

ABSTRACT

Title of Thesis: FEASIBILITY ANALYSIS OF COUPLING FDS MODELING
WITH MACHINE LEARNING FOR SITUATIONAL
AWARENESS IN AIRCRAFT HANGARS

Alison M. Davis
Master of Science, 2022

Thesis Directed by: Dr. James A. Milke, Advisor
Department of Fire Protection Engineering
Dr. Arnaud Trouvé, Co-Advisor
Department of Fire Protection Engineering

Situational awareness is a critical factor in maintaining the safety of firefighters and can be largely improved in buildings using distributed sensors that provide real-time data. A two-phase approach is used to increase situational awareness in aircraft hangars. Phase I consists of modeling a hangar with an open door in Fire Dynamics Simulator (FDS), with a high density of smoke, temperature, CO and CO₂ sensors located at the ceiling. Fuels of interest including Douglas fir, polyethylene, paper, JP-8, and propane are modeled in six potential fire locations, with five locations along the centerline of the hangar and one in the corner of the hangar. Additionally, wind and beams at the ceiling are added to the simulation to determine the impact on the products of combustion that the sensors pick up.

Phase II uses the data acquired from the FDS simulations to inform and build machine learning models that utilize supervised learning techniques to identify the location of the fire, the magnitude of the fire and the composition of the fuel that is burning. It is determined that temperature and smoke are the key products of combustion needed for these analyses. The location of the fire is identified within a circular area with a 5 m radius by using temperature measurements, thus reducing the amount of input data needed for the machine learning models. The magnitude of the fire is predicted using temperature as inputs to a heat

release rate (HRR) model using a fully connected, three-layer, feed forward neural network. The composition of the fuel is predicted using a linear support vector machine that supports multi-class classification, using products of temperature and smoke obscuration as inputs. The location model is 80% accurate, the HRR model is 85% accurate and the fuel composition model varies between 62% and 91% accuracy depending on the classification goals. These results prove the feasibility of machine learning applications in an aircraft hangar setting.

FEASIBILITY ANALYSIS OF COUPLING FDS MODELING
WITH MACHINE LEARNING FOR SITUATIONAL
AWARENESS IN AIRCRAFT HANGARS

by

Alison M. Davis

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2022

Advisory Committee:

Dr. James A. Milke, Chair/Advisor
Dr. Arnaud Trouvé, Co-Advisor
Dr. Bridgett Ashley

© Copyright by
Alison M. Davis
2022

Acknowledgements

I am extremely grateful for all of the support I have received throughout this past year on this research project. I would first like to thank my advisors, Dr. Milke and Dr. Trouvé for their expertise and guidance. Dr. Milke has been a great mentor during my five years as a FPE student, providing me with research experiences, connecting me with industry professionals, imparting his advice and providing words of encouragement. His expertise in detection, specifically in aircraft hangars has been a great benefit to this project. Dr. Trouvé has been a great resource for all of my FDS questions and his insight into new ways to approach a problem was always appreciated.

I would also like to express my sincerest gratitude to the Air Force and Battelle for funding this research project. Specifically, Dr. Bridgett Ashley, who is also serving as a member on my committee, Dr. Heather Luckarift and Julie Bryant for their contributions to help improve this research.

A special thank you goes out to Dr. Mark Fuge, who has been my go-to resource for all things machine learning, answering my questions and providing insight, even when I was no longer a student in his class.

I would like to acknowledge the University of Maryland supercomputing resources (<http://hpcc.umd.edu>) made available for conducting the research reported in this paper.

Finally, I would like to thank my friends and family for their support during my journey. To my fire friends - thank you for making my undergraduate and graduate experience what it was. It would not have been the same without your support and laughs we shared along the way. To my family - thank you to my siblings for helping to remind me of my potential. Lastly, thank you to my mom and dad for always believing in me, for supporting me throughout my entire education and in my pursuit of higher education. I would not have made it to the finish line without your constant love and encouragement.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Computational Modeling Background	3
1.3	Machine Learning Background	4
1.4	Scope of Work	4
2	Literature Review	6
2.1	Inverse Modeling	6
2.2	History of Machine Learning and Fire Detection	8
2.2.1	Recent Machine Learning Applications in Fire Protection	13
3	FDS Models and Validations	18
3.1	NIST Simulation Validation	18
3.1.1	Detector Activation Analysis	19
3.2	Grid Size	21
3.2.1	Critical Length Scale Analysis	21
3.3	Hand Calculations	24
3.3.1	Heat Transfer Coefficient	24
4	Model Inputs	29
4.1	Compartment Geometry	29
4.2	Instrumentation Devices and Spacing	30
4.3	Fuel Model and Design Fires	31
4.3.1	Combustion and Pyrolysis Model Description	31

4.3.2	Design Fire	33
4.3.3	Fuel Characteristics	35
4.4	Differing Conditions	41
4.4.1	Wind	41
4.4.2	Complex Geometry	42
4.5	Model Limitations	44
5	Machine Learning	45
5.1	Location Identification Methodology	45
5.1.1	Enclosure Fire Physics	46
5.1.2	Temperature Approach	48
5.2	Heat Release Rate and Temperature Algorithm	50
5.2.1	Final Architecture	51
5.2.2	Training on Individual Fuels	53
5.2.3	Combining Fuels into Single Feed Forward Neural Network	55
5.3	Fuel Discrimination Algorithm	57
5.3.1	No Time Dependency	58
5.3.2	Time Dependency with Discrimination of All Fuels	61
5.3.3	Time Dependency with Binary Discrimination Based on Suppression Agent	62
6	Results	64
6.1	Location Identification	64
6.2	HRR Model	67
6.2.1	Overall Results	76
6.3	Fuel Discrimination Model	78
6.3.1	All Fuels Approach	78
6.3.2	Binary Fuel Approach	79
6.3.3	Comparison Between Classification Methods	80
7	Conclusion and Future Work	82
A	FDS Input Files	87
A.1	Locations	87

A.2	No Wind Case (Location B)	87
A.3	Wind Case (Location F)	93
A.4	Beam Case (Location A)	100
A.5	Design Fire Ramp-Up Values	108
A.5.1	Douglas fir	108
A.5.2	Polyethylene	109
A.5.3	Paper	109
A.5.4	JP-8	110
A.5.5	Propane	111
A.5.6	Polyurethane Foam	111
B	Python Code	113
C	Graphs	140

List of Tables

1.1	Test Matrix	5
3.1	Comparison Between FDS and NIST Smoke Detector Activation Times	21
5.1	Final Mean Absolute Error for Fuels Trained on an Individual Basis	54
5.2	Manual Subset of Fuels for Training	56
5.3	Manual Subset of Fuels for Testing	57
5.4	Parameters in SVM	58
6.1	Final Loss Values	68
6.2	Critical HRR Analysis Test Matrix	72
6.3	HRR Category Agreement	77
6.4	Total “All Fuels” seen	79
6.5	Total Binary Fuels seen	80

List of Figures

2.1	Flowchart of inverse HRR search procedure as described by Overholt.	7
2.2	Neural network diagram.	11
2.3	Description of how a perceptron works.	11
2.4	A simple CNN representation.	12
2.5	ReLU activation function.	14
2.6	Representation of transfer learning. The circles represent outputs from fire simulations, rectangles represent ANNs, black arrows represent direct training and blue arrows represent knowledge transfer.	15
3.1	Heat (H) and smoke detector (P) locations, plan view 15 m high facility.	20
3.2	PyroSim visualization of a side view into the hangar. The fire is located in the center of the hangar and the meshes that correspond to the fire plume and ceiling jet are highlighted in yellow.	22
3.3	Comparison of plume and ceiling jet velocity for different grid sizes.	23
3.4	Comparison of temperature values when the adjacent mesh is increased from a factor two to five.	23
3.5	Heat transfer coefficient from eight locations where the total heat flux was measured. The (x,y) coordinates in meters for each location are shown in the legend.	26
3.6	Temperature inputs into MQH equation.	27
3.7	Comparison of prescribed HRR in FDS for JP-8 versus derived HRR using total heat flux from FDS in MQH correlation.	27

3.8	Comparison of prescribed HRR (when reduced by a factor of four from the original HRR) in FDS for JP-8 versus derived HRR using heat transfer coefficient from full scale HRR total heat flux and reduced HRR in FDS in MQH correlation.	28
4.1	Smokeview animation of hangar configuration; view from the outside looking into the hangar through the open door.	30
4.2	Smokeview animation of hangar configuration; side view from the hangar with the hangar on the right and the outdoor space on the left, no wind case.	30
4.3	Sensor locations at the ceiling, relative to the corner of the north and east wall.	31
4.4	Comparison of HRR curve from literature vs. derived HRR curve.	36
4.5	Douglas fir HRR curve.	36
4.6	Polyethylene HRR curve.	37
4.7	Propane HRR curve.	38
4.8	Large box with paper HRR curve.	39
4.9	JP-8 HRR curve.	40
4.10	Polyurethane foam GM21 HRR curve.	41
4.11	PyroSim visualization of hangar configuration with beams, with sensor of interest circled in red.	43
4.12	Comparison of CO concentration with beams at the ceiling versus a flat ceiling.	44
5.1	Labeled sensor locations.	45
5.2	Raw versus smoothed data for JP-8 fire in location F with wind.	46
5.3	Stratification of smoke at the ceiling.	47
5.4	Temperature versus position, from JP-8 fire at location C, at the end of the simulation.	47
5.5	JP-8 fire at location C, with smoothed temperatures, taken along the diagonal of the hangar.	48
5.6	Flowchart of code breakdown.	50
5.7	Performance of L1 (MAE) versus L2 (MSE) loss.	53
5.8	Scaled testing MAE when each fuel is trained individually.	54
5.9	Smoke versus temperature at various times.	59
5.10	SVM applied at various times.	60
5.11	Accuracy when trained on a single instance in time (150 seconds) and tested across the entirety of the simulation.	61
5.12	Smoke obscuration versus temperature versus time for five main fuels of interest.	62

5.13	Smoke obscuration versus temperature versus time for binary classification	63
6.1	Results of location identification model with gray circle representing the 5 m radius of interest; circles = Douglas fir, triangles = propane, squares = paper, plus sign = JP-8; blue = no wind, green = wind, red = beams.	65
6.2	Polyethylene temperatures, compared to ambient.	66
6.3	Paper fire with wind at location F.	67
6.4	Train and test loss curves.	68
6.5	HRR curve predictions based on training set.	69
6.6	HRR curve predictions.	71
6.7	Proportional error curves.	73
6.8	Scaled HRR curve predictions.	75
6.9	Polyurethane foam fire.	76
6.10	HRR confusion matrix.	78
6.11	Confusion matrix of classification on all five fuel types.	79
6.12	Confusion Matrix of binary classification of JP-8 versus other fuels.	80
6.13	Critical accuracy comparison of all fuels versus binary classification.	81
C.1	Pairplot at 30 seconds.	141
C.2	Pairplot at 150 seconds.	142
C.3	Pairplot at 300 seconds.	143

List of Abbreviations

AI: Artificial Intelligence

ANN: Artificial Neural Network

CFAST: Consolidated Model of Fire and Smoke Transport

CFD: Computational Fluid Dynamics

CNN: Convolutional Neural Network

DL: Deep Learning

DTW: Dynamic time warping

FDS: Fire Dynamics Simulator

FT-IR: Fourier Transform Infrared

GA: Genetic Algorithm

HRR: Heat Release Rate

HRRPUA: Heat Release Rate Per Unit Area

HTM: Hierarchical temporal memory

IFM: Inverse Fire Modeling

IoT: Internet of Things

LES: Large eddy simulation

LVQ: Learning Vector Quantization Network

MAE: Mean Absolute Error

ML: Machine Learning

MLR: Mass loss rate

MLP: Multilayer Perceptron

MPI: Message-Passing Interface

MQH: McCaffrey, Quintiere, Harkleroad

MSE: Mean Squared Error

NIST: National Institute of Standards and Technology

NN: Neural Network

PCA: Principal Component Analysis

PVC: Polyvinyl-chloride

ReLU: Rectified linear unit

RNN: Recurrent Neural Network

SFPE: Society of Fire Protection Engineers

SGD: Stochastic Gradient Descent

SVM: Support vector machine

UL FSRI: Underwriter's Laboratory Fire Safety Research Institute

WSN: Wireless Sensor Network

Chapter 1

Introduction

Aircraft hangars store expensive aircraft, and these valuable planes and corresponding aeronautical equipment need to be protected accordingly. Detection of fires has historically been done with standard, stationary sensors that do not communicate with one another. Firefighters collect information manually by making their way around or through the building in order to be able to confidently assess what is burning, the location and the severity of the fire. This is time consuming and also requires firefighters to enter the area where the fire is located. Improvements can be made through the use of contemporary stationary sensors and machine learning (ML).

This report explores the feasibility of coupling contemporary sensors with ML techniques to provide situational awareness for fires in the space of interest. This report will be divided into two phases. The first involves simulations of fires in an aircraft hangar using a computational fluid dynamics model to depict the fire-driven fluid flow and create a set of fire signatures. Simulations will monitor the conditions produced in the upper smoke layer of the hangar to compile data on temperature, gases, and soot. Fuel type, location, environmental conditions, and ceiling geometry will differ for the variety of the simulations.

The second phase of this research will conduct an analysis of the array of parameters describing fire conditions created within the hangar simulations. This analysis will be conducted using ML, to condense the multitude of data associated with the fire signatures into succinct statements that describe the location and size of the fire and provide an indication of the commodity involved in the fire.

1.1 Motivation

In a fire scenario, incident commanders need to know the fire location and severity (determined by conditions produced by the fire) in order to propose an effective strategy for fire control. A way that this is done in the field is referred to as “size-up”, which is defined as “the ongoing evaluation of problems confronted with a fire situation”.[1] This process starts when there is indication of an alarm, by asking fundamental questions such as, “what is the type of emergency?”, “what’s on fire...?” and “what resources are needed?”[2] and continues until the fire is under control. Strategic factors like location and extent of the fire in addition to weather conditions [1] are taken into account, which pertain to the scope of this research. Prior knowledge and answers to these questions allows for better preparation when responding to a fire scene.

This information can be gathered through real-time data collected from distributed sensors located in the affected space. Buildings include a wide variety of sensors, however only a marginal amount of sensors provide information regarding the presence of a fire. The information from contemporary sensors is limited to determining whether or not a fire is present in the space, but does not provide additional information as to the conditions produced by the fire. Most fire detectors and sensors are discrete, meaning that they only operate at a predetermined threshold value, but with modifications these sensors could be adjusted to collect continuous information from the fire scenario. Data from these sensors can be utilized to understand the severity of the fire via the products of combustion and conditions produced, thus providing firefighters with more information on how to best approach the fire scenario.

It is inefficient for firefighters to survey an entire building during the fire incident. This is a time consuming process that can take away from the ability to quickly identify and put out the fire - assuming intervention is needed. This is an inherently unsafe process because if there is a large magnitude fire, firefighters could be unknowingly subjecting themselves to extreme heat and harmful products of combustion. Increasing situational awareness is one of the critical factors in maintaining the safety of the fire service and anyone involved in fires, as this was cited as one of the leading factors identified by firefighters in near-miss reports and is defined as “understanding the current environment and being able to accurately anticipate future problems to enable effective action”.[3] Additional aspects of situational awareness include the ability to perceive, process, predict, decide, take action and communicate or coordinate. All of these aspects, aside from the ability to take action, can be automated and/or simplified through the use of sensor data and ML.[4]

This integration of technology can be likened to that of smart buildings. Buildings are now utilizing

information beyond the equipment and sensors present in the space, and are connecting this information in a way that allows for the building to be responsive and interact with the building operators and occupants to give practical information.[5]

1.2 Computational Modeling Background

The most recent version of Fire Dynamics Simulator, FDS version 6.7.6, is used for computational modeling purposes. FDS is developed by the National Institute of Standards and Technology (NIST). Smokeview, version 6.7.15, which is also developed by NIST, is used for visualization purposes.

FDS is a computational fluid dynamics (CFD) model that decomposes the fire compartment into numerous control volumes and describes fire-driven fluid flow through solving a version of the Navier-Stokes equations for low-speed, thermally driven flow. FDS is more widely utilized for smoke and heat transfer from fires. The large eddy simulation (LES) approach is used to describe turbulence, in order to properly resolve the physics while decreasing computational time. The combustion model utilized by FDS and for the purpose of this research is a “single step, mixing-controlled chemical reaction which uses three lumped species.” Air, fuel and products are defined as the lumped species by default.[6] Any additional species or reactions that are not mixing-controlled will be defined in the input file as needed.

Parallel Processing can be implemented with multiple mesh domains and each mesh is assigned to a single core processor. In order to benefit from parallel processing, the number of computational cells per mesh must be distributed relatively equally. For clusters of computers, such as UMD’s high performance computing cluster, Deepthought2 (<https://hpcc.umd.edu/>) which is utilized for this project, FDS uses the Message Passing Interface (MPI), where the flow field in each mesh is computed as a MPI process. [6]

Validation involves comparing model predictions with experimental measurements and quantifying the differences when uncertainties between model inputs and measurements arise. FDS has been validated for many purposes, but specific to this research are smoke detection and combustion models. Comparison of smoke transport with UL 217 test data and further validation with full-scale multi-compartment tests indicates that FDS is capable of properly predicting smoke detector activation, when coupled with proper lag corrections. In addition, research from Cleary demonstrated that FDS can predict conditions that a sensor might experience in a real fire.[7]

1.3 Machine Learning Background

“Artificial Intelligence” (AI) and ML tend to be used interchangeably, however in a more technical sense, ML is a branch of AI that allows the machine, which in the scope of this research is a computer, to “automatically learn” from data without needing to explicitly program the results. AI can be achieved through ML.[8]

ML can be broadly separated into three categories - supervised learning, unsupervised learning and reinforcement learning. Supervised learning is when data is given to the algorithm with labels and the model is trained on this labeled data to learn how to predict the label of a sample that has never been seen. An unsupervised learning approach does not have labeled data, and the model is tasked to identify patterns in the data based on the tools in the model itself. Lastly, reinforcement learning is given no prior knowledge, therefore learning and adjusting the policy as the model progresses. Rewards and penalties are received to allow for the model to adjust towards the path that gives the greatest reward.

Different problems lend themselves better to different categories of ML. Supervised learning will be used for this research, as the data has known labels, i.e. heat release rate, location and fuel type, which can link these labels to FDS outputs.

Some types of ML problems can fall under the category of regression (supervised learning), classification (supervised learning) and clustering (unsupervised learning). Regression and classification are the most pertinent for the scope of this research. Regression is defined as technique that helps to identify the relationship between variables and provides a numeric estimate, whereas classification identifies a category that an observation fits in. Clustering does not use predefined classes for categorization purposes.

1.4 Scope of Work

This work is focused on modeling a representative array of fuels in various locations for an aircraft hangar. The information used in the fire simulations is gathered from the scientific literature and adapted to fit the desired length and criteria outlined in the simulation.

A total of six fuel packages are analyzed at six locations, with these locations being labeled A, B, C, D, E and F. Locations A through E are along the center line of the hangar and location F is in the corner closest to the hangar door. Among these locations and fuel types, conditions vary from flaming combustion, flaming combustion with wind and flaming combustion with beams located at the ceiling. Smoldering fires are not modeled for this research due to the very specific fuel properties that are not readily available in literature. These simulations are outlined via the test matrix in Table 1.1 and a corresponding ML algorithm will be

applied to the simulated data.

Table 1.1: Test Matrix

Flaming Combustion WITHOUT Wind						
Fuels	A	B	C	D	E	F
Douglas Fir	X		X		X	X
0.5xHRR			X			
0.25xHRR			X			
Polyethylene	X		X		X	X
2xHRR			X			
4xHRR			X			
Propane	X		X		X	X
0.5xHRR			X			
0.25xHRR			X			
Large Box with Paper	X		X		X	X
0.5xHRR			X			
0.25xHRR			X			
JP-8	X		X		X	X
0.5xHRR			X			
0.25xHRR			X			
Polyurethane Foam GM21			X			
Flaming Combustion WITH Wind						
Fuels	A	B	C	D	E	F
Douglas Fir	X	X	X	X	X	X
Polyethylene	X	X	X	X	X	X
Propane	X	X	X	X	X	X
Large Box with Paper	X	X	X	X	X	X
JP-8	X	X	X	X	X	X
Flaming Combustion with Beams						
Fuels	A	B	C	D	E	F
Douglas Fir	X	X	X	X	X	X

Chapter 2

Literature Review

A literature review was conducted to provide an overview of current knowledge and relevant strategies that incorporate CFD modeling and/or ML in the fire sciences. In addition, background will be discussed that will allow for an understanding of relevant ML terminology. This is not intended to be an all-encompassing review.

2.1 Inverse Modeling

Inverse modeling calculates a set of observations from the factors that produced them and can be seen as a first approach to backing out information from a set of input data. Andrew Neviackas studied this topic in his Master of Science thesis in 2007 [9]. Fire image technology combined with fire modeling was performed for real-time analysis, also aiding in the purpose of situational awareness.

An inverse fire modeling algorithm (IFM) uses MATLAB with the objective being to estimate the HRR from the upper layer temperatures, specifically is the fire in the growth, steady state or decay phase. The data needed for these calculations is obtained through thermal imaging cameras.

The McCaffrey, Quintiere, and Harklerod (MQH) correlation is a useful way to estimate the relationship between HRR and temperature, but pertinent information might be unknown for a given scenario, which is where genetic algorithms (GAs) are utilized. This research uses GAs which work on the principles of Social Darwinism. The algorithm is seeded with an initial population and the population is made up of parameters such as mass loss rate (MLR), door width, door height, window width and window height. A measure of the quality is obtained and a new population is created from the old. The better fitted parameters “survive”

and “reproduce” until a threshold is reached. Single and multiple compartment tests were examined with a ceiling height of 3 m and estimation success was obtained when there was only one parameter that was changing, for example, the vent configuration was known and the MLR was not.

Single compartment tests successfully converge, especially when there is only one unknown parameter. However, the IFM could not track the jump in fire size that FDS predicted. This methodology is mainly used for steady state fires and needs to be developed for transient fires. [9]

Overholt and Ezekoye acknowledge Neviackas’ work, while noting that it is limited to a constant time averaged HRR and is computationally expensive, so they developed a new inverse HRR solution methodology [10]. The HRR is parametrized using a piecewise linear function, allowing the problem to be treated as a least-squares problem where error between the measured and predicted values is minimized. The vector of a single HRR variable is desired and the vector of HRR contains transient HRRs. A predictor-corrector method was used with input times and the hot gas layer temperatures. Overholt and Ezekoye also used the MQH correlation, as this is a linear problem that is easy to invert, which was also seen in Neviackas’ work. The steps for inverse HRR search are described below and seen in Figure 2.1. A zone model approach using CFAST was used with a methane design fire for data acquisition. Python was used for the implementation of the model.

1. Predictor step computes a $\Delta\dot{Q}$ for a temperature difference between measured and predicted at every time by using $\frac{dT}{d\dot{Q}}$ found from the MQH correlation. The next value is computed using $\dot{Q}^{k+1} = \dot{Q}^k + \frac{dT}{d\dot{Q}}^{-1} * \Delta T$.
2. Corrector step is performed by running the CFAST model with HRR values derived from MQH to find the temperature at the next instance in time.
3. If the error is less than 10^{-3} , then \dot{Q} is returned. If not, then steps 1 and 2 are repeated.

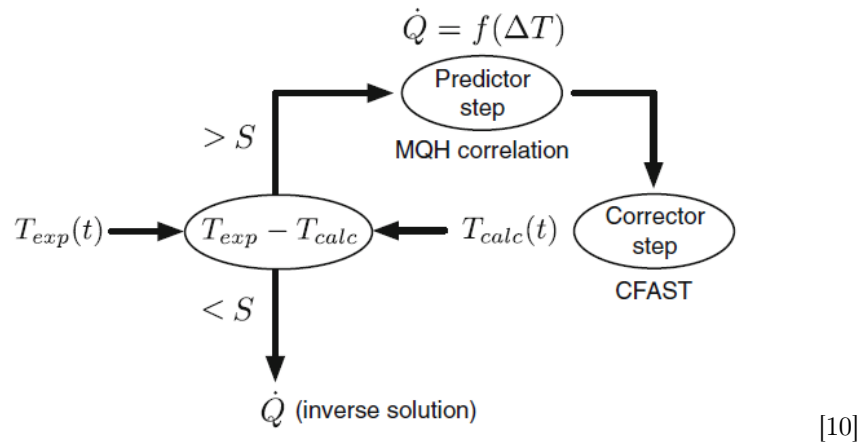


Figure 2.1: Flowchart of inverse HRR search procedure as described by Overholt.

The predictor-corrector method allows the inverse methodology to be used on problems that exceed the limitations of already existing correlations, however this method still overpredicts when there is a “near sudden step change” in the HRR.

There is good agreement with this method when there is a complex HRR curve. The results for the steady state compartment data are close to/within the uncertainty bounds of the experimental data and experimental data from the University of Texas at Austin on transient compartment fires yields relative errors between nominal and inverse HRR between 7 % and 19 % [10]. However, the main limitation is that the inverse HRR solution depends on the boundary conditions, so they must be known and prescribed.

2.2 History of Machine Learning and Fire Detection

ML techniques allow for the ability to determine information about a situation with less explicit inputs, such as boundary conditions. Okayama et al. were one of the first research groups that used neural networks for fire detection, using a combination of odor sensors and back propagation neural networks in 1992. They were able to distinguish between smoldering fires and environmental sources, however this only resulted in limited success. [11] Other first approaches can be seen in the work of Ishii et al. A time-delay neural network was used that allowed for a better ability to capture the transient nature of fire. A time delay-neural network is an extension of a multilayer perceptron (MLP), allowing a neuron to store the history of the input signal based on time delays. [12] Each neuron can access the input value at $T+1$ (where T is the number of time delays Δt) points in time, allowing for the neuron to detect relationships between current and former input values, thus being able to adapt to a set and/or sequence of patterns. [13] Temperature, smoke and CO concentrations were used to calculate inputs to the neural network (such as HRR) and is similar to the “hybrid first-principles-neural network method”. This method includes available prior knowledge about the process with a neural network which allows for the ability to interpolate and extrapolate easier and provides promising methods for fire detection. [14]

This was built upon by Serio et al. in 2000, where actual experiments are performed and coupled with ML. The tests were conducted in a 600 m^3 laboratory combustion facility, with the use of Fourier Transform Infrared (FT-IR) spectroscopy. It is stated that a typical advanced neural network has three layers of processing nodes with weighted connections between the layers. A Learning Vector Quantization Network (LVQ) was utilized with the Kohonen hidden layer to learn and perform classification. The inputs to the network were the concentrations from the FT-IR measurements and the outputs were the classification of

the fire type. LVQ has the ability to extrapolate if well trained and this example correctly predicted 96% of the fires, with the main difficulty being in classifying the transition from smoldering to flaming fires. These results are promising for the purpose of early fire detection. [15]

There are many different ML algorithms that are utilized in fire science. The most common use of ML is for detecting the presence of a fire in a space - for example, are there anomalous conditions in the room? Principal component analysis (PCA) is a useful tool when there are measurements from many sensors at once, and allows for the ability to determine and use the most influential sensors. A study by McAvoy et al. explored how PCA could be applied to fire experiments, for the use of fire detection. Three PCA components were used and allowed for increased detection of smoldering fires and reduced detection time in both flaming and smoldering fires. [16] PCA is a dimensionality reduction method that seeks to retain the relevant information of the data set. It projects the data onto a plane and attempts to minimize the distance of the data points to the line or maximizes the distance from the projected point to the origin. The slope of the first principal component indicates what variable is more important and this process is repeated until the data is sufficiently reduced, which can be determined by finding the “explained variance” of each component, which indicates the value of the information content of each component. When the proportion of explained variance reaches a predetermined threshold value, the data-set can be said to be sufficiently reduced. This approach is beneficial for discrimination between flaming fire, smoldering fire and nuisance sources, which is not of interest for this study. However, the use of PCA is beneficial for a multi-sensor approach and could be useful in determining what sensor locations are indicative of the fire location and which sensors can be ignored, and this approach could be taken at a later time.

Most recently, ML has been implemented with visual fire detection mechanisms and coupled with already existing cameras present in a particular space. For example, algorithms examine specific characteristics such as color, contrast, brightness variation and movement. The most common use of ML is applied to image analysis of fire and non fire scenarios, which provides information on if a fire is present in the space and not an indication of the magnitude of the fire scenario, which does not assist with situational awareness. This quicker detection clearly provides a benefit in that there is a reduction of the chance of fire spread, but could be extrapolated for the benefit of identifying fire location.

Another main driving force behind incorporating ML into fire detection is to reduce false alarms and create a dependable fire detection system with high accuracy and reactivity, which requires the use of a dynamic threshold for applying rule-based algorithms. The values of ML are fixed after training, so flexibility must be incorporated in order to resolve this issue by introducing a multi-functional AI framework - essentially

taking a “learn as you go” approach. When only looking at visual data there is an issue with blind spots, as the fire could be shielded from the view of the camera, confusing abnormal bright light sources with fire and detection delay due to the small flame size at the start of the fire that are too sensitive for the cameras to detect. The proposed system for this research includes Internet of Things (IoT) gateways, a fire detection server and a software-defined networking controller. The IoT is “a system of interrelated, internet-connected objects that are able to collect and transfer data over a wireless network without human intervention”[17], where the IoT gateways send temperature, humidity, gas sensors and camera data to the AI framework.

If forecasting or prediction is the goal, then either recurrent neural network (RNN) or Deep learning (DL) should be implemented. A RNN takes the outputs of the model and feeds them back in as inputs. Forecasting future sensor output is a supervised learning technique that utilizes artificial, recurrent or convolutional neural networks, in addition to DL architecture. Supervised learning is frequently used in ML and it involves finding the best values of the parameters to minimize the cost function.[18]

Saeed et al. describe convolutional neural network based early fire detection in greater detail [19], but first discuss wireless sensor networks (WSN) and why this approach is not always ideal. WSN incorporate spatially dispersed sensors that monitor environmental conditions and forward the data to a central location. WSN strategies to detect fire need a large amount of data. As for most detection methodologies, lag time and small fire size should be considered as a hindrance for early fire detection. When utilizing video fire detection, the whole image is required in addition to an intensely accurate algorithm. The combination of these two requirements takes time to minimize false alarms, thus making it less ideal for early fire detection.

A way to improve upon this is method uses the first proposed model, Adaptive Boosting (Adaboost) in combination with many multilayer perceptrons (MLP). The data utilized in this model is from fire sensors. Adaboost is used to combine many weak classifiers into a single, stronger classifier and this is done by putting more weight on “difficult to classify” instances and less weight on those that were handled well.[20] A MLP is a class of feed forward neural network.

A neural network consists of nodes and connections between the nodes. It starts with unknown parameters that are estimated using back propagation. The inputs, which can be seen as a vector x , are multiplied by a vector of weights by $W^T x$ to get a value. This value is run through the activation function and scaled by adding a bias term. Each node in the hidden layer uses a different portion of the activation function. Weights are parameters that are multiplied and biases are parameters that are added. A diagram of a simple three layer, feed forward neural network can be seen in Figure 2.2. Each circle within the hidden layer h_n represents a node.

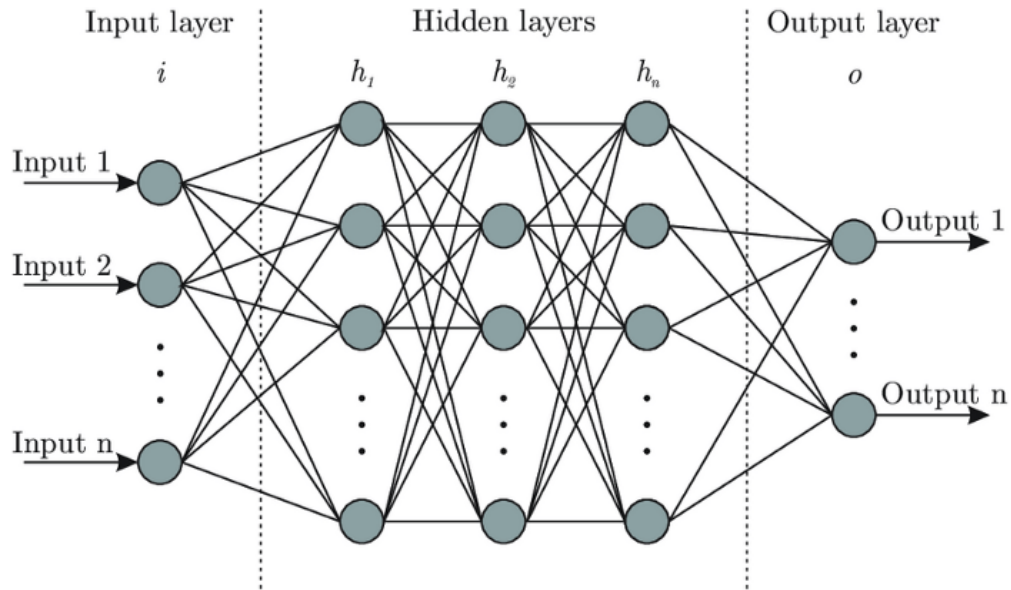
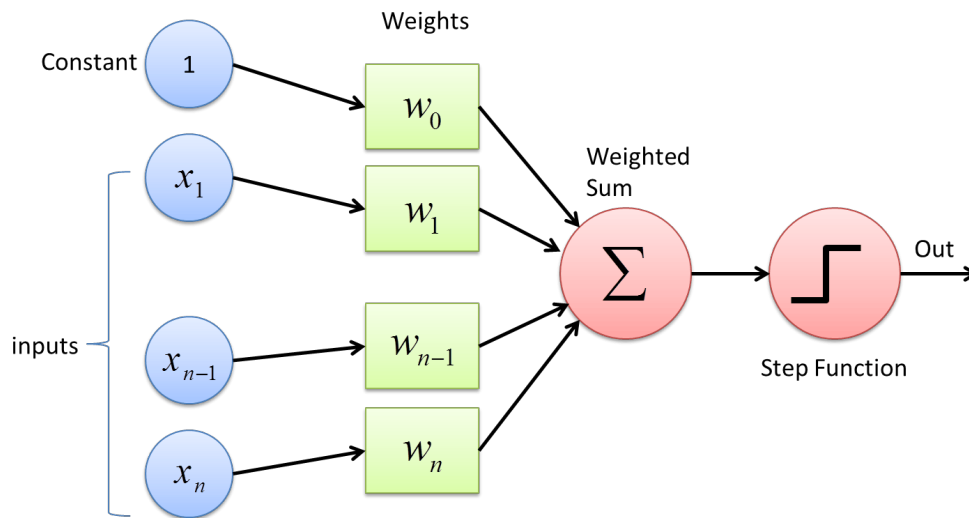


Figure 2.2: Neural network diagram.

A perceptron is similar to a node and consists of input values, weights and biases, a net sum of said weights and biases, and are passed through an activation function, seen in Figure 2.3. The larger the weight value, the more importance the model places on the value and the specific activation function determines if the data will get transferred to the next layer of the network.



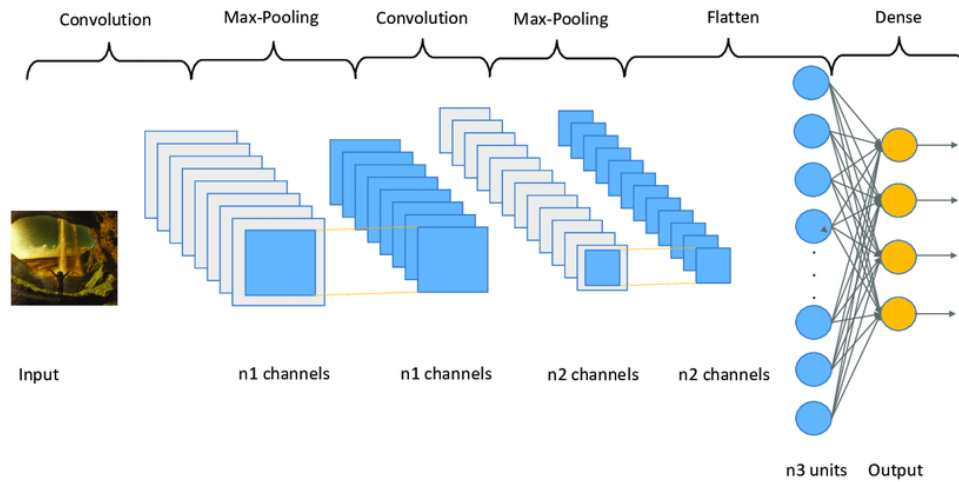
[21]

Figure 2.3: Description of how a perceptron works.

The steps for this method include pre-processing the data to get rid of the noise and then using the Ad-

aboost algorithm. Four different learning algorithms were used - gradient descent with adaptive learning and backpropagation, gradient descent with momentum and adaptive learning rate backpropagation, conjugate gradient descent with backpropagation and fletcher reeves, and lastly broaden Fletcher Goldfarb Shannon. The purpose was to evaluate accuracy against different parameters.

The second model is Adaboost-LBP (local binary patterns) which is also used to assist the deep CNN (which is the third model). Adaboost LBP is a visual descriptor used for classification that utilizes images and videos for detection. It was developed in 1994 and has been useful for text data classification. Convolutional Neural Networks (CNN) divide the problem into sub-parts until the final solution is reached. CNNs are better for fixed windows of time, as opposed to the usage of RNNs for unknown windows of time. CNNs have a “sliding window” of weights instead of having weights for each individual time point. Every point in a convolutional network is a “receptive field which is set of units of previous layers.”[19] Figure 2.4. shows a representation of a simple CNN when the input is an image. The results of Saeed et als research shows that boosting yielded better prediction results.



[22]

Figure 2.4: A simple CNN representation.

The research and techniques discussed above show that previous attempts to incorporate ML have been done experimentally or utilizing video/image analysis. The use of experimental data allows for discrimination between smoldering and flaming, however the current data in the literature is not detailed or accurate enough to model the physics of smoldering in the CFD framework.

2.2.1 Recent Machine Learning Applications in Fire Protection

The following section will discuss the modern applications of ML, specifically the data used for inputs, techniques, results and what can be built upon.

