securely transmitted to desired destinations through various wireless or wired networks. The use of these data is believed to be a promising solution to overcome the current difficulties for fire fighting [1]. But, in spite of the recent efforts developed to the IoT technologies in providing both reliable data acquisition and efficient data transfer, none of the real-time data has been utilized in any significant degree by the fire safety community, particularly in the area of firefighting, where the lack of information on the fire-ground is known to be one of the leading factors to incorrect decision making for fire response strategies in which can exacerbate property losses and/or casualties.

Instantaneous data extraction to reveal trends, unseen patterns, hidden relationship, and/or new insights is computationally complex. In a typical building environment, a vast amount of data can be generated from various devices/systems. Since the data is collected from a wide range of sources, the characteristics of the data are highly unstructured and heterogenous [2]. Given a specific situation, there will be a need for faster response to the data. When a larger building environment is involved, the data volume, variety, and velocity (response requirement) increase at scale. Due to the nature of the data itself (high volume, high variety, and high velocity), traditional data analytic algorithms, such as predictive modeling and statistical analysis, are incapable of providing any constructive insights within seconds [3]. For that, a robust and computationally efficient algorithm that can analyze both structured and unstructured data to provide trustworthy insights into decision-making processes regardless of source, size, and type is vital for enhancement of situation awareness, operational effectiveness, and safety for fire fighting.

Machine learning paradigm (ML) [4] is among the top methods that can provide real-time prediction. Chenebert et al. [5] provided a decision tree (DT) classifier being trained based on image data for flame detection in outdoor environments. Using imagery as training data, Yin and his co-workers [6] developed a deep neural network (DNN) framework for smoke detection and they provided hand-crafted features that can help improve detection rate above 96% on their image dataset. Recently, more advanced ML architectures are proposed. A well-established fast object detection convolutional neural network (CNN) with 12 layers was used to identify flames for high-resolution videos given in residential building settings [7]. Aslan et al. [8] attempted to use the state-of-the-art ML architecture, generative adversarial networks (GANs), to classify flame and non-flame objects for videos from surveillance cameras. Also, a saliency detection method [9] based on DNN was proposed to provide wildfire detection for unmanned aerial vehicle and reliable accuracy was reported. Results from these research works show that the use of ML paradigm can overcome the real-time fire predictions that statistical based [10] and physics-based models [11] might not be able to handle. However, to best of our knowledge, the use of ML paradigm for data in time series in building fires does not currently exist.

Indeed, the primary challenge is the scarcity of real-world data for building sensors with fire events. The data problem has been raised in different literature [12]. For the fire safety community, it can be noted that acquiring the desired sensor data is not trivial because 1) fire events do not happen frequently, 2) time series associated with fire event in building environments are not available to the public data warehouse [13], and 3) physically conducting full-scale fire experiments in buildings is costly and time-consuming. Moreover, no prior research work has been carried out to provide guidance on the data requirement for ML applications. With that, there can be a high possibility in which the experimental data is not usable. When conventional ML paradigm demands a large amount of training data (in the order of million sets), we, therefore, propose a learning-by-synthesis approach to generate simulated data to facilitate the use of ML paradigm for prediction of building fire hazards. The research outcome is intended to enhance situational awareness for firefighting in buildings.

In the following sections, the overview of Fire Data Generator (FD-Gen) and its capabilities will be presented. An example case will be given. Preliminary results for the example will be shown and key finding will be highlighted.

Overview of FD-Gen and its capabilities

Fire Data Generator (FD-Gen) is a computational tool with its front-end written in Python[1]. The code is developed to generate time series for typical devices/sensors (i.e. heat detector, smoke

[1] Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the procedures adequately. Such identification is not intended to imply recommendation or endorsement by the National

detector, and other targets) in user-specified building environments. There are four fundamental elements associated with FD-Gen: 1) basic input handler, 2) distribution function module, 3) sampling module, and 4) simulation engine. It should be noted that the code is written to execute multiple runs with different configurations on the Fire Research Division computer cluster at NIST in parallel. Depending the availability on the computational resources, a maximum of ten thousand simulation cases can be completed in a single day.

Basic input handler

The growth and the spread of a fire depend on many factors. For FD-Gen, the basic input handler is capable of handling input parameters accounting for the effect associated with different geometric and/or thermal configurations for the building and various types of fires. Table 1 provides the overall statistics and the assumed distribution functions for a list of important parameters that were found in a previous study associated with residential buildings [14]. Users can utilize the information from the table as the range of inputs to configure the simulation cases. User-specified values will be needed in case of missing statistics or assumed distribution functions.

In addition to the parameters shown in the table, other input parameters include: a) the opening condition, wall density and wall specific heat of the buildings; b) the peak HRR, total energy and its plateau for fires; and c) the thermal properties, locations, and the normal direction for detectors. For commercial buildings, the current version of FD-Gen does not have the statistics and the assumed distribution functions for any input parameters. For that, user-specified values are required. However, this information can be obtained from [15] and might be considered in the future updates.

Parameter	Units	Min	Mean	Max	Assumed Distribution Function
Floor area	m^2	12.8	18.2	34.6	Lognormal
Ceiling height	m	2.13	2.61	3.66	Lognormal
Opening width	m	0.81	2.03	3.24	Lognormal
Opening height	m	1.93	2.27	NA	Lognormal
Wall conductivity	W/m-K	NA	1.03	NA	NA
Wall thickness	mm	13.5	14.3	15.9	Lognormal
Fire location[2]	--	1	2	4	Normal

Table 1: Statistics on selected input parameters for residential buildings in FD-Gen.

Distribution function module

In order to produce sampling values of the input parameters based on their assumed probability distribution, SciPy [16] is implemented. SciPy is a module written in Python that consists of a library with well-known probability distribution functions. In the current version, the code can account for the following univariate distributions such as Binomial, Exponential, Logistic, Lognormal, Normal, Poisson, Triangular, Uniform, Weibull, Gamma, and Beta. Based on recent literature [14], the Normal, Lognormal, and Uniform distribution functions will mostly be used.

Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

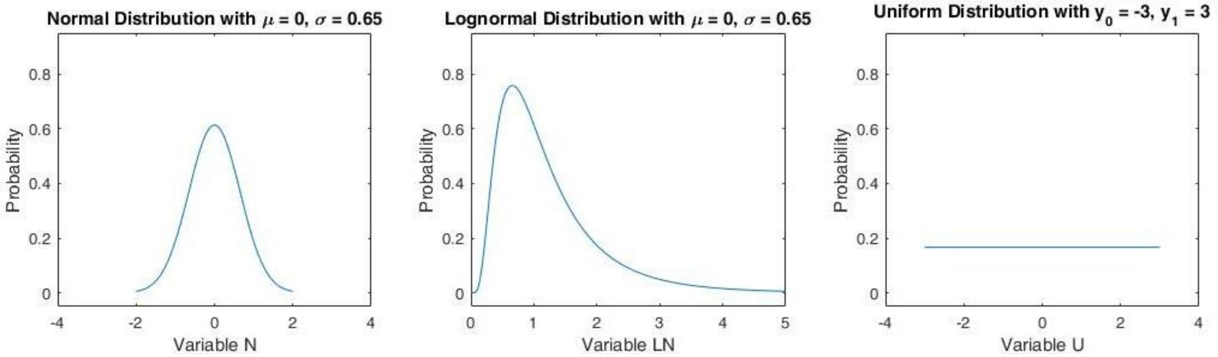
[2] The values indicate location of the fire with 1 = in the center of the room, 2 = against a wall, and 4 = at a corner.

The mathematical formulation of these distribution functions is provided below and Figures 1 show the profiles for the 3 distribution functions for reference:

$$Normal(x;\ \mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

$$Lognormal(x;\ \mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{[\ln(x)-\mu]^2}{2\sigma^2}} \tag{2}$$

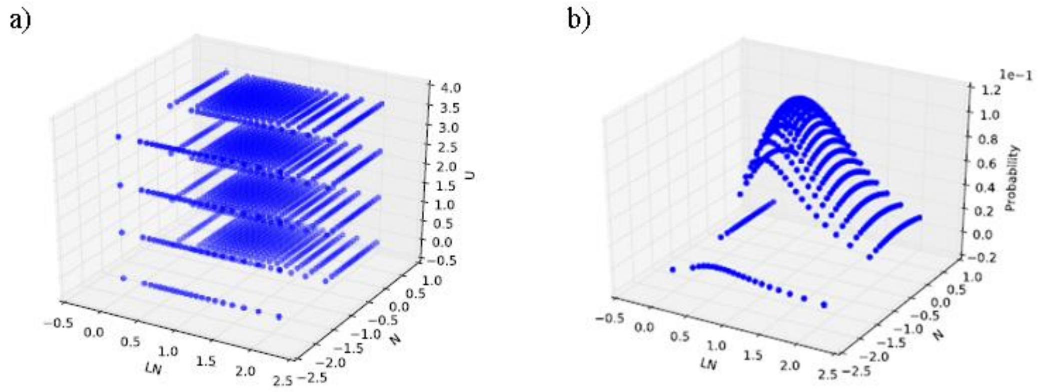
$$Uniform(x;\ y_0, y_1) = \frac{[G(x-y_0) - G(x-y_1)]}{y_1} \tag{3}$$



Figures 1: Profile for the 3 distribution functions with given μ, σ, y_0, and y_1.

Sampling module

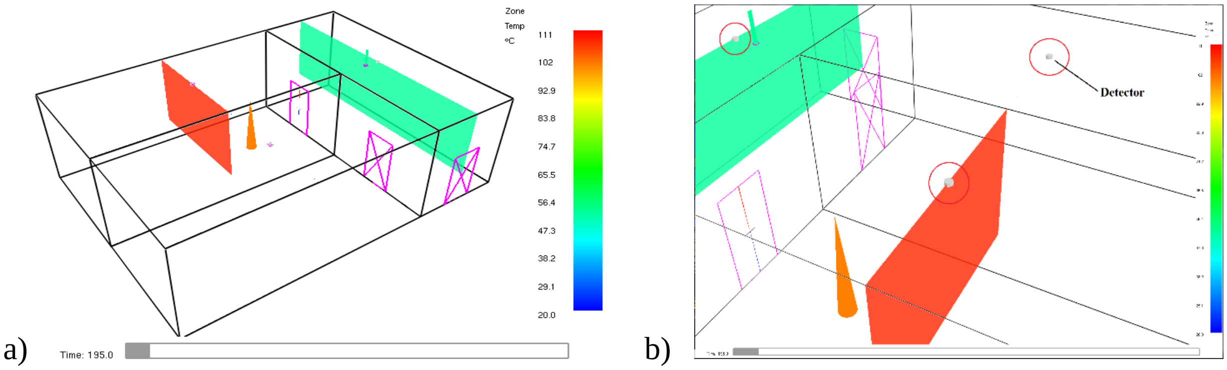
In FD-Gen, a sampler connects a set of variables to their corresponding distributions and produces a sequence of points in the input space. This is an important feature for FD-Gen as it would greatly affect the cost of computational resources. For this current version, the code only supports 1 sampler and it is the grid sampler. Using Eqns. 1 to 3 with μ, σ, y_0 and y_1 being 0, 0.5, -3, and 3, respectively, Figures 2 show the sampling coverage in 3-D space. For grid sampler with 20 points over the Normal and Lognormal distribution equally and 4 points over the uniform distribution equally spaced in the variable values, a total number of 1600 sampling will be needed. In the future, Monte Carlo and/or Latin Hypercube Sampling can be implemented to provide the sampling flexibility and enhance numerical efficiency.



Figures 2: a) 3-D dispersion of the sampling point coordinate and b) the probability map.

