UNIVERSITÉ LIBRE DE BRUXELLES

Bruface
Brussels Faculty of Engineering

VRIJE UNIVERSITEIT BRUSSEL

Machine Learning for Combustion Data Analysis

Abdennacer ZARGUIT

Master thesis submitted under the supervision of
Prof. dr. ir. Alessandro PARENTE

The co-supervision of
Gianmarco AVERSANO

In order to be awarded the Master's Degree in
Electromechanical Engineering
Option Vehicle Technology and Transport

Academic year
2018–2019

Acknowledgment

I would like to thank my supervisor Professor Alessandro PARENTE for having given me the opportunity to do this master thesis, for his experience and knowledge in the field of combustion systems, for his quality teaching among other professors of the different master and bachelor courses that greatly contributed to my scientific background.

Next, I would like to express my sincere gratitude to my co-supervisor Gianmarco AVERSANO for the continuous support of my Master Thesis, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I would also like to express my sincere thanks to the members of the jury for agreeing to read and evaluate my work.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Machine Learning for Combustion Data Analysis

Abdennacer ZARGUIT
Master's Degree in Electromechanical Engineering
Option Vehicle Technology and Transport
2018–2019

Abstract

The aim of the present thesis is to analyze multidimensional combustion datasets from premixed and non-premixed laminar flame simulations of a series of well documented piloted flames with inhomogeneous inlets. The datasets are studied using different algorithms of Machine Learning (ML). Those algorithms are presented through state of art which reviews several scientific methods including clustering analysis and dimensionality reduction. The main part of this work consists to reduce the dimensionality of the combustion data with no prior knowledge using Principal Component Analysis (PCA) and Autoencoders. Unsupervised clustering algorithms are then utilized to classify features into relative groups based on the structural similarities of the original data. Results using PCA show that the first 10 principal components contain 90% of the total variance in the dataset with Auto scaling method and only 5 principal components are needed to have 90% of the total variance in the dataset using Range scaling method. Using Local PCA and K-Means clustering algorithms, the dataset is clustered into 5 clusters and the characteristics of each cluster are given. Supervised classification algorithm called Random Forest is used in order to select the most relevant features within the dataset. Indeed, selecting only the most relevant features makes the model easier to interpret and extract the most important information from it.

Keywords: *ML, Dimensionality Reduction, PCA, Clustering, Supervised, Unsupervised.*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the increase in computational power, it has become commonplace that data resulting from combustion simulations occupy hundreds of gigabytes of storage and contain tens of hundreds of chemical species [23]. To explore and analyze this large, high dimensional data, conventional visualization techniques such as scatter plots, histograms and pair plots are limited. Since human visual perception cannot visualize data in more than three dimensions of space, calling for new data visualization techniques has therefore become a necessity [9].

In addition to visualization, finding dominating structures and patterns in the data, analyzing it from different perspectives and summarizing it into useful information has become the most challenging task in many research fields.

One of the most promising field of science to tackle data is the field of Machine Learning (ML) [21] which is a subset of Artificial Intelligence (AI) that focuses on teaching computers how to learn without the need to be programmed for specific tasks. The key idea behind ML is the possibility to create algorithms that learn from and infer unknowns from data. The most common methods of Machine Learning used to detect the main features of data with no prior knowledge are dimensionality reduction and clustering.

A popular dimensionality reduction method is Principal Component Analysis (PCA) [23], [4], [7]. PCA projects data to a lower dimensional space with orthogonal components representing maximum variance [17]. PCA has been employed in many fields including combustion modeling, computer security, music analysis, computer aided diagnosis, bioinformatics and electroencephalography [18, 9]. In this work, an application of dimensionality reduction is developed for large-scale data generated from simulation of combustion phenomena using PCA and Autoencoders (AE). Developing unsupervised clustering model to group this dataset into clusters using Local Principal Component Analysis [22, 24] and K-Means [1, 2] algorithms is another objective of this study.

The present thesis has the following structure: The first part is an overview of Machine Learning, starting by presenting the foundations of Machine Learning and its types (supervised, unsupervised and reinforcement learning). This part also contains the literature review exposing previous studies made on this subject.