2.2.1.1 Hierarchical Temporal Memory

Hierarchical temporal memory (HTM) ML algorithms take in real-time data that enables the detection and modeling of spatial and temporal patterns, making this an unsupervised learning approach. They are based on memory and are considered a “biological neural network” that can adapt to changing statistics. Ryder et al. used HTMs for fire state determination, and used experimental data as the inputs. HTMs are an example of continuous learning and are configured to determine if the system is experiencing anomalous conditions before the analysis of the next time step. Variations in threshold values may occur for specific fire scenarios, building environments and sensor locations, which makes it necessary to have an approach that can deal with real-time data. Learning can be “turned off” which could be beneficial for real world applications, so the system does not learn abnormal conditions as normal - for example when maintenance is being performed. Sensor location was determined to be important because the majority of the anomalies that were not detected were from sensors below the ceiling or on the wall, as they were not able to pick up changes in temperature and relative humidity as early. This method does best with detecting anomalies in larger fires. [23]

2.2.1.2 Artificial Neural Networks

ML has allowed for ways to combine the accuracy of CFD with the computational speed of zone models, as experiments can be difficult and expensive to perform. The following approach explores the ability to use FDS for inverse problems, determining what simulation inputs produced the outputs.

An artificial neural network (ANN) can be trained to determine transient HRRs from transient thermocouple measurements. This training and mapping of FDS outputs to inputs makes the inversion problem quicker to solve. Buffington et al. created a model of a 6 m by 5 m by 2.6 m compartment, with a door that was represented by a hole in the wall. This model was used to pretrain the ANNs that eventually learned to predict FDS results. It is ideal to create “non-parametric” description of HRR curves, therefore the functional form of the HRR was not specified ahead of time, which allows the design fires to capture the variability in burn tests. This approach can create unphysical, noisy results, so a Gaussian process is used

with the goal being to describe a probability distribution of reasonably smooth functions to create a training set. This results in transient HRRs centered around 150 kW with a standard deviation of 90 kW. The bandwidth was set to 30 seconds for smooth HRR curves and 60 seconds for more rapidly changing HRR curves. Post-processing restrictions were implemented, such as taking the absolute value of the HRR and forcing the HRR curve to zero after the end time has been reached, indicating extinction. The Python package Scipy was used to “draw random HRR ramps, write them into FDS or CFAST and run the simulations”

This research predicted both temperature from HRR and HRR from temperature and the process for both approaches are the same. The neural network architecture that was utilized for this specific problem incorporates an input and output layer that contains 91 nodes which is the length of the time array. Inputs and outputs are either time series temperature or HRR data. Four hidden layers were used in this feed forward neural network, each with 128 perceptrons, with the rectified linear unit (ReLU) being the activation function of choice. ReLU returns the input if it is positive, and otherwise returns zero and helps to deal with the “vanishing gradient” problem. This occurs for certain activation functions when applying very large or small weights, as the derivative is calculated as zero and the network is not able to learn further. [24] ReLU incorporates a non-linearity and behaves well with extreme weights, as the derivative at any value larger than zero is the same. This activation can be seen in Figure 2.5. In general, it is beneficial to have an activation function with a nonzero derivative at large values. Mathematically, $\frac{\delta \sigma}{\delta w^T x} > 0$ where σ is the activation function and $w^T x$ represents the weights applied to the inputs.

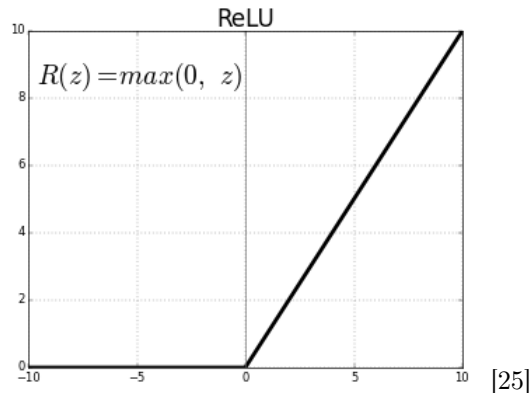


Figure 2.5: ReLU activation function.

Continuing with the discussion of the work of Buffington et al., the network is trained using Adam (adaptive moment estimator) as the optimizer, which reduces the time to train the model. This uses a combination of stochastic gradient descent (SGD) with momentum, accelerating the gradients in the proper

direction. Additionally, an adaptive step size is used which is helpful for regions with flat gradients. This allows the optimizer to get out of the “slow regions” by taking larger step sizes, and then changing back to the previous smaller step sizes for the remainder of the optimization.

The model is trained with a batch size of 31 and the validation split is 20 % of the data. Buffington et al. discussed the benefits of transfer learning for reducing the required number of iterations through the training set. This case uses 20,000 CFAST cases for initial training. Without transfer learning, the ANN is initialized with random weights and biases at the start of training. However, if using transfer learning, once the first emulator is trained its weights and biases are used to initialize the rest and assumes that “an agent can learn a task more efficiently if it has already been trained to perform a similar task” [InverseProblemsML] and can be seen in Figure 2.6.

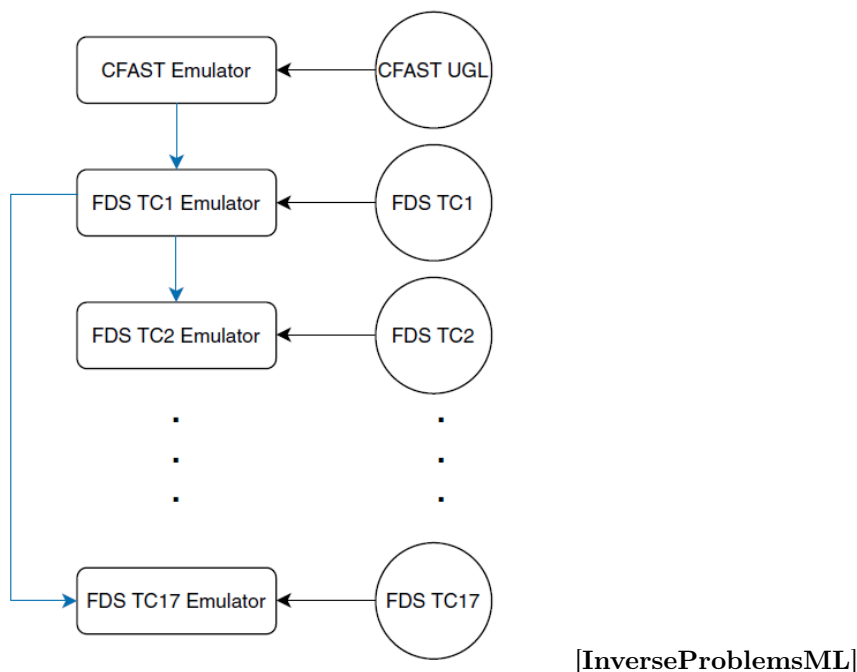


Figure 2.6: Representation of transfer learning. The circles represent outputs from fire simulations, rectangles represent ANNs, black arrows represent direct training and blue arrows represent knowledge transfer.

ANNs are trained using CFAST predicted upper gas layer temperatures. The mean absolute error (MAE) is 6 % of the average HRR which could be due to the fact that there is no guarantee that a temperature output implies a unique temperature input. Additionally, there is difficulty in telling when the HRR goes to zero since the temperatures in the compartment remain elevated for a while after the fire is extinguished and most error occurs when slope of HRR curve rapidly changes signs.

A key takeaway of this study is that HRR inversion appears more accurate for burner fires than pool fires since FDS is based on burner fires. Further research can be done on the effect of compartment geometry on the impact of transfer learning, in addition to incorporating fine mesh results. [InverseProblemsML]

2.2.1.3 Support Vector Machines

Another approach to incorporating ML with fire detection is through the use of a support vector machine with dynamic time warping kernel function (SVM-DTWK). [26] Dynamic time warping (DTW) can deal with different length sequences that are shifted by time to determine the best relationship between two elements in a given sequence. A support vector machine (SVM) is a supervised learning algorithm that can solve classification problems and can overcome “overfitting” and the “curse of dimensionality”. Overfitting occurs when the model learns the details/noise of the training data so well that it negatively impacts the performance on unseen data. The “curse of dimensionality” refers to the error increasing with an increasing number of features.

The basic form of the decision boundary of the SVM is

$$y(x) = w^T x + b \quad (2.1)$$

where x is an input vector, w is the coefficients of the input vector and b is the bias. The end goal is to maximize the margin between unique regions and correctly classify the largest number of points.

Baek et al., consider temporal dynamics in multiple sensors that pick up different fire signals, allowing for the differentiation between flaming, smoldering and heating fires. Certain sensors react better for certain fires but the type of fire that can occur in a given setting is not known ahead of time. The use of video detection can result in late detection time since pixels rely on the fire being highly developed to pick it up. However, a multi-sensor fire detection approach that combines temperature, CO, CO₂ and smoke production allows for more accurate prediction. For example, flaming fires have a high increase in temperature over a short amount of time compared to smoldering and heating fires

The training phase of the SVM constructs multiple classifiers for each sensor type and utilizes information without prior knowledge of the fire type. The level of comparison is fire detection time and false alarm rate. Experimental data from NIST is used to conduct a description of a real-life fire data set, resulting in the conclusion that hard SVM-DTWK performs better than thresholding methods utilizing PCA and ANNs, which have poor fire detection performance, as PCA techniques may lose the temporal dependency since

each sensor value is considered an independent feature. This approach does not explicitly incorporate the correlation between time-series data and including the dependency among fire sensor signals might help increase the accuracy. [26]

There are more approaches that have been taken, such as classification through the use of decision trees, which involve selecting splitting features and continuing to split until all instances are sorted into the appropriate class, or attempting a random forest approach, which is a combination of multiple decision trees. [27] Techniques have been performed to detect, categorize and identify the HRR of a fire, however the room for improvement such as incorporating temporal dependency has been identified. The utilization of more comprehensive models such as FDS will be explored for the purpose of training a ML model to determine the HRR and fuel type of the fire present in the space. The ability to build upon previous research and its applicability will be further analyzed for large, open spaces, such as a hangar.

Chapter 3

FDS Models and Validations

The purpose of the following chapter is to validate that the devices used in this analysis are working as intended and that the geometry is well-resolved.

3.1 NIST Simulation Validation

Proof of concept was required before beginning large scale simulations of the design hangar of interest. The National Institute of Standards and Technology (NIST) performed an “analysis of high bay hangar facilities for fire detector sensitivity and placement” in 1997. The purpose of this study was to determine if new approaches to fire detection and suppression could lead to quicker response, and thus less damage to the aircraft and equipment present in the hangar. A total of 33 full-scale experiments were conducted using JP-5 and JP-8 (jet propulsion fuel type 5 and type 8) in different geographical locations. The experiments and location that were of greatest interest were JP-8 fires in Barber’s Point Hawaii, reducing the experiments of interest to a total of 13. This “warm climate” experiment in a 15m high facility most closely replicates the environmental conditions of the 12 m tall design hangar of interest.

The hangar dimensions in Hawaii were 97.8 m by 73.8 m by 14.9 m. The roof slopes from 14.9 m at the center of the hangar to 13.4 m on the east and west walls. The 5 degree slope of the ceiling was found to exhibit only small effects on the smoke flow down the slope after verification using a CFD model [28]. It should be noted that the design hangar does not have a sloped ceiling. Analog, addressable, spot-type photoelectric detectors were used and the location of sprinklers were ignored for this validation test. The sensitivity set for the smoke detectors that triggered activation of the alarm was 8.2 %/m smoke obscuration.

[6]. Heat detectors were analog spot type and were programmed to operate at an activation threshold of 57.2 °C.

Draft curtains were installed to enclose an area of 24.4 m x 18.3 m and were 3.7 m deep. Draft curtains essentially reduced the ceiling area and trapped in the smoke over a smaller area before the smoke layer could billow under the draft curtains and begin filling the rest of the hangar ceiling. Detectors installed within the limits of the draft curtain responded quicker than those outside of the boundaries. Five detector fire tests were conducted in the facility, with three of the five tests having the draft curtain in place.

These experiments lead to the conclusion that the zone models and simple correlations that are used to approximate detection response times do not provide good approximations for high bay applications or with the large jet fuel fires. Most of the models are based on experiments with lower ceiling heights or smaller fires, which results in a large difference between buoyancy-driven plume entrainment from that in an aircraft hangar. [28]

3.1.1 Detector Activation Analysis

A diagram of the sensor configuration can be seen below with the addition of the draft curtains can be seen in Figure 3.1. The labels of the smoke detectors in Table 3.1 correspond to the labels seen in this figure, i.e. SD11 corresponds to P11.

A Cleary Photoelectric P1 detector was used in FDS with $\alpha_e = 1.8$, $\beta_e = -1.0$, $\alpha_c = 1.0$, $\beta_c = -0.8$, where α and β are used to calculate the characteristic filling or “lag” times. The default activation obscuration in FDS is 3.24 %/m, however the activation obscuration was input as 8.2 %/m to match the threshold value used by NIST. Heat detectors were not analyzed because for fire sizes less than 2 MW, none of the heat detectors responded.

There are a total of 17 smoke detectors placed in the modeled hangar space, located in the corresponding North, East, South and West locations indicated in the Figure 3.1. To decrease computational time, the only area of interest that was modeled was the area directly enclosed by the draft curtains. The draft curtains were treated as a solid obstruction, since heat loss to the boundaries did not matter for smoke detector activation. The space below the draft curtains was treated as “open” to allow for the smoke to theoretically escape into the remainder of the larger enclosure, had that been included in the model.

The test of interest is Test 2, which is a 0.6 m x 0.6 m square pan of JP-5 fuel with a HRR of 0.5 MW, therefore the fire itself was a 0.6 m x 0.6 m square vent with a HRR per unit area applied to the surface of the vent. The ramp-up time was prescribed according to what was defined in the NIST report. The fire

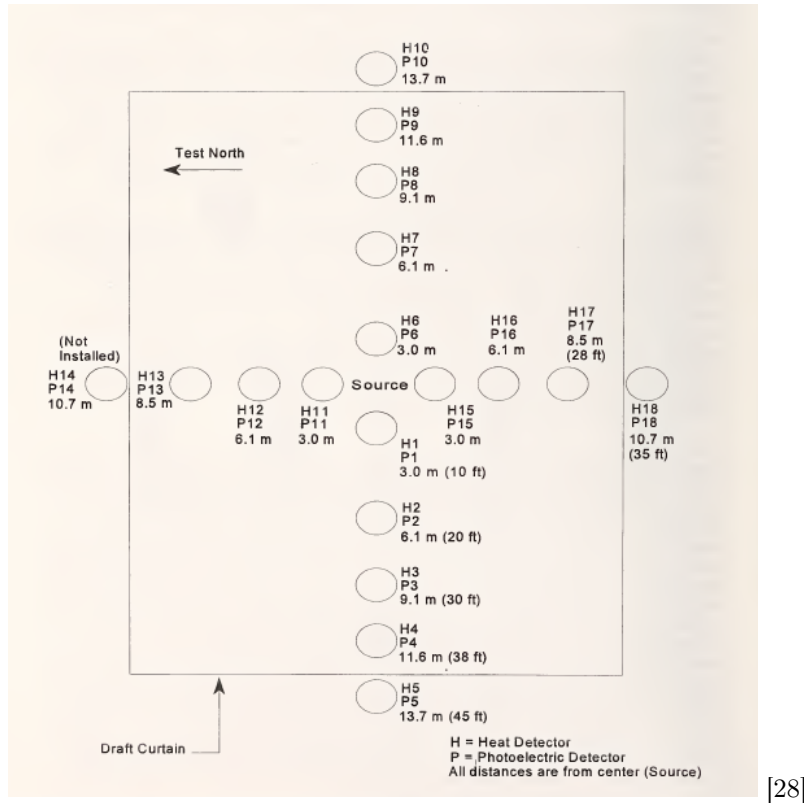


Figure 3.1: Heat (H) and smoke detector (P) locations, plan view 15 m high facility.

was “assumed to grow linearly from zero at the start of each test and reach its steady state value 60 seconds after full pan involvement”. [28] Full pan involvement for Test 2 was 10 seconds.

There is a 1 m x 1 m x 0.5 m volume encompassing the burner and a region that extends 1.5 m from the ceiling that is well resolved to properly account for the development of the products of combustion directly above the fire and at the ceiling, where it matters for precision of detection. The grid resolution inside the well-resolved regions ranges from 0.05 m above the burner to 0.125 m at the ceiling. The reason for this jump in grid size was to decrease computational time, while still allowing for the details to be captured. The grid resolution outside of this region was 0.25 m.

The results for the FDS model versus the experimental NIST results is seen in Table 3.1. The minimum percent error is 9.4 % and the maximum percent error is 90 %.

Table 3.1: Comparison Between FDS and NIST Smoke Detector Activation Times

Location	Label	FDS (s)	NIST (s)	% Error
North	SD11	40.9	27	51.5
-	SD12	50.1	44	13.9
-	SD13	76.0	40	90.0
East	SD6	37.3	23	62.2
-	SD7	52.6	35	48.9
-	SD8	88.3	-	-
-	SD9	-	73	-
South	SD15	42.3	32	32.2
-	SD16	52.7	60	12.2
-	SD17	54.5	48	13.5
West	SD1	36.5	31	17.7
-	SD2	43.8	60	27.0
-	SD3	60.5	-	-
-	SD4	73.4	81	9.4

Draft curtains significantly shifted the center of the plume to the “experimental east”, which was verified experimentally through the higher temperature reading in the thermocouples [28]. It is assumed that there is a difference in experimental activation versus CFD activation due to ventilation and weather effects in the space that could not be accounted for in the simulation. Note, that the doors were closed for Test 2. The slope in the ceiling was not accounted for in the FDS simulation because the small slope was assumed to be negligible, however this is not the case for fires, as even a slight slope in the ceiling configuration matters for smoke movement.

3.2 Grid Size

3.2.1 Critical Length Scale Analysis

This section will now turn to a discussion of the main hangar configuration of interest. Grid size is important for modeling in FDS, as a grid size that is too large will not resolve the physics of the problem correctly, but too small of a mesh/grid size will greatly increase computational time. Computational cost scales with grid size to the fourth power. There are certain areas of the model that require a finer level of resolution, which requires the identification of relevant length scales.

The relevant length scale, L , for the purposes of this report can be identified as the burner and the door opening. A rule of thumb to ensure the proper design of the computational grid is $L/\Delta x \geq 10$. $L = 1\text{ m}$ for the burner and $L = 7\text{ m}$ to represent the height of the hangar door. The most restrictive length scale is

that of the burner and a grid size of 0.1 m was used to resolve the portions of the hangar that were most critical to capture the physics. This grid size meets the constraints of the rule of thumb, where $10 = 10$. The portions where the finer grid resolution was needed were identified to be the fire plume and the ceiling jet since the smoke and products of combustion would be constrained in these regions. The transport of the smoke and travel across the ceiling is important for detection. Further grid validation was performed in FDS, comparing a smaller grid size of 0.05 m to 0.1 m to determine if the chosen grid size was sufficient. The well resolved regions of the hangar are highlighted in yellow in Figure 3.2. The fire plume region is 7 m x 7 m x 11 m and was designed to have a large enough cross sectional area to capture the entire fire plume even with the addition of wind that will cause the plume to shift or bend. The ceiling jet extends 1 m below the ceiling, with total dimensions of 26 m x 26 m x 1 m. The well-resolved region does not have to capture the entire upper layer of smoke, that will descend and be limited by the height of the door opening.

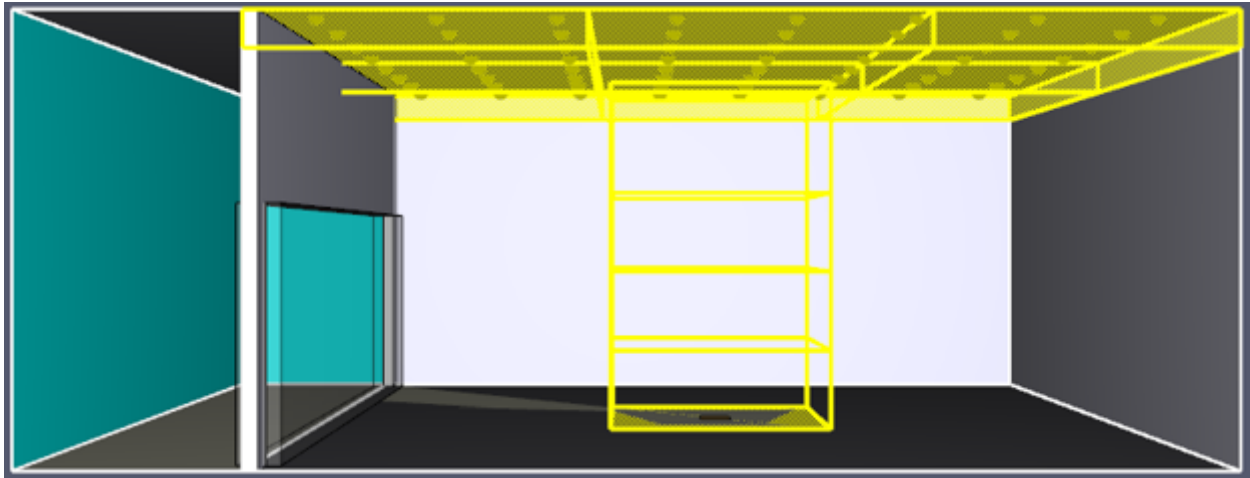
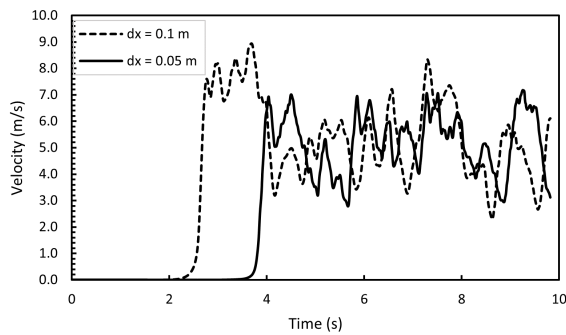


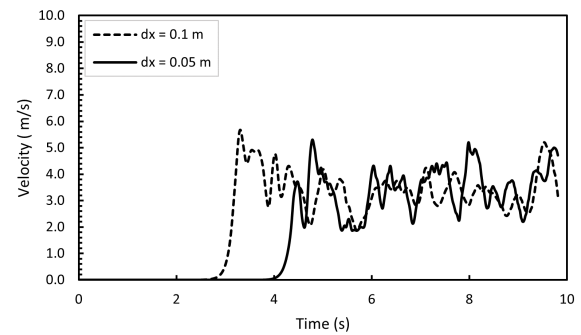
Figure 3.2: PyroSim visualization of a side view into the hangar. The fire is located in the center of the hangar and the meshes that correspond to the fire plume and ceiling jet are highlighted in yellow.

The plume and the ceiling jet velocity are shown in Figure 3.3, with a grid size of 0.1 m plotted with a dashed line and the finer grid size of 0.05 m plotted with a solid line. There is a difference in lag time as the velocity with the finer mesh is picked up approximately less than two seconds later, but when looking at the magnitude of the velocities, the grid resolution does not have a strong impact. The oscillations in the detected velocity for both the plume and ceiling jet are alternating and there does not seem to be the need for a correction to make these velocities align depending on the grid resolution. The maximum and minimum velocities are approximately equal for both resolutions, though they do occur at different times. This leads to the conclusion that the finer grid resolution does not perform better and the maximum grid resolution of

0.1 m is sufficient for use in this problem. It is recommended to avoid a jump in grid resolution from one mesh to another by more than a factor of two. A comparison was made between a less resolved region with a grid size of 0.5 m and 0.2 m to see if this recommendation could be bypassed. Due to the overlap in the measured quantity in Figure 3.4, it was determined that the grid size in the mesh in the non-plume/ceiling jet region could be increased from a factor two, to a factor of five. Therefore, a grid size of 0.5 m was used in all other regions in the hangar to decrease the computational time. The resolution from one mesh to another increased by a factor of five because the space was very large and the physics outside of the plume and the ceiling jet were not as sensitive to the model specifics.



(a) Plume Velocity, taken 4 m above the center of the fire



(b) Ceiling Jet Velocity, taken at the ceiling 6m above the fire, in line with the perimeter of the burner

Figure 3.3: Comparison of plume and ceiling jet velocity for different grid sizes.

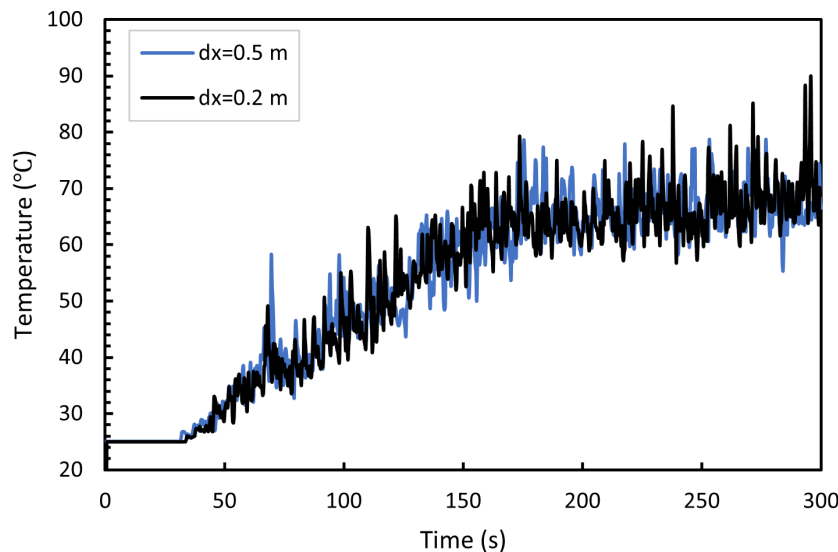


Figure 3.4: Comparison of temperature values when the adjacent mesh is increased from a factor two to five.

3.3 Hand Calculations

The McCaffrey, Quintiere, Harkleroad correlation, known as the MQH correlation is widely used but has limited applications. The correlation is limited to temperatures less than 600 °C, or pre-flashover conditions. The correlation is not preferred for use in large enclosures, as the time for the smoke layer to descend to the opening height needs to be short for the MQH correlation to be applied and be a good fit [29].

However, it was determined to be of value to see if there was any correlation between temperature and HRR for particularly sooty fuels, and see if there was a way to relate the physics of fire growth in a large space to this simple equation that relates temperature and HRR,

$$\Delta T_g = 6.85 \left(\frac{\dot{Q}^2}{A_o \sqrt{H_o} h_k A_T} \right)^{1/3} \quad (3.1)$$

where ΔT_g is the change in temperature of the gas in the upper layer, \dot{Q} is HRR in kW, A_o is the area of the ventilation opening in m^2 , H_o is the height of the ventilation opening in m, h_k is the effective heat transfer coefficient of the boundaries in $\frac{W}{m^2 K}$ and A_T is the solid wall surface area in m^2 .

The equation was rearranged to solve for \dot{Q} , that way the HRR can be backed out and solved for as a function of temperature.

The values used in equation 3.2 are $A_o = 168 m^2$, $H_o = 7 m$ and $A_T = 2,432 m^2$.

$$\dot{Q} = \left(\left(\frac{\Delta T_g}{6.85} \right)^3 (A_o \sqrt{H_o} h_k A_T) \right)^{1/2} \quad (3.2)$$

There is good agreement with the MQH correlation and larger scale fires, such as JP-8, as compared to the other fuel sources of interest for this report. JP-8 is a sooty fuel so there is a distinct, thick smoke layer that is created when the fuel burns. Soot drives radiation heat transfer and increases the temperature of the gas.

3.3.1 Heat Transfer Coefficient

The heat transfer coefficient, h , when used in the MQH correlation is calculated as

$$h = \sqrt{k \rho c / t} \quad (3.3)$$

where k is the thermal conductivity, ρ is the density and c is the specific heat of the material of interest. This indicates that h is only a function of the material properties of the walls and/or ceiling of the enclosure. The rate of heat transfer is limited by the slowest process, which according to the derivation of the MQH correlation, is conduction inside the solid walls. Gas transfers heat through convection and radiation and heat is transferred through the solid via conduction. This simplification of the heat transfer coefficient is not applicable for the intended use of MQH for this research, because convection and radiation play a dominate role in the heat transfer, and therefore is not calculated using equation 3.3.

Convection and radiation play an important role in the physics of heat transfer and are taken into account by adding eight heat flux gauges to representative locations across the ceiling in the FDS simulation. They were not located close to the fire plume, as to not be skewed by the initial rise of hot gases from the fuel source. Total heat flux gauges were used which captured the following quantities:

$$\dot{q}_{net}'' = \epsilon_s(\dot{q}_{inc,rad}'' - \sigma T_s^4) + h(T_{gas} - T_s) \quad (3.4)$$

where $\dot{q}_{inc,rad}''$ is the incident radiative heat flux, ϵ_s is the surface emissivity, h is the convective heat transfer coefficient, T_s is the surface temperature and T_{gas} is the temperature of the gas near the surface. This equation incorporates the radiative and convective energy absorbed at the solid surface. The net heat flux was rearranged to solve for h by dividing \dot{q}_{net}'' by $(T_{gas} - T_s)$, assuming that the radiative portion of this equation was negligible. The calculation for h was performed as a function of time, so ΔT was taken at each time step in the FDS output file. Typical values of h are on the order of magnitude of $0.01 kW/m^2 K$ and the derived value of the heat transfer coefficient for a full-scale JP-8 fire reached a steady state value of approximately $0.01 kW/m^2 K$. The values of h , calculated at the 8 locations where the heat flux is measured is shown in Figure 3.5. The values for h are approximately equivalent regardless of the location at the ceiling. However, when h was calculated for smaller fires, the value changes, which indicates that MQH might not be quantitatively accurate for smaller fires that do not produce as much heat or smoke. There is a limiting case for applicability. This newly derived value of h derived from FDS is then used in equation 3.2 to back out the HRR.

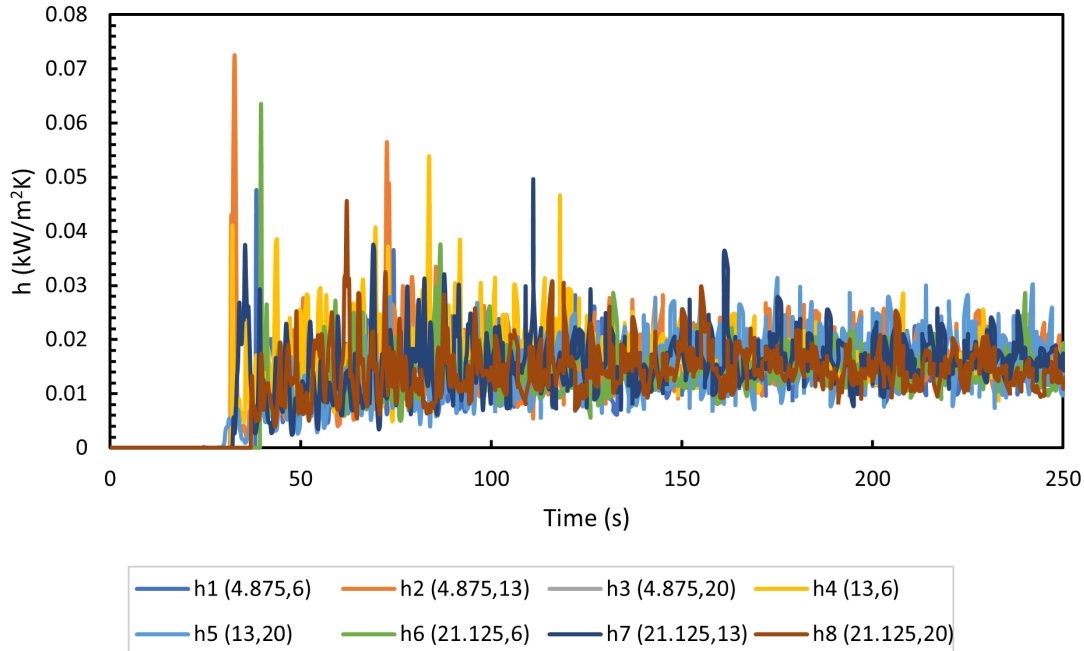


Figure 3.5: Heat transfer coefficient from eight locations where the total heat flux was measured. The (x,y) coordinates in meters for each location are shown in the legend.

Figure 3.6 shows the time variations of the smoke temperature used in the MQH correlation seen in Figure 3.7. Figure 3.7 shows a graph of the prescribed HRR from FDS seen in black, versus the derived HRR using the MQH correlation, in gray. There is almost identical agreement between these curves, with the slight deviation during the initial growth phase.

Figure 3.8 shows the limited applicability of the MQH correlation for backing out the HRR if the HRR is not large enough to begin with. A JP-8 fire was analyzed to limit the number of changing variables, i.e. fuel composition. The HRR was scaled down by a factor of four and an identical analysis was performed as was described above. It can be seen that there is a deviation between the prescribed HRR and the derived HRR for this lower magnitude. An attempt to correct this and allow for better agreement was made using the convective heat transfer coefficient calculated from the full scale JP-8 fire. This increased the HRR but not enough to where it would be considered an acceptable way to estimate this quantity.

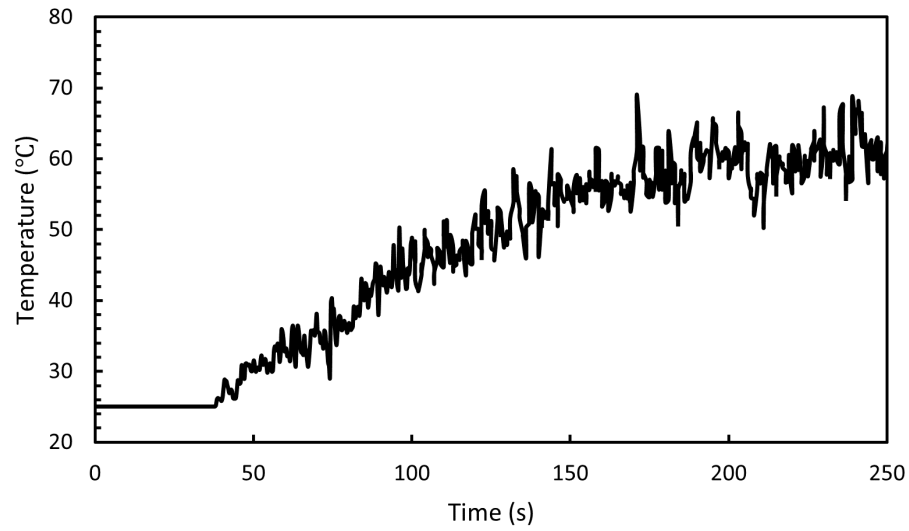


Figure 3.6: Temperature inputs into MQH equation.

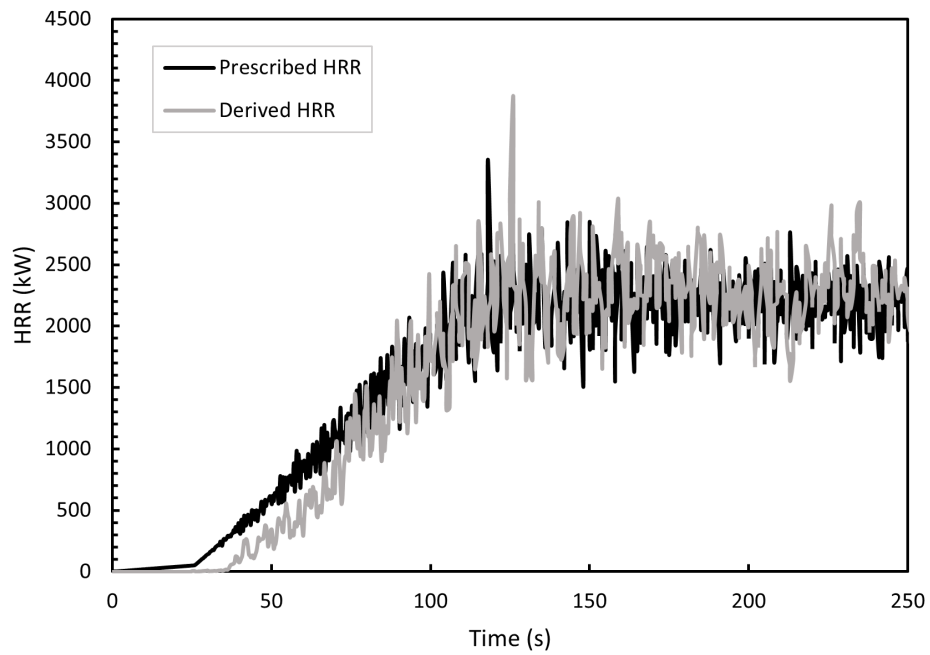


Figure 3.7: Comparison of prescribed HRR in FDS for JP-8 versus derived HRR using total heat flux from FDS in MQH correlation.

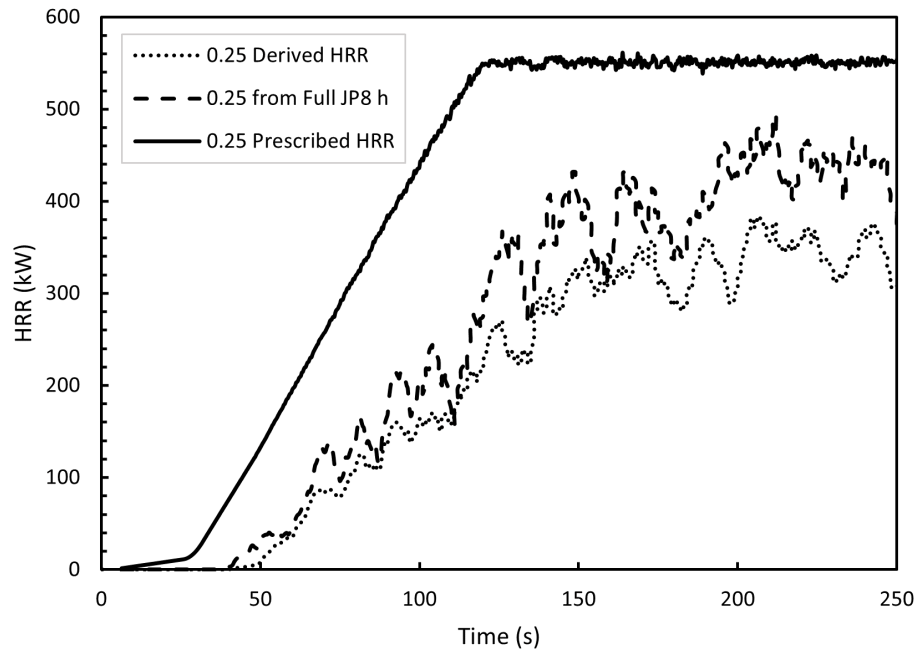


Figure 3.8: Comparison of prescribed HRR (when reduced by a factor of four from the original HRR) in FDS for JP-8 versus derived HRR using heat transfer coefficient from full scale HRR total heat flux and reduced HRR in FDS in MQH correlation.