Simulation engine

CFAST [17] is used as the simulation engine for FD-Gen. In general, CFAST is a fire simulation program that divides compartments into two zones. Each zone includes a gas mixture/soot medium bounded by a ceiling or a floor, and four surfaces. Thermal conditions of each zone are assumed to be uniform. When there is a fire, a hot layer will form and the medium can be divided into an upper layer and a lower layer. If the fire persists, the upper layer increases in depth and the temperature will rise. When openings exist, there will be natural flow through the openings allowing air exchange between different zones. Figure 3a shows a typical simulation case with a user-specified fire in a zone for 3-compartments building structure.



Figures 3: a) Visualization of a typical CFAST simulation and b) exploded view for detectors.

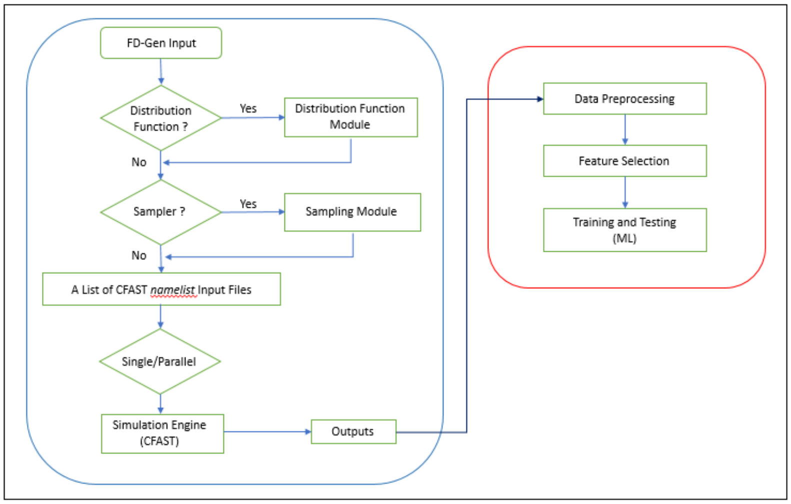


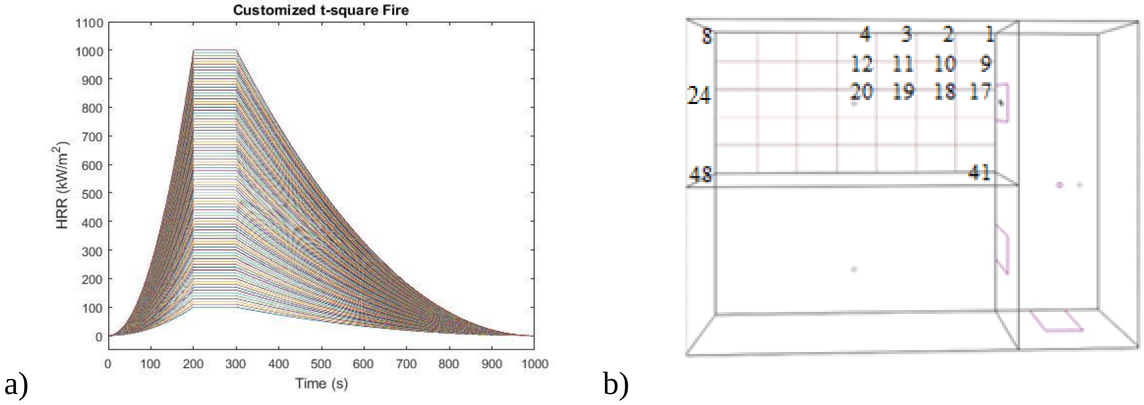
Figure 4: The workflow for FD-Gen.

In CFAST, devices, such as heat detectors, smoke detectors, and other targets can be specified. Although the simulation problem is based on zone method, empirical equations are implemented to determine the local parameters such as gas temperature, smoke concentration, and total radiative heat transfer for heat detectors, smoke detectors, and other targets, respectively. Most importantly, CFAST is verified and validated such that the simulation results obtained from the program are reliable for a wide range of input conditions. Physically, verification ensures a given model is translated correctly into the computer program, and validation ensures the model accurately represents the phenomena of interest. Appendix A provides the full validated range of input parameters and Appendix B shows the modeling uncertainty for all output quantities of interest. Based on Appendix B, it is worth noting that CFAST is conservatively biased, in comparison to the experimental data, for all output quantities expect those highlighted in yellow.

In summary, Figure 4 demonstrates an overall workflow for FD-Gen. An example case is provided in the next section for reference.

Preliminary Results and Discussion

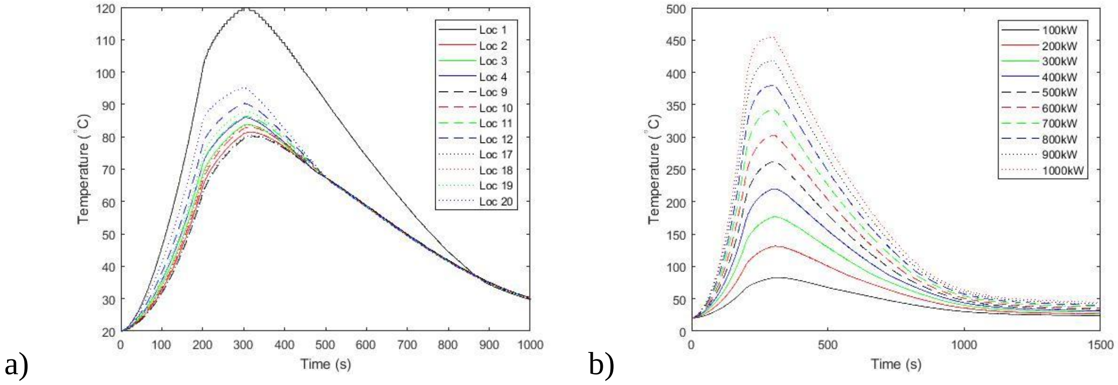
Consider a single-story building with 3 compartments similar to that shown in Figure 3a. There are 2 offices and 1 corridor. The dimensions of the two offices are identical and they are 8m x 4m x 3.5m for length, width, height, respectively. The corridor dimensions are 3m x 8m x 3.5m. The material of the surfaces, ceiling, and floor is concrete. The office located in the upper left-hand corner is denoted as Office1 and the other office is denoted as Office2. It can be shown that there are 3 openings: one is in between Office1 and corridor, one is in between Office2 and corridor, and one is connected for corridor and the outdoor environment. The dimensions of the 3 openings are 2m in width and 1m in height. Initially, the openings are all closed, when the normal incident heat flux to the center of the opening is greater than 2.5kW/m^2, the opening will then be opened. Heat detectors are located at the center of the ceiling in each compartment. The outdoor conditions are normal with the temperature maintained at 20˚C in atmospheric pressure.