The next part presents the practical case study for this thesis. It is a case study of combustion data generated from premixed and non-premixed laminar flame simulations of a series of well documented piloted flames with inhomogeneous inlets [9].

Throughout this part, a step-by-step explanation of the methodology used in this work is presented and achieved, including the implementation of the model. Furthermore, a comparative

study of the different algorithms used for dimensionality reduction and clustering to build this model is carried out and presented. Then, the results of the model are given, and the interpretation of these results is discussed and explained.

The last part presents a supervised learning method called Random Forest used for Feature Selection [21, 15]. Indeed, since the considered dataset has high number of features it will not be possible to analyze them all. Identifying only the most relevant features makes the model easier to interpret and extract the most important information from it. Finally, conclusions are given at the end.

Chapter 2

Machine Learning Fundamentals

2.1 Introduction to Machine Learning

Machine Learning is a branch of Artificial Intelligence (AI) that allows computer systems to learn directly from examples, data and experience [13]. By allowing computers to intelligently perform specific tasks, Machine Learning systems can perform complex processes by learning from data rather than following pre-programmed rules [20]. Recent years have seen interesting progress in Machine Learning, which has allowed it to increase its capabilities across a suite of applications. While the increase in data availability has allowed Machine Learning to be used in many applications, computer processing power has also supported the analytical capabilities of this field. Advances in algorithms have also been made in the field itself, which has enabled Machine Learning to gain momentum. As a result of these advances, systems that operated a few years ago at levels significantly below those of humans can now outperform human performance in specific tasks.

While Artificial Intelligence builds intelligent machines that think and act like human beings, Machine Learning algorithms automatically build a mathematical model using sample data also known as "training data" to make decisions without being specifically programmed to make those decisions. The term Machine Learning was coined by Arthur Samuel in 1959, a researcher in the field of computer gaming and Artificial Intelligence and stated that "Machine Learning gives computers the ability to learn without being explicitly programmed".

Another interesting branch of Artificial Intelligence is known as Deep Learning (DL). This specific field of Artificial Intelligence can handle significant amount of data using artificial neural networks. The distinction between Artificial Intelligence, Machine Learning and Deep Learning is clearly illustrated in Figure 2.1. As one can see from this Figure, Deep Learning is subset of Machine Learning which is also a subset of Artificial Intelligence [13].

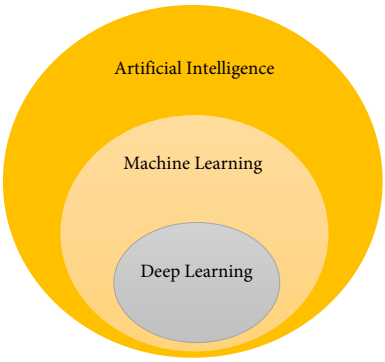


Figure 2.1 | Relationship Between: AI, ML and DL [13].

As it can be seen from Figure 2.2, Machine Learning is divided into three main categories [21]:

- Supervised learning: In supervised learning, a system is trained with data that has been labelled. The system learns how this data is structured and uses this to predict the categories of new data.
- Unsupervised learning: Unsupervised learning is learning without labels. It aims to detect the characteristics that make data points more or less similar to each other. The system is able to recognize patterns, similarities and anomalies, taking into consideration only the input data.
- Reinforcement learning: Reinforcement learning focuses on learning from experience and lies between unsupervised and supervised learning. Decisions are made by the system based on reward/punishment received for the last action performed.

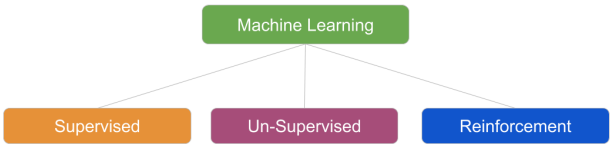


Figure 2.2 | Types of Machine Learning [21].