Chapter 4

Model Inputs

The following section details the model inputs in FDS that will be used for data acquisition for training machine learning models. All simulations were run in parallel on the UMD High Performance Computing cluster, Deepthought2, using one node and up to 15 processors. It was designed that an individual mesh would be assigned to a certain MPI processor. The run time for each simulation varied between 5 and 30 hours.

4.1 Compartment Geometry

The hangar configuration of interest is an F-35 aircraft maintenance hangar. The dimensions are 26 m x 26 m x 12m with a 24 m x 7m opening on one side to represent the door opening. To accurately account for entrainment effects through the door, and later wind effects, an outdoor mesh of dimensions 26 m x 6 m x 12 m was added, which was extended to 10 m in the y-direction for wind. The visualization of the hangar configurations are provided in Figure 4.1 and Figure 4.2. Further material properties can be found in Appendix B.

Note that the hangar is empty and there is no representation of a plane in the hangar space since thermal insult to the aircraft is not a parameter of interest. The only concern is the measurement of the products of combustion at the ceiling.



Figure 4.1: Smokeview animation of hangar configuration; view from the outside looking into the hangar through the open door.

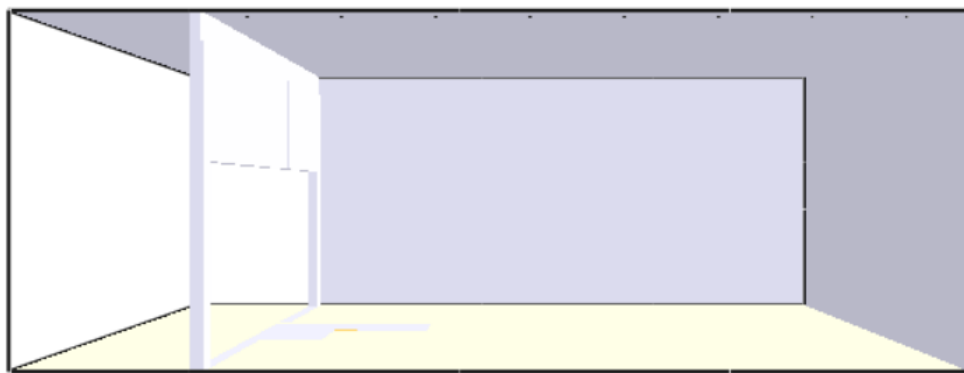


Figure 4.2: Smokeview animation of hangar configuration; side view from the hangar with the hangar on the right and the outdoor space on the left, no wind case.

4.2 Instrumentation Devices and Spacing

The spacing for each device is the same regardless of the type of device, which include smoke detectors modeled with the Cleary Photoelectric P1 detector, temperature sensors, and CO and CO₂ detectors that measure based on volume fraction. The devices are located at the ceiling and spaced in a grid-like pattern which results in an eight by eight grid with a total of 64 of each type of sensor located at the ceiling. It is understood that this is a high density of sensors that is likely not indicative of reality, however, since this is a feasibility study, the high density is assumed to be appropriate.

The array of sensors will be located 1.625 m from each wall and subsequently spaced 3.25 m apart in the x and y direction. The location of the sensors relative to each other and the wall can be seen in Figure 4.3.

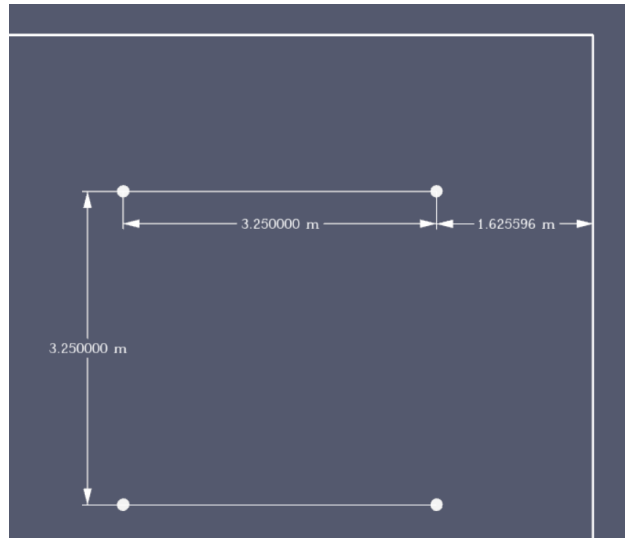


Figure 4.3: Sensor locations at the ceiling, relative to the corner of the north and east wall.

4.3 Fuel Model and Design Fires

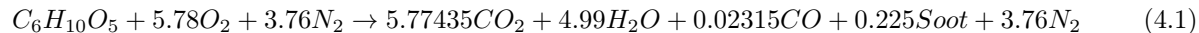
4.3.1 Combustion and Pyrolysis Model Description

Combustion is a gas phase process that refers to the “reaction of fuel vapor and oxygen”, as opposed to pyrolysis, which is a solid phase process that refers to the “generation of fuel vapor at a solid or liquid fuel surface”. In the default simple chemistry, mixing-controlled combustion model, pyrolysis and combustion can be defined the same way. Only one gaseous fuel can be prescribed at a time, which makes it so there is a single gas phase combustion reaction that represents all potential fuels. This reaction is infinitely fast, resulting in the entire chemical reaction taking place instantaneously at the flame sheet [6]. The mixing of the fuel and the oxygen govern the combustion process.

The lumped species approach reduces the number of transport equations that need to be solved by combining species into groups that transport and react together. Fuel, air and products are considered the lumped species. Carbon monoxide and soot yields must be prescribed, otherwise they are zero by default and assume complete combustion conditions. The yields for CO and soot are defined as $y_i = \frac{\text{mass } i \text{ in products}}{\text{mass of fuel reacted}}$. FDS uses the prescribed information from the REAC line to calculate the stoichiometric coefficients. The calculated yields are for well-ventilated, post-flame conditions. Further details can be found in the FDS User

Guide [6] and the Technical Reference Guide [30].

The balanced equation with the proper stoichiometric coefficients for Douglas fir can be seen in Equation 4.1. Details on how to calculate these coefficients can be found in the FDS User Guide [6]



The REAC lines that are used to create the balanced stoichiometric reaction for each fuel are seen below. These reactions occur at the surface of a 1 m² burner inline with the floor.

```
&REAC FUEL = 'Douglas_Fir',  
FORMULA='C6H10O5',  
HEAT_OF_COMBUSTION=16400,  
SOOT_YIELD=0.015,  
CO_YIELD=0.004/
```

```
&REAC FUEL = 'POLYETHYLENE',  
FORMULA='C2H4',  
HEAT_OF_COMBUSTION=43400,  
SOOT_YIELD=0.06,  
CO_YIELD=0.024/
```

```
&REAC FUEL = 'Paper_Box',  
FORMULA='C6H10O5',  
HEAT_OF_COMBUSTION=12500,  
SOOT_YIELD=0.002,  
CO_YIELD=0.029/
```

```
&REAC FUEL = 'JP-8',  
FORMULA='C11H21',  
HEAT_OF_COMBUSTION=42800,  
SOOT_YIELD=0.03,  
CO_YIELD=0.01/
```

```
&REAC FUEL = 'PROPANE',
```

FORMULA='C3H8',
HEAT_OF_COMBUSTION=46300,
SOOT_YIELD=0.024,
CO_YIELD=0.005/

&REAC FUEL = 'POLYURETHANE',
FORMULA='CH1.8O0.3N0.05',
HEAT_OF_COMBUSTION=26200,
SOOT_YIELD=0.131,
CO_YIELD=0.010/

4.3.2 Design Fire

There are five main design fires of interest for this analysis. The sections below will explain why each fire was chosen and why it is believed to be a fuel of interest for an aircraft hangar. The fuel characteristics will be described in greater detail in section 4.3.3.

4.3.2.1 Design Fire 1: Douglas Fir

Douglas Fir represents a transient fuel package that could be present in the hangar on any given day. When receiving large shipments, items will be delivered on wooden pallets and distributed throughout the hangar. It is assumed that the pallets will likely not be immediately discarded and that stacks of pallets will accumulate in the hangar for an extended period of time, therefore posing a fire hazard.

4.3.2.2 Design Fire 2: Polyethylene

Electrical fires caused 21.6 % of reported fire incidents that led to a foam discharge suppression effort between the years of 2004-2019. [31] Electrical cables and cabinets can ignite due to electrical current. There are electrical cabinets in the hangar, so this fuel/ignition source is present in the space. These electrical cables are typically covered in either polyvinyl-chloride (PVC) or polyethylene. Polyethylene is chosen as it is less stable than PVC so it represents a less predictable fire and a greater potential threat.

4.3.2.3 Design Fire 3: Large Box of Paper

The large box of paper represents another transient fuel type. Paper trash as well as paper files can be stored in the hangar or moved around, so this is a representative fuel type. There is a high possibility for multiple items to be present in the trash, however a trash can with multiple fuel types would be beyond the scope of this research. The type of trash can vary based on the day and it is difficult to discriminate fuel type if multiple items are burning at once, even though such would be realistic. Assuming that the paper is the only fuel source present at the time is a conservative approach.

4.3.2.4 Design Fire 4: JP-8

JP-8 is military grade jet propulsion fuel, used in Air Force aircraft, therefore this is an ever-present fuel source in the hangar and is justified to be a design fire. Additionally, a fire involving JP-8 was a design fire of concern when selecting extinguishing systems for hangars. [32] A full F-35 fuel tank contains 10,900 L of fuel. However, the fire will be defined on a heat release rate per unit area (HRRPUA) basis so the driving factor in this model will be the characteristics as opposed to the quantity of the fuel.

4.3.2.5 Design Fire 5: Propane

Propane heaters are not up to code for use in an aircraft hangar, so it is unlikely that a propane tank would be found in the hangar space. However, propane is considered a design fire of interest because it has the same HRR as JP-8, but produces less soot so it provides more of a challenge for the model to discriminate between the two.

4.3.2.6 Additional Considerations

Polyurethane foam is added as an additional design fire of interest, that will be used solely as a test fire to see how well the model learns from the temperature profiles it has seen to extrapolate and predict the HRR of an untrained fuel. Polyurethane foam GM21 is a fast-growth rate fire, which results in a new design curve. In addition, polyurethane foam is a typical material of choice for furniture padding, so if there is any upholstered furniture present in the hangar this fuel would be justifiable.

4.3.3 Fuel Characteristics

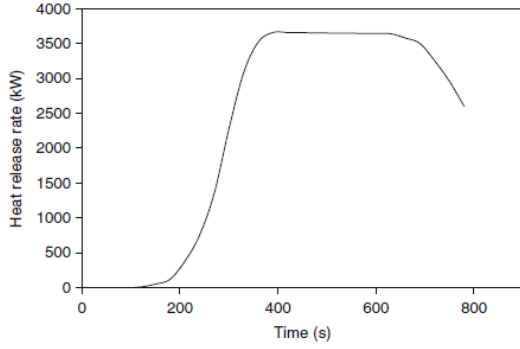
A simple pyrolysis model approach was taken to model all of the fuel sources due to the complex composition of the fuel sources. This model greatly simplifies the pyrolysis model by neglecting the material properties of the fuel. This requires an explicit definition of the gas phase reaction. Each fire was specified via a prescribed HRR, therefore material properties were not needed to be listed in the FDS input file. This treated the model as an ejection of gaseous products from a solid surface. The units of HRRPUA are kW/m^2 .

HRR is a time-dependent function and as such, temperature and concentrations are inherently time-dependent variables because they are influenced by the HRR. When the simulation starts, HRR values are ramped up to model the natural growth of the fire, as it is not realistic for the fire to instantaneously reach a steady state value. Typical growth for a fire can be defined as a t^2 fire and defined in FDS using TAU_Q . However, the fire growth curves from the literature were more complex than a t^2 fire, so a function was created and prescribed as the input.

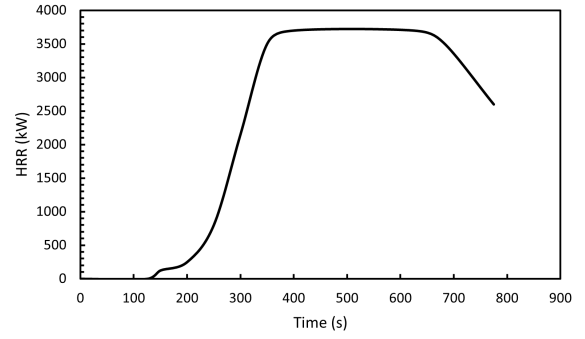
$RAMP_Q$ is specified since the HRR is what is changing as a function of time. T designates the time and F indicates the fraction of the HRR that should be applied at that given time, T. The intermediate time points are accounted for using linear interpolation. Values for F were calculated by taking the HRR at the time of interest and dividing it by the peak HRR from the simulated burn. When F was equal to 1 it is assumed to be burning at the peak value. The specific ramp-up values for T and F can be seen in Appendix B.

4.3.3.1 Douglas Fir

Information on the ramp-up function was extrapolated from Figure 26.73 of the SFPE Handbook which corresponds to the HRR of a typical wood pallet stack with the dimensions 1.22 m x 1.22 m x 1.22 m high. The peak HRR was identified as 3700 kW and the times/fractions were obtained via visual inspection of the HRR curve in the handbook. This method can be deemed appropriate as can be see in a comparison of the curve from the literature versus the recreated/derived curve in Figure 4.4. There is acceptable agreement between these two curves.



(a) From SFPE



(b) From Derived

Figure 4.4: Comparison of HRR curve from literature vs. derived HRR curve.

HRR measurements are from Krasner (1968), where he performed multiple tests where the burning rate of pallets were measured. A relatively constant burning can be seen if the stack is reasonably high, which is the case for the test pallet that was modeled. When applied to nonstandard pallet sizes, the following equation for HRR on a per-unit-pallet-floor area basis can be used, $\dot{q}'' = 919(1 + 2.14h_p)(1 - 1.03M)$ where h_p is stack height in m and M is moisture content in %. However, if h_p is less than 0.5 m this equation will overpredict the HRR.

The entire duration of burning, including growth, steady state and decay will not be included in the FDS simulation. The truncated curve is seen in Figure 4.5.

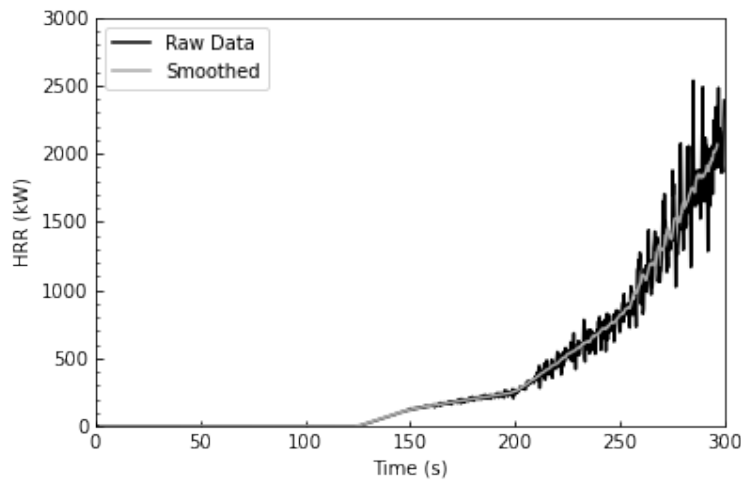


Figure 4.5: Douglas fir HRR curve.

4.3.3.2 Polyethylene (Electric Cables)

Information for the ramp-up function was obtained from the SP Design Fire report for a vertical configuration with no ventilation. Vertically versus horizontally positioned material impacts burning because there is the influence of gravity through melting or “disrupting of the material.” [33] There were three vertical configurations that were compared - open, half open and closed. The closed vertical situation was chosen because the HRR for this scenario fell in between the HRR for the open and half open scenario in magnitude. The HRR reached a steady state value in the configuration of interest once the entire tray was burning, and the fire is ventilation controlled. Additionally, “flame spread will depend on level of thermal attack” [33]. The type of cable was chosen because it had the fastest flame spread, and therefore the poorest fire performance. The HRR curves were obtained from large scale tests and flame spread was observed at 100kW. [33]

The peak HRR for the entire duration of the fire was approximated as 285 kW, however the truncated curve that was used in the FDS model can be seen in Figure 4.6. The peak HRR for the simulation is on the order of 50 kW due to the slow growth phase of the electric cable fires.

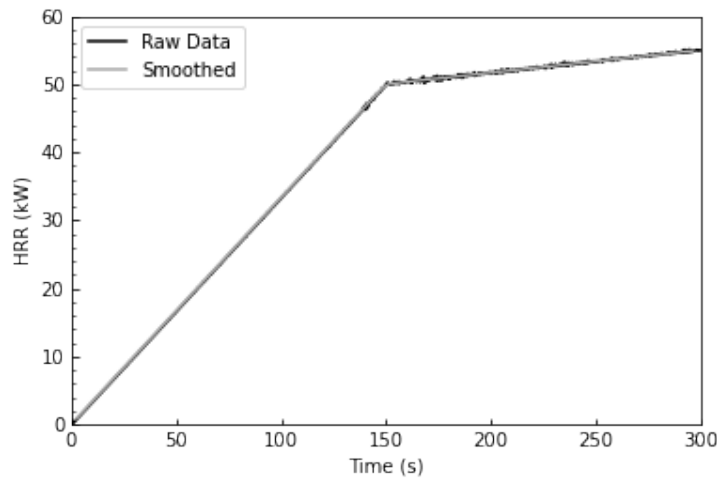


Figure 4.6: Polyethylene HRR curve.

4.3.3.3 Propane

Propane and JP-8 have similar fuel characteristics and for the purpose of this research, the HRR ramp-up values for propane are identical to that of JP-8. The goal is to explore the differences between fuels with identical HRRs and different soot yields.

The design curve input in FDS is seen in Figure 4.7.

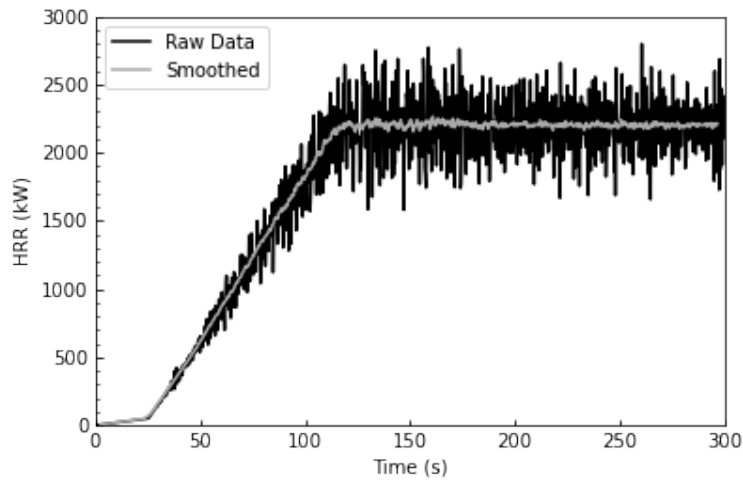


Figure 4.7: Propane HRR curve.

4.3.3.4 Large Box with Paper

Information from the ramp-up values for paper were taken from Test 009 from Nuclear Regulatory Commission (NRC) and NIST. [34] The ignition source was a heptane wick inside the corner of the box. The wick was a piece of nylon rope with 5 mL of heptane applied to it and was ignited with the lighter right after placement in the box. This ignition method is used for items that are easily ignited, but need the ignition source inside the test item itself. “The most common ignition methods involving transient combustible material were welding and cutting, overheated wires and shorts, work lights and smoking materials.” The wick represents these sources. The 0.227 m^3 cardboard box was filled 50% with shredded paper. The peak HRR was measured as 436 kW and the total heat released was 29.95 MJ. [34]

The truncated curve was recreated via visual inspection from the design curve in the NRC/NIST report and is seen in Figure 4.8 below.

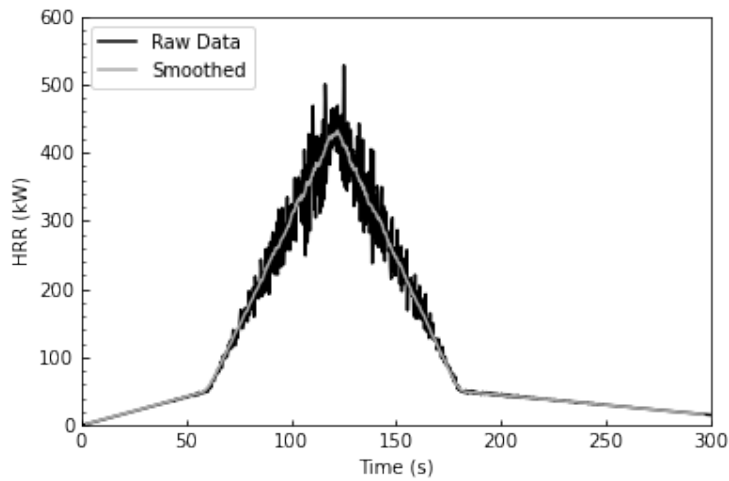


Figure 4.8: Large box with paper HRR curve.

4.3.3.5 JP-8

Information for ramp-up values for JP-8 were obtained from the NIST report where the reference for the smoke detector activation was modeled from. The material properties were obtained from a series of laboratory experiments performed at NIST, using a cone calorimeter and NIST large scale calorimeter. It should be noted that the tests met Military Specifications MIL-T-5624P. [28]

Thirteen fire tests were performed in the NIST large scale calorimeter and HRR as a function of pan-size was determined. Per table 3 in the NIST report, the HRR for JP-8 in a circular pan with a diameter of 1.2 m is 2200 kW. [28]

The design fire curve that showed the HRR for a 1.2 m diameter pool fire from Figure 3 in the report was for JP-5 only. Because JP-5 closely mirrors that of JP-8, the JP-5 curve was scaled up to match the maximum HRR of a 1.2 m diameter JP-8 pool fire. The adjustment was deemed necessary because the pool fire pan diameter was 1.2 m, whereas the design fire curve for JP-8 was given for a 1 m pool fire.[28] The larger the diameter of the fuel the larger the HRR. The truncated HRR curve, recreated from visual inspection, is seen in Figure 4.9.

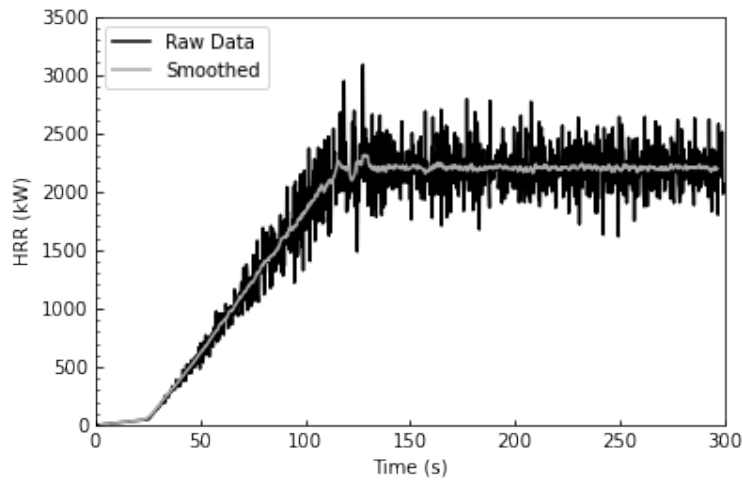


Figure 4.9: JP-8 HRR curve.

4.3.3.6 Polyurethane Foam GM21

Information for the ramp-up values for polyurethane foam was obtained from the SFPE Handbook. [35] Values from Figure 26.114 were visually estimated using curve F21, which is a single upholstered chair comprised of polyurethane foam. The goal of including this simulation is to compare the results to that of the Douglas fir pallet because both are characterized as fast growth fires.

The chair used in curve F21 was made of polyurethane foam that complied with the 1975 California TB 117 standard and was covered in polyolefin fabric. Ignition was achieved using a 50 kW burner placed at the side of the chair which represented a small trashcan fire and provided the smallest time between ignition and peak HRR. Figure 4.10 shows the design curve.

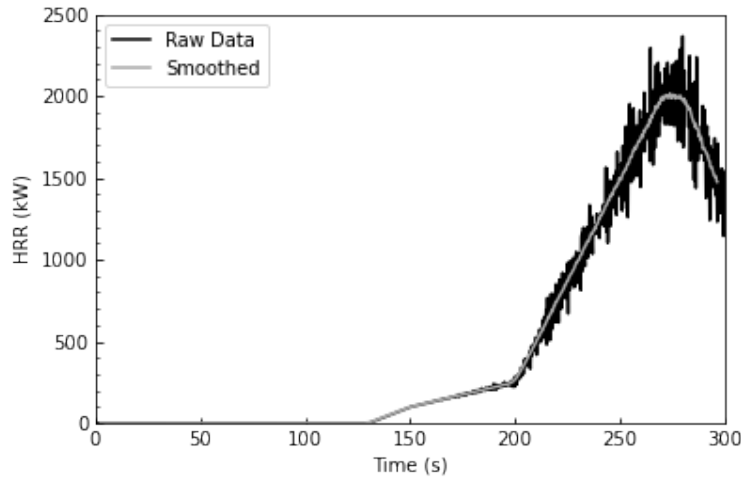


Figure 4.10: Polyurethane foam GM21 HRR curve.

4.4 Differing Conditions

Environmental conditions can change depending on the time of day or year, and as a result these changing conditions needed to be taken into account. They can impact how the fire behaves in a space, which is imperative for analysis in a hangar. The two factors that will be explored are wind and complex geometry in the form of a beamed ceiling.

4.4.1 Wind

Wind data for Tyndall Air Force Base from the Handbook of Smoke Control Engineering [36] was given as 8.8 m/s which was the extreme 1 % design value. This wind velocity was tested for Douglas fir fires in four locations in the hangar. The velocity was too high, disrupting the flames (which could be visually seen in Smokeview) and fire dynamics to the point where the products of combustion that the sensors picked up were greatly impacted. Since this wind speed was the extreme design value, a choice was made to reduce it by a factor of two. The velocity of interest for all flaming combustion simulations with wind was 4.4 m/s, as this allowed for wind effects to play a role, but didn't completely disturb the structure of the flame.

There are three methods for adding wind to FDS according to the FDS User Guide. The “wall of wind” [6] approach was used for adding wind into the hangar. The wind was added on the exterior boundary of the domain, which is the “outdoor” boundary that runs parallel to the hangar door. Additionally, the

outdoor region will be moved 10 m from the hangar door to allow for sufficient time for the wind profile to develop before entering the hangar. The wind will be blowing directly into the hangar using a power law atmospheric wind profile. The default height above the ground, z_o where the wind speed is measured is 10 m and the atmospheric profile power, p is 0.14 for airports, as this environment is roughly open terrain [36]. The equation utilized for the wind velocity using the atmospheric wind profile is

$$u = u_o \left(\frac{z}{z_o} \right)^p \quad (4.2)$$

Construction of an aircraft hangar is assumed to have very loose leakage rates which, according to table 50.2 of the SFPE Handbook, corresponds to an exterior building wall ratio of $A/A_{tot} = 0.0012 \text{ m}^2/\text{m}^2$. [35] The total wall area is 312 m^2 which corresponds to a leakage area of 0.3744 m^2 per wall. The leakage was added into the FDS simulation in the form of an “open” type vent. The total leakage area per wall was divided by four and distributed into 4 separate vents. The vents were offset one meter from the top and bottom edge of the wall and 4 m from the left and right of the wall. Each vent was 0.09 m^2 to give a total leakage area of 0.36 m^2 which is the most appropriate way to distribute the area and that is closest to 0.3744 m^2 .

Specifics of how the wind was added is provided in the FDS input file in Appendix B.

4.4.2 Complex Geometry

Beams can impact smoke development and movement across the ceiling depending on their depth and configuration within the space, which can cause a difference in the time that it takes the sensors to pick up either temperature, smoke obscuration, CO or CO₂ production. W16x31 beams (which have a depth of 40.34 cm) were assumed to be present throughout the hangar. The beams were not spaced exactly according to industry standards and instead were used to divide the ceiling into nine sections. The beams were 9 m from each wall, which lead to a separation of 8 m between adjacent parallel beams which can be seen in Figure 4.11.

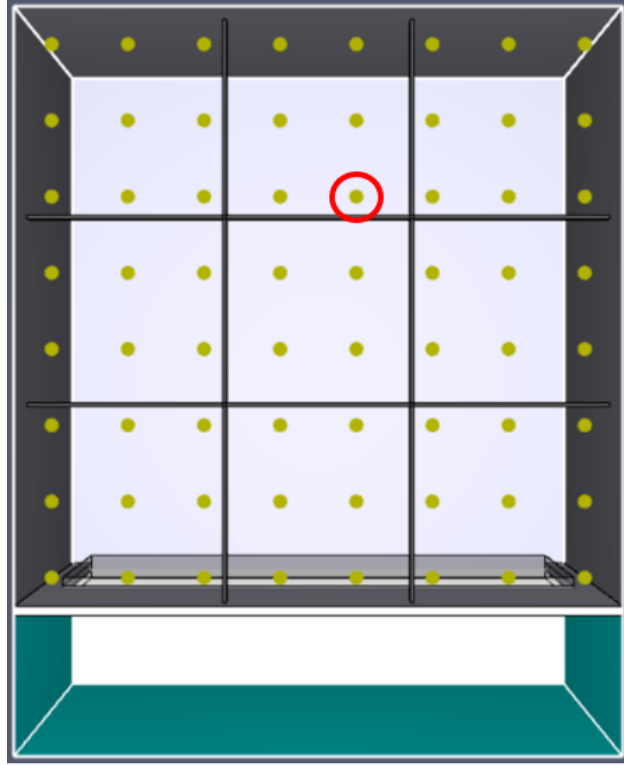


Figure 4.11: PyroSim visualization of hangar configuration with beams, with sensor of interest circled in red.

To determine the effect of beams, the development of combustion products will be compared for beam and no beam cases in Figure 4.12. A Douglas fir fire located in the center of the hangar will be analyzed. The sensor of interest, circled in red in Figure 11, was located outside of the immediate vicinity of the plume, in an adjacent “grid” created by the beams to directly analyze if the time it takes the smoke to fill up the grid directly above the plume and spill into the neighboring grids is impacted.

The trend seen in a single product is representative of the other products of combustion, therefore only CO concentration is shown and Figure 4.12 shows the twenty-point running average of the FDS sensor data. The effects of beams were considered negligible because there is oscillations between the concentrations in the beam and no beam cases. Douglas fir fires were the only simulations that included beams at the ceiling once the impact of beams was concluded to be negligible.

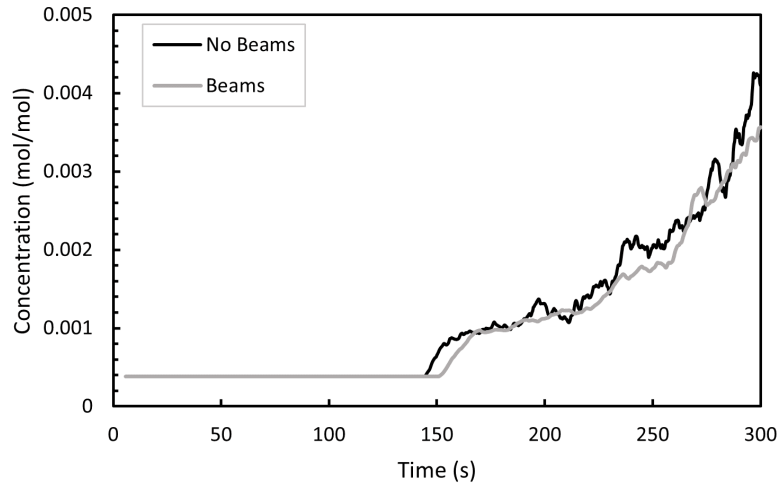


Figure 4.12: Comparison of CO concentration with beams at the ceiling versus a flat ceiling.

4.5 Model Limitations

FDS is only as realistic as the inputs to the simulations are. The most important user-imposed limitations as opposed to inherit model limitations are summarized below.

1. HRRPUA - controlling the burning rate instead of allowing the material to pyrolyze based on the surrounding environmental conditions. The fuel will not achieve burnout unless specified. [6]
 - (a) No spread rate was taken into account.
 - (b) HRR doesn't change when wind is added; only the temperatures at the ceiling are affected.
2. All critical length scales are not fully resolved and those that are fully resolved are only resolved in small areas (these areas being the plume and the ceiling jet). As a result, there is some loss of accuracy with this hybrid grid approach.
3. The description of carbon monoxide in FDS has been overly simplified, but has been validated.
 - (a) CO and soot have fixed yields, which means that the simulation can only be as good as the value utilized from the literature

Chapter 5

Machine Learning

5.1 Location Identification Methodology

The location methodology is the basis for the ML models used in the fire magnitude and fuel composition portions of this analysis. Figure 5.1 shows a top-down view of the hangar in PyroSim. The numbered sensors, with the numbers starting at zero to correspond to the indices used in Python, are represented by yellow points at the ceiling and possible locations of the 1 m^2 burner fires are represented by the white squares, labeled A through F, in Figure 5.1.

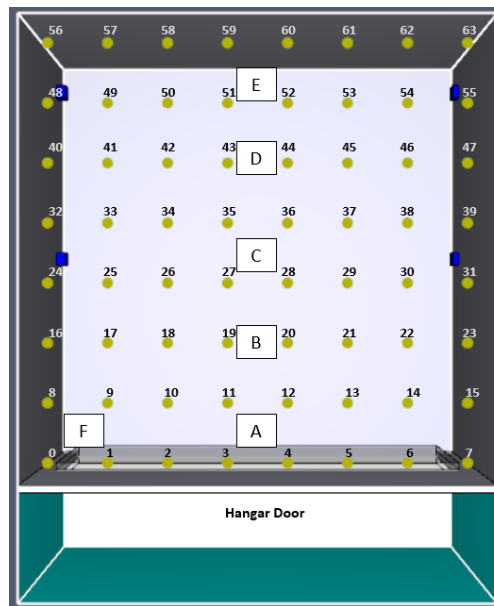


Figure 5.1: Labeled sensor locations.

Preprocessing of the data was required. This approach began by taking the FDS _devc.csv output files and extracting the temperature and smoke columns for each simulation, which included different fuels, locations, wind and beam cases. There are many fluctuations in the raw data, as was seen in chapter 4, so the data was smoothed using a twenty-point running average, which is beneficial for identifying true trends in the data, as opposed to instantaneous spikes. This twenty-point running average equates to approximately 6 seconds of simulated time, as the simulation runs for 300 seconds with measurements taken 1000 times over the course of the run-time, for a measurement frequency of about 3.3 Hz. A comparison can be seen between the raw data and smoothed data in Figure 5.2.

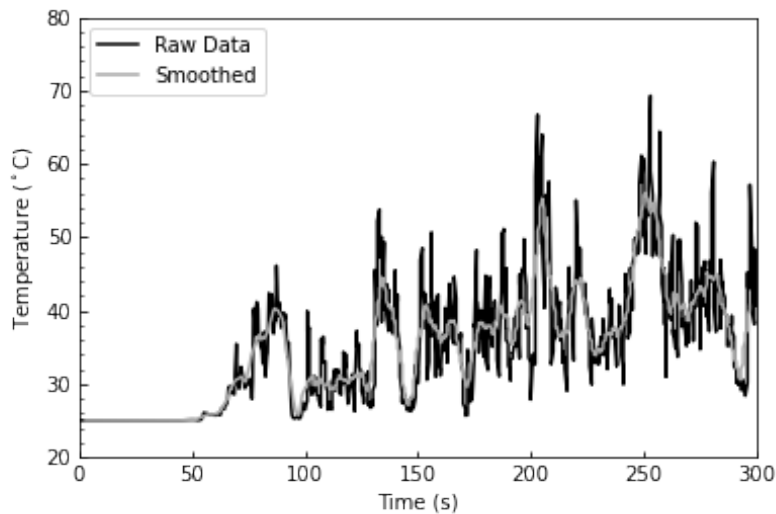


Figure 5.2: Raw versus smoothed data for JP-8 gire in location F with wind.

5.1.1 Enclosure Fire Physics

There is background on the physics of fire that guides the approach for identifying location. The sensors are located at the ceiling because of the buoyancy effects. As the fire burns it entrains air and the temperature of the air and products of combustion increases due to the heat of the combustion reaction. The density of the air in the fire plume decreases with the increase in temperature. This buoyant fluid, which contains air, CO, CO₂ and smoke rises vertically to the ceiling. The ceiling jet distributes the smoke across the ceiling, where it then begins to accumulate, seen in Figure 5.3. The maximum depth of the upper layer is determined by the hangar door opening.



Figure 5.3: Stratification of smoke at the ceiling.

There are variations in the products of combustion in the smoke layer as a function of distance from the point of impact of the fire plume onto the ceiling. This occurs because of the mixing with ambient air, which “dilutes” the products of combustion whether that be concentration of CO, CO₂, smoke concentration or temperature. The further away from the plume, the more cool, ambient air that can be mixed with the smoke layer. The relationship between distance from the center of the plume and temperature is seen in Figure 5.4. This temperature decreases with increasing distance and this trend can be seen in the other products as well, as mixing with ambient air dilutes the concentration of the products of combustion. This highlights that sensors at the ceiling can impart information on the fire location due to the effect of buoyancy and mixing.