Figures 5: a) 100 t-square fire profiles and b) 48 fire locations in Office1.

Given the compartment settings provided above, simulation runs are executed for a fire with a wide range of peak HRR at different locations in Office1. Figures 5 show the profile of 1000 HRR curves for fires and the 48 locations of a fire with a given HRR curve. Using a t-square relationship provided in CFAST and the limits of peak HRR (i.e. the lower bound to be 100kW and the upper bound to be 1000kW), the varying HRR curves can be generated using grid sampling (with 10kW increment) on peak HRR. Similarly, since the floor dimensions of Office1 are specified, the 48 locations of a fire with the HRR curve are generated using the grid sampler. It can be seen that 8 fire locations are specified in the horizontal direction and 6 fires are specified in the vertical direction. In total, detector temperature for 4800 simulation runs are obtained. The simulation time is set for 3600 seconds and output intervals are set to be 1 second. For this example, the total computational time is approximately 1.5 hours.

Figures 6 show the detector temperature profiles for a 100kW peak HRR fire for all 48 locations in Office1 and the detector temperature profiles for all t-square files at a location. The relative distance in between the fire location and detector location is 1.71m in horizontal direction and 1.20m in vertical direction (Loc 10). In Figure 6a, that peak detected temperature for Loc 1 is the highest. The observation is probably due to the corner effect associated with the fire. It should be noted that other output quantities such as smoke concentration and total incident heat flux can be obtained. For simplicity, this paper will only focus on temperature at selected locations.



Figures 6: Temperature profiles a) for a fire with 100kW peak HRR at different locations and b) for fires with different HRR ranging from 100kW to 1000kW at Loc 10.

Machine Learning

Given a synthetic dataset, the use of machine learning paradigm requires 3 additional processes: 1) preprocessing, 2) feature extraction, and 3) training and evaluation. Since the focus of this paper is on FD-Gen, only a brief discussion on the use of machine learning paradigm will be included. It should be noted that the information provided in the subsections is only for demonstration. Actual development of a machine learning model will require more careful consideration. For simplicity, the below example would only take into account the data generated for cases with the same HRR profile (100kW in Figure 6b) for 48 different fire locations (refer to Figure 5b). In total, the dataset contains detector temperature profiles for 48 simulation runs.

Labels	Location Range (m)	Number of profiles
Class 1: very close to the detector	0 - 1	2
Class 2: close to the detector	1 - 2	4
Class 3: a distance from the detector	2 - 3	6
Class 4: far away from the detector	3 - 4	6
Class 5: very far away from the detector	4 - 4.75	4
Class 6: at corner	4.75	2

Table 2: New labels for the example dataset.

Preprocessing

There are three important matters that the data preprocessing will have to consider in this example case. The first one is to categorize target labels such that a machine learning model can be trained efficiently. In Figure 6a, it can be seen that the change of detector temperature profiles depends primarily on the relative distance between the fire and the detector. The figure shows the closer the fire to the detector, the higher the maximum detector temperature is. An exception is observed for the case with a corner fire. Physically, this is due to the corner effect of the fire that yields a ceiling jet with much higher temperature. Based on the data behavior obtained from the data analysis and numerical experiment, the target labels for the current dataset will be modified and categorized into different "classes". Table 2 shows the breakdown information about the new labels. For label in *Class 1*, the fire is considered to be located in a distance ranging from 0 to 1 m.

The second matter would be the removal of duplicate information in the dataset. As shown in Figure 5b, symmetry along the centerline across the horizontal direction of Office1 is observed. Since CFAST is a zone model and with the identical boundary and initial conditions for the cases, the resulting detector temperature profiles for two separate fire case (even with same HRR profile) at two different locations (i.e. Loc 12 and Loc 28) will be identical. This type of duplicate information is problematic when a machine learning model is being evaluated for its prediction accuracy. Physically, this can be explained with the fact that a well-trained model based on a dataset will have perfect predictions (~100 % accuracy) on the same dataset itself. However, when the model is applied, the prediction accuracy for such model might be very low. This behavior is considered as *overfitting*. To counter overfitting, unseen data will have to be used for testing. For that, only data associated with fire locations from Loc 1 to Loc 24 will be considered.

Lastly, a segmentation on the dataset is required for real-time applications. Specifically, the entire detector temperature profile for a specific case as shown in Figure 6a will have to be divided into smaller time series. It can be understood that a machine learning model will be able to encode more precise information (i.e. trend, rate or change, etc.) about the fire using the entire detector temperature profile as an instance for a feature vector. In this scenario, it is highly likely that the model will have better prediction performance. However, such model would lose the ability of real-time detection, as it needs to wait for the complete temperature profile. In this example, 3600 seconds of data are needed. In order to build a model that is feasible to be used in real-time, the dataset is suggested to be divided into smaller segments using a rolling time window. In this scenario, only partial information about the temperature profile will be needed. In this pilot study, a parametric study will be carried out to examine the prediction performance for different lengths of the time window, *W*. The window size will vary from 0 s to 120 s (with increment of 10 s).