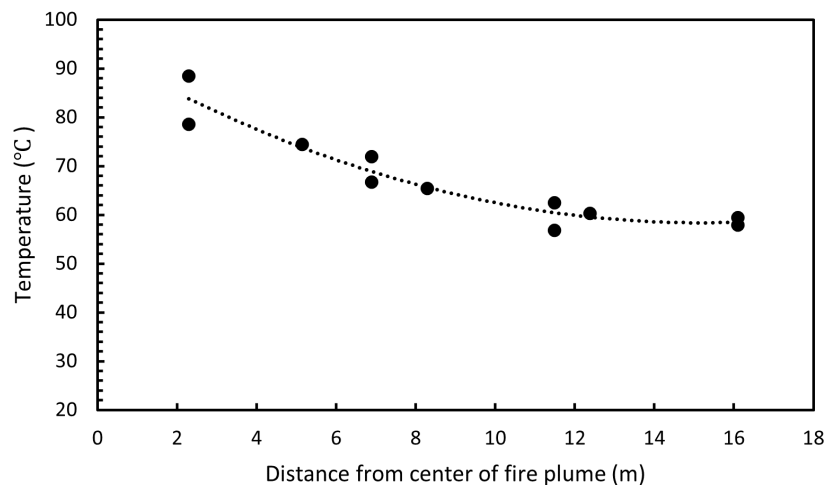


Figure 5.4: Temperature versus position, from JP-8 fire at location C, at the end of the simulation.

5.1.2 Temperature Approach

The temperature along the diagonal of the hangar for a JP-8 fire burning in location C was taken and plotted in Figure 5.5, where sensor 56 is the top left corner of the hangar and sensor 7 is located at the bottom right of the hangar. Sensor 35, in yellow and sensor 28, in light blue, are the closest sensors to the fire and therefore achieve the greatest temperature. As was previously discussed, when the sensor is further away from the fire, the temperature of the sensor decreases because it gets further away from the plume and the large development of the products of combustion. These temperature differences are first noted in the initial growth phases of the fire and remain for the entire duration of the combustion reaction. Figure 5.5 highlights that these trends hold true for any instance in time, not just the end of the simulation when the products of combustion have had ample time to develop. The maximum temperature value and the corresponding sensor that picks this up needs to be identified. It is not important to identify the growth of the fire at this stage of the project, as the first instance in time where this maximum temperature occurs is of interest, allowing for the quickest detection time. Heat will be the indicative product that will be used to determine the range in which the fire is predicted to be located.

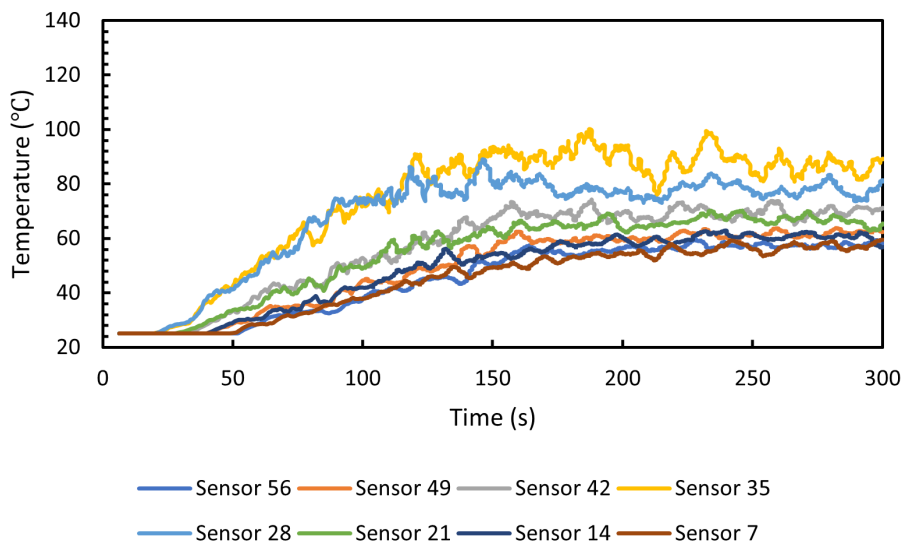


Figure 5.5: JP-8 fire at location C, with smoothed temperatures, taken along the diagonal of the hangar.

The time-dependent temperatures were analyzed for each fuel package, resulting in a (64,982) matrix where there are 64 sensors and 982 instances in time when the measurement is taken. An array was created to store the maximum temperature value at every instance in time. This paired down the matrix, on a

simulation basis, from shape (64, 982) to (1, 982), identifying the single sensor at each instance in time that had the maximum temperature, and storing that temperature value in an array. .

It is necessary to determine if the temperature that the sensor is reading is truly greater than the ambient temperature, which is 25 °C. Therefore, a $\pm 15\%$ uncertainty was applied to both the ambient and the measured temperature and the temperatures were compared to ensure that there was no overlap. The justification for the $\pm 15\%$ is based on the methodology that Underwriter’s Laboratory Fire Safety Research Institute (UL FSRI) uses. Radiation will impact the temperature of the thermocouple readings, as there will be radiative losses or gains depending on where the thermocouple is located in relation to the flame [37]. The justification continues by stating that expanded uncertainty is approximately $\pm 15\%$ due to the effect of radiative heat transfer to the thermocouples. Even though there are temperature sensors at the ceiling, which act more like a typical heat detector instead of thermocouples, the thermocouple uncertainty can be assumed to be applicable to the temperature sensors. The ambient temperature + 15% was subtracted from the measured temperature -15% and is referred to as the “overlap” value in the code in Appendix B.

If the comparison of the ambient and measured temperature range results in a positive value, the temperature that the sensor is picking up is confirmed to be from the fire and not random fluctuations in the space. This means that the lower range of the measured temperature, denoted by T, $(T-0.15T)$ is greater than the larger ambient temperature (28.75 ° C) and the intervals do not overlap. The index of the positive values of the overlap are stored in an array. The value of interest is the first instance of positive overlap, which is used to get the coordinate location of the sensor that is assumed to be closest to the fire. This sensor location is stored and plotted in relation to the actual fire location. It is assumed that the fire is located within a circular area, centered at the first sensor activation with a radius of 5 m. This approach reduces the amount of sensors of interest from 64 to a maximum of nine, depending on the location of the sensor. The labels of the sensors that fall within this circular region are stored for use in the subsequent algorithms. The flow chart that represents this process is seen in Figure 5.6, with the detailed code found in Appendix B. Results will be discussed in chapter 6.

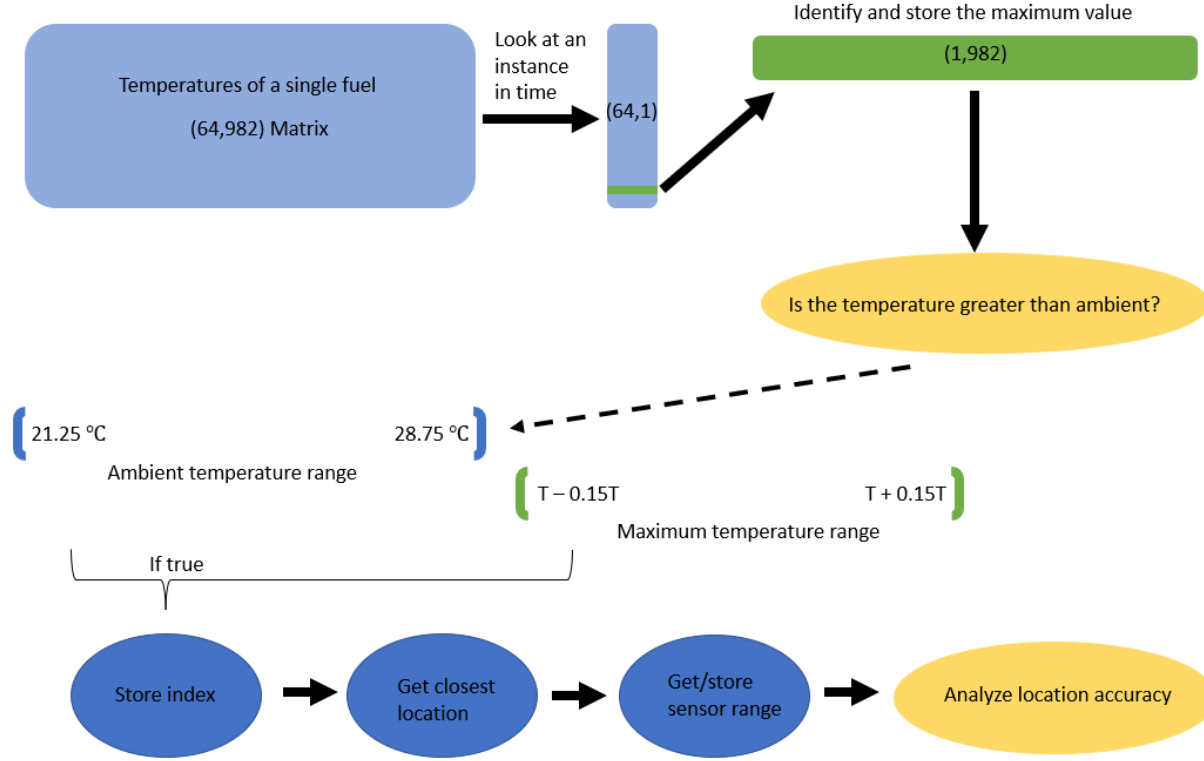


Figure 5.6: Flowchart of code breakdown.

5.2 Heat Release Rate and Temperature Algorithm

Several iterations were attempted for the development of the HRR and temperature algorithm to determine the magnitude of the fire, but they all involve the same preprocessing of the data. The (up to nine) sensors of interest were stored for the 56 unique simulations. There will be less than nine sensors in the area of interest if the sensor is located close to the wall of the hangar. If the location was identified, the time-series temperature data was extracted, averaged and stored in a matrix. The average temperature across the sensors of interest was taken because it is unknown exactly where the fire falls within this hypothesized circular region, and a single sensor should not be driving the analysis in one direction over another. If the sensor location was not identified, then the average temperature across all 64 sensors at the ceiling was taken.

HRR values were needed to train the neural network and were obtained from the `_hrr.csv` FDS output files. If the network is given temperature data, the simulation must be trained to map the temperatures to the known HRR, making this a supervised learning problem. Therefore, a twenty-point running average was taken over the HRRs, resulting in an array of length 981. The time array had to be shortened from 982 to

981 to match the size of the smoothed HRR data.

5.2.1 Final Architecture

The overall steps in the neural network are as follows:

1. Give the model inputs, where the inputs are temperature values
2. Give the model targets to map to, where the targets are the HRRs that correspond to the temperature
3. Do a forward pass and make predictions
4. Compute the loss of the function
5. Update the model by
 - (a) Resetting the optimizer gradients
 - (b) Computing the gradient of the loss function with back propagation
 - (c) Taking an optimization step
6. Run through another epoch and repeat until completion

A forward pass refers to calculating the values of the outputs from the inputs. Back propagation can be described by taking a first guess as to the weights applied to the inputs and initializing the final bias terms to zero. The loss is calculated and the derivative of the loss with respect to the bias term is found. The weight and bias terms are continually updated by subtracting the step size (slope multiplied by the learning rate) from the old bias value. The function continues to calculate until the value where the slope approximately equals zero is found. Gradient descent is used to optimize the unknown parameters.

Various parameters were explored, such as number of nodes and hidden layers, to determine the best architecture for the finalized neural network. After trial and error, the optimized architecture was a 3-layer, fully connected, feed forward neural network with ReLU activation, created using Pytorch,[38] with an input and output size of 981 that corresponds to the length of the time array, and 400 nodes in each of the hidden layers. This was determined to be a combination that did not overfit to the training data. Adam optimizer was used with a learning rate of 1e-5 and a weight decay of 0.01. The learning rate controls the step size at each iteration in an attempt to minimize the loss function. The smaller the learning rate, the longer it takes the model to adapt to the problem, but if the learning rate is too high the model could diverge. Weight

decay prevents the weights from getting too large, additionally preventing overfitting. 20,000 epochs were used, as this resulted in the ideal tradeoff between computational time and reduced loss. An epoch is “the number of times the algorithm sees the complete data set”, for example, the model will see all of the fires in the training set, perform the required calculations, then run through another epoch, seeing the training set again after performing updates to the model.

Gradient descent refers to the calculation of the slope of the error and then moving in the direction opposite of the increase, toward the direction of the minimum. The model is trained using stochastic gradient descent, where the batch size (the number of samples the model needs to work through before updating the parameters) is equal to one. A sample is equal to one simulated fire scenario, which is why the input to the neural network is 981, as this equates to the number of measurements in one sample. This implementation can be specifically seen in the for loop in the training code in Appendix B.

An important parameter for the design of the neural network is the criterion used to train the network, which specifies the type of loss used when computing the gradients. An L1-norm loss function, which uses the mean absolute error (MAE), minimizes the sum of the absolute differences between the target and the estimated value and is typically more resistant to outliers. This results in the model focusing on the bulk of the training data and ignoring any difficult or noisy data. The mean squared error (MSE) is an L2-norm loss function and minimizes the sum of the square of the differences between the target and estimated values, which gives high weight to outliers. The difference between training the model with the specified architecture on each loss criterion is explored.

The first training criterion that was explored utilized PyTorch’s `nn.L1Loss()` [38]. This measures the MAE to compute the gradient of the loss function, and continues to train throughout the remainder of the epochs. In order to have a direct comparison between the L1 and the L2 losses, the performance when training with MAE had to be evaluated on the test quantities using both MAE and MSE.

This loss function is calculated using

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (5.1)$$

where n is the total number of data points y is the true value and \hat{y} is the predicted value. Calculating the loss in this way gives the error in more tangible units of kW, as opposed to the units of MSE, which are kW².

The second criterion used to train the network was PyTorch’s `nn.MSELoss()`, calculated as

$$MSE = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (5.2)$$

The network was set into testing mode and the MSE performance was evaluated on MAE and MSE. The performance can be evaluated in Figure 5.7. It is beneficial to plot the loss across epochs because the loss function is calculated across every data point and will give the “quantitative loss”. The log-loss curve for MAE is reduced to a lower value when evaluated on both criteria metrics, indicating that the L1-norm loss function, MAE, is the optimal criterion to train and evaluate this specific subset of fuels on, as the closer the loss is to zero, the better the model performed. From this point on, when referring to loss, MAE is the loss function that is used. It is important to note that the optimal parameters that are chosen for the final architecture are not universal and will not apply to scenarios that contain different data sets. Additionally, further optimization could continue to be performed.

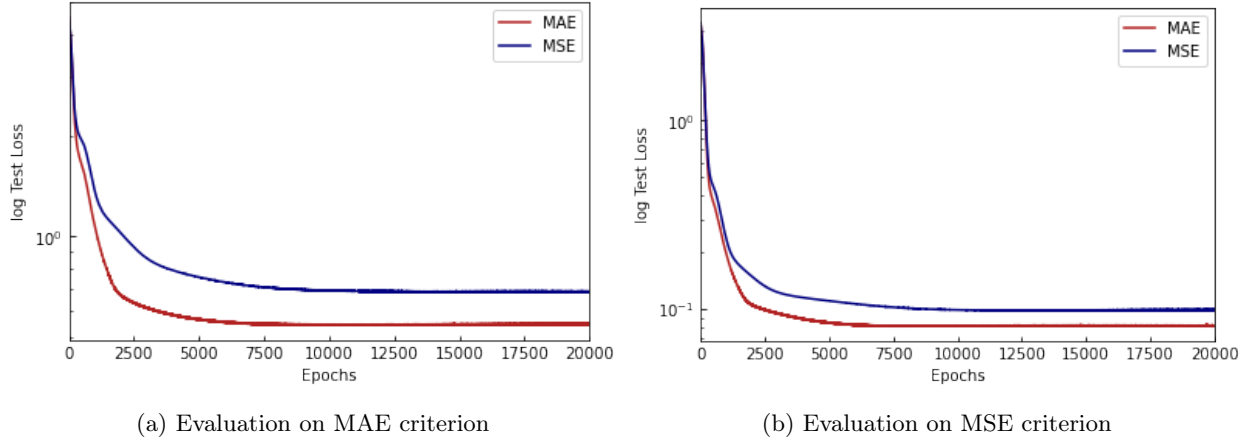


Figure 5.7: Performance of L1 (MAE) versus L2 (MSE) loss.

5.2.2 Training on Individual Fuels

In order to get insight into the model performance, the neural network was trained and evaluated on each fuel individually to see how the results from each fuel perform, compared to when the fuels are combined into a single feed forward neural network.

The sensors of interest that were identified in the location identification model were also used for this approach. The temperatures and the HRRs were only scaled relative to the individual fuel of interest, using Sklearn’s `MinMaxScaler()`. This translates each feature, for example, the combined temperatures of all of

the propane fires, individually by identifying the minimum and maximum value within the feature range and scaling the range from 0 to 1 relative to these values. Further detail on how this scaling is performed can be found on the Sklearn website. A random 80/20, train/test split was performed using Sklearn's `train_test_split`. [39]

The same neural network architecture and model parameters that were utilized for the entire fuel set explained above were also utilized for each individual fuel. The test losses were calculated for each fuel and plotted on the same graph in Figure 5.8 as a function of the number of epochs. The scaled losses were compared because of the order of magnitude difference of HRR between polyethylene and JP-8, for example. The asymptotic values of the scaled and unscaled MAE are reported in Table 5.1. Again, the closer the scaled MAE is to zero, the better the model performed, as the goal is to minimize the loss, which is equivalent to the derivative of the loss being zero. The unscaled MAE indicates the minimum kW loss, but it needs to be considered that this value is relative to the peak HRR that this fuel produces. For example, polyethylene has one of the lowest unscaled MAEs, but the peak value of polyethylene is about 50 kW, so a loss of 26.25 kW out of about 50 kW yields a high error, which is seen in the scaled MAE.

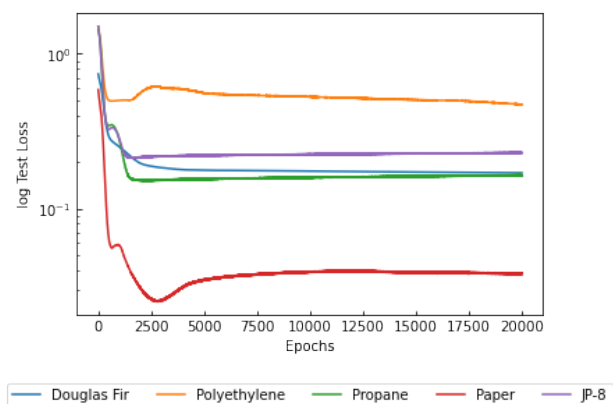


Figure 5.8: Scaled testing MAE when each fuel is trained individually.

Table 5.1: Final Mean Absolute Error for Fuels Trained on an Individual Basis

Fuel	Scaled MAE (kW/kW)	MAE (kW)
Douglas Fir	0.1701	362.10
Polyethylene	0.4703	26.25
Propane	0.1638	384.07
Paper	0.0379	18.49
JP-8	0.2295	535.17

Every fuel reaches an asymptotic value for loss, indicating that these fuels have been sufficiently trained

on. The loss for polyethylene is the highest in comparison to the other fuels, indicating that this was the most difficult for the model to fit, which can be hypothesized to be true when the fuels are combined into a single, large model. A possibility for this behavior is that the polyethylene has such a low HRR relative to the other fuels of interest. This hypothesis will be assessed in greater detail in later sections.

5.2.3 Combining Fuels into Single Feed Forward Neural Network

If this model were to be deployed in an actual hangar, the fuel would not be known ahead of time, therefore all of the potential fuels of interest would need to be trained on at the same time to allow for the variety in the identification of potential fires.

As was seen in Chapter 4, the HRR of these fuels vary in order of magnitude, with polyethylene having an order of magnitude of 10 and fuels like JP-8, Douglas fir and propane being on an order of magnitude of 10^3 . The input data (temperature) and the output data (HRR) were scaled using `MinMaxScaler()` [39]. The feature range in this scenario is the matrix of average temperatures for the 56 individual fires and the matrix of HRRs for the 56 fires. The HRRs are scaled from 0 to 1 relative to the entire set and the temperatures are scaled from 0 to 1 relative to the entire set.

It was important to have an equal distribution of each fuel in the training set. If there is significantly more of one fuel in the training set than another then the model will overfit to the more frequently seen fuel. Instead of using the `train_test_split` from Sklearn [39], the training and testing set were manually created, with 40 out of the 56 fires going to train and the remaining 16 being used for testing. This resulted in an approximate 70/30, train/test split. A random subset (based on environmental conditions and location) of eight of each fuel type were selected to train on. The specific fuels utilized in the training and testing set are listed in Table 5.2 and 5.3. It should be noted that there is not an equal split of each fuel in the testing set because there were more Douglas fir simulations that were initially run due to the analysis of the effect of beams at the ceiling of the hangar. However, the unequal distribution of fuel type in the testing set is not of concern for the performance of the model. The temperatures and corresponding scaled HRRs were put into input and output, training and testing matrices and the model was trained.

Table 5.2: Manual Subset of Fuels for Training

Training Set				
Fire	Fuel	Location	Wind	Beams
0	Douglas Fir	A	No	No
1	Douglas Fir	C	No	No
2	Douglas Fir	E	No	No
4	Douglas Fir	A	Yes	No
7	Douglas Fir	D	Yes	No
8	Douglas Fir	E	Yes	No
11	Douglas Fir	B	No	Yes
13	Douglas Fir	D	No	Yes
16	Polyethylene	A	No	No
17	Polyethylene	C	No	No
18	Polyethylene	E	No	No
20	Polyethylene	A	Yes	No
21	Polyethylene	B	Yes	No
22	Polyethylene	C	Yes	No
23	Polyethylene	D	Yes	No
25	Polyethylene	F	Yes	No
26	Propane	A	No	No
27	Propane	C	No	No
28	Propane	E	No	No
30	Propane	A	Yes	No
31	Propane	B	Yes	No
32	Propane	C	Yes	No
33	Propane	D	Yes	No
35	Propane	F	Yes	No
36	Paper	A	No	No
37	Paper	C	No	No
38	Paper	E	No	No
40	Paper	A	Yes	No
41	Paper	B	Yes	No
42	Paper	C	Yes	No
43	Paper	D	Yes	No
45	Paper	F	Yes	No
46	JP-8	A	No	No
47	JP-8	C	No	No
48	JP-8	E	No	No
50	JP-8	A	Yes	No
51	JP-8	B	Yes	No
52	JP-8	C	Yes	No
53	JP-8	D	Yes	No
55	JP-8	F	Yes	No

Table 5.3: Manual Subset of Fuels for Testing

Testing Set				
Fire	Fuel	Location	Wind	Beams
3	Douglas Fir	F	No	No
5	Douglas Fir	B	Yes	No
6	Douglas Fir	C	Yes	No
9	Douglas Fir	F	Yes	No
10	Douglas Fir	A	No	Yes
12	Douglas Fir	C	No	Yes
14	Douglas Fir	E	No	Yes
15	Douglas Fir	F	No	Yes
19	Polyethylene	F	No	No
24	Polyethylene	E	Yes	No
29	Propane	F	No	No
34	Propane	E	Yes	No
39	Paper	F	No	No
44	Paper	E	Yes	No
49	JP-8	F	No	No
54	JP-8	E	Yes	No

5.3 Fuel Discrimination Algorithm

Prior to determining what method to use for fuel type classification, a series of visualizations were performed to determine the ideal trends to focus on. Every product of combustion was compared relative to one another and for each of the fires at a given time step using a PairPlot from Seaborn implementation in Python, resulting in a total of 16 plots. These graphs and further detail are in Appendix C.

It was determined that the relationship between smoke and temperature yielded the strongest separability between different fuel types. This is a beneficial relationship to explore because temperature relates to HRR and MLR, and smoke production is based on MLR. There are multiple ways to perform classification, but based on the visualizations it was determined that a support vector machine (SVM) utilized for the supervised classification of the fuel type would be the best approach. This was implemented using `sklearn.svm.SVC` [39].

There are three main approaches that were taken for the training of the SVM, however the parameters and training set are the same for each approach. The same training and testing set that was created for the neural network was used for training and testing of the following SVMs. This achieved the same purpose of making sure that the SVM was not over-trained on certain fuel types. The parameters in the SVM are in Table 5.4.

Table 5.4: Parameters in SVM

Kernel	C	decision_function_shape
Linear	1	ovo

The visualizations indicated that the trends were linearly separable, therefore the kernel was set to linear. When this kernel is used, the data is not projected to a higher dimension and the following equation is applied,

$$K(x, y) = x^T y + c \quad (5.3)$$

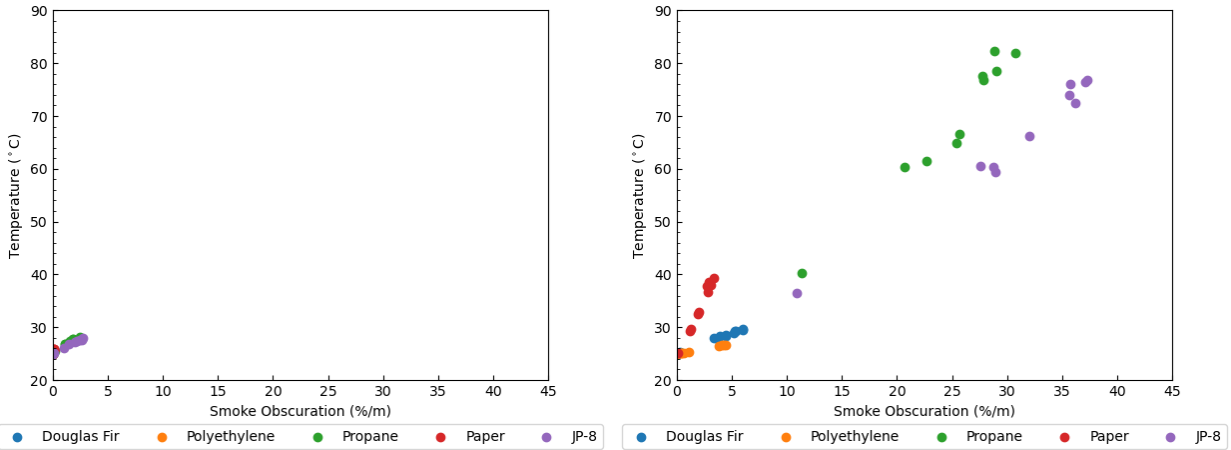
where x and y are separate data points.

C is the regularization parameter, and is set to 1 as the default. The strength of the regularization is inversely proportional to C , which in the case of the default, still remains 1. The penalty applied using this parameter is a squared L2 penalty. The parameter 'ovo', which stands for "one versus one" is specified because this is a multi-class classification approach, since there are five fuels of interest. If binary classification is intended, this parameter will be ignored and the "one versus rest" approach is taken.

5.3.1 No Time Dependency

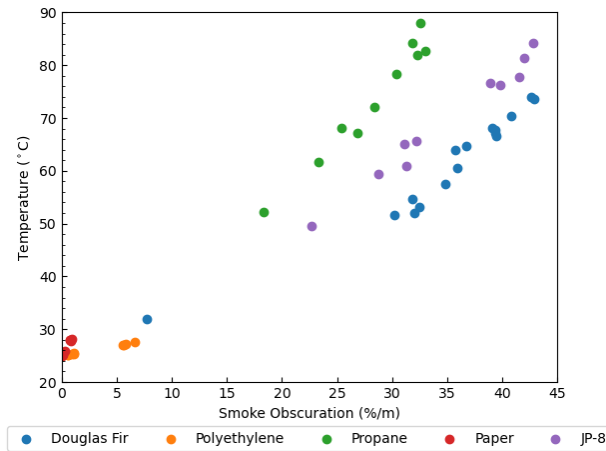
The SVM was initially trained on a specific instance in time, that was believed to be a representative segment in time for training, as this time allowed for sufficient development of the fires and the fuels became more separable at this instance.

The results were not ideal when testing the accuracy using this methodology, because the products of combustion (smoke and temperature) are time dependent quantities that scale according to the how the HRR changes in time. This can be seen in Figure 5.9 which identifies the relationship between temperature and smoke obscuration at 30 seconds, 150 seconds and 300 seconds which is the end of the simulation.



(a) Smoke versus temperature at approximately 30 seconds.

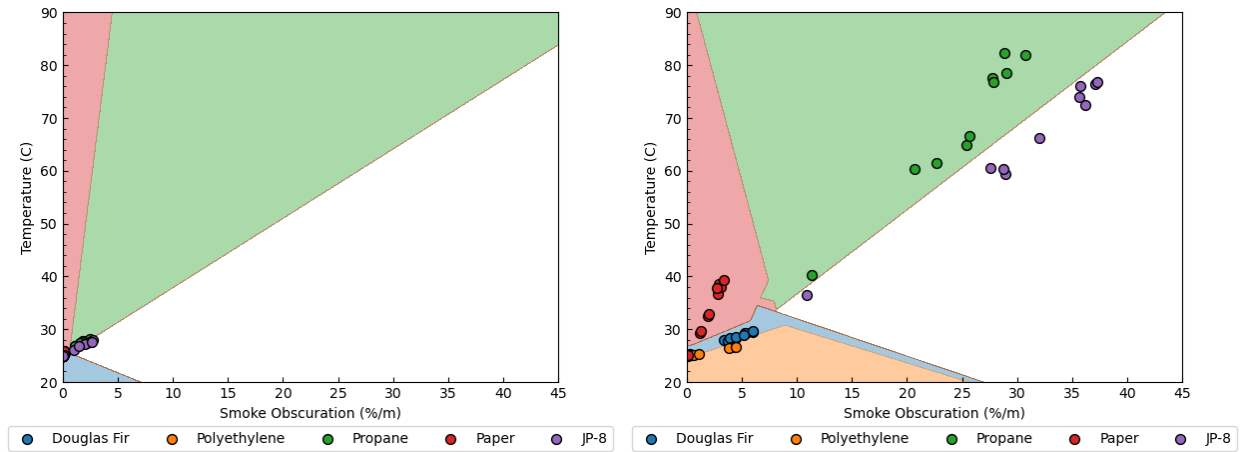
(b) Smoke versus temperature at approximately 150 seconds.



(c) Smoke versus temperature at approximately 300 seconds.

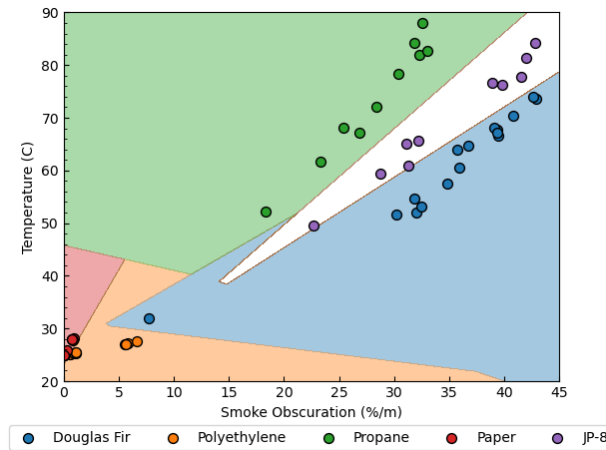
Figure 5.9: Smoke versus temperature at various times.

An example of the support vector machine being trained on and applied to the specific times of interest is shown in Figure 5.10, which shows a visual representation of the separable hyperplanes. Hyperplanes are considered the decision boundaries between unique groups of data points, essentially drawing a line between different regions in if the data set is two-dimensional. The colored segments indicate the region of interest in which the fuel is classified. For example, it is predicted that propane fires fall within the green segment. The relationship between smoke obscuration and temperature results in relatively linear trends. This indicates that visually, the regions of the hyperplanes do not need to extend to the area shown in Figure 5.10 and can be made narrower to allow for more precision and the potential addition of more fuels.



(a) SVM applied at approximately 30 seconds.

(b) SVM applied at approximately 150 seconds.



(c) SVM applied at approximately 300 seconds.

Figure 5.10: SVM applied at various times.

Training on a specific instance in time did well when the SVM was applied to the same subset on which it was trained on, or a similar time range. However, when trained on a particular time and tested on a drastically different time, the classification was poor, as it can be seen in Figure 5.10 that the hyperplanes change over each instance in time. Figure 5.11 shows the SVM being trained on the time 150 seconds and applied to every instance in time from 0 to 300 seconds. The accuracy has large fluctuations, and only performs well between 100 and 160 seconds, which is why it was determined that time had to be explicitly taken into account, as there is not a critical instance in time that represents the entire burning duration of the fire.

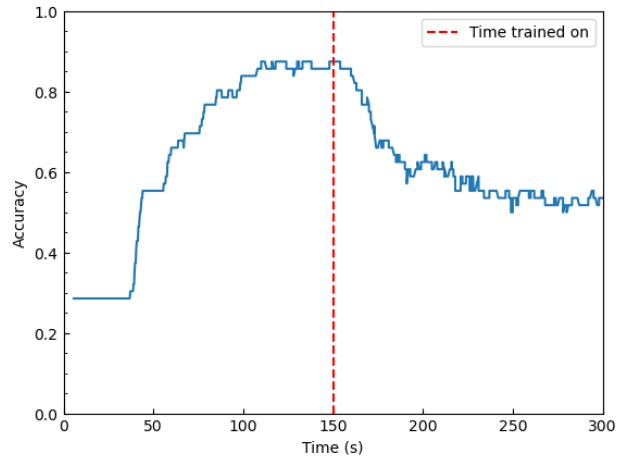


Figure 5.11: Accuracy when trained on a single instance in time (150 seconds) and tested across the entirety of the simulation.

5.3.2 Time Dependency with Discrimination of All Fuels

Prior to including time, the input to the SVM appeared as a two-dimensional plot. However, incorporating time changes this to a three-dimensional visualization, in Figure 5.12. The input to the SVM will now have three columns as opposed to two. If a “slice” of the graph in Figure 5.12 is taken along the time axis, this would lead to the “no time dependency” visualization.

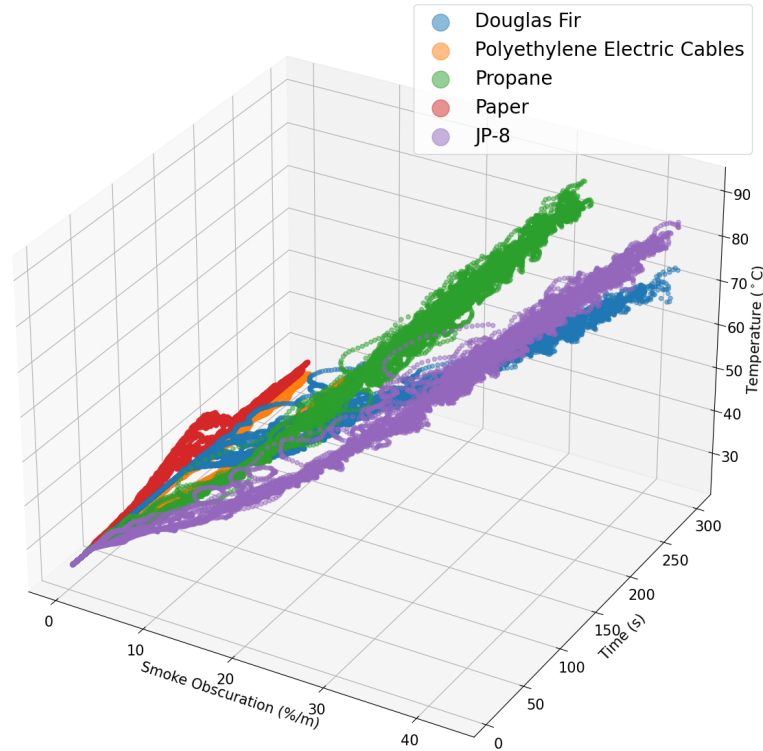


Figure 5.12: Smoke obscuration versus temperature versus time for five main fuels of interest.

Smoke, temperature and time have different numeric values, so they were each scaled using the `MinMaxScaler()` [39]. This scaling was performed to significantly speed up the training of the SVM, as the goal is to maximize the distances from the hyperplane, in case the data set becomes complex, which is quicker when the distances range from zero to one instead of, zero to 300 in the case of time. After the scaling was performed, the smoke, temperature and time columns will be appended and create a (54992,3) matrix. The matrix will then be split up into a training and a testing set, which is the same set identified for the HRR model and the corresponding fuel labels will be set as the target value. After the preprocessing of the data, the SVM will be trained and results will be analyzed in following chapter.

5.3.3 Time Dependency with Binary Discrimination Based on Suppression Agent

The approach for the time binary discrimination of fuel types is determined based on suppression agent, where foam suppression can be utilized for JP-8 fires. The methodology is similar to that of all fuels, except the target labels have been adjusted. The labels for Douglas fir, polyethylene, propane and paper have been

replaced with “other” for the training of the binary SVM. This can be seen in the Figure 5.13, with JP-8 in purple and “other” fuels in gray.

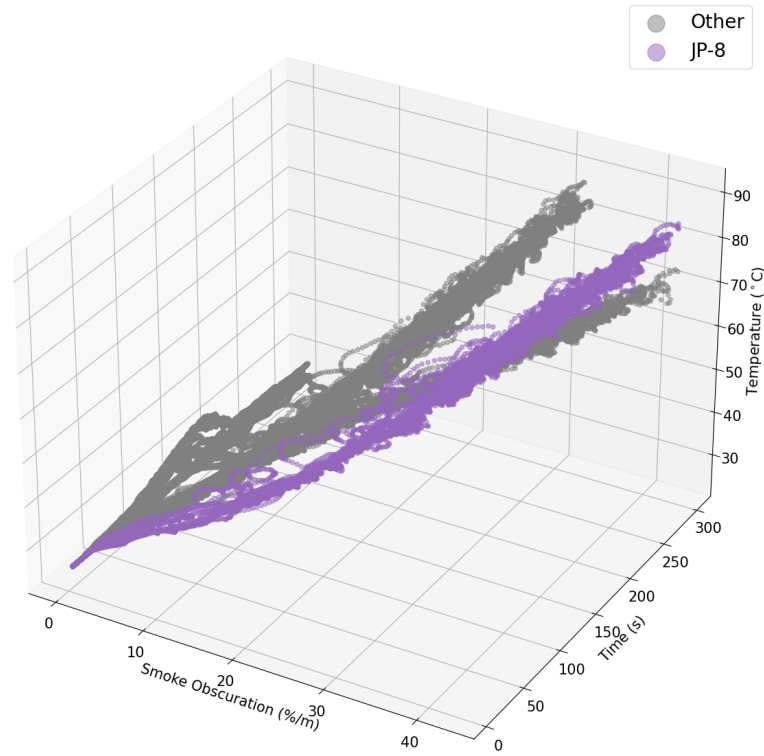


Figure 5.13: Smoke obscuration versus temperature versus time for binary classification

Chapter 6

Results

6.1 Location Identification

The training set represents a total of 56 unique fire scenarios. The model identified 45 out of 56 of these fires correctly, resulting in about 80% accuracy. The results of the location identification are in Figure 6.1. It is considered a success when the fire falls within or on the perimeter of the gray area of interest. The numbers in the legends of the graphs indicate the first sensor that activated. Some fires identified the same first sensor activation, so there will be overlap of the symbols in the graph.

The remaining eleven fires that were not captured with this model were not misidentified, rather they were not able to be located at all. The location of all ten polyethylene fire scenarios were not able to be identified. The HRR and subsequent temperature that the sensors pick up for this fuel type are order of magnitudes smaller than that of the other fuel types, so when the $\pm 15\%$ uncertainty is applied, the measured and ambient temperatures always overlap. This is shown in Figure 6.2, where the lower bound of maximum temperatures of the no-wind polyethylene cases are plotted and compared to the upper bound of the ambient temperature. If the uncertainty threshold was decreased from $\pm 15\%$ then the polyethylene fires without wind would have been correctly identified, but the presence of wind greatly impacted the location identification regardless of how low the threshold was set.

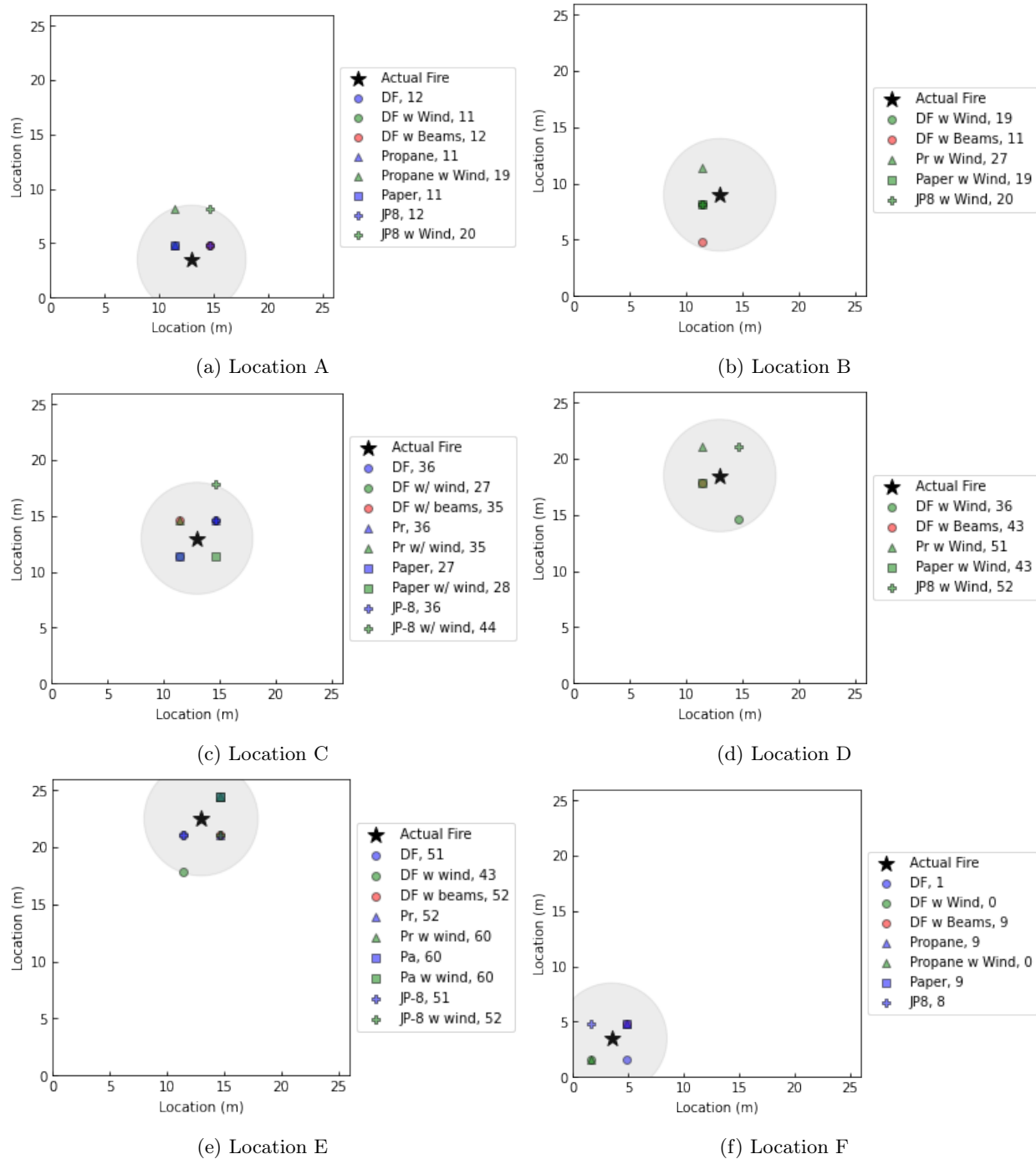


Figure 6.1: Results of location identification model with gray circle representing the 5 m radius of interest; circles = Douglas fir, triangles = propane, squares = paper, plus sign = JP-8; blue = no wind, green = wind, red = beams.

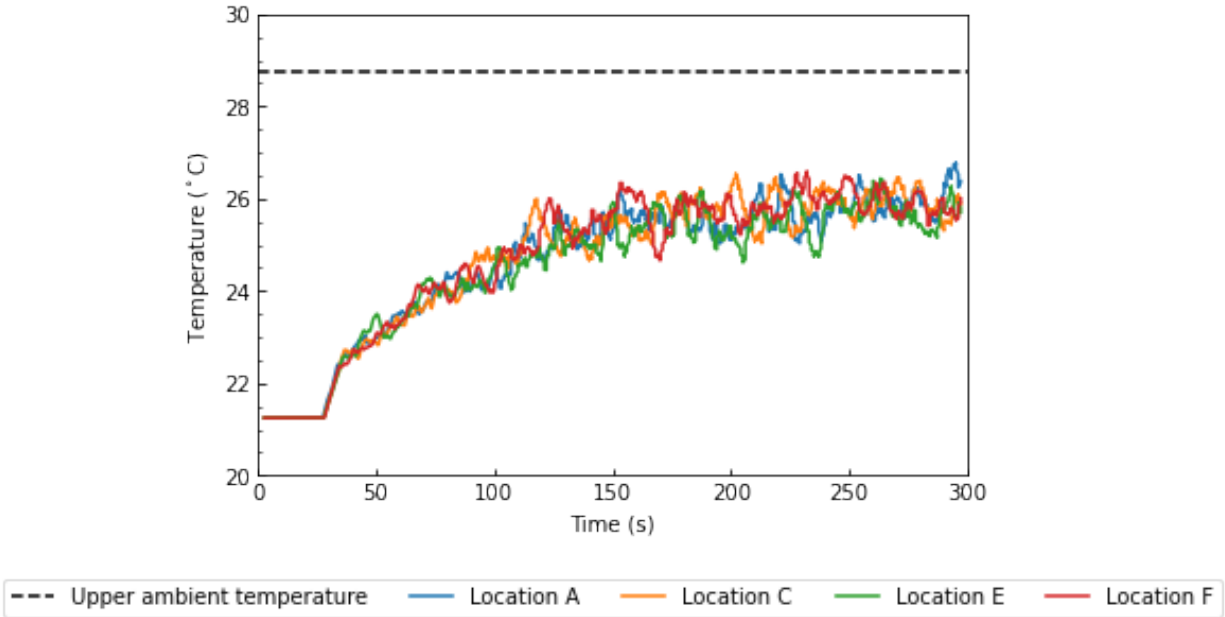


Figure 6.2: Polyethylene temperatures, compared to ambient.

The other fire that could not be identified was the paper fire, with wind, in location F. The paper fire had a peak HRR that was about a factor of ten less than Douglas Fir, JP-8 and propane. Additionally, paper has a lower soot yield, at 0.002 g/g. The combination of lower HRR and soot yield made the fire more susceptible to the impacts of wind, which were uniquely seen in location F. The location in the corner of the space closest to the hangar door caused entrainment effects from the wind that greatly impacted the structure of the flame and the distribution of products of combustion to the detectors, which is shown in Figure 6.3a, where it can be seen that the smoke is being taken out of the hangar through the door. Figure 6.3b illustrates what a vector slice of these entrainment effects look like, with the scale going from -6.0 m/s to 5.0 m/s. It can be seen that there is outflow at the bottom of the hangar door near the base of the fire. Figure 6.3c shows the complex flow structure through the hangar door, with velocities ranging from -6.0 m/s to 5.0 m/s. It is confirmed that there is a strong negative outflow as the wind interacts with the side of the hangar doors, shown in blue. The stronger positive inflow is seen toward the upper middle portion of the hangar door. All images were captured at 120 seconds, when the paper fire reached its peak HRR.

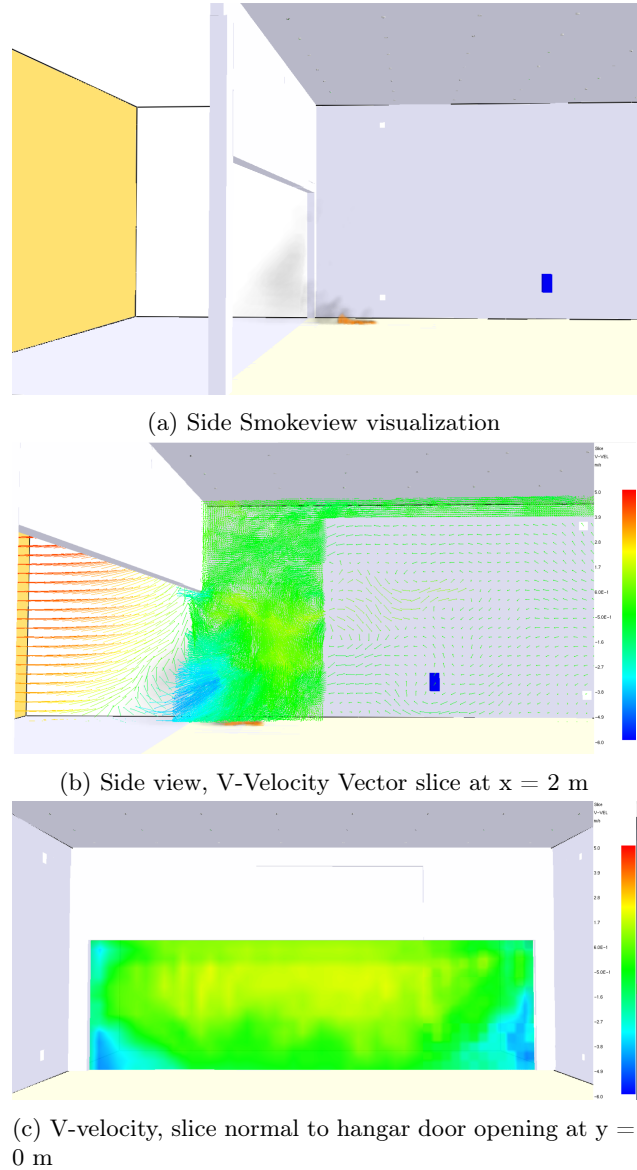


Figure 6.3: Paper fire with wind at location F.

6.2 HRR Model

The relative high success of identifying the location of the fire directly impacts the performance of the HRR model in a positive way. The scaled train and test results are seen in Figure 6.4a, which is plotted on a log scale to differentiate between the similar train and test final values. The unscaled train and test results for the combined fuel neural network are seen in Figure 6.4b. The train loss does not appear to reach an

asymptotic value when plotted on a log y-axis, indicating that the neural network could be trained for more epochs in order to sufficiently reduce the loss, while the test loss does reach an asymptotic value. However, the loss curves look as expected, as the model performs better on training than it does testing. The final loss values at the end of training and testing are in Table 6.1.

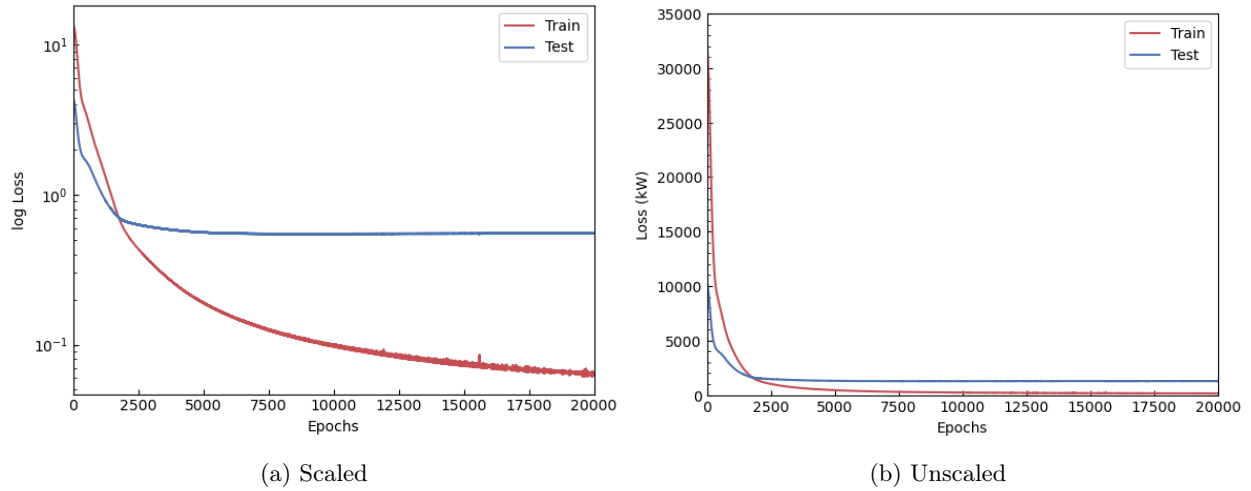
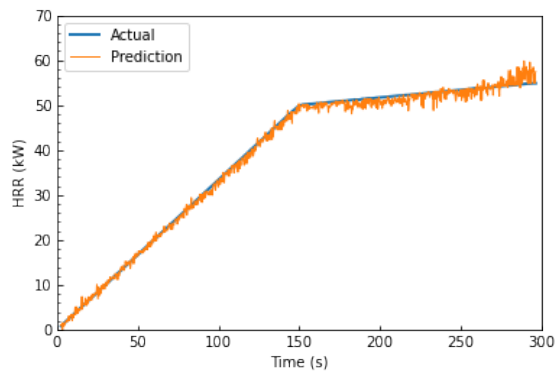


Figure 6.4: Train and test loss curves.

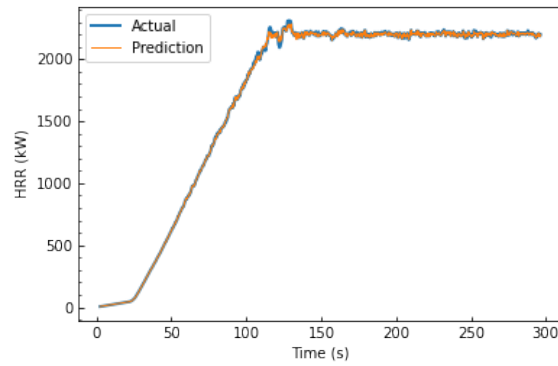
Table 6.1: Final Loss Values

Scaled	Train (kW)	Test (kW)
Yes	0.0649	0.5513
No	150.45	1278.16

The train loss reaches a small value, but to get a better idea of how well the training did at predicting the HRR curve, a few representative training graphs are shown in Figure 6.5. The model was used to predict the HRR curve from the given temperature data and it can be seen that the prediction was able to track the actual HRR curve with good agreement. The results that best indicate performance, specifically on unseen data will be seen with the testing data.



(a) Polyethylene, location B, wind.

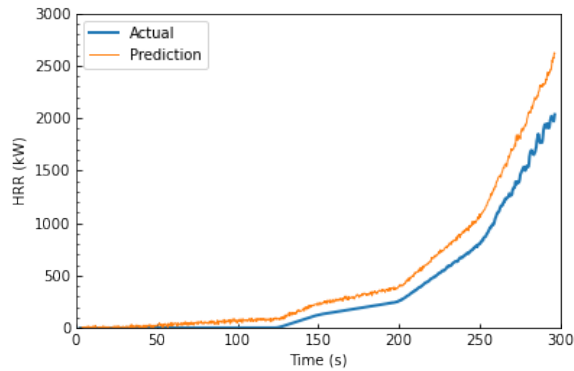


(b) JP-8, location A, no wind.

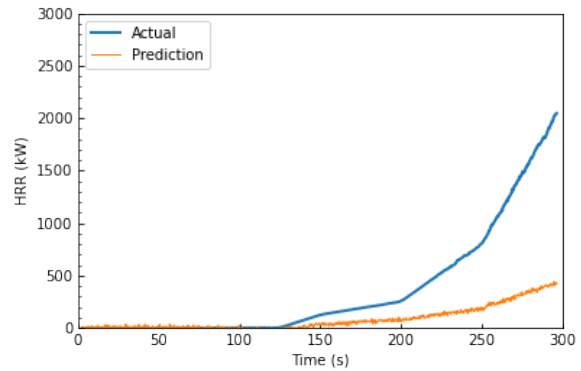
Figure 6.5: HRR curve predictions based on training set.

6.2.0.1 Test Results on Different Location and Wind

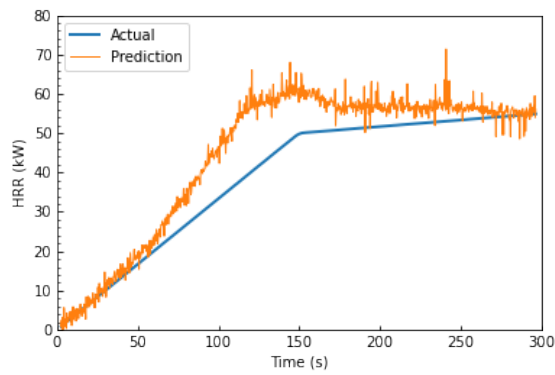
It can be seen that the model does an overall good job at predicting the magnitude of the fire in Figure 6.6, which is the main interest to the firefighters responding to the fire.



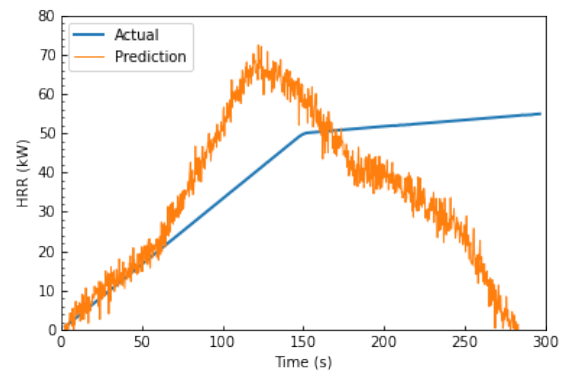
(a) Douglas fir, location F, no wind.



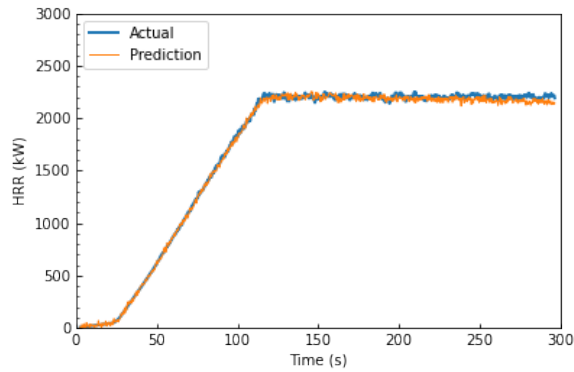
(b) Douglas fir, location F, wind.



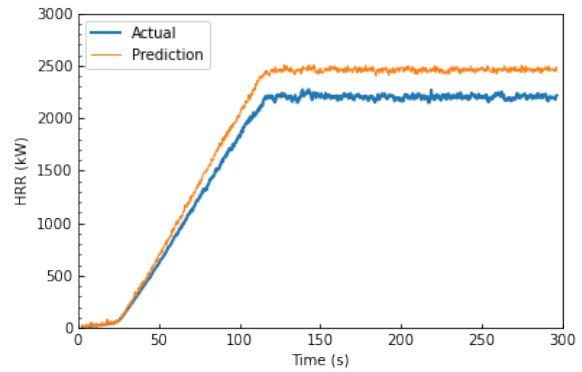
(c) Polyethylene, location F, no wind.



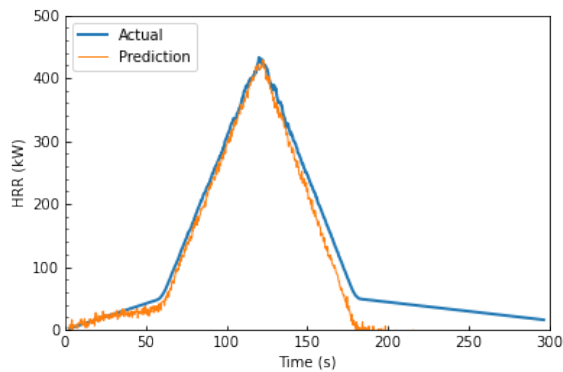
(d) Polyethylene, location E, wind.



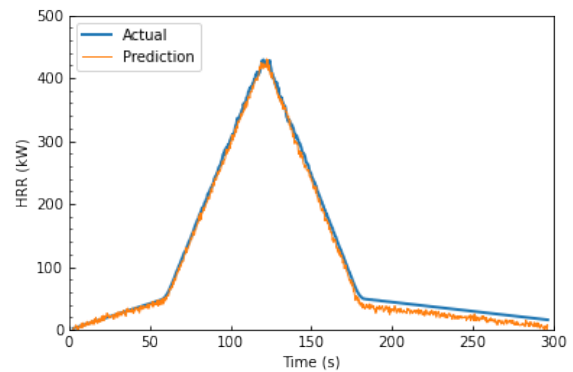
(e) Propane, location F, no wind.



(f) Propane, location E, wind.



(g) Paper, location F, no wind.



(h) Paper, location E, wind.

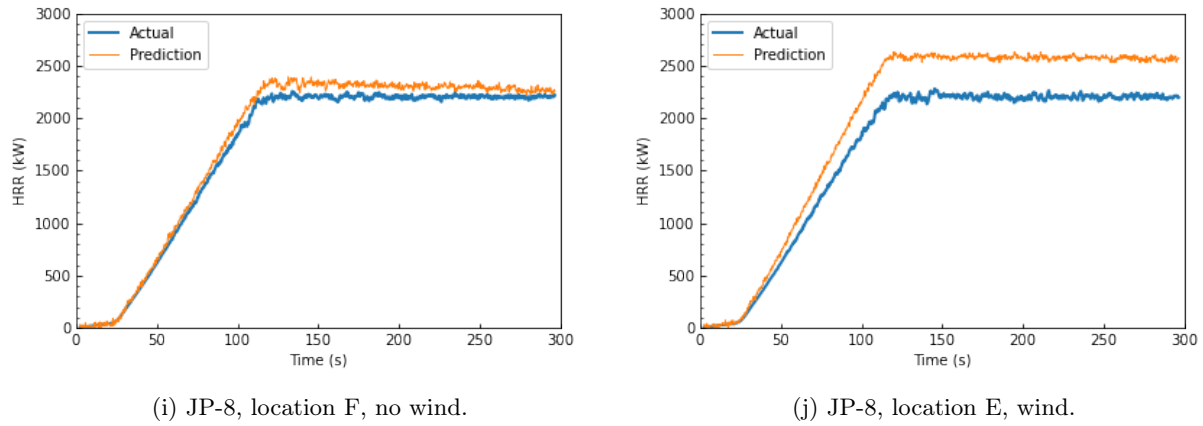


Figure 6.6: HRR curve predictions.

It is seen that the paper fire in location F predicts a negative HRR starting around 180 seconds. It is understood that there cannot be negative HRRs, so the fire is assumed to have extinguished at this point. This predicts early extinction of the paper fire by about two minutes, however the trend in the growth and initial decay phase is captured well which is of main interest to firefighters. When extrapolating and analyzing overall trends in the HRR predictions, there are discrepancies specifically with the polyethylene fire when wind is present. There are difficulties predicting the HRR of polyethylene fires for the same reasons that the model was unable to predict the location of these fires. The HRR is drastically lower than the other fuels in the training set, and when wind is added, this significantly drives down the temperature that the sensors are picking up. As previously mentioned, the HRR was prescribed as a HRRPUA, so the addition of wind does not impact the HRR of the fire. The addition of wind does impact the temperature and can be seen to largely impact the temperature of low HRR fires. This causes issues for the model, as it is now trying to map two different temperature profiles to the same HRR profile, which is why Figure 6.6d does not follow the trend of the actual HRR curve and predicts extinction of the fire. The relationship between critical HRR with the addition of wind is of interest for the use of this model.

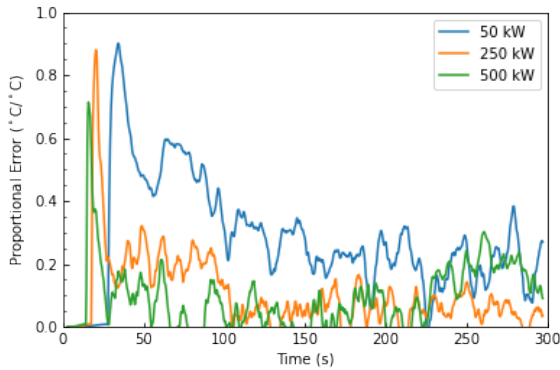
The critical HRR is explored by increasing the HRR of polyethylene fires by a factor of five and a factor of ten. The peak HRR for the duration of the simulation is approximately 50 kW, so the scaled up fires will be considered 250 kW and 500 kW. A total of 12 simulations, all in Location A, were used for this analysis, as the wind would have a greater effect on the fires closest to the hangar door opening. The combinations of simulations are seen in Table 6.2. The temperatures of the nine sensors of interest within the 5 m radius of the fire were averaged and used in the following analysis.

Table 6.2: Critical HRR Analysis Test Matrix

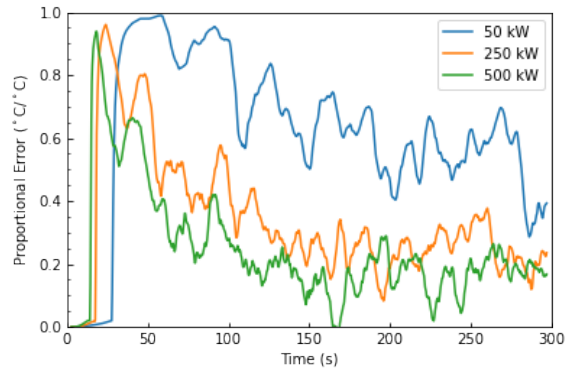
HRR	No Wind	4.4 m/s	2.2 m/s	1.1 m/s
50 kW	X	X	X	X
250 kW	X	X	X	X
500 kW	X	X	X	X

The temperatures were scaled using the `MinMaxScaler()` [39], in order for the temperatures and calculations to be directly comparable to each other. Proportional error is calculated using the “no wind” case as the reference. The difference between the “no wind” case and a wind case for a given HRR is calculated at each instance in time and then divided by the reference case at that same time. This calculation defines the error based on the amount the temperature deviates from the no wind case, therefore a smaller proportional error is better as this indicates wind did not greatly impact the temperature.

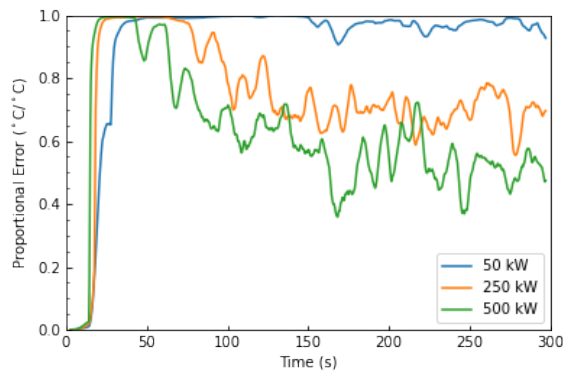
Figures 6.7a, 6.7b and 6.7c show the proportional error for each individual wind case. There is overlap between the 250 and 500 kW curves, indicating that the prescribed wind speeds do not greatly impact the temperature difference between these two fires. There is a larger deviation in the trends with the 50 kW fire, as there is a larger error present and less overlap with the other HRR curves. This indicates that the critical HRR, that works well for this designed model and that does not get largely impacted by the presence of wind is somewhere between 50 kW and 250 kW.



(a) Quarter wind, 1.1 m/s.



(b) Half wind, 2.2 m/s.

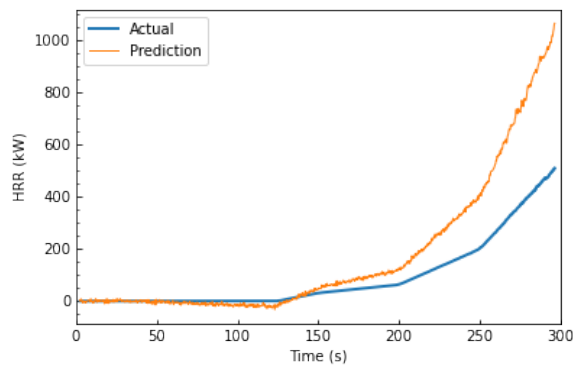


(c) Full wind, 4.4 m/s.

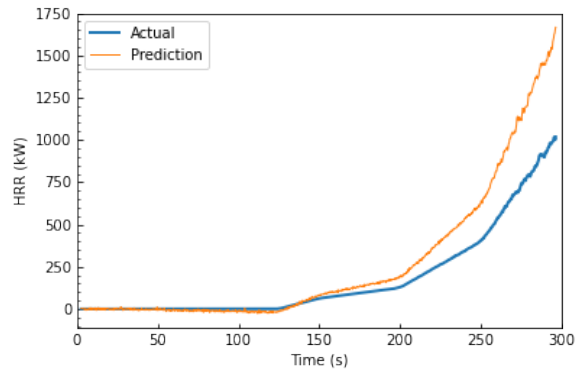
Figure 6.7: Proportional error curves.

6.2.0.2 Test Results on Different HRR and New Fuel

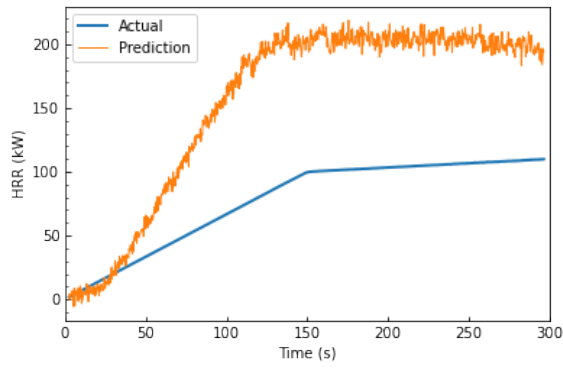
A secondary testing set was used to further validate the model's performance. The graphs in Figure 6.8 show how well the model did at predicting completely unseen HRRs. The testing above showed how the model performed when looking at different fires, when the difference was the location and the environmental conditions. The subset below has scaled the HRR up or down by a factor of two and four to determine if the model was sufficiently trained in that it could extrapolate a prediction of the HRR curve. The model did relatively well at predicting similar magnitude HRR curves. These scenarios would be indicative of if a larger or smaller fuel package was present in the hangar.



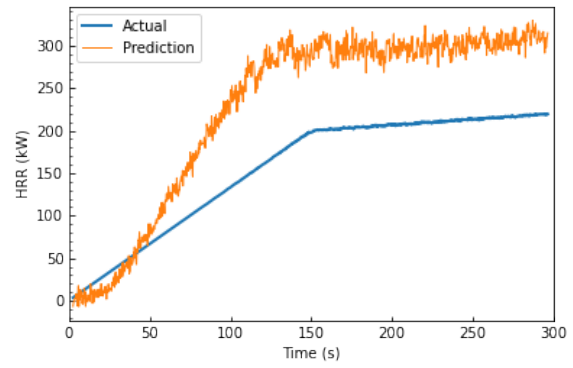
(a) DF 0.25xHRR



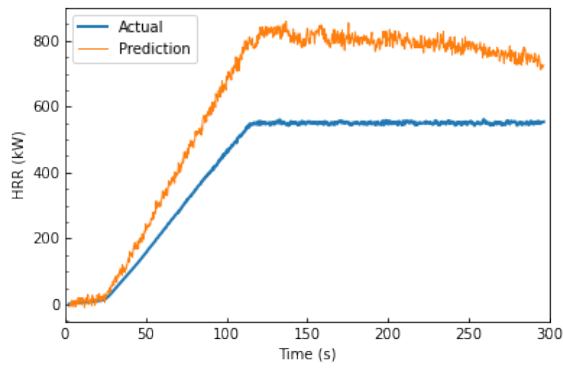
(b) DF 0.5xHRR



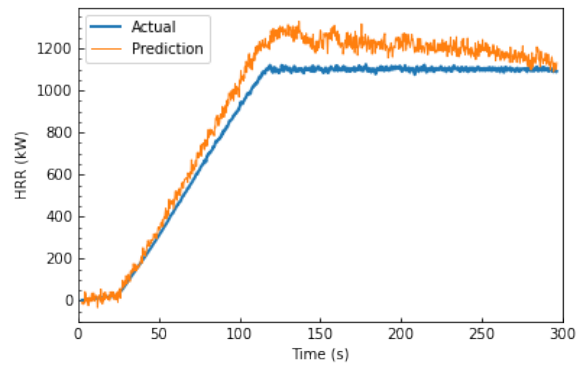
(c) PE 2xHRR



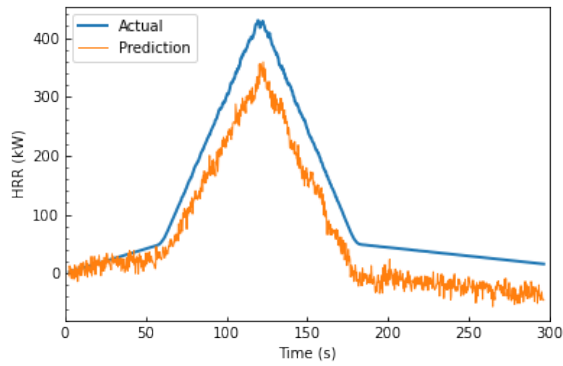
(d) PE 4xHRR



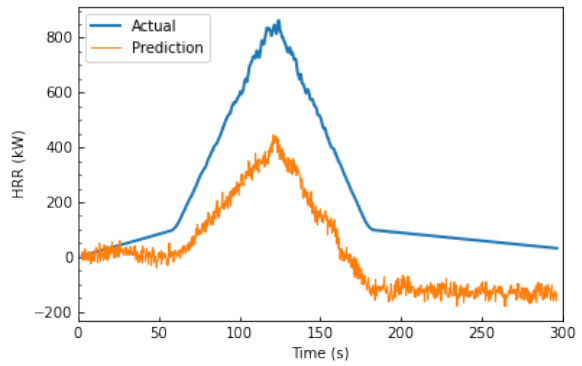
(e) Propane 0.25xHRR



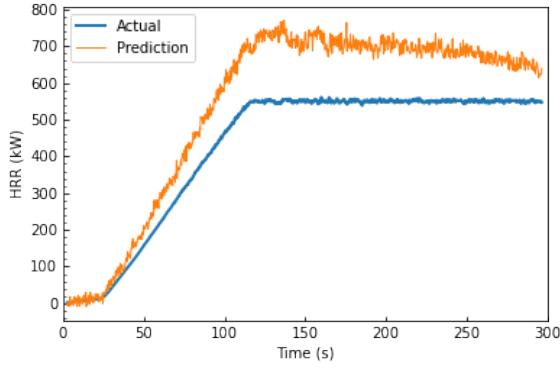
(f) Propane 0.5xHRR



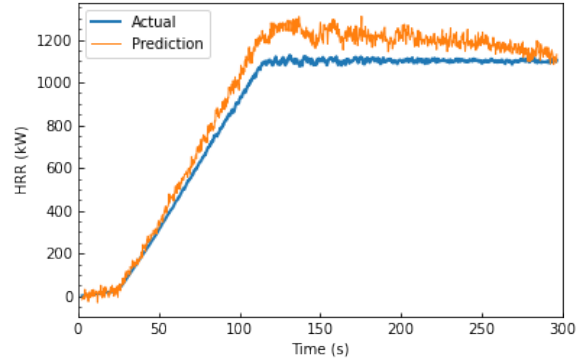
(g) Paper 0.25xHRR



(h) Paper 0.5xHRR



(i) JP-8 0.25xHRR



(j) JP-8 0.5xHRR

Figure 6.8: Scaled HRR curve predictions.

The final test to determine the performance of the model is to see how it predicts on a completely unseen fuel type, which is polyurethane foam. All fires that were previously trained on were either increasing, reached a steady state value, or declined (with this fire having a lower peak HRR). Therefore, predicting on the polyurethane foam fire provides a large challenge. This ultra-fast t^2 growth phase of the fire was able to be captured, but the decay phase was not. The prediction seems to mirror the Douglas fir design curve, however the model was able to predict the shift in time of the growth phase of the polyurethane foam fire. A hypothesis as to why the model was not able to predict the decay phase of the fire is because the temperature in the space was already elevated and that heat remained present in the hangar. The abrupt decrease in HRR did not correlate to an abrupt decrease in temperature. The HRR that was seen at the end of the decay phase before the simulation cut off was around 1500 kW which is still quite high compared to the growth and decay that was captured in the paper fire. This hypothesis has been made by other research groups when accounting for the prediction of abrupt changes in the HRR of larger fires.