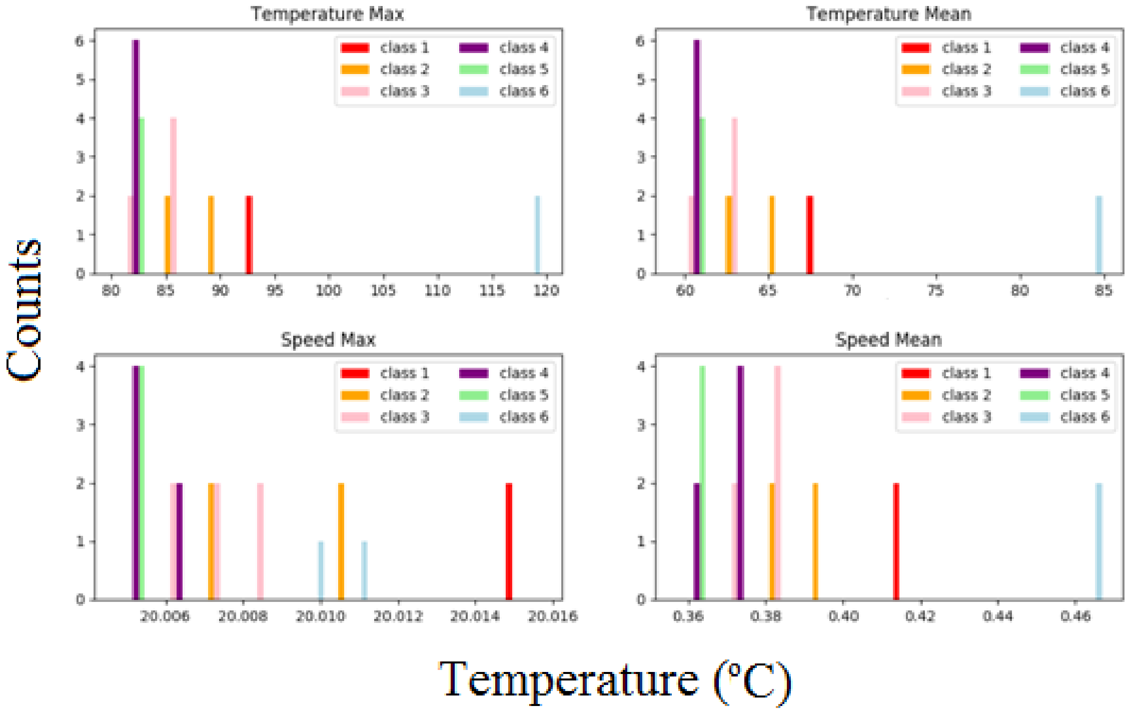
Feature Extraction

Feature extraction is a critical process for obtaining feature vectors that can include important information about the data. A good feature extraction will also help increasing the prediction performance for a machine learning model. In this example, we aim to extract features based on the temperature profiles for fire location detection. For each fire event, we describe temperature signal as: $T = [T_1, T_2, \dots, T_n]$, where T_i is the detected temperature at time stamp, *t*, of i. In addition, we further extract speed signal: $S = [T_1/t_1, T_2/t_2, \dots, T_n/t_n]$, which indicates the average temperature growth rate ($^\circ\mathrm{C}/s$). Statistical features such as maximum and mean of these signals are also extracted. Moreover, since the data is given in time series, the first order derivatives of these signals are also considered. Table 3 summarizes the signals and features that will be used in our model.

Signal	**Features**
Temperature	Maximum, mean, median
Speed	
First order derivative of Temperature	
First order derivative of Speed	

Table 3: Signals and their features.

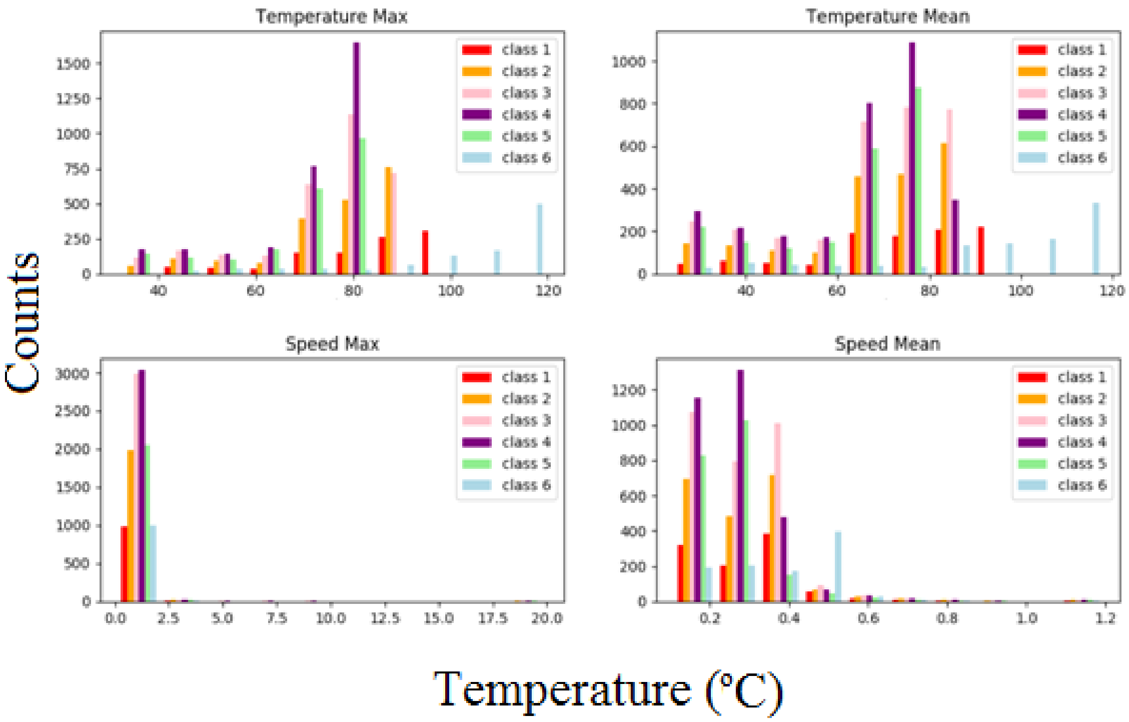
Figures 7 show the data distribution for the statistical (max and mean) temperature and speed signals based on the entire time series and four distinct behaviors are observed. It can be seen that 1) the fire located at the corner (*Class 6*) has a much larger maximum (refer to upper left plot) and average temperature (refer to upper right plot). The fire that is very close to the detector (*Class 1*) has the largest maximum temperature growth rate (refer to lower left plot). And for the fire that is far (*Class 4*) and very far away (*Class 5*) from the detector, they both have a lower maximum and lower average temperature (refer to upper plots). Though, cases associated with fire of *Class 5* tend to have lower average temperature growth rates than that of *Class 4*. Figures 8 show the data distribution for the statistical signals for temperature and speed segmented based on the rolling window technique. In this figure, the window size is considered to be 120 s. Similar to that of seen in Figures 7, clear distinctions associated with each feature are observed. For that, it can be expected that the use of these features for building a model will likely lead to good performance.