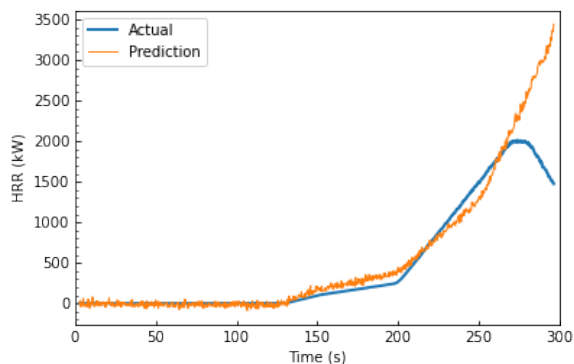


Figure 6.9: Polyurethane foam fire.

6.2.1 Overall Results

It is promising to see that the overall trends in the growth rate of the fire are captured through the use of this neural network and that for certain fires, the HRR curve is predicted with almost perfect agreement. However, for the purpose of situational awareness, the exact value of the HRR to the nearest kW is not of interest. It is more important to know the magnitude of the fire in a categorical sense, giving the firefighters a relative idea of the size of fire they need to put out. The categories have been created to quantify the risk and the equipment needed to extinguish the fire, with the risk increasing with each category. These categories are:

- Category 1: 0 - 250 kW
- Category 2: 251 - 500 kW
- Category 3: 501 - 1000 kW
- Category 4: > 1000 kW

When looking at the testing results, in Table 6.3, the model was able to correctly identify 85% of the categories. Meeting the category criteria meant that the predicted peak HRR value fell in the same category as the actual peak HRR value and did not look at the agreement in the trends throughout the duration of the fire simulation. This performance can be further analyzed by the confusion matrix shown in Figure 6.10. This provides detail on where/what type of classification errors are being made, giving insight on how to better improve the model. The x-axis corresponds to the true category (either 1, 2, 3 or 4) and the y-axis

corresponds to the predicted category (either 1, 2, 3 or 4). If the classification was performed perfectly, then the 45 degree diagonal would contain all 100% values. If there is a misclassification of the fire category, it is better to overpredict than underpredict, as the firefighter can be prepared to take a conservative approach to fighting the fire. Practically speaking, there is no harm if a firefighter is prepared to fight a larger fire than what is present in the space, so in this case, predicting a larger fire might not be considered an explicit failure in the model. However, 50% of the misclassified fires underpredict the magnitude and risk of the fire by one category, which is a cause for concern and could be detrimental to firefighter safety and the subsequent protection of the hangar, so this discrepancy needs to be resolved.

Table 6.3: HRR Category Agreement

Fire	Predicted Category	True Category	Correct?
3	4	4	Yes
5	4	4	Yes
6	4	4	Yes
9	4	2	No
10	4	4	Yes
12	4	4	Yes
14	4	4	Yes
15	4	4	Yes
19	1	1	Yes
24	1	1	Yes
29	4	4	Yes
34	4	4	Yes
39	2	2	Yes
44	2	2	Yes
49	4	4	Yes
54	4	4	Yes
Douglas fir 0.25xHRR	3	4	No
Polyethylene 2xHRR	1	1	Yes
Polyethylene 4xHRR	1	2	No
Paper 0.25xHRR	2	2	Yes
Paper 0.5xHRR	3	2	No
Propane 0.25xHRR	3	3	Yes
JP-8 0.25xHRR	3	3	Yes
JP-8 0.5xHRR	4	4	Yes
Douglas fir 0.5xHRR	4	4	Yes
Propane 0.5xHRR	4	4	Yes
Polyurethane foam	4	4	Yes

Fire 9 is a Douglas fir fire with wind, in location F. As previously discussed with the inability to locate the paper fire in location F, wind plays a greater role in the disturbance of the flame structure and temperature when the fire is located in the corner of the hangar, near the open hangar door. The corner effects weren't large enough to impact the location model, but they were large enough to reduce the temperature in a

way that the model believes that the HRR associated with this temperature is much lower than the actual HRR. The polyethylene 4xHRR fire and the Douglas fir 0.25xHRR fire were very close to being correctly categorized.

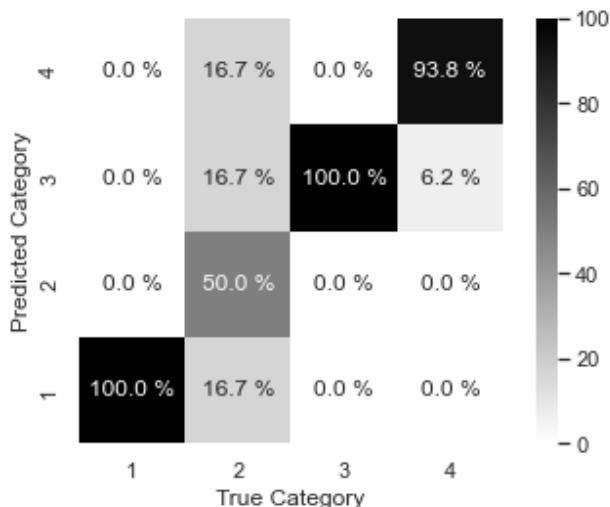


Figure 6.10: HRR confusion matrix.

6.3 Fuel Discrimination Model

The two approaches that will be discussed in the following section are the “all fuels” identification and the binary fuels identification.

6.3.1 All Fuels Approach

The test accuracy is what quantifies how well the SVM did on the classification. This approach resulted in 62% test accuracy. This accuracy can be further analyzed by the confusion matrix shown in Figure 6.11.

The x-axis corresponds to the true fuel, while the y-axis corresponds to the predicted fuel. Douglas fir is denoted as “DF”, polyethylene is “PE”, propane is “Pr” and paper is “Pa”. Table 6.4 shows how many instances a certain fuel was tested on. Insight from the matrix shows that the largest misclassification error is occurring with Douglas fir fires, as 58.8% of these fires are predicted as being polyethylene. This is due to the slow growth of the HRR in the Douglas fir fire over the course of the simulation. The HRR does not ramp up until the simulation time is between 125 and 150 seconds. It is understood that the HRR of the polyethylene is orders of magnitude lower than that of the other fuels, so the SVM gets confused in this

sense. It has learned that during the 125 seconds of the simulation, polyethylene has a small, close to zero HRR, therefore the nonexistent HRR of Douglas fir during this time frame poses issues for classification, as the model thinks that Douglas fir could be polyethylene.

Table 6.4: Total “All Fuels” seen

Fuel	Instances
Douglas Fir	7856
Polyethylene	1964
Propane	1964
Paper	1964
JP-8	1964

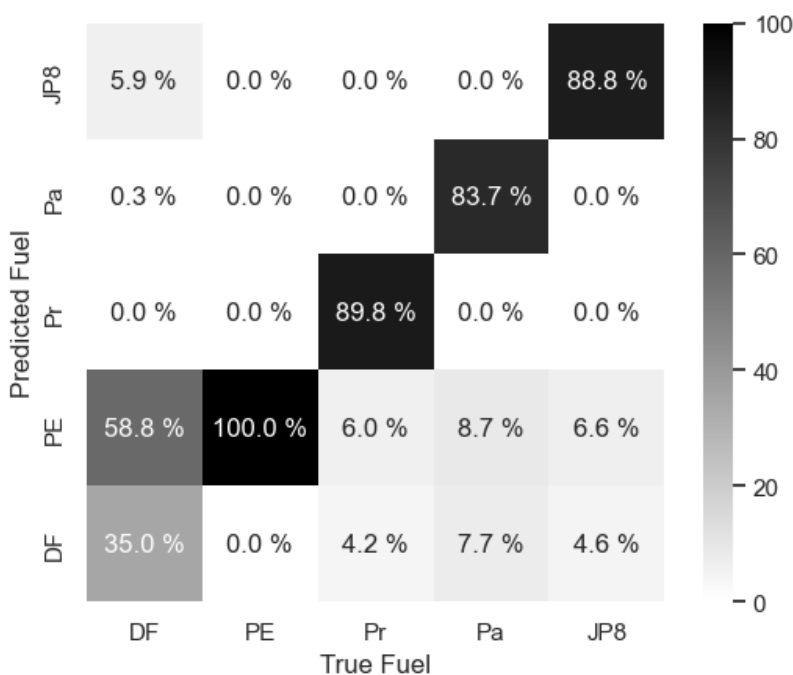


Figure 6.11: Confusion matrix of classification on all five fuel types.

6.3.2 Binary Fuel Approach

The approach to differentiate fuel type based on suppression agent yielded better result than the all fuels approach, as there was difficulty in classifying similar fuel sources. There was 91% testing accuracy when classifying on a binary basis and the details of this model’s performance is seen in Figure 6.12, with the total number of instances of test cases in Table 6.5.

Table 6.5: Total Binary Fuels seen

Fuel	Instances
Other	13748
JP-8	1964

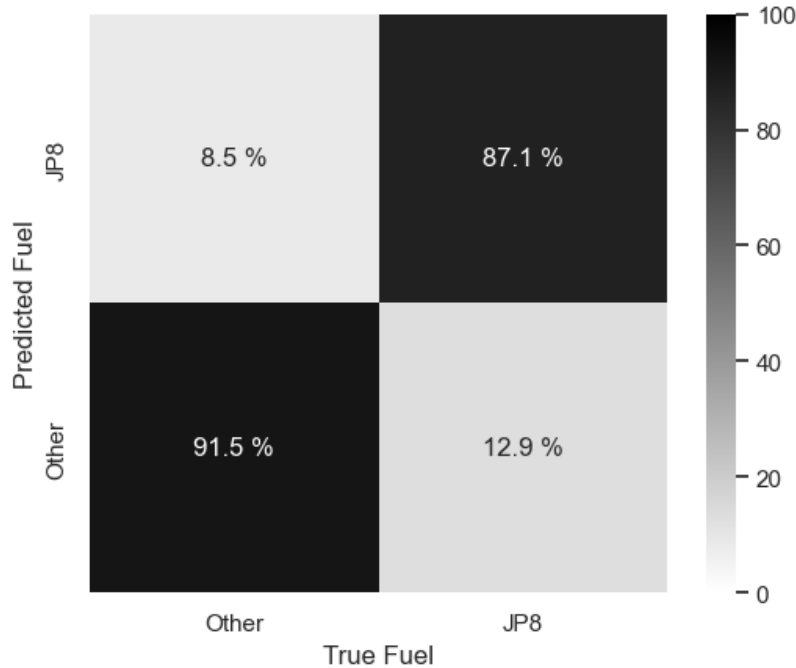


Figure 6.12: Confusion Matrix of binary classification of JP-8 versus other fuels.

6.3.3 Comparison Between Classification Methods

It is of interest to determine how the accuracy of the model changes as a function of time. This relationship is explored in Figure 6.13. The accuracy was determined at 30 second intervals in time, starting from 0 to 30 seconds, and ending with 240 to 270 seconds. The corresponding SVM was used to predict the classification accuracy for each of the nine, 30 second intervals, which were then plotted as a point. The same approach was taken to predict the classification on a binary set of fuels.

The dashed line represents the average test accuracy on the entire training set. The testing accuracy on all fuels tended to increase with time, while the accuracy of the binary classification remained relatively stable, mainly decreasing in the final time frames.

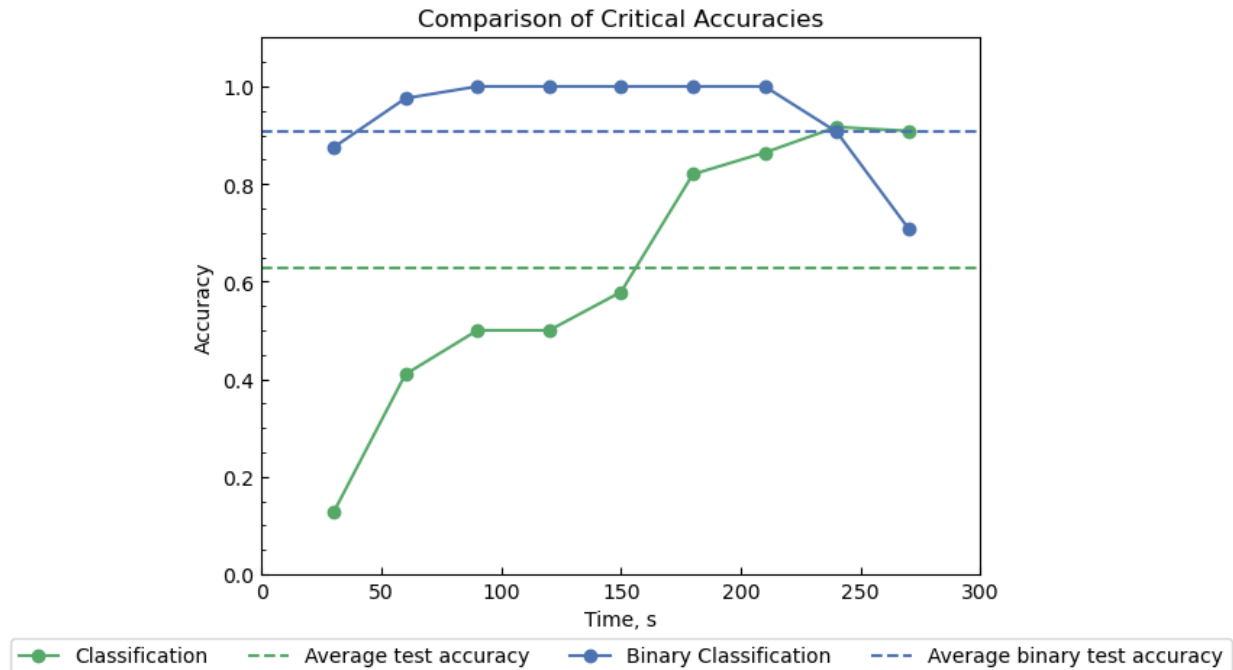


Figure 6.13: Critical accuracy comparison of all fuels versus binary classification.

The time at which the model is trained and tested yields different results. It was determined that, for this specific fuel set, that there is better performance once the SVM starts training at later instances in time. When the SVM started training on all fuels at one minute, there was 71% accuracy, and when it was trained starting at two minutes, the accuracy increased to 80%. This is because the later in time, the more time the products have to develop and differentiate from the other fuels, specifically in the initial growth stages.

Chapter 7

Conclusion and Future Work

The limitations in getting the bulk of the information about the fire scenario upon arrival to the scene and the manual process of surveying the scene of the fire are addressed with machine learning techniques. This information is obtained by adjusting the sensors to collect continuous data, that the models can then use to make predictions. These techniques allow for the size-up process to be automated, providing the firefighter with pertinent information before getting to the scene of the fire, which allows for quicker intervention and suppression of the fire.

This research determined that it is feasible to use machine learning techniques to determine where a fire is located, the size of the fire and the composition of the fuel that is burning. FDS simulations were performed on a combination of fuels, locations and environmental conditions. The models made use of sensor data from FDS, specifically temperature and smoke obscuration, as inputs to answer the key questions posed in the scope of this project, and provide better situational awareness. The location identification method reduced the amount of data needed for the HRR model, which predicted the design curve and magnitude of the fire using a three-layer feed forward neural network, and did the same for the fuel discrimination model, which used a linear support vector machine that considered time dependency.

Increased situational awareness is needed for incident commanders to be better prepared to deal with a fire scenario, and is especially important for aircraft hangars, where the monetary loss in addition to loss of productivity is extremely high. The size-up process is performed with sensors and machine learning from the indication of alarm until firefighters enter the hangar and begin fighting the fire. Each methodology yielded promising results, with over 60% accuracy for all models.

The location methodology was able to identify all fires with a HRR exceeding a minimum threshold,

which nominally equates to HRRs larger than 50 kW to 250 kW, as a critical range rather than a specific critical HRR was identified. There is the exception of the paper fire in location F in the corner, so it is determined that fires of a certain size can be largely affected by entrainment effects in the corner of the hangar, which could likely be extended to entrainment effects when the fire is located by the wall. In a large space like a hangar, it is beneficial to have the location of the fire narrowed down from a large area (such as 676 m² in the case of the hangar used for the FDS models) to a circular area of approximately 78.5 m². This reduces the time it takes the incident commander to survey the building, looking for the fire. If they are given the coordinates or region of the hangar where the fire is suspected to be, they can go straight there, reducing the potential time the fire is able to grow and spread. The location method had 80% accuracy, which means there is an 80% chance that the fire is identified faster, and the time on the scene can be spent on fighting rather than finding the fire.

It is beneficial from a situational awareness standpoint to know the relative size of the fire that is burning, as a firefighter is approaching the scene of the fire. As mentioned before, knowing the exact size of the fire, for example, to the nearest 10 kW, is not important. Knowing a general category (small, medium, large or extra large) in which the fire size falls in is of interest to determine the plan of attack to extinguish the fire, and further determine if this is a routine or hazardous fire. The larger the fire, the more equipment and people will be needed. The HRR model from temperature yields good results because it is now 85% likely that the firefighters will be better prepared for the severity of the fire they are facing. It is important to clarify that the model is only able to give real-time updates of the predicted HRR and was not designed to predict the HRR at future times. Therefore the model is only beneficial for the duration of time it is growing. However, the simulations demonstrate that the fuels of interest grow and then decay within five minutes, or grow and reach a steady state value in this time frame. Assuming the fire does not spread to other objects in the hangar, the incident commander will know the largest size of the fire as they are approaching the hangar, as they are likely not at the scene of the fire within five minutes.

It is beneficial to identify the individual fuel packages from the lens of academia, and a success rate of 62% is very promising. However, when asking the questions during size-up like “what is burning?” this can be seen almost as asking “what resources are needed?” because specific fuels require specific suppression agents. This is why classifying on a binary basis and considering the suppression agent needed is of greater interest to firefighters, as determining, for example, is the fire paper or Douglas fir is not imperative as both fires can be fought using water. JP-8 fires are most effectively fought with foam suppression agents, and 91% accuracy of classification when identifying JP-8 versus others answers what resources are needed. This

combined increased knowledge of the fire scene allows for a better understanding of the current environment and automates and simplifies the processing of data needed to obtain proper situational awareness.

These models prove successful for the feasibility of coupling sensors with machine learning in a hangar configuration, however there are limitations to the domain of validity of this model. Future work and further validation will need to be performed if this model were to be deployed in a hangar, or similar space.

The HRR was defined as HRRPUA and was not impacted by the presence of wind. It would be beneficial to broaden the scope of applicability and make the model more realistic by defining the combustion reaction using an evaporation model. Defining the reaction this way becomes more computationally expensive, as there is a need for better grid resolution to properly account for convection. Heat losses to the system and the surroundings need to be properly described in order to not over account for the heat feedback present in the system. On paper, this reaction is possible to define, but in practice may prove difficult which is why it is recommended to begin this approach with liquid fuels of interest, before extending this to the pyrolysis of solid fuels. This model accounts for the variation in evaporation in time based on thermal feedback, which allows for this model to be extended for the consideration of defining the potential fire spread to other objects as a function of applied heat flux.

This leads to the next area of future work, which is the consideration of how the model will work if the fire spreads as the simulation progresses. The model is currently designed for a fixed size (1 m^2) fire, which may or may not be realistic depending on the initial size of the fuel package or the other potential fuel sources in the neighboring area, though it is assumed that items in a hangar are typically well spread out. It would be of interest to determine how to adjust the location identification model to account for this growth, as the model would have to pick a different sensor location as a function of time and broaden the region used for analysis of the HRR as the fire spreads throughout the space. This spread can be defined through the use of a predefined spread rate or through utilization of the evaporation/pyrolysis model, which will take into account applied heat flux from the primary to the secondary fuel sources.

Another way to incorporate more data is to expand the SVM to incorporate CO and/or CO₂, which results in a visualization of a potential four or five-dimensional graph. This might allow for greater identification of trends and more separability when looking at an all fuels approach, as this approach had a greater challenge for the first half of the simulation, specifically when Douglas fir was still developing. Tracking these additional products might allow for different trends to be seen and allow for polyethylene to be differentiated from Douglas fir early on. A version of PCA was performed when using temperature, smoke and time, as these were manually determined to explain most of the variance within the data set. However it will be beneficial

to explore the effect of incorporating CO and CO₂ as these products could still have some proportion of explained variance.

A ML model is only as good as the data set that it is trained on. Therefore, it is beneficial to increase the size of the training set if this model were to be deployed in a hangar, to ensure the confidence in its performance. Suggestions for an increased data set include exploring other locations within the hangar. Locations along the centerline of the hangar and the corner closest to the hangar door were explored for this feasibility study. The model would become more robust if other locations were analyzed and it is believed that fires closest to the wall and corners will provide the greatest deviation from the fires burning in the interior of the hangar. It would also be of benefit to adjust the wind velocity to address the variation in wind patterns as different wind profiles can be simulated. The training set used for this study consisted of fires when the hangar door was open, so it would be of interest to study the effect of the hangar door being closed. It is not expected that a fire in the hangar will become underventilated due to the extremely loose leakage and overall large volume of the space, however the lack of wind or natural airflow from the outside through the 168 m² opening could impact the fire. To provide a “safety factor” for this model it would be beneficial to train on a marginal amount of “unexpected” fuels for the occupancy. This will increase the chances that the HRR and fuel package will be properly predicted, even on abnormal days. It should be understood that an infinite number of fuels could be added to the training set, so there needs to be a cut-off at some point, as a sufficiently large training set, with accepted fuels of interest will allow the model to learn the patterns in the data and extrapolate easier to unseen fires. Additionally, it is understood that FDS is not a perfect model and is a qualitative tool, therefore the ML models should be experimentally validated if this were to be deployed in a hangar. The fuels of interest could be burned in a laboratory setting to provide the most accurate data, and to allow for the effect of natural environmental factors to be seen, such as variations in temperature and ventilation conditions in the lab. Experiments could also be performed in a large open space like a hangar to further confirm the applicability. Using a model to validate another model is a good first step, and the FDS simulations can be used to narrow down what fuels should be studied experimentally and would help cut down on associated monetary costs, but this needs to go a step further for this model to be sufficiently validated and for there to be total confidence in its performance.

The models have been confirmed to work in an open space with a tall ceiling, but they could be expanded to another type of building. The building can be more compartmentalized, in which case the location search would be even more helpful. If the space of interest had a taller ceiling height, the models might have to be adjusted to consider the smoke transport. Additionally, a new building requires a new subset of fuels of

interest to be identified and modeled, as the fuels used for this research were assumed to be present in a hangar and will not be applicable to other occupancies.

The last suggestions for future research deal with the structure of the neural network. The first is performing a grid search, which is a systematic method used to analyze combinations of various hyperparameters to identify the optimal hyperparameters based on the model performance. This could include adjusting the model architecture, learning rate, number of epochs, the optimizer and more. Grid search can be “automated” or performed manually, but either way this is a time extensive process when training a deep neural network. The final suggestion is creating a more complex model like a CNN or RNN that can predict how the HRR will develop in time and use the outputs of the network to inform the inputs. The feed forward neural network takes in real-time data and makes predictions at the same time. However, it could be beneficial to develop this model to be able to predict the HRR a minute or so out, to increase the level of information that firefighters get, allowing for an increase in situational awareness, specifically when it comes to predicting the risk associated with the fire.

Appendix A

FDS Input Files

It is important to note, that the mesh needs to be redefined with the new location to consider the 7 m x 7 m fire plume changing location.

A.1 Locations

The vent and surface IDs for the 6 locations of interest are listed below.

Location A: &VENT XB = 12.5,13.5,3,4,0,0, SURF_ID='FIRE' /

Location B: &VENT XB = 12.5,13.5,8.5,9.5,0,0, SURF_ID='FIRE' /

Location C: &VENT XB = 12.5,13.5,12.5,13.5,0,0, SURF_ID='FIRE' /

Location D: &VENT XB = 12.5,13.5,18,19,0,0, SURF_ID='FIRE' /

Location E: &VENT XB = 12.5,13.5,22,23,0,0, SURF_ID='FIRE' /

Location F: &VENT XB = 3,4,3,4,0,0, SURF_ID='FIRE' /

A.2 No Wind Case (Location B)

/dx =0.1–0.5

&MESH ID='FirePlume1 ', IJK=70,70,25, XB=0,7,0,7,0,2.5/ dx=0.1

&MESH ID='FirePlume2 ', IJK=70,70,25, XB=0,7,0,7,2.5,5/ dx=0.1

&MESH ID='FirePlume3 ', IJK=70,70,25, XB=0,7,0,7,5,7.5/ dx=0.1

&MESH ID='FirePlume4 ', IJK=70,70,35, XB=0,7,0,7,7.5,11/ dx=0.1

```

&MESH ID='SmokeLayer1' , IJK=130,90,10, XB=0,13,0,9,11,12/ dx=0.1
&MESH ID='SmokeLayer2' , IJK=130,90,10, XB=0,13,9,18,11,12/ dx=0.1
&MESH ID='SmokeLayer3' , IJK=130,80,10, XB=0,13,18,26,11,12/ dx=0.1
&MESH ID='SmokeLayer4' , IJK=130,90,10, XB=13,26,0,9,11,12/ dx=0.1
&MESH ID='SmokeLayer5' , IJK=130,90,10, XB=13,26,9,18,11,12/ dx=0.1
&MESH ID='SmokeLayer6' , IJK=130,80,10, XB=13,26,18,26,11,12/ dx=0.1

```

```

&MESH ID='Mesh01' , IJK=38,22,22, XB=7,26,0,11,0,11/ dx=0.5
&MESH ID='Mesh02' , IJK=14,38,22, XB=0,7,7,26,0,11/ dx=0.5
&MESH ID='Mesh03' , IJK=38,30,22, XB=7,26,11,26,0,11/ dx=0.5

```

```

&MESH ID='Outdoors1' , IJK=52,12,24,XB=0,26,-6,0,0.0,12.0/ dx = 0.5

```

```

&TIME T_END=300/

```

```

&MISC TMPA = 25, HUMIDITY = 65 /

```

```

&MATL ID = 'CONCRETE' ,
    DENSITY = 2200 ,
    SPECIFIC_HEAT = 1.04 ,
    CONDUCTIVITY = 1.8 /

```

```

&SURF ID = 'FLOOR' ,
    MATL_ID = 'CONCRETE'
    THICKNESS = 0.1
    COLOR = 'BEIGE' /

```

```

&MATL ID = 'CORRUGATED METAL' ,
    CONDUCTIVITY = 52 ,

```

SPECIFIC_HEAT = 0.47 ,
DENSITY = 7800 /

&SURF ID = 'WALL' ,
THICKNESS = 0.2 ,
DEFAULT = .TRUE. ,
MATL_ID = 'CORRUGATED METAL'
COLOR = 'LAVENDER' /

&MATL ID = 'Carbon Steel' ,
FYI = '1.5 % carbon steel from NCFS database' ,
CONDUCTIVITY = 36 ,
SPECIFIC_HEAT = 0.486 ,
DENSITY = 7753 /

&SURF ID = 'ELECTRICAL PANEL' ,
THICKNESS = 0.05 ,
MATL_ID = 'Carbon Steel' ,
COLOR='MELON' /

/ create hangar space
&OBST XB = 0,26,0,26,0,0, SURF_ID = 'FLOOR' / main floor
&OBST XB = 0,26,0,0.4,0,12, SURF_ID = 'WALL' / front wall

&HOLE XB = 1,25,-0.4,0.8,0,7/ hangar door

/open outdoor space
&VENT XB = 0,26,-6,-6,0,12, SURF_ID = 'OPEN' /
&VENT XB = 26,26,-6,0,0,12, SURF_ID = 'OPEN' /
&VENT XB = 0,0,-6,0,0,12, SURF_ID = 'OPEN' /

&VENT XB = 0,26,-6,0,0,0, SURF_ID = 'FLOOR' /

/ electrical cabinet(s)

&OBST XB = 0,0.6,12.8,13.6,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='BLUE'
' /

&OBST XB = 0,0.6,23.6,24.4,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='BLUE'
' /

&OBST XB = 25.6,26,12.8,13.6,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='BLUE'
' /

&OBST XB = 25.6,26,23.6,24.4,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='BLUE'
' /

/smoke detectors cleary model

&DEVC ID = 'YZ=1.625,12', XB=1.625,24.375,1.625,1.625,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID = 'YZ=4.875,12', XB=1.625,24.375,4.875,4.875,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID = 'YZ=8.125,12', XB=1.625,24.375,8.125,8.125,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID = 'YZ=11.375,12', XB=1.625,24.375,11.375,11.375,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID = 'YZ=14.625,12', XB=1.625,24.375,14.625,14.625,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID = 'YZ=17.875,12', XB=1.625,24.375,17.875,17.875,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID = 'YZ=21.125,12', XB=1.625,24.375,21.125,21.125,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID = 'YZ=24.375,12', XB=1.625,24.375,24.375,24.375,12,12, PROP_ID='Cleary

Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&PROP ID='Cleary Photoelectric P1', QUANTITY='CHAMBER OBSCURATION', ALPHA_E
=1.8, BETA_E=-1.0,
ALPHA_C=1.0, BETA_C=-0.8, ACTIVATION_OBSCURATION=3.24/

/temperature sensors that act as heat detectors

&DEVC ID ='HD YZ=1.625,12', XB=1.625,24.375,1.625,1.625,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=4.875,12', XB=1.625,24.375,4.875,4.875,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=8.125,12', XB=1.625,24.375,8.125,8.125,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=11.375,12', XB=1.625,24.375,11.375,11.375,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=14.625,12', XB=1.625,24.375,14.625,14.625,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=17.875,12', XB=1.625,24.375,17.875,17.875,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=21.125,12', XB=1.625,24.375,21.125,21.125,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=24.375,12', XB=1.625,24.375,24.375,24.375,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

/CO Detectors

&DEVC ID ='CO YZ=1.625', XB=1.625,24.375,1.625,1.625,12,12, QUANTITY='VOLUME
FRACTION', SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='CO YZ=4.875', XB=1.625,24.375,4.875,4.875,12,12, QUANTITY='VOLUME
FRACTION', SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='CO YZ=8.125', XB=1.625,24.375,8.125,8.125,12,12, QUANTITY='VOLUME

FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=11.375', XB=1.625,24.375,11.375,11.375,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=14.625', XB=1.625,24.375,14.625,14.625,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=17.875', XB=1.625,24.375,17.875,17.875,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=21.125', XB=1.625,24.375,21.125,21.125,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=24.375', XB=1.625,24.375,24.375,24.375,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/

/CO2 Detectors

&DEVC ID ='CO2 YZ=1.625', XB=1.625,24.375,1.625,1.625,12,12, QUANTITY='VOLUME
 FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=4.875', XB=1.625,24.375,4.875,4.875,12,12, QUANTITY='VOLUME
 FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=8.125', XB=1.625,24.375,8.125,8.125,12,12, QUANTITY='VOLUME
 FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=11.375', XB=1.625,24.375,11.375,11.375,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=14.625', XB=1.625,24.375,14.625,14.625,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=17.875', XB=1.625,24.375,17.875,17.875,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=21.125', XB=1.625,24.375,21.125,21.125,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=24.375', XB=1.625,24.375,24.375,24.375,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/

```

&REAC FUEL = 'Douglas_Fir',
      FORMULA='C6H10O5',
      HEAT_OF_COMBUSTION=16400,
      SOOT_YIELD=0.015,
      CO_YIELD=0.004/

&SURF ID='FIRE'
      HRRPUA=3700,
      RAMP_Q='fireramp' / HRRPUA assuming the area of the fire is 1m^2

&RAMP ID='fireramp', T=0, F=0/
&RAMP ID='fireramp', T=125, F=0/
&RAMP ID='fireramp', T=150, F=0.0338/
&RAMP ID='fireramp', T=200, F=0.0676/
&RAMP ID='fireramp', T=250, F=0.216/
&RAMP ID='fireramp', T=300, F=0.581/
&RAMP ID='fireramp', T=350, F=0.946/
&RAMP ID='fireramp', T=400, F=1/
&RAMP ID='fireramp', T=625, F=1/
&RAMP ID='fireramp', T=680, F=0.953/
&RAMP ID='fireramp', T=775, F=0.703/

&VENT XB = 3,4,3,4,0,0, SURF_ID='FIRE' /

&TAIL/

```

A.3 Wind Case (Location F)

```

/dx =0.1-0.5

&MESH ID='FirePlume1', IJK=70,70,25, XB=0,7,0,7,0,2.5/ dx=0.1
&MESH ID='FirePlume2', IJK=70,70,25, XB=0,7,0,7,2.5,5/ dx=0.1
&MESH ID='FirePlume3', IJK=70,70,25, XB=0,7,0,7,5,7.5/ dx=0.1

```

```

&MESH ID='FirePlume4' , IJK=70,70,35, XB=0,7,0,7,7.5,11/ dx=0.1

&MESH ID='SmokeLayer1' , IJK=130,90,10, XB=0,13,0,9,11,12/ dx=0.1
&MESH ID='SmokeLayer2' , IJK=130,90,10, XB=0,13,9,18,11,12/ dx=0.1
&MESH ID='SmokeLayer3' , IJK=130,80,10, XB=0,13,18,26,11,12/ dx=0.1
&MESH ID='SmokeLayer4' , IJK=130,90,10, XB=13,26,0,9,11,12/ dx=0.1
&MESH ID='SmokeLayer5' , IJK=130,90,10, XB=13,26,9,18,11,12/ dx=0.1
&MESH ID='SmokeLayer6' , IJK=130,80,10, XB=13,26,18,26,11,12/ dx=0.1

&MESH ID='Mesh01' , IJK=38,22,22, XB=7,26,0,11,0,11/ dx=0.5
&MESH ID='Mesh02' , IJK=14,38,22, XB=0,7,7,26,0,11/ dx=0.5
&MESH ID='Mesh03' , IJK=38,30,22, XB=7,26,11,26,0,11/ dx=0.5

&MESH ID='Outdoors1' , IJK=52,20,24,XB=0,26,-10,0,0.0,12.0/ dx = 0.5

&TIME T_END=300/

&MISC TMPA = 25, HUMIDITY = 65 /

&MATL ID = 'CONCRETE' ,
      DENSITY = 2200 ,
      SPECIFIC_HEAT = 1.04 ,
      CONDUCTIVITY = 1.8 /

&SURF ID = 'FLOOR' ,
      MATL_ID = 'CONCRETE'
      THICKNESS = 0.1
      COLOR = 'BEIGE' /

&MATL ID = 'CORRUGATED METAL' ,

```

CONDUCTIVITY = 52 ,
SPECIFIC_HEAT = 0.47 ,
DENSITY = 7800 /

&SURF ID = 'WALL' ,
THICKNESS = 0.2 ,
DEFAULT = .TRUE. ,
MATL_ID = 'CORRUGATED METAL'
COLOR = 'LAVENDER' /

&MATL ID = 'Carbon Steel' ,
FYI = '1.5 % carbon steel from NCFS database' ,
CONDUCTIVITY = 36 ,
SPECIFIC_HEAT = 0.486 ,
DENSITY = 7753 /

&SURF ID = 'ELECTRICAL PANEL' ,
THICKNESS = 0.05 ,
MATL_ID = 'Carbon Steel' ,
COLOR='MELON' /

/ create hangar space
&OBST XB = 0,26,0,26,0,0, SURF_ID = 'FLOOR' / main floor
&OBST XB = 0,26,0,0.4,0,12, SURF_ID = 'WALL' / front wall

/Leakage on walls and at ceiling
&VENT XB=0,0,4,4.3,1,1.3, SURF_ID='OPEN' / left wall
&VENT XB=0,0,21.7,22,1,1.3, SURF_ID='OPEN' / left wall
&VENT XB=0,0,4,4.3,10.7,11, SURF_ID='OPEN' / left wall
&VENT XB=0,0,21.7,22,10.7,11, SURF_ID='OPEN' / left wall

```

&VENT XB=4,4.3,26,26,1,1.3, SURF_ID='OPEN' / back wall
&VENT XB=21.7,22,26,26,1,1.3, SURF_ID='OPEN' / back wall
&VENT XB=4,4.3,26,26,10.7,11, SURF_ID='OPEN' / back wall
&VENT XB=21.7,22,26,26,10.7,11, SURF_ID='OPEN' / back wall

```

```

&VENT XB=26,26,4,4.3,1,1.3, SURF_ID='OPEN' / right wall
&VENT XB=26,26,21.7,22,1,1.3, SURF_ID='OPEN' / right wall
&VENT XB=26,26,4,4.3,10.7,11, SURF_ID='OPEN' / right wall
&VENT XB=26,26,21.7,22,10.7,11, SURF_ID='OPEN' / right wall

```

```

&HOLE XB = 1,25,-0.4,0.8,0,7 / hangar door

```

```

/OUDOOR SPACE WITH WIND (wall of wind approach)
&SURF ID='inlet ', VEL=-4.4, PROFILE='ATMOSPHERIC', Z0=10, PLE=0.14/
&VENT XB = 0,26,-10,-10,0,12, SURF_ID = 'inlet' /
&VENT XB = 26,26,-10,0,0,12, SURF_ID = 'OPEN' /
&VENT XB = 0,0,-10,0,0,12, SURF_ID = 'OPEN' /
&VENT XB = 0,26,-10,0,12,12, SURF_ID='OPEN' /

```

```

/Visualize velocity
&SLCF PBX=0, QUANTITY='VELOCITY', VECTOR=T/
&SLCF PBX=2, QUANTITY='VELOCITY', VECTOR=T/
&SLCF PBX=9, QUANTITY='VELOCITY', VECTOR=T/
&SLCF PBX=11, QUANTITY='VELOCITY', VECTOR=T/
&SLCF PBX=13, QUANTITY='VELOCITY', VECTOR=T/
&SLCF PBX=15, QUANTITY='VELOCITY', VECTOR=T/
&SLCF PBX=17, QUANTITY='VELOCITY', VECTOR=T/

```

&SLCF PBX=24, QUANTITY='VELOCITY', VECTOR=T/

&SLCF PBX=26, QUANTITY='VELOCITY', VECTOR=T/

&SLCF PBY=26, QUANTITY='VELOCITY', VECTOR=T/

/ electrical cabinet(s)

&OBST XB = 0,0.6,12.8,13.6,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='BLUE'
'/

&OBST XB = 0,0.6,23.6,24.4,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='BLUE'
'/

&OBST XB = 25.6,26,12.8,13.6,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='BLUE'
BLUE' /

&OBST XB = 25.6,26,23.6,24.4,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='BLUE'
BLUE' /

/smoke detectors cleary model

&DEVC ID ='YZ=1.625,12', XB=1.625,24.375,1.625,1.625,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='YZ=4.875,12', XB=1.625,24.375,4.875,4.875,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='YZ=8.125,12', XB=1.625,24.375,8.125,8.125,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='YZ=11.375,12', XB=1.625,24.375,11.375,11.375,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='YZ=14.625,12', XB=1.625,24.375,14.625,14.625,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='YZ=17.875,12', XB=1.625,24.375,17.875,17.875,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='YZ=21.125,12', XB=1.625,24.375,21.125,21.125,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='YZ=24.375,12', XB=1.625,24.375,24.375,24.375,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T/

&PROP ID='Cleary Photoelectric P1', QUANTITY='CHAMBER OBSCURATION', ALPHA_E
=1.8, BETA_E=-1.0,
ALPHA_C=1.0, BETA_C=-0.8, ACTIVATION_OBSCURATION=3.24/

/temperature sensors that act as heat detectors

&DEVC ID ='HD YZ=1.625,12', XB=1.625,24.375,1.625,1.625,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=4.875,12', XB=1.625,24.375,4.875,4.875,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=8.125,12', XB=1.625,24.375,8.125,8.125,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=11.375,12', XB=1.625,24.375,11.375,11.375,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=14.625,12', XB=1.625,24.375,14.625,14.625,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=17.875,12', XB=1.625,24.375,17.875,17.875,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=21.125,12', XB=1.625,24.375,21.125,21.125,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

&DEVC ID ='HD YZ=24.375,12', XB=1.625,24.375,24.375,24.375,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T/

/CO Detectors

&DEVC ID ='CO YZ=1.625', XB=1.625,24.375,1.625,1.625,12,12, QUANTITY='VOLUME

FRACTION' ,SPEC_ID='CARBON MONOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=4.875' , XB=1.625,24.375,4.875,4.875,12,12 , QUANTITY='VOLUME
 FRACTION' ,SPEC_ID='CARBON MONOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=8.125' , XB=1.625,24.375,8.125,8.125,12,12 , QUANTITY='VOLUME
 FRACTION' ,SPEC_ID='CARBON MONOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=11.375' , XB=1.625,24.375,11.375,11.375,12,12 , QUANTITY='
 VOLUME FRACTION' ,SPEC_ID='CARBON MONOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=14.625' , XB=1.625,24.375,14.625,14.625,12,12 , QUANTITY='
 VOLUME FRACTION' ,SPEC_ID='CARBON MONOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=17.875' , XB=1.625,24.375,17.875,17.875,12,12 , QUANTITY='
 VOLUME FRACTION' ,SPEC_ID='CARBON MONOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=21.125' , XB=1.625,24.375,21.125,21.125,12,12 , QUANTITY='
 VOLUME FRACTION' ,SPEC_ID='CARBON MONOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=24.375' , XB=1.625,24.375,24.375,24.375,12,12 , QUANTITY='
 VOLUME FRACTION' ,SPEC_ID='CARBON MONOXIDE' , POINTS=8, TIME_HISTORY=T/

/CO2 Detectors

&DEVC ID ='CO2 YZ=1.625' , XB=1.625,24.375,1.625,1.625,12,12 , QUANTITY='VOLUME
 FRACTION' ,SPEC_ID='CARBON DIOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=4.875' , XB=1.625,24.375,4.875,4.875,12,12 , QUANTITY='VOLUME
 FRACTION' ,SPEC_ID='CARBON DIOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=8.125' , XB=1.625,24.375,8.125,8.125,12,12 , QUANTITY='VOLUME
 FRACTION' ,SPEC_ID='CARBON DIOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=11.375' , XB=1.625,24.375,11.375,11.375,12,12 , QUANTITY='
 VOLUME FRACTION' ,SPEC_ID='CARBON DIOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=14.625' , XB=1.625,24.375,14.625,14.625,12,12 , QUANTITY='
 VOLUME FRACTION' ,SPEC_ID='CARBON DIOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=17.875' , XB=1.625,24.375,17.875,17.875,12,12 , QUANTITY='
 VOLUME FRACTION' ,SPEC_ID='CARBON DIOXIDE' , POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=21.125' , XB=1.625,24.375,21.125,21.125,12,12 , QUANTITY='

VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO2 YZ=24.375', XB=1.625,24.375,24.375,24.375,12,12, QUANTITY='
 VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/

&REAC FUEL = 'Paper_Box',
 FORMULA='C6H10O5',
 HEAT_OF_COMBUSTION=12500,
 SOOT_YIELD=0.002,
 CO_YIELD=0.029/

&SURF ID='FIRE'
 HRRPUA=436,
 RAMP_Q='fireramp' / HRRPUA assuming the area of the fire is 1m^2

&RAMP ID='fireramp', T=0, F=0/
 &RAMP ID='fireramp', T=60, F=0.1148/
 &RAMP ID='fireramp', T=120, F=1/
 &RAMP ID='fireramp', T=180, F=0.1147/
 &RAMP ID='fireramp', T=300, F=0.0344/
 &RAMP ID='fireramp', T=600, F=0.01147/
 &RAMP ID='fireramp', T=720, F=0/

&VENT XB = 3,4,3,4,0,0, SURF_ID='FIRE' /

&TAIL/

A.4 Beam Case (Location A)

/dx =0.1-0.5
 &MESH ID='FirePlume1', IJK=70,70,25, XB=9.5,16.5,9.5,16.5,0,2.5/
 &MESH ID='FirePlume2', IJK=70,70,25, XB=9.5,16.5,9.5,16.5,2.5,5/
 &MESH ID='FirePlume3', IJK=70,70,25, XB=9.5,16.5,9.5,16.5,5,7.5/

&MESH ID='FirePlume4' , IJK=70,70,35 , XB=9.5,16.5,9.5,16.5,7.5,11/

&MESH ID='SmokeLayer1' , IJK=130,90,10 , XB=0,13,0,9,11,12/

&MESH ID='SmokeLayer2' , IJK=130,90,10 , XB=0,13,9,18,11,12/

&MESH ID='SmokeLayer3' , IJK=130,80,10 , XB=0,13,18,26,11,12/

&MESH ID='SmokeLayer4' , IJK=130,90,10 , XB=13,26,0,9,11,12/

&MESH ID='SmokeLayer5' , IJK=130,90,10 , XB=13,26,9,18,11,12/

&MESH ID='SmokeLayer6' , IJK=130,80,10 , XB=13,26,18,26,11,12/

&MESH ID='Mesh01' , IJK=19,52,22 , XB=0,9.5,0,26,0,11/ dx=0.5

&MESH ID='Mesh02' , IJK=33,19,22 XB=9.5,26,0,9.5,0,11/ dx=0.5

&MESH ID='Mesh03' , IJK=19,14,22 , XB=16.5,26,9.5,16.5,0,11/ dx=0.5

&MESH ID='Mesh04' , IJK=33,19,22 , XB=9.5,26,16.5,26,0,11/ dx=0.5

&MESH ID='Outdoors1' , IJK=52,12,24,XB=0,26,-6,0,0.0,12.0/ dx = 0.5

&TIME T_END=300/

&MISC TMPA = 25 , HUMIDITY = 65 /

&MATL ID = 'CONCRETE' ,

DENSITY = 2200 ,

SPECIFIC_HEAT = 1.04 ,

CONDUCTIVITY = 1.8 /

&SURF ID = 'FLOOR' ,

MATL_ID = 'CONCRETE'

THICKNESS = 0.1

COLOR = 'BEIGE' /

&MATL ID = 'CORRUGATED METAL',
CONDUCTIVITY = 52,
SPECIFIC_HEAT = 0.47,
DENSITY = 7800 /

&SURF ID = 'WALL',
THICKNESS = 0.2,
DEFAULT = .TRUE.,
MATL_ID = 'CORRUGATED METAL'
COLOR = 'LAVENDER' /

&MATL ID = 'Carbon Steel',
FYI = '1.5 % carbon steel from NCFS database',
CONDUCTIVITY = 36,
SPECIFIC_HEAT = 0.486,
DENSITY = 7753 /

&SURF ID = 'ELECTRICAL PANEL',
THICKNESS = 0.05,
MATL_ID = 'Carbon Steel'
COLOR = 'MELON' /

&MATL ID = 'STEEL SN400',
FYI = 'SN400 International Journal of Steel Structures',
CONDUCTIVITY_RAMP = 'k',
SPECIFIC_HEAT_RAMP = 'c',
DENSITY = 7851 /

&RAMP ID = 'c', T = 20, F = 0.52/
&RAMP ID = 'c', T = 200, F = 0.535/

&RAMP ID = 'c' , T = 500, F = 0.709/

&RAMP ID = 'k' , T = 20, F = 53.7/

&RAMP ID = 'k' , T = 200, F = 48.3/

&RAMP ID = 'k' , T = 500, F = 42.1/

&SURF ID = 'Structural Steel',
THICKNESS = 0.006,
MATL_ID = 'STEEL SN400'
COLOR='DIM GRAY'/

/ create hangar space

&OBST XB = 0,26,0,26,0,0, SURF_ID = 'FLOOR' / main floor

&OBST XB = 0,26,0,0.4,0,12, SURF_ID = 'WALL' / front wall

&HOLE XB = 1,25,-0.4,0.8,0,7/ hangar door

/ perpendicular beams (to door)

&OBST ID='web', XB=8.9965,9.0035,0.5587,25.4413,11.608,11.989, SURF_ID='

Structural Steel'/

&OBST ID='top flange', XB = 8.9298,9.0702,0.5587,25.4413,11.989,12.0, SURF_ID

='Structural Steel' /

&OBST ID='bottom flange', XB = 8.9298,9.0702,0.5587,25.4413,11.597,11.608,

SURF_ID='Structural Steel'/

&OBST ID = 'web 1', XB=16.9965,17.0035,0.5587,25.4413,11.608,11.989, SURF_ID='

Structural Steel'/

&OBST ID = 'top flange 1', XB=16.9298,17.0702,0.5587,25.4413,11.989,12.0,

SURF_ID='Structural Steel'/

&OBST ID = 'bottom flange 1', XB=16.9298,17.0702,0.5587,25.4413,11.597,11.608,

```

SURF_ID='Structural Steel'/

/ parallel beams (to door)
&OBST ID='par web', XB=0.5587,25.4413,8.9965,9.0035,11.608,11.989, SURF_ID='
Structural Steel'/
&OBST ID='par top flange', XB=0.5587,25.4413,8.9298,9.0702,11.989,12.0,
SURF_ID='Structural Steel'/
&OBST ID = 'par bottom flange', XB=0.5587,25.4413,8.9298,9.0702,11.587,11.608,
SURF_ID='Structural Steel'/

&OBST ID='par web 1', XB=0.5587,25.4413,16.9965,17.0035,11.608,11.989, SURF_ID
='Structural Steel'/
&OBST ID='par top flange 1', XB=0.5587,25.4413,16.9298,17.0702,11.989,12.0,
SURF_ID='Structural Steel'/
&OBST ID='par bottom flange 1', XB
=0.5587,25.4413,16.9298,17.0702,11.587,11.608, SURF_ID='Structural Steel'/

/open outdoor space
&VENT XB = 0,26,-6,-6,0,12, SURF_ID = 'OPEN' /
&VENT XB = 26,26,-6,0,0,12, SURF_ID = 'OPEN' /
&VENT XB = 0,0,-6,0,0,12, SURF_ID = 'OPEN' /
&VENT XB = 0,26,-6,0,0,0, SURF_ID = 'FLOOR' /

/ electrical cabinet(s)

&OBST XB = 0,0.6,12.8,13.6,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='BLUE
'/
&OBST XB = 0,0.6,23.6,24.4,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='BLUE
'/

```

&OBST XB = 25.6,26,12.8,13.6,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='
BLUE' /

&OBST XB = 25.6,26,23.6,24.4,1.6,2.4, SURF_ID = 'ELECTRICAL PANEL', COLOR='
BLUE' /

/smoke detectors cleary model

&DEVC ID = 'YZ=1.625,12', XB=1.625,24.375,1.625,1.625,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T /

&DEVC ID = 'YZ=4.875,12', XB=1.625,24.375,4.875,4.875,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T /

&DEVC ID = 'YZ=8.125,12', XB=1.625,24.375,8.125,8.125,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T /

&DEVC ID = 'YZ=11.375,12', XB=1.625,24.375,11.375,11.375,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T /

&DEVC ID = 'YZ=14.625,12', XB=1.625,24.375,14.625,14.625,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T /

&DEVC ID = 'YZ=17.875,12', XB=1.625,24.375,17.875,17.875,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T /

&DEVC ID = 'YZ=21.125,12', XB=1.625,24.375,21.125,21.125,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T /

&DEVC ID = 'YZ=24.375,12', XB=1.625,24.375,24.375,24.375,12,12, PROP_ID='Cleary
Photoelectric P1', POINTS=8, TIME_HISTORY=T /

&PROP ID='Cleary Photoelectric P1', QUANTITY='CHAMBER OBSCURATION', ALPHA_E
=1.8, BETA_E=-1.0,

ALPHA_C=1.0, BETA_C=-0.8, ACTIVATION_OBSCURATION=3.24 /

/temperature sensors that act as heat detectors

&DEVC ID = 'HD YZ=1.625,12', XB=1.625,24.375,1.625,1.625,12,12, QUANTITY='
TEMPERATURE', POINTS=8, TIME_HISTORY=T /

&DEVC ID ='HD YZ=4.875,12', XB=1.625,24.375,4.875,4.875,12,12, QUANTITY='TEMPERATURE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='HD YZ=8.125,12', XB=1.625,24.375,8.125,8.125,12,12, QUANTITY='TEMPERATURE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='HD YZ=11.375,12', XB=1.625,24.375,11.375,11.375,12,12, QUANTITY='TEMPERATURE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='HD YZ=14.625,12', XB=1.625,24.375,14.625,14.625,12,12, QUANTITY='TEMPERATURE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='HD YZ=17.875,12', XB=1.625,24.375,17.875,17.875,12,12, QUANTITY='TEMPERATURE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='HD YZ=21.125,12', XB=1.625,24.375,21.125,21.125,12,12, QUANTITY='TEMPERATURE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='HD YZ=24.375,12', XB=1.625,24.375,24.375,24.375,12,12, QUANTITY='TEMPERATURE', POINTS=8, TIME_HISTORY=T/

/CO Detectors

&DEVC ID ='CO YZ=1.625', XB=1.625,24.375,1.625,1.625,12,12, QUANTITY='VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=4.875', XB=1.625,24.375,4.875,4.875,12,12, QUANTITY='VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=8.125', XB=1.625,24.375,8.125,8.125,12,12, QUANTITY='VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=11.375', XB=1.625,24.375,11.375,11.375,12,12, QUANTITY='VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=14.625', XB=1.625,24.375,14.625,14.625,12,12, QUANTITY='VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=17.875', XB=1.625,24.375,17.875,17.875,12,12, QUANTITY='VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/
 &DEVC ID ='CO YZ=21.125', XB=1.625,24.375,21.125,21.125,12,12, QUANTITY='VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/


```

&DEVC ID = 'CO YZ=24.375', XB=1.625,24.375,24.375,24.375,12,12, QUANTITY='
    VOLUME FRACTION',SPEC_ID='CARBON MONOXIDE', POINTS=8, TIME_HISTORY=T/

/CO2 Detectors
&DEVC ID = 'CO2 YZ=1.625', XB=1.625,24.375,1.625,1.625,12,12, QUANTITY='VOLUME
    FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
&DEVC ID = 'CO2 YZ=4.875', XB=1.625,24.375,4.875,4.875,12,12, QUANTITY='VOLUME
    FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
&DEVC ID = 'CO2 YZ=8.125', XB=1.625,24.375,8.125,8.125,12,12, QUANTITY='VOLUME
    FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
&DEVC ID = 'CO2 YZ=11.375', XB=1.625,24.375,11.375,11.375,12,12, QUANTITY='
    VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
&DEVC ID = 'CO2 YZ=14.625', XB=1.625,24.375,14.625,14.625,12,12, QUANTITY='
    VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
&DEVC ID = 'CO2 YZ=17.875', XB=1.625,24.375,17.875,17.875,12,12, QUANTITY='
    VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
&DEVC ID = 'CO2 YZ=21.125', XB=1.625,24.375,21.125,21.125,12,12, QUANTITY='
    VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/
&DEVC ID = 'CO2 YZ=24.375', XB=1.625,24.375,24.375,24.375,12,12, QUANTITY='
    VOLUME FRACTION',SPEC_ID='CARBON DIOXIDE', POINTS=8, TIME_HISTORY=T/

&REAC FUEL = 'Douglas_Fir',
    FORMULA='C6H10O5',
    HEAT_OF_COMBUSTION=16400,
    SOOT_YIELD=0.015,
    CO_YIELD=0.004/

&SURF ID='FIRE'
    HRRPUA=3700,
    RAMP_Q='fireramp' / HRRPUA assuming the area of the fire is 1m^2

```

```

&RAMP ID='fireramp ', T=0, F=0/
&RAMP ID='fireramp ', T=125, F=0/
&RAMP ID='fireramp ', T=150, F=0.0338/
&RAMP ID='fireramp ', T=200, F=0.0676/
&RAMP ID='fireramp ', T=250, F=0.216/
&RAMP ID='fireramp ', T=300, F=0.581/
&RAMP ID='fireramp ', T=350, F=0.946/
&RAMP ID='fireramp ', T=400, F=1/
&RAMP ID='fireramp ', T=625, F=1/
&RAMP ID='fireramp ', T=680, F=0.953/
&RAMP ID='fireramp ', T=775, F=0.703/

&VENT XB = 12.5,13.5,12.5,13.5,0,0, SURF_ID='FIRE'/

&TAIL/

```

A.5 Design Fire Ramp-Up Values

The entire duration of the ramp function may or may not be taken into account during the simulation, as the simulation is only 300 seconds long.

A.5.1 Douglas fir

```

&REAC FUEL = 'Douglas_Fir ',
      FORMULA='C6H10O5 ',
      HEAT_OF_COMBUSTION=16400,
      SOOT_YIELD=0.015,
      CO_YIELD=0.004/

&SURF ID='FIRE'
      HRRPUA=3700,
      RAMP_Q='fireramp '/ HRRPUA assuming the area of the fire is 1m^2

```

```

&RAMP ID='fireramp ', T=0, F=0/
&RAMP ID='fireramp ', T=125, F=0/
&RAMP ID='fireramp ', T=150, F=0.0338/
&RAMP ID='fireramp ', T=200, F=0.0676/
&RAMP ID='fireramp ', T=250, F=0.216/
&RAMP ID='fireramp ', T=300, F=0.581/
&RAMP ID='fireramp ', T=350, F=0.946/
&RAMP ID='fireramp ', T=400, F=1/
&RAMP ID='fireramp ', T=625, F=1/
&RAMP ID='fireramp ', T=680, F=0.953/
&RAMP ID='fireramp ', T=775, F=0.703/

```

A.5.2 Polyethylene

```

&REAC FUEL = 'POLYETHYLENE',
      FORMULA='C2H4',
      HEAT_OF_COMBUSTION=43400,
      SOOT_YIELD=0.06,
      CO_YIELD=0.024/

&SURF ID='FIRE'
      HRRPUA=285,
      RAMP_Q='fireramp '/ HRRPUA assuming the area of the fire is 1m^2
&RAMP ID='fireramp ', T=0, F=0/
&RAMP ID='fireramp ', T=150, F=0.1754/
&RAMP ID='fireramp ', T=300, F=0.193/

```

A.5.3 Paper

```

&REAC FUEL = 'Paper_Box',
      FORMULA='C6H10O5',
      HEAT_OF_COMBUSTION=12500,
      SOOT_YIELD=0.002,

```

CO_YIELD=0.029/

&SURF ID='FIRE'

HRRPUA=436,

RAMP_Q='fireramp' / HRRPUA assuming the area of the fire is 1m²

&RAMP ID='fireramp' , T=0, F=0/

&RAMP ID='fireramp' , T=60, F=0.1148/

&RAMP ID='fireramp' , T=120, F=1/

&RAMP ID='fireramp' , T=180, F=0.1147/

&RAMP ID='fireramp' , T=300, F=0.0344/

&RAMP ID='fireramp' , T=600, F=0.01147/

&RAMP ID='fireramp' , T=720, F=0/

A.5.4 JP-8

&REAC FUEL = 'JP-8',

FORMULA='C11H21' ,

HEAT_OF_COMBUSTION=42800,

SOOT_YIELD=0.03,

CO_YIELD=0.01/

&SURF ID='FIRE'

HRRPUA=2200,

RAMP_Q='fireramp' / HRRPUA assuming the area of the fire is 1m²

&RAMP ID='fireramp' , T=0, F=0/

&RAMP ID='fireramp' , T=25, F=0.0227/

&RAMP ID='fireramp' , T=45, F=0.227/

&RAMP ID='fireramp' , T=65, F=0.4545/

&RAMP ID='fireramp' , T=75, F=0.568/

&RAMP ID='fireramp' , T=85, F=0.682/

&RAMP ID='fireramp' , T=115, F=1/

&RAMP ID='fireramp' , T=300, F=1/

A.5.5 Propane

```
&REAC FUEL = 'PROPANE' ,  
      FORMULA='C3H8' ,  
      HEAT_OF_COMBUSTION=46300 ,  
      SOOT_YIELD=0.024 ,  
      CO_YIELD=0.005/  
  
&SURF ID='FIRE'  
      HRRPUA=2200 ,  
      RAMP_Q='fireramp' / HRRPUA assuming the area of the fire is 1m^2  
  
&RAMP ID='fireramp' , T=0, F=0/  
&RAMP ID='fireramp' , T=25, F=0.0227/  
&RAMP ID='fireramp' , T=45, F=0.227/  
&RAMP ID='fireramp' , T=65, F=0.4545/  
&RAMP ID='fireramp' , T=75, F=0.568/  
&RAMP ID='fireramp' , T=85, F=0.682/  
&RAMP ID='fireramp' , T=115, F=1/  
&RAMP ID='fireramp' , T=300, F=1/
```

A.5.6 Polyurethane Foam

```
&REAC FUEL = 'POLYURETHANE' ,  
      FORMULA='CH1.8O0.3N0.05' ,  
      HEAT_OF_COMBUSTION=26200 ,  
      SOOT_YIELD=0.131 ,  
      CO_YIELD=0.010/  
  
&SURF ID='FIRE'  
      HRRPUA=2000 ,  
      RAMP_Q='fireramp' / HRRPUA assuming the area of the fire is 1m^2  
  
&RAMP ID='fireramp' , T=0, F=0/
```

&RAMP ID='fireramp ', T=130, F=0/
&RAMP ID='fireramp ', T=140, F=0.025/
&RAMP ID='fireramp ', T=150, F=0.05/
&RAMP ID='fireramp ', T=200, F=0.125/
&RAMP ID='fireramp ', T=270, F=1/
&RAMP ID='fireramp ', T=280, F=1/
&RAMP ID='fireramp ', T=300, F=0.675/

Appendix B

Python Code

The following code has been developed to create the location, HRR and fuel identification models.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.pyplot import figure
from numpy import genfromtxt

import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import torch
import torch.nn as nn

from sklearn import svm
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix,
    accuracy_score

import matplotlib.ticker as tck
from matplotlib.ticker import MultipleLocator
```

Import Excel files for temperature here

```
combine = np.hstack((DF_A,DF_C,DF_E,DF_F,DF_A_W,DF_B_W,DF_C_W,DF_D_W,DF_E_W,
    DF_F_W,DF_A_Bm,DF_B_Bm,DF_C_Bm,DF_D_Bm,DF_E_Bm,DF_F_Bm,PE_A,PE_C,PE_E,PE_F,
    PE_A_W,PE_B_W,
    PE_C_W,PE_D_W,PE_E_W,PE_F_W,Pr_A,Pr_C,Pr_E,Pr_F,Pr_A_W,
    Pr_B_W, Pr_C_W,Pr_D_W,Pr_E_W,Pr_F_W,Pa_A,Pa_C,Pa_E,Pa_F,
    Pa_A_W,Pa_B_W, Pa_C_W,Pa_D_W,Pa_E_W,Pa_F_W,
    JP8_A,JP8_C,JP8_E,JP8_F,JP8_A_W,JP8_B_W,JP8_C_W,JP8_D_W,
    JP8_E_W,JP8_F_W))

print(combine.shape)

smokeCols = np.arange(1,65)
tempCols = np.arange(65,129)
COCols = np.arange(129,193)
CO2Cols = np.arange(193,257)

def running_mean(x, N):
    cumsum = np.cumsum(np.insert(x, 0, 0))
    return (cumsum[N:] - cumsum[:-N]) / float(N)

RAsmoke = np.zeros((982,64))
RAtemp = np.zeros((982,64))
brackets = np.arange(0,14649,257)
loopRange = np.arange(0,55974,982)
allTemp = np.zeros((54992,64))
allSmoke = np.zeros((54992,64))
for i in range(56):
    getFuel = combine[1:,brackets[i]:brackets[i+1]] # look at each fire
    individually
    smoke = getFuel[:,1:65]
    temp = getFuel[:,65:129]
    for j in range(64):
        RAsmoke[:,j] = running_mean(smoke[:,j],20) # smooth the data for each fire
        by taking the running average
```



```

        RAtemp[:,j] = running_mean(temp[:,j],20)
    allTemp[loopRange[i]:loopRange[i+1],:] = RAtemp # put all of the temperatures
        into 1 array
    allSmoke[loopRange[i]:loopRange[i+1],:] = RAsmoke # put all of the
        temperatures into 1 array

# transpose the matrices so it works for later portions
allTemp = allTemp.T
print(allTemp.shape)
allSmoke = allSmoke.T
print(allSmoke.shape)

#account for uncertainty
unc = 25*0.15 # ambient temperature multiplied by 15%
amb = [25-unc,25+unc]
tempCols = np.arange(65,129)

maximum = np.zeros((56,982))
brackets = np.zeros((57)) # should be dimension x+1
for i in range(56):
    brackets[i+1] = 982 + 982*i #allows for each fuel/condition to be looked at
        separately

maxs = np.zeros(54992)
check = np.zeros(54992)

for i in range(56):
    x = np.arange(brackets[i],brackets[i+1])
    for j in range(982):
        temp = allTemp[:,int(x[j])] # gets the temperature at every sensor for a
            specific fuel/condition at a single instance in time
        maxs[int(x[j])] = np.argmax(temp) # gives the maximum values at every
            instance in time
    a = maxs[int(x[j])]

```

```

        check[int(x[j])] = temp[int(a)] # stores the maximum smoke obscuration
        into a large 1D array

# Fires and Corresponding Indices
fireTitle =['DF_A','DF_C','DF_E','DF_F','DF_AW','DF_BW','DF_CW','DF_DW','DF_EW','
            DF_FW','DF_AB','DF_BB','DF_CB','DF_DB','DF_EB',
            'DF_FB','PE_A','PE_C','PE_E','PE_F','PE_AW','PE_BW','PE_CW','
            PE_DW','PE_EW','PE_FW','Pr_A','Pr_C','Pr_E','Pr_F',
            'Pr_AW','Pr_BW','Pr_CW','Pr_DW','Pr_EW','Pr_FW','Pa_A','Pa_C',
            'Pa_E','Pa_F','Pa_AW','Pa_BW','Pa_CW','Pa_DW','Pa_EW','
            Pa_FW',
            'JP8_A','JP8_C','JP8_E','JP8_F','JP8_AW','JP8_BW','JP8_CW','
            JP8_DW','JP8_EW','JP8_FW']

overlap = np.zeros(54992)

# check to make sure the smoke is above a critical threshold and identify this as
the location - start with something more sensitive so when it picks up a non-
zero smoke obscuration
for i in range(len(maxs)):
    CI_temp = check[i]*0.15
    temp_range = [check[i]-CI_temp,check[i]+CI_temp]
    overlap[i] = temp_range[0]-amb[1]
print(overlap.shape)

# Coordinate locations of the sensors in the hangar
X_Loc = np.zeros(8)
Y_Loc = np.zeros(8)
for i in range(0,8):
    X_Loc[i] = 1.625 + 3.25*i
    Y_Loc[i] = 1.625 + 3.25*i

y1 = np.arange(0,8)
y2 = np.arange(8,16)
y3 = np.arange(16,24)
y4 = np.arange(24,32)

```

```

y5 = np.arange(32,40)
y6 = np.arange(40,48)
y7 = np.arange(48,56)
y8 = np.arange(56,64)
Y_range = np.zeros(64)

for i in range(0,8):
    Y_range[y1[i]] = Y_Loc[0] # 0-7
    Y_range[y2[i]] = Y_Loc[1] # 8-15
    Y_range[y3[i]] = Y_Loc[2] # 16-23
    Y_range[y4[i]] = Y_Loc[3] # 24-31
    Y_range[y5[i]] = Y_Loc[4] # 32-39
    Y_range[y6[i]] = Y_Loc[5] # 40-47
    Y_range[y7[i]] = Y_Loc[6] # 48-55
    Y_range[y8[i]] = Y_Loc[7] # 56-63

X_range = np.hstack((X_Loc,X_Loc,X_Loc,X_Loc,X_Loc,X_Loc,X_Loc,X_Loc))
coord = np.zeros((64,2))

for i in range(64):
    coord[i,:] = np.hstack((X_range[i],Y_range[i]))

#actual fire location coordinates
LocA = [13,3.5] # 1,5,11,17,20,26,29,32,35
LocB = [13,9] #6,12,21,36
LocC = [13,13] #2,7,13,18,22,27,30,33,37
LocD = [13,18.5] #8,14,23,38
LocE = [13,22.5] #3,9,15,24,39
LocF = [3.5,3.5] #4,10,16,19,25,28,31,34,40

actual = [LocA,LocC,LocE,LocF,LocA,LocB,LocC,LocD,LocE,LocF,LocA,LocB,LocC,LocD,
          LocE,LocF,LocA,
          LocC,LocE,LocF,LocA,LocB,LocC,LocD,LocE,LocF,LocA,LocC,LocE,LocF,LocA,
          LocB,LocC,LocD,

```

```

LocE,LocF,LocA,LocC,LocE,LocF,LocA,LocB,LocC,LocD,LocE,LocF,LocA,LocC,
LocE,LocF,LocA,LocB,LocC,LocD,LocE,LocF]

```

```

sensors = np.zeros((56,9))
valStore = np.zeros((56,2))
for i in range(56):
    #look at each fuel/condition at a time
    a = int(brackets[i])
    b = int(brackets[i+1])
    comp = overlap[a:b]
    maxrange = maxs[a:b]

    if comp[i].all()>0: # only go into this loop if the overlap yields a positive
        value
        array = np.where(comp>0)
        array0 = np.array(array[0])
        if len(array0)<10 and len(array0)>0: #when the temperatures are smoothed
            out, nothing works for the PE cables...
            idx = array0[0]
            loc = maxrange[idx]
            sensors[i,:] = [loc-9,loc-8,loc-7,loc-1, loc, loc+1, loc+7, loc+8, loc
                +9]
            val = coord[int(loc)]
            valStore[i,0] = val[0]
            valStore[i,1] = val[1]
            interest = actual[i]
            fig, ax=plt.subplots()
            ax.tick_params(axis='both',which='both',direction='in')
            ax.yaxis.set_minor_locator(tck.AutoMinorLocator())
            ax.set_aspect('equal')
            label = (('Fire',i+1,'Location_Prediction'))
            plt.title(label)
            ax.set_xlim([0,26])
            ax.set_ylim([0,26])

```

```

fire = plt.Circle((val[0],val[1]),5,color='red',alpha=0.5,label='
    Potential_Fire_Location') # changed to 5 for testing
ax.add_patch(fire)
plt.scatter(val[0],val[1],color = 'blue',label='First_Sensor_
    Activation')
plt.scatter(interest[0],interest[1],color='black',marker='*',label='
    Actual_Location')
plt.legend()
plt.show()
print('Fire',i+1,'is_closest_to_sensor',loc)
print('The fire is assumed to be located within a circular region
    centered at (' , val[0], 'm', ' , val[1], 'm) with a radius of 5m' )
if len(array0)>10:
    for j in range(len(array0)):
        idx = array0[j]
        loc = maxrange[idx]
        sensors[i,:] = [loc-9,loc-8,loc-7,loc-1, loc, loc+1, loc+7, loc+8,
            loc+9]
        val = coord[int(loc)]
        valStore[i,0] = val[0]
        valStore[i,1] = val[1]
        interest = actual[i]
        fig, ax=plt.subplots()
        ax.tick_params(axis='both',which='both',direction='in')
        ax.yaxis.set_minor_locator(tck.AutoMinorLocator())
        ax.set_aspect('equal')
        label = (('Fire',i+1,'Location_Prediction'))
        plt.title(label)
        ax.set_xlim(0,26)
        ax.set_ylim(0,26)
        fire = plt.Circle((val[0],val[1]),5,color='red',alpha=0.5,label='
            Potential_Fire_Location') # changed to 5 for testing
        ax.add_patch(fire)

```

```

plt.scatter(val[0],val[1],color = 'blue',label='First_Sensor_Activation')

plt.scatter(interest[0],interest[1],color='black',marker='*',label='Actual_Location')

plt.legend()

plt.show()

print('Fire',i+1,'is_closest_to_sensor',loc)

print('The fire is assumed to be located within a circular region centered at (' ,val[0] , 'm' , ' ,val[1] , 'm') with a radius of 5m' )

break

elif len(array0)==0:

    print('Unable_to_identify_location_of_Fire',i+1)

```

Get sensors of interest

```

AVG = np.zeros((56,982))
AVGsmoke = np.zeros((56,982))
e = 9
pos_count, neg_count, great_count = 0,0,0
for i in range(56):
    a = int(brackets[i])
    b = int(brackets[i+1])
    pos_count, neg_count, great_count = 0,0,0
    for j in range(9):
        if sensors[i][j] >= 0:
            pos_count +=1
            e = 9
        f_i = sensors[i][0:e]
        if sensors[i][j] > 63:
            great_count +=1
            e = pos_count-great_count
            f_i = sensors[i][0:e]
        f_i = np.zeros((pos_count))
        s = abs(pos_count-9)

```

```

        f_i = sensors[i][s:e]
    x = len(f_i)
    if all(f_i == 0):
        var = allTemp[:,a:b]
        varSmoke = allSmoke[:,a:b]
    else:
        var = np.zeros((x,982))
        varSmoke = np.zeros((x,982))
        for k in range(x):
            var[k,:] = allTemp[:,a:b][int(f_i[k]),:]
            varSmoke[k,:] = allSmoke[:,a:b][int(f_i[k]),:]

    avg = np.average(var,axis=0)
    avgSmoke = np.average(varSmoke,axis=0)
    AVG[i,:] = avg # takes the average value of the sensors of interest and puts
                    # them into an array for each fire - the order of the temps are in the order
                    # that the fuels were sampled
    AVGsmoke[i,:] = avgSmoke

```

Import Excel files for HRR here

```

allHRR = np.hstack((DF_A_HRR,DF_C_HRR,DF_E_HRR,DF_F_HRR,DF_A_W_HRR,DF_B_W_HRR,
                    DF_C_W_HRR,DF_D_W_HRR,DF_E_W_HRR,DF_F_W_HRR,
                    DF_A_Bm_HRR,DF_B_Bm_HRR,DF_C_Bm_HRR,DF_D_Bm_HRR,DF_E_Bm_HRR,
                    DF_F_Bm_HRR,PE_A_HRR,PE_C_HRR,PE_E_HRR,PE_F_HRR,
                    PE_A_W_HRR,PE_B_W_HRR,PE_C_W_HRR,PE_D_W_HRR,PE_E_W_HRR,
                    PE_F_W_HRR,Pr_A_HRR,Pr_C_HRR,Pr_E_HRR,Pr_F_HRR,Pr_A_W_HRR,
                    Pr_B_W_HRR,Pr_C_W_HRR,Pr_D_W_HRR,Pr_E_W_HRR,Pr_F_W_HRR,
                    Pa_A_HRR,Pa_C_HRR,Pa_E_HRR,Pa_F_HRR,Pa_A_W_HRR,Pa_B_W_HRR,
                    Pa_C_W_HRR,Pa_D_W_HRR,Pa_E_W_HRR,Pa_F_W_HRR, JP8_A_HRR,
                    JP8_C_HRR,JP8_E_HRR,JP8_F_HRR,JP8_A_W_HRR,JP8_B_W_HRR,
                    JP8_C_W_HRR,JP8_D_W_HRR,JP8_E_W_HRR,JP8_F_W_HRR))

bracketsHRR = np.zeros((57)) # should be dimension x+1
totalHRR = np.zeros((56,1000))

```

```

for i in range(56):
    bracketsHRR[i+1] = 13 + 13*i
    c = int(bracketsHRR[i])
    d = int(bracketsHRR[i+1])
    value = allHRR[:,c:d]
    totalHRR[i,:] = value[2:,1]

RAhrr = np.zeros((56,981))

for i in range(56):
    RAhrr[i,:] = running_mean(totalHRR[i],20)

scaler = MinMaxScaler()
scaler.fit(AVG.reshape(-1,1))
print(scaler.data_max_)
print(scaler.data_min_)
xtrans = scaler.transform(AVG.reshape(-1,1))
tempdata = np.zeros((56,981))
r = np.zeros(57)
for i in range(56):
    r[i+1] = 981*i + 981
    tempdata[i,:] = xtrans[int(r[i]):int(r[i+1])].reshape(1,-1)

scaler2 = MinMaxScaler()
scaler2.fit(RAhrr.reshape(-1,1))
print(scaler2.data_max_)
print(scaler2.data_min_)
ytrans = scaler2.transform(RAhrr.reshape(-1,1))

hrrdata = np.zeros((56,981))
r = np.zeros(57)
for i in range(56):
    r[i+1] = 981*i + 981
    hrrdata[i,:] = ytrans[int(r[i]):int(r[i+1])].reshape(1,-1)

want =
    [0,1,2,4,7,8,11,13,16,17,18,20,21,22,23,25,26,27,28,30,31,32,33,35,36,37,38,

```