Figures 7: Data distribution for scenarios using the entire profile.

Training and Evaluation

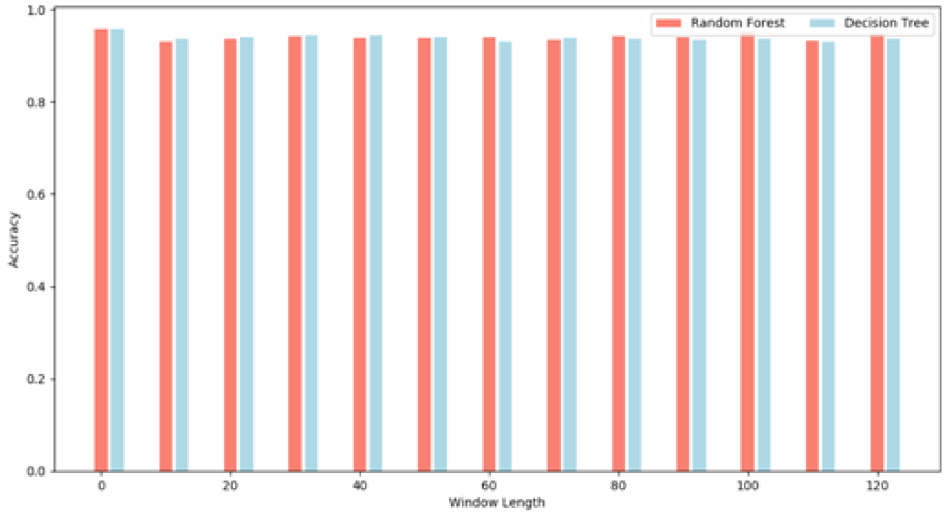
Given a dataset with feature vectors (refer to Table 3) and the corresponding labels (refer to Table 2), we can use machine learning paradigm to classify fire relative location. In this example, we build two machine learning models, one for Random Forest and one for Decision Tree. These models are used because of its numerical efficiency and its capability to handle the complexity of the problem. For problems with more complex data being involved (i.e. prediction of tenability for another room at a given time), more advanced models, such as SVM and/or neural network, can be used.



Figures 8: Data distribution for scenarios with real-time detection (W = 120 s).

For evaluation purpose, we divide the dataset into two subsets: a training set and a testing set. We train the detection model based on the training set and evaluate the performance of the model using testing set. Since the dataset for this example is relatively small, we adopt leave-one-experiment-out cross-validation to evaluate our model. Specifically, we divide the dataset into 24 subsets where each subset contains only the data instances from one particular simulation run. We then train the classifier with the 23 sets among them (the training set) and carry out evaluation on the remaining one (the testing set). This is repeated 24 times for all subsets. The overall average accuracy across the 24 evaluations is reported as the final performance.

Figure 9 shows the model performance for use of the entire temperature profiles (denoted as $W = 0$ s) and the use of rolling window technique with different window sizes (denoted as $W > 0$ s). Preliminary results indicate that our proposed machine learning models are capable of detecting fire relative location with reasonable accuracy. For the scenario using the entire profiles, the accuracy is 95.83%. Only one instance is wrongly classified: a C*lass 2* (close to the detector) instance is wrongly detected as C*lass 3* (medium distance to the detector). For the scenario using the rolling window technique, our models achieve reasonable accuracy even with partial information. The best performance is observed to be 95.08% and that is for W = 100 s using Random Forest. In general, these preliminary results demonstrate promising results for using machine learning paradigm to detect fire location based on detector temperature profiles in real-time.



Figures 9: Performance of fire location detection for the scenario of using the entire profile ($W = 0s$) and the scenario of real time detection ($W > 0s$).

Conclusions

The workflow of the Fire Data Generator (FD-Gen) is presented. The functionalities associated with the basic input handler, distribution function module, sampling module, and simulation engine are discussed. Using FD-Gen, heat detector temperature data are obtained for 4800 cases in a single-story building with three compartments. As a demonstration, these data are used to train two simple machine learning algorithms. Preliminary results shown that the proposed models can predict the relative location of the fire with reasonable accuracy. It can be seen that these real-time predictions will help to develop a system to enhance situational awareness for fire fighting.

References

[1] Hamins, A.P., Bryner, N.P., Jones, A.W. and Koepke, G.H., 2015. Research Roadmap for Smart Fire Fighting (No. Special Publication (NIST SP)-1191).

[2] Qolomany, B., Al-Fuqaha, A., Gupta, A., Benhaddou, D., Alwajidi, S., Qadir, J. and Fong, A.C., 2019. Machine Learning, Big Data, And Smart Buildings: A Comprehensive Survey. *arXiv preprint arXiv:1904.01460*.

[3] Mahdavinejad, Mohammad Saeid, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi, and Amit P. Sheth. "Machine learning for Internet of Things data analysis: A survey." *Digital Communications and Networks* 4, no. 3 (2018): 161-175.

[4] Bishop, C.M., 2006. Pattern recognition and machine learning. springer.

[5] Chenebert, A., Breckon, T.P. and Gaszczak, A., 2011, September. A non-temporal texture driven approach to real-time fire detection. In *2011 18th IEEE International Conference on Image Processing* (pp. 1741-1744). IEEE.

[6] Yin, Z., Wan, B., Yuan, F., Xia, X. and Shi, J., 2017. A deep normalization and convolutional neural network for image smoke detection. *Ieee Access*, 5, pp.18429-18438.

[7] Shen, D., Chen, X., Nguyen, M. and Yan, W.Q., 2018, April. Flame detection using deep learning. In *2018 4th International Conference on Control, Automation and Robotics* (ICCAR) (pp. 416-420). IEEE.

[8] Aslan, S., Güdükbay, U., Töreyin, B.U. and Çetin, A.E., 2019. Deep Convolutional Generative Adversarial Networks Based Flame Detection in Video. *arXiv preprint arXiv*:1902.01824.

[9] Zhao, Y., Ma, J., Li, X. and Zhang, J., 2018. Saliency detection and deep learning-based wildfire identification in UAV imagery. *Sensors*, 18(3), p.712.

[10] Celik, T., Demirel, H., Ozkaramanli, H. and Uyguroglu, M., 2007. Fire detection using statistical color model in video sequences. *Journal of Visual Communication and Image Representation*, 18(2), pp.176-185.

[11] McGrattan, K., Hostikka, S., McDermott, R., Floyd, J., Weinschenk, C. and Overholt, K., 2013. Fire dynamics simulator user's guide. *NIST special publication*, 1019(6).

[12] Sharma, J., Granmo, O.C., Goodwin, M. and Fidje, J.T., 2017, August. Deep convolutional neural networks for fire detection in images. In *International Conference on Engineering Applications of Neural Networks* (pp. 183-193). Springer, Cham.

[13] https://www.data.gov/ [Online; accessed 2019-08-26].

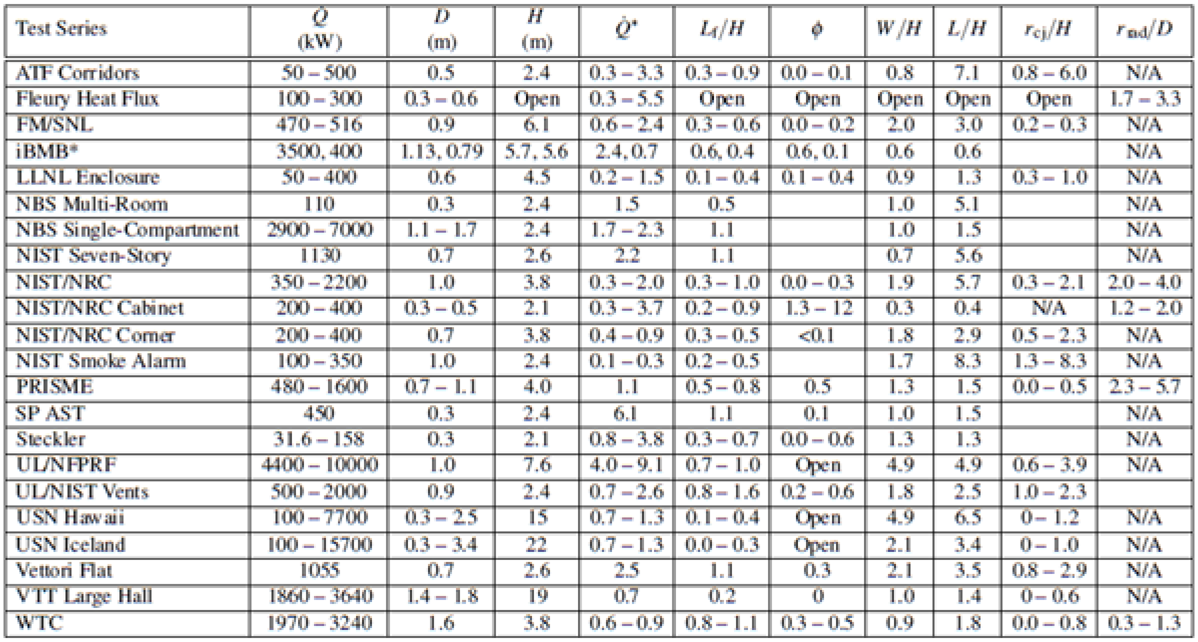
[14] Bruns, M.C., 2018. Estimating the flashover probability of residential fires using Monte Carlo simulations of the MQH correlation. *Fire technology*, 54(1), pp.187-210.

[15] https://www.eia.gov/consumption/commercial/data/2012/ [Online; accessed 2019-08-26].

[16] Jones, E., Oliphant, E., and Peterson, P., 2001. SciPy: Open Source Scientific Tools for Python, http://www.scipy.org/ [Online; accessed 2019-08-26].

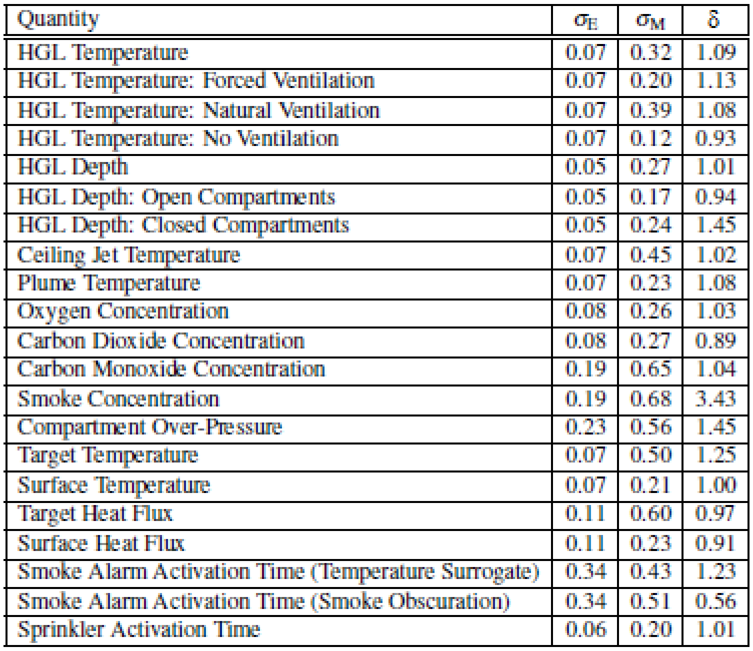
[17] Peacock, R.D., Reneke, P.A. and Forney, G.P., 2017. CFAST–Consolidated Model of Fire Growth and Smoke Transport (Version 7) Volume 2: User's Guide. *NIST Technical Note 1889v2*.