```

40,41,42,43,45,46,47,48,50,51,52,53,55]
wantTest = [3,5,6,9,10,12,14,15,19,24,29,34,39,44,49,54]

xTrainManual = np.zeros((len(want),981))
yTrainManual = np.zeros((len(want),981))
xTestManual = np.zeros((len(wantTest),981))
yTestManual = np.zeros((len(wantTest),981))

for i in range(len(want)):
    xTrainManual[i,:] = tempdata[want[i]]
    yTrainManual[i,:] = hrrdata[want[i]]

for i in range(len(wantTest)):
    xTestManual[i,:] = tempdata[wantTest[i]]
    yTestManual[i,:] = hrrdata[wantTest[i]]

```

Neural Network Architecture

```

class Net(nn.Module):
    def __init__(self, input_size, hidden_size):
        super(Net, self).__init__()

        self.fc1 = nn.Linear(input_size, hidden_size)
        self.ReLU = nn.ReLU()
        self.fc2 = nn.Linear(input_size, hidden_size)
        self.ReLU2 = nn.ReLU()
        self.fc3 = nn.Linear(input_size, hidden_size)
        self.ReLU3 = nn.ReLU()
        self.fc4 = nn.Linear(hidden_size, input_size)

    # Forward pass builds the model prediction from the inputs
    def forward(self, x):
        #####
        out = self.fc1(x)

```

```

        out = self.ReLU(out)
        out = self.fc2(x)
        out = self.ReLU2(out)
        out = self.fc3(x)
        out = self.ReLU3(out)
        out = self.fc4(out)

    pass

    return out

```

```

model = Net(input_size=981, hidden_size=400)

criterion = nn.L1Loss()

learning_rate = 0.00001
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate, weight_decay
                              =0.01)

```

```

num_epochs = 20000
MAE_train = np.zeros((num_epochs))
MAE_test = np.zeros((num_epochs))
MAE_train_us = np.zeros((num_epochs))
MAE_test_us = np.zeros((num_epochs))

for epoch in range(num_epochs):
    total_loss=0
    test_total_loss = 0

    for i in range(xTrainManual.shape[0]):
        inputs = torch.from_numpy(xTrainManual[i,:].astype(np.float32))
        targets = torch.from_numpy(yTrainManual[i,:].astype(np.float32))

        ## Do Forward pass
        # Make predictions
        outputs = model(inputs)

        # Compute the loss function

```

```

        loss = criterion(outputs, targets)

        total_loss+=loss

# Update the model
# Reset the optimizer gradients
optimizer.zero_grad()
# Compute the gradient of the loss function
total_loss.backward()
# Do an optimization step
optimizer.step()

MAE_train[epoch] = total_loss.item()
MAE_train_us[epoch] = scaler2.inverse_transform(MAE_train[epoch].reshape(-1,1)
        ) #unscale the MSE to get an actual idea of the value in kW

# Print the loss
if (epoch+1) % 1000 == 0:
    print ('Epoch_{:4}/{:4}], Loss: {:.4f}'.format(epoch+1, num_epochs,
            total_loss.item()))

model.eval()
with torch.no_grad(): # don't add to the computational graph
    for j in range(xTestManual.shape[0]):
        test_inputs = torch.from_numpy(xTestManual[j,:].astype(np.float32))
        test_targets = torch.from_numpy(yTestManual[j,:].astype(np.float32))
        test_outputs = model(test_inputs)
        test_loss = criterion(test_outputs, test_targets)
        test_total_loss+=test_loss

MAE_test[epoch] = test_total_loss.item()
MAE_test_us[epoch] = scaler2.inverse_transform(MAE_test[epoch].reshape(-1,1))
        #unscale the MSE to get an actual idea of the value in kW

```

```

#set the model back into training mode
model.train()

fig, ax = plt.subplots()
ax.tick_params(axis='both', which='both', direction='in')
ax.yaxis.set_minor_locator(tck.AutoMinorLocator())

plt.plot(np.arange(num_epochs), MAE_train, 'r', label='Train')
plt.plot(np.arange(num_epochs), MAE_test, 'b', label='Test')
plt.xlabel('Epochs')
plt.ylabel('log10 Loss')
plt.yscale('log')
plt.xlim(0, 20000)
plt.legend()
plt.savefig('logMAE.png', bbox_inches='tight')
plt.show()

print('MAETrain:', MAE_train[num_epochs-1])
print('MAETest:', MAE_test[num_epochs-1])

fig, ax = plt.subplots()
ax.tick_params(axis='both', which='both', direction='in')
ax.yaxis.set_minor_locator(tck.AutoMinorLocator())
print(MAE_train.shape)
plt.plot(np.arange(num_epochs), MAE_train_us, 'r', label='Train')
plt.plot(np.arange(num_epochs), MAE_test_us, 'b', label='Test')

plt.xlabel('Epochs')
plt.ylabel('Loss10 (kW)')
plt.ylim(0, 35000)
plt.xlim(0, 20000)
plt.legend()
plt.savefig('MAE.png', bbox_inches='tight')

```

```

plt.show()

print('MAETrain:', MAE_train_us[num_epochs-1])
print('MAETest:', MAE_test_us[num_epochs-1])

time = DF_A_HRR[10:991,0]
print(time.shape)

from sklearn.metrics import mean_absolute_error
for i in range(xTrainManual.shape[0]):
    figure()
    fig, ax = plt.subplots()
    ax.tick_params(axis='both', which='both', direction='in')
    ax.yaxis.set_minor_locator(tck.AutoMinorLocator())
    train = xTrainManual[i,:]
    actual = scaler2.inverse_transform(yTrainManual[i,:].reshape(-1,1))
    pred = model(torch.from_numpy(train.reshape(1,-1).astype(np.float32))).detach
        ().numpy()
    predScale = scaler2.inverse_transform(pred).reshape(-1,1)
    if i<8:
        ax.set_ylim(0,2500)
        ax.set_xlim(0,300)
    if 7<i<16:
        ax.set_ylim(0,70)
        ax.set_xlim(0,300)
    if 15<i<24:
        ax.set_ylim(0,2500)
        ax.set_xlim(0,300)
    if 23<i<32:
        ax.set_ylim(0,500)
        ax.set_xlim(0,300)
    if i>32:
        ax.set_ylim(0,2500)
        ax.set_xlim(0,300)

```

```

plt.plot(time, actual, linewidth=2, label = 'Actual')
plt.plot(time, predScale, linewidth=1, label = 'Prediction')
plt.xlabel('Time_(s)')
plt.ylabel('HRR_(kW)')
plt.legend(loc='upper_left')
# plt.title(want[i])
plt.savefig("Fire"+str(want[i])+"Prediction"+" .png", format="PNG")
plt.show()

# find the individual MAEs
train_mae = mean_absolute_error(pred, yTrainManual[i,:].reshape(1,-1))
us_train_mae = mean_absolute_error(predScale, actual)
print('Fire', want[i], 'Individual_SCALED_Training_MSE:', train_mae)
print('Fire', want[i], 'Individual_UNSCALED_Training_MSE:', us_train_mae)

for i in range(xTestManual.shape[0]):
    figure()
    fig, ax = plt.subplots()
    ax.tick_params(axis='both', which='both', direction='in')
    ax.yaxis.set_minor_locator(tck.AutoMinorLocator())
    test = xTestManual[i,:]
    actualtest = scaler2.inverse_transform(yTestManual[i,:].reshape(-1,1))
    predtest = model(torch.from_numpy(test.reshape(1,-1).astype(np.float32))).
        detach().numpy()
    predScaletest = scaler2.inverse_transform(predtest).reshape(-1,1)
    if i<8:
        ax.set_ylim(0,3000)
        ax.set_xlim(0,300)
    if 7<i<10:
        ax.set_ylim(0,80)
        ax.set_xlim(0,300)
    if 9<i<12:
        ax.set_ylim(0,3000)

```

```

        ax.set_xlim(0,300)
    if 11<i<14:
        ax.set_ylim(0,500)
        ax.set_xlim(0,300)
    if i>13:
        ax.set_ylim(0,3000)
        ax.set_xlim(0,300)

    plt.plot(time,actualtest,linewidth=2,label = 'Actual')
    plt.plot(time,predScaletest,linewidth=1,label = 'Prediction')
    plt.xlabel('Time(s)')
    plt.ylabel('HRR(kW)')
    plt.legend(loc='upper_left')
    plt.savefig("Fire"+str(wantTest[i])+"TestPrediction"+" .png",format="PNG")
    plt.show()
    figure()

    test_mae = mean_absolute_error(predtest,yTestManual[i,:].reshape(1,-1))
    us_test_mae = mean_absolute_error(predScaletest,actualtest)

    print('Fire', wantTest[i], 'Individual_SCALED_Testing_MAE:', test_mae)
    print('Fire', wantTest[i], 'Individual_UNSCALED_Testing_MAE:', us_test_mae)

```

Fuel classification portion

```

# Make 3d graph
stack = np.zeros((54992,3))
timeClass = DF_A_HRR[10:992,0]
plt.rcParams.update(plt.rcParamsDefault)

for i in range(56):
    a = 982*i
    b = 982*(i+1)

    fuelSmoke = AVGsmoke[i,:].reshape(-1,1)
    fuelTemp = AVG[i,:].reshape(-1,1)

```

```

fuelTime = timeClass.reshape(-1,1)

stack[a:b,:] = np.hstack((fuelSmoke,fuelTemp,fuelTime))

DFstack = stack[0:15712,:]
PEstack = stack[15712:25532,:]
Prstack = stack[25532:35352,:]
Pastack = stack[35352:45172,:]
Jp8stack = stack[45172:54992,:]

print(stack[:,0].shape)

fireTitleNumeric =

    [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,3,3,
3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4]

fireTitleNumeric = np.array(fireTitleNumeric).reshape(-1,1)

labels = np.zeros((54992))

labels_w_time = np.array(fireTitleNumeric).reshape(-1,1)

for i in range(56):
    for j in range(982):
        a = 982*i
        b = 982*(i+1)
        labels[a:b] = labels_w_time[i]

labels = [int(i) for i in labels]

labels = np.array(labels)

fig = plt.figure(figsize=(15,15))
ax = fig.add_subplot(projection='3d')
y = labels

print(labels.shape)

```



```

ax.scatter(DFstack[:,0],DFstack[:,2],DFstack[:,1],alpha=0.5,label='Douglas_Fir')
ax.scatter(PEstack[:,0],PEstack[:,2],PEstack[:,1],alpha=0.5,label='Polyethylene')
ax.scatter(Prstack[:,0],Prstack[:,2],Prstack[:,1],alpha=0.5,label='Propane')
ax.scatter(Pastack[:,0],Pastack[:,2],Pastack[:,1],alpha=0.5,label='Paper')
ax.scatter(Jp8stack[:,0],Jp8stack[:,2],Jp8stack[:,1],alpha=0.5,label='JP-8')

ax.set_xlabel('Smoke_Obscuration_(%/m)')
ax.set_zlabel('Temperature_(C)')
ax.set_ylabel('Time_(s)')

ax.legend(markerscale=4, fontsize=20)
plt.savefig('3dPlot.png',bbox_inches='tight')
plt.show()

```

```

# already scaled the temperature so just need to scale the smoke and the time
scalerSmoke = MinMaxScaler()
scalerSmoke.fit(AVGsmoke.reshape(-1,1))
smoketrans = scalerSmoke.transform(AVGsmoke.reshape(-1,1))

scalerTime = MinMaxScaler()
scalerTime.fit(stack[:,2].reshape(-1,1))
timetrans = scalerTime.transform(stack[:,2].reshape(-1,1))

#put these scaled values into an array
scaled = np.hstack((smoketrans,xtrans.reshape(-1,1),timetrans))

```

```

fireRanges = np.arange(0,55974,982)
xManUS = np.zeros((39280,3))
yManUS = np.zeros((39280)) # this should be the labels for the fires (0 to 4)

for i in range(len(want)):

```

```

        instance = want[i]
        c = fireRanges[instance]
        d = fireRanges[instance+1]
        group = scaled[c:d,:]
        xManUS[fireRanges[i]:fireRanges[i+1],:] = group
        yManUS[fireRanges[i]:fireRanges[i+1]] = fireTitleNumeric[instance]

xManUSTest = np.zeros((15712,3))
yManUSTest = np.zeros((15712))
for i in range(len(wantTest)):
    instanceTest = wantTest[i]
    cTest = fireRanges[instanceTest]
    dTest = fireRanges[instanceTest+1]
    groupTest = scaled[cTest:dTest,:]
    xManUSTest[fireRanges[i]:fireRanges[i+1],:] = groupTest
    yManUSTest[fireRanges[i]:fireRanges[i+1]] = fireTitleNumeric[instanceTest]

linear = svm.SVC(kernel='linear',C=1,decision_function_shape='ovo').fit(xManUS,
yManUS)

predTrain = linear.predict(xManUS)
print('Training Accuracy', accuracy_score(yManUS,predTrain)*100)

predTest = linear.predict(xManUSTest)
accTest = accuracy_score(yManUSTest,predTest)
print('Testing Accuracy', accuracy_score(yManUSTest,predTest)*100)

# Confusion Matrix to see what was misclassified
results = confusion_matrix(yManUSTest,predTest)
categories=['DF','PE','Pr','Pa','JP8']

percent = np.zeros((results.shape[0],results.shape[0]))
for i in range(results.shape[0]):
    for j in range(results.shape[0]):

```

```

percent[i,j]= results[i,j]/sum(results[i,:])

print(results)

sns.set()
fig, ax0 = plt.subplots(1,1)
categories=['DF','PE','Pr','Pa','JP8']
ax = sns.heatmap(percent*100, annot=True, fmt = '.1f',xticklabels=categories,
                  yticklabels=categories, square=1,cmap='binary')
for t in ax.texts: t.set_text(t.get_text() + "%")
# plt.title('All Fuels Classification')
plt.xlabel('Predicted_Fuel')
plt.ylabel('True_Fuel')
plt.savefig('AllFuelsConfusion.png',bbox_inches='tight')
plt.show()

```

Find critical instance in time for accuracy

```

# get chunks of 30 seconds which corresponds to approximately 100 instances
a = np.zeros((9,16))
b = np.zeros((9,16))
test = np.zeros((1600,3))
acc = np.zeros(9)
testii_ranges = np.arange(0,1700,100)
yTestii = np.zeros((1600))
for i in range(9):
    for j in range(16):
        a[i,j] = 982*j+100*i
        b[i,j] = a[i,j]+100
        test[(j)*100:(j+1)*100,:] = xManUSTest[int(a[i,j]):int(b[i,j]),:]
        instanceTest = wantTest[j]
        yTestii[testii_ranges[j]:testii_ranges[j+1]] = fireTitleNumeric[
            instanceTest]

    pred = linear.predict(test)
    acc[i] = accuracy_score(yTestii,pred)

```

[illegible]

```

labelsBiTest = np.zeros((15712))
for i in range(16):
    for j in range(982):
        a = 982*i
        b = 982*(i+1)
        labelsBiTest[a:b] = yManBiTest[i]

linear_binary = svm.SVC(kernel='linear',C=1,decision_function_shape='ovo').fit(
    xManUS,labelsBiTrain)

predTrainBi = linear_binary.predict(xManUS)
print('Training Accuracy Manual Binary', accuracy_score(labelsBiTrain,predTrainBi)
    *100)

predTestBi = linear_binary.predict(xManUSTest)
accTestBi = accuracy_score(labelsBiTest,predTestBi)
print('Testing Accuracy Manual', accuracy_score(labelsBiTest,predTestBi)*100)

# Confusion Matrix to see what was misclassified
resultsbi = confusion_matrix(labelsBiTest,predTestBi)
categoriesbi = ['Other','JP8']

percentBi = np.zeros((resultsbi.shape[0],resultsbi.shape[0]))
for i in range(resultsbi.shape[0]):
    for j in range(resultsbi.shape[0]):
        percentBi[i,j]= resultsbi[i,j]/sum(resultsbi[i,:])

sns.set()
fig, ax0 = plt.subplots(1,1)
categories=['DF','PE','Pr','Pa','JP8']
ax = sns.heatmap(percentBi*100, annot=True, fmt = '.1f',xticklabels=categoriesbi,
    yticklabels=categoriesbi, square=1,cmap='binary',vmin=0, vmax=100)
for t in ax.texts: t.set_text(t.get_text() + "%")
# plt.title('Binary Fuel Classification')

```

```

plt.xlabel('Predicted_Fuel')
plt.ylabel('True_Fuel')
plt.savefig('BinaryConfusion.png',bbox_inches='tight')
plt.show()

print(resultsbi)

# get chunks of 30 seconds which corresponds to approximately 100 instances
aBi= np.zeros((9,16))
bBi = np.zeros((9,16))
testBi = np.zeros((1600,3))
accBi = np.zeros(9)
yTestiiBi = np.zeros((1600))

for i in range(9):
    for j in range(16):
        aBi[i,j] = 982*j+100*i
        bBi[i,j] = aBi[i,j]+100
        testBi[(j)*100:(j+1)*100,:] = xManUSTest[int(aBi[i,j]):int(bBi[i,j]),:]
        yTestiiBi[testii_ranges[j]:testii_ranges[j+1]] = yManBiTest[j] # takes the
                                numeric label and appends it 100 times for that chunk of time and
                                moves to the rest
        predBi = linear_binary.predict(testBi)
        accBi[i] = accuracy_score(yTestiiBi,predBi)

```

Put both critical accuracies on same plot and compare

```

figure()
plt.rcParams.update(plt.rcParamsDefault)
fig, ax = plt.subplots()
ax.tick_params(axis='both',which='both',direction='in')
ax.yaxis.set_minor_locator(tck.AutoMinorLocator())
plt.title('Comparison_of_Critical_Accuracies')
plt.xlabel('Time,s')
plt.ylabel('Accuracy')
xTime = np.arange(30,300,30)

```

```

plt.plot(xTime,acc,color = 'g', marker='o',label='Classification')
plt.axhline(y=accTest, color='g', linestyle='--',label='Average_test_accuracy')
plt.plot(xTime,accBi,color='b',marker='o',label='Binary_Classification')
plt.axhline(y=accTestBi, color='b', linestyle='--',label='Average_binary_test_
accuracy')
plt.xlim(0,300)
plt.ylim(0,1.1)
plt.legend(loc='upper_center',bbox_to_anchor=(0.5,-0.1),ncol=5)
# plt.legend(loc='center left',bbox_to_anchor=(1,0.5))
plt.savefig('critAccuracy.png',bbox_inches='tight')
plt.show()

```

Start training at one minute and see accuracy for all 5 fuels

```

newRanges = np.zeros(114)
arrayRange = np.arange(0,32062,782)
for i in range(57):
    j = 2*i
    newRanges[j] =200+(982*i)
    newRanges[j+1] = newRanges[j]+782

xOne = np.zeros((31280,3))
yOne = np.zeros((31280)) # this should be the labels for the fires (0 to 4)

for i in range(len(want)):
    ins1 = want[i]
    c1 = newRanges[ins1*2]
    d1 = newRanges[(2*ins1)+1]
    group1 = scaled[int(c1):int(d1),:]
    xOne[arrayRange[i]:arrayRange[i+1],:] = group1
    yOne[arrayRange[i]:arrayRange[i+1]]= fireTitleNumeric[ins1]

xOneTest = np.zeros((12512,3))
yOneTest = np.zeros((12512)) # this should be the labels for the fires (0 to 4)

```

```

for i in range(len(wantTest)):
    ins1T = wantTest[i]
    c1T = newRanges[ins1T*2]
    d1T = newRanges[(2*ins1T)+1]
    group1T = scaled[int(c1T):int(d1T),:]
    xOneTest[arrayRange[i]:arrayRange[i+1],:] = group1T
    yOneTest[arrayRange[i]:arrayRange[i+1]]= fireTitleNumeric[ins1T]

linear_one_min = svm.SVC(kernel='linear',C=1,decision_function_shape='ovo').fit(
    xOne,yOne)

predTrainOne = linear_one_min.predict(xOne)
print('Training Accuracy', accuracy_score(yOne,predTrainOne)*100)

predTestOne = linear_one_min.predict(xOneTest)
print('Testing Accuracy', accuracy_score(yOneTest,predTestOne)*100)

resultsOne = confusion_matrix(yOneTest,predTestOne)
percentOne = np.zeros((resultsOne.shape[0],resultsOne.shape[0]))
for i in range(resultsOne.shape[0]):
    for j in range(resultsOne.shape[0]):
        percentOne[i,j]= resultsOne[i,j]/sum(resultsOne[i,:])
print(percentOne*100)

```

Start training at two minutes

```

# starts at 400 for this portion
newRanges2 = np.zeros(114)
arrayRange2 = np.arange(0,23862,582)
for i in range(57):
    j = 2*i
    newRanges2[j] =400+(982*i)
    newRanges2[j+1] = newRanges2[j]+582

xTwo = np.zeros((23280,3))
yTwo = np.zeros((23280)) # this should be the labels for the fires (0 to 4)

```



```

for i in range(len(want)):
    ins2 = want[i]
    c2 = newRanges2[ins2*2]
    d2 = newRanges2[(2*ins2)+1]
    group2 = scaled[int(c2):int(d2),:]
    xTwo[arrayRange2[i]:arrayRange2[i+1],:] = group2
    yTwo[arrayRange2[i]:arrayRange2[i+1]]= fireTitleNumeric[ins2]

xTwoTest = np.zeros((9312,3))
yTwoTest = np.zeros((9312)) # this should be the labels for the fires (0 to 4)

for i in range(len(wantTest)):
    ins2T = wantTest[i]
    c2T = newRanges2[ins2T*2]
    d2T = newRanges2[(2*ins2T)+1]
    group2T = scaled[int(c2T):int(d2T),:]
    xTwoTest[arrayRange2[i]:arrayRange2[i+1],:] = group2T
    yTwoTest[arrayRange2[i]:arrayRange2[i+1]]= fireTitleNumeric[ins2T]

linear_two_min = svm.SVC(kernel='linear',C=1,decision_function_shape='ovo').fit(
    xTwo,yTwo)

predTrainTwo = linear_two_min.predict(xTwo)
print('Training Accuracy', accuracy_score(yTwo,predTrainTwo)*100)

predTestTwo = linear_two_min.predict(xTwoTest)
print('Testing Accuracy', accuracy_score(yTwoTest,predTestTwo)*100)

```

Appendix C

Graphs

This section shows the pair plots (Seaborn implementation) that were used to determine what two products of combustion had the strongest relationship and should be used for fuel identification. The diagonal shows the distribution of a single variable and the scatter plots show the potential relationship between variables. The four products of combustion are plotted on the x and y axes, for a combination of 16 plots.

The axes bounds are fixed across each pair plot in time.



Figure C.1: Pairplot at 30 seconds.

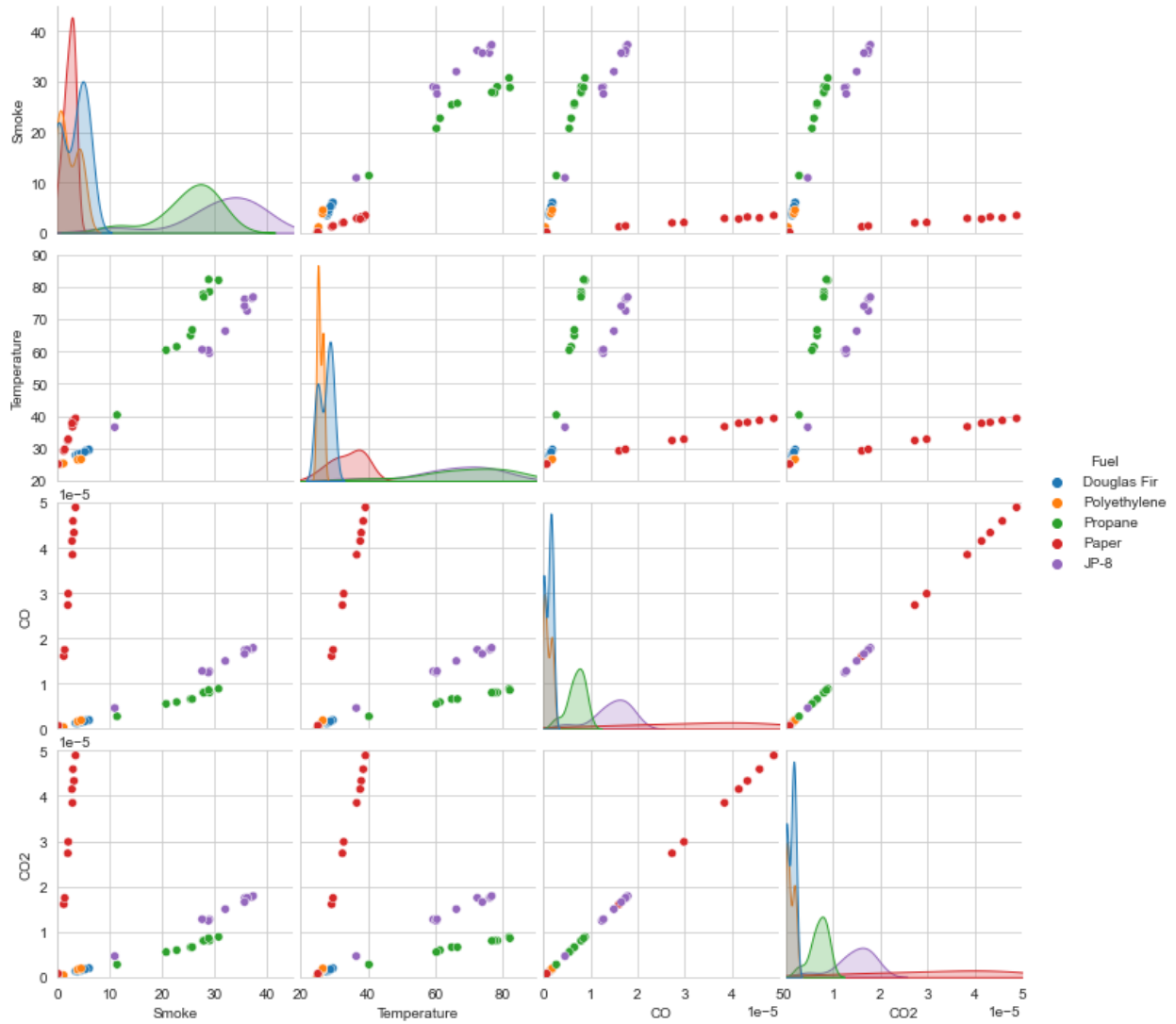


Figure C.2: Pairplot at 150 seconds.

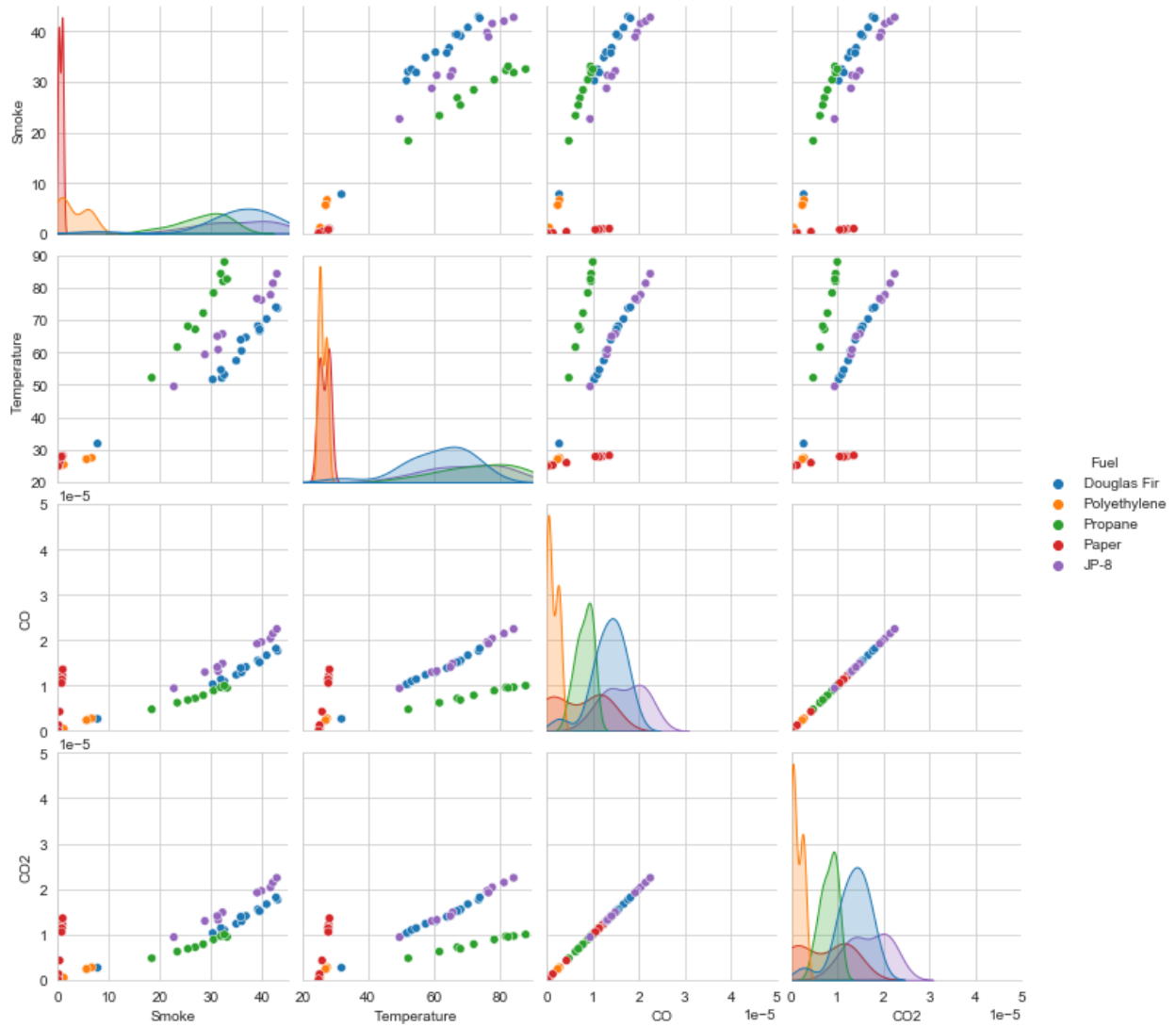


Figure C.3: Pairplot at 300 seconds.

References

- [1] Dugan, M. M. (April 11, 2003). Size-up. <https://www.firehouse.com/photo-story/article/10559778/sizeup>
- [2] Behnam, D., & Lincoln, R. (June 1, 2021). Size-up critical to safety. <https://www.fireengineering.com/leadership/size-up-critical-to-safety/>
- [3] Dow, M., Garis, L., & Thomas, L. (2013). *Reframing situational awareness within the fire service culture* (tech. rep.). University of the Fraser Valley, Centre for Social Research.
- [4] Frieze, G. (February 15, 2017). Importance of situational awareness process for firefighters. <https://www.firerescue1.com/situational-awareness/articles/importance-of-situational-awareness-process-for-firefighters-HWUS4JoADKGBqxRx/>
- [5] *What is a smart building?* (April 5, 2011). <https://buildingefficiencyinitiative.org/articles/what-smart-building>
- [6] McGrattan, K., Hostikka, S., Floyd, J., McDermott, R., & Vanella, M. (2021a). *Fire dynamics simulator user's guide* (tech. rep.). https://github.com/firemodels/fds/releases/download/FDS6.7.6/FDS_User_Guide.pdf
- [7] McGrattan, K., Hostikka, S., Floyd, J., McDermott, R., & Vanella, M. (2021b). *Fire dynamics simulator technical reference guide volume 3: Validation* (tech. rep.). https://github.com/firemodels/fds/releases/download/FDS6.7.6/FDS_Validation_Guide.pdf
- [8] Iriondo, R. (October 15, 2018). Machine learning (ml) vs. artificial intelligence (ai) - crucial differences. <https://pub.towardsai.net/differences-between-ai-and-machine-learning-and-why-it-matters-1255b182fc6>
- [9] Neviackas, A. (2007). *Inverse fire modeling to estimate the heat release rate of compartment fires* (Master's thesis). University of Maryland, College Park.

- [10] Overholt, K. J., & Ezekoye, O. A. (2012). Characterizing heat release rates using an inverse fire modeling technique. *Fire Technology*, 48, 893–909.
- [11] Okayama, Y. (1991). Approach to detection of fires in their very early stage by odor sensors and neural net. *Proceedings of the 3rd International Symposium of Fire Safety Science*, 955–964.
- [12] Ishii, H., Ono, T., Yamauchi, Y., & Ohtani, S. (1993). Fire detection system by multi-layered neural network with delay circuit. *Proceedings of the 4th International Symposium of Fire Safety Science*, 761–772.
- [13] Kaiser, M. (1994). Time-delay neural networks for control [Fourth IFAC Symposium on Robot Control, Capri, Italy, September 19-21, 1994]. *IFAC Proceedings Volumes*, 27(14), 967–972. [https://doi.org/https://doi.org/10.1016/S1474-6670\(17\)47423-4](https://doi.org/https://doi.org/10.1016/S1474-6670(17)47423-4)
- [14] Psychogios, D. C., & Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10), 1499–1511.
- [15] Chen, Y., Serio, M., & Sathyamoorthy, S. (2000). Development of a fire detection system using ft-ir spectroscopy and artificial neural networks. *Fire Safety Science*, 6, 791–802. <https://doi.org/10.3801/IAFSS.FSS.6-791>
- [16] McAvoy, T. J., Milke, J., & Kunt, T. A. (1996). Using multivariate statistical methods to detect fires. *Fire Technology*, 32, 6–24. <https://doi.org/https://doi.org/10.1007/BF01040755>
- [17] *What is iot?* (n.d.). <https://www.aeris.com/in/what-is-iot/>
- [18] Namuduri, S., Narayanan, B. N., Davuluru, V. S. P., Burton, L., & Bhansali, S. (2020). Review - deep learning methods for sensor based predictive maintenance and future perspectives for electrochemical sensors. *Journal of The Electrochemical Society*, 167.
- [19] Saeed, F., Paul, A., Karthigaikumar, P., & Nayyar, A. (2020). Convolutional neural network based early fire detection. *Multimedia Tools and Application*, 79, 9083–9099. <https://doi.org/https://doi.org/10.1007/s11042-019-07785-w>
- [20] Desarda, A. (January 17, 2019). Understanding adaboost. <https://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe>
- [21] *What is a perceptron?* (n.d.). <https://deepai.org/machine-learning-glossary-and-terms/perceptron>
- [22] Garcia-Ordas, M. T., Benitez-Andrades, J. A., Garcia-Rodriguez, I., Benavides, C., & Alaiz-Moreton, H. (2020). Detecting respiratory pathologies using convolutional neural networks and variational autoencoders for unbalancing data.

- [23] Ryder, N., Geiman, J., & Weckman, E. (2021). Hierarchical temporal memory continuous learning algorithms for fire state determination. *Fire Technology*, 57. <https://doi.org/10.1007/s10694-020-01055-0>
- [24] Brownlee, J. (January 11, 2019). How to fix the vanishing gradients problem using the relu. <https://machinelearningmastery.com/how-to-fix-vanishing-gradients-using-the-rectified-linear-activation-function/>
- [25] Sarkar, K. (2018). *Relu: Not a differentiable function: Why used in gradient based optimization? and other generalizations of relu*. <https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec>
- [26] Baek, J., Alhindi, T. J., Jeong, M. K., Jeong, Y.-S., Seo, S., Kang, J., Choi, J., & Chung, H. (2021). Real-time fire detection algorithm based on support vector machine with dynamic time warping kernel function. *Fire Technology*, 57, 2929–2953.
- [27] Tam, W. C., Fu, E., Peacock, R., Reneke, P., Wang, J., Li, J., & Cleary, T. (2020). Generating synthetic sensor data to facilitate machine learning paradigm for prediction of building fire hazard. *Fire Technology*. <https://doi.org/10.1007/s10694-020-01022-9>
- [28] Gott, J. E., Lowe, D. L., Notarianni, K. A., & Davis, W. (1997). *Analysis of high bay hangar facilities for fire detector sensitivity and placement* (tech. rep.). Building, Fire Research Laboratory, National Institute of Standards, and Technology.
- [29] Sharma, P., & Quintiere, J. (2010). Compartment fire temperatures. *Journal of Fire Protection Engineering - J FIRE PROT ENG*, 20, 253–271. <https://doi.org/10.1177/1042391510380139>
- [30] McGrattan, K., Hostikka, S., Floyd, J., McDermott, R., & Vanella, M. (2021c). *Fire dynamics simulator technical reference guide volume 1: Mathematical model* (tech. rep.). https://github.com/firemodels/fds/releases/download/FDS6.7.7/FDS_Technical_Reference_Guide.pdf
- [31] Milke, J., Behera, S., Lee, K., & Slingluff, C. (2019). *Review of foam fire suppression system discharges in aircraft hangars*. University of Maryland, College Park.
- [32] Milke, J., & Poole, J. (2022). *Performance criteria for aircraft hangar fire protection systems*. Fire Protection Research Foundation.
- [33] Hertzberg, T., Sundstrom, B., & van Hees, P. (2003). Design fires for enclosures: A first attempt to create design fires based on euroclasses for linings. *SP Fire Technology*.
- [34] *Heat release rate and fire characteristics of fuels representative of typical transient fire events in nuclear power plants* (No. 2232). (March, 2020). U.S. Nuclear Regulatory Commission.

- [35] Hurley, M. J., Gottuk, D., Jr, J. R. H., Harada, K., Kuligowski, E., Puchovsky, M., Torero, J., Jr., J. M. W., & Wieczorek, C. (Eds.). (2016). *Sfpe handbook of fire protection engineering* (5th). Springer.
- [36] Klote, J. H., Milke, J. A., Turnbull, P. G., Kashef, A., & Ferreira, M. J. (2012). *Handbook of smoke control engineering*. American Society of Heating, Refrigerating; Air-Conditioning Engineers.
- [37] Blevins, L. G. (August 15, 1999). *Behavior of bare and aspirated thermocouples in compartment fires*. National Heat Transfer Conference, 33rd. Proceedings. HTD99-280, Albuquerque, NM. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=916775
- [38] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., . . . Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [39] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.