Appendix A: Summary of important experimental parameters.



Test Series	$\dot{Q}$ (kW)	D (m)	H (m)	$\dot{Q}^*$	L_f/H	ϕ	W/H	L/H	r_{cj}/H	r_{nd}/D
ATF Corridors	50 – 500	0.5	2.4	0.3 – 3.3	0.3 – 0.9	0.0 – 0.1	0.8	7.1	0.8 – 6.0	N/A
Fleury Heat Flux	100 – 300	0.3 – 0.6	Open	0.3 – 5.5	Open	Open	Open	Open	Open	1.7 – 3.3
FM/SNL	470 – 516	0.9	6.1	0.6 – 2.4	0.3 – 0.6	0.0 – 0.2	2.0	3.0	0.2 – 0.3	N/A
iBMB*	3500, 400	1.13, 0.79	5.7, 5.6	2.4, 0.7	0.6, 0.4	0.6, 0.1	0.6	0.6		N/A
LLNL Enclosure	50 – 400	0.6	4.5	0.2 – 1.5	0.1 – 0.4	0.1 – 0.4	0.9	1.3	0.3 – 1.0	N/A
NBS Multi-Room	110	0.3	2.4	1.5	0.5		1.0	5.1		N/A
NBS Single-Compartment	2900 – 7000	1.1 – 1.7	2.4	1.7 – 2.3	1.1		1.0	1.5		N/A
NIST Seven-Story	1130	0.7	2.6	2.2	1.1		0.7	5.6		N/A
NIST/NRC	350 – 2200	1.0	3.8	0.3 – 2.0	0.3 – 1.0	0.0 – 0.3	1.9	5.7	0.3 – 2.1	2.0 – 4.0
NIST/NRC Cabinet	200 – 400	0.3 – 0.5	2.1	0.3 – 3.7	0.2 – 0.9	1.3 – 12	0.3	0.4	N/A	1.2 – 2.0
NIST/NRC Corner	200 – 400	0.7	3.8	0.4 – 0.9	0.3 – 0.5	<0.1	1.8	2.9	0.5 – 2.3	N/A
NIST Smoke Alarm	100 – 350	1.0	2.4	0.1 – 0.3	0.2 – 0.5		1.7	8.3	1.3 – 8.3	N/A
PRISME	480 – 1600	0.7 – 1.1	4.0	1.1	0.5 – 0.8	0.5	1.3	1.5	0.0 – 0.5	2.3 – 5.7
SP AST	450	0.3	2.4	6.1	1.1	0.1	1.0	1.5		N/A
Steckler	31.6 – 158	0.3	2.1	0.8 – 3.8	0.3 – 0.7	0.0 – 0.6	1.3	1.3		N/A
UL/NFPRF	4400 – 10000	1.0	7.6	4.0 – 9.1	0.7 – 1.0	Open	4.9	4.9	0.6 – 3.9	N/A
UL/NIST Vents	500 – 2000	0.9	2.4	0.7 – 2.6	0.8 – 1.6	0.2 – 0.6	1.8	2.5	1.0 – 2.3	
USN Hawaii	100 – 7700	0.3 – 2.5	15	0.7 – 1.3	0.1 – 0.4	Open	4.9	6.5	0 – 1.2	N/A
USN Iceland	100 – 15700	0.3 – 3.4	22	0.7 – 1.3	0.0 – 0.3	Open	2.1	3.4	0 – 1.0	N/A
Vettori Flat	1055	0.7	2.6	2.5	1.1	0.3	2.1	3.5	0.8 – 2.9	N/A
VTT Large Hall	1860 – 3640	1.4 – 1.8	19	0.7	0.2	0	1.0	1.4	0 – 0.6	N/A
WTC	1970 – 3240	1.6	3.8	0.6 – 0.9	0.8 – 1.1	0.3 – 0.5	0.9	1.8	0.0 – 0.8	0.3 – 1.3

Appendix B: Summary of statistics for all quantities of interest.



Quantity	σ_E	σ_M	δ
HGL Temperature	0.07	0.32	1.09
HGL Temperature: Forced Ventilation	0.07	0.20	1.13
HGL Temperature: Natural Ventilation	0.07	0.39	1.08
HGL Temperature: No Ventilation	0.07	0.12	0.93
HGL Depth	0.05	0.27	1.01
HGL Depth: Open Compartments	0.05	0.17	0.94
HGL Depth: Closed Compartments	0.05	0.24	1.45
Ceiling Jet Temperature	0.07	0.45	1.02
Plume Temperature	0.07	0.23	1.08
Oxygen Concentration	0.08	0.26	1.03
Carbon Dioxide Concentration	0.08	0.27	0.89
Carbon Monoxide Concentration	0.19	0.65	1.04
Smoke Concentration	0.19	0.68	3.43
Compartment Over-Pressure	0.23	0.56	1.45
Target Temperature	0.07	0.50	1.25
Surface Temperature	0.07	0.21	1.00
Target Heat Flux	0.11	0.60	0.97
Surface Heat Flux	0.11	0.23	0.91
Smoke Alarm Activation Time (Temperature Surrogate)	0.34	0.43	1.23
Smoke Alarm Activation Time (Smoke Obscuration)	0.34	0.51	0.56
Sprinkler Activation Time	0.06	0.20	1.01

Note that the term δ is a calculated bias factor representing the degree to which the model over-predicted or under-predicted experimental data, the term σ_M is a measure of model uncertainty, and the term σ_E is a measure of experimental uncertainty. δ > 0 means the model over-predicted the observations, and $\sigma_M < \sigma_E$ means that the model uncertainty is within experimental uncertainty.