2.2 Applications of Machine Learning

Many people interact daily with systems based on Machine Learning algorithms, for example in image recognition systems, such as those used on social media; voice recognition systems, used by virtual personal assistants; and recommender systems, such as those used by online retailers. In healthcare, Machine Learning is creating systems that can help doctors give more accurate or effective diagnoses for certain conditions. In transport, it is supporting the development of autonomous vehicles, and helping to make existing transport networks more efficient. Furthermore, Machine Learning is helping to make sense of the vast amount of data available to researchers today, offering new insights into biology, physics, medicine, the social sciences, and more [20]. Examples of Machine Learning tasks in real life situations include [16]:

- *Identifying the zip code from handwritten digits on an envelope*

  Image processing algorithms are used to read destination addresses from letters (envelope). This automatic system is currently used by various postal companies. The input is a scan of the handwriting, then, classifiers are used to recognize handprinted digits and give the desired output which is the actual digits in the zip code.

- *Detecting fraudulent activity in credit card transactions*

  It is important that credit card companies should be able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. This

can be achieved with help of Machine Learning models specifically with the help of Deep Learning frameworks.

- *Segmenting customers into groups with similar preferences*

  This is often used in Market Segmentation, given a set of customer records, the main goal is to identify the common characteristics that customers share and whether there are groups of customers with similar preferences.

2.3 Supervised Learning

Supervised learning is one of the most commonly used and successful type of Machine Learning. In this section, supervised learning will be described in more detail and several popular supervised learning categories will be explained [16].

Supervised learning is the task of inferring a function f from labelled training data. The training data has already enough details and labels which allow the algorithm to use positions/distances of data points to infer a relationship between multiple variables. In supervised learning, each example is a pair consisting of an input object X (also called the feature) and a desired output value Y (also called the target) [21]. They can be expressed by the following relationship:

$$Y = f(X) \tag{2.1}$$

The goal is to approximate the mapping function f so that the output variable Y can be predicted when new input data X are available. Supervised Learning can be divided into 2 categories i.e Classification and Regression.

2.3.1 Principle

In order to solve a given problem of supervised learning, one has to perform the following steps (also shown in Figure 2.3) [21]:

- Data collection: This is the essential first step because the quality and quantity of the gathered data will directly determine how good the predictive model can be.
- Data preparation: Once the data is collected, it's time to assess the condition of it, including looking for trends, correlation, outliers, exceptions, incorrect, inconsistent, or missing information. Data preparation also includes feature scaling and centering.
- Choosing a model: The next step is choosing a model. There are tradeoffs between several characteristics of algorithms, such as speed of training, memory processing occupation and accuracy of the algorithm. Some of these algorithms are very well suited for image data, others for sequences, some for numerical data, others for text-based data.
- Training: The training is done by using the selected model. The model is trained in order to understand patterns and learn from the data.
- Evaluation: Evaluation allows to test the model against data that has never been used for training.

- Parameter Tuning: Once the evaluation has been done, it's possible to see if the training can further be improved. this can be achieved by feature selection and determining the feature importance.
- Prediction: The final step is prediction; it is the use of the model to predict the outcomes for new data points.

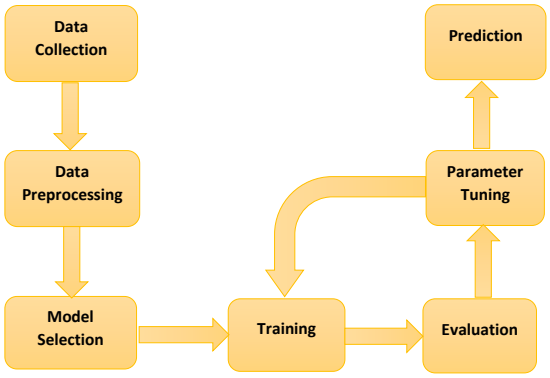


Figure 2.3 | Supervised Learning process workflow [21].

2.3.2 Regression

Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent predictor variable X and an independent response variable Y. One of the most common type of regression is linear regression, linear regression consists of finding the best-fitting straight line through the points. The best-fitting line is called a regression line [21, 16].

An example of linear regression (shown in Figure 2.4) is predicting house prices based on the features of the house like the size, number of rooms, etc. Predicting the target variable (price) is done by finding the best fit line so that the error between points is minimized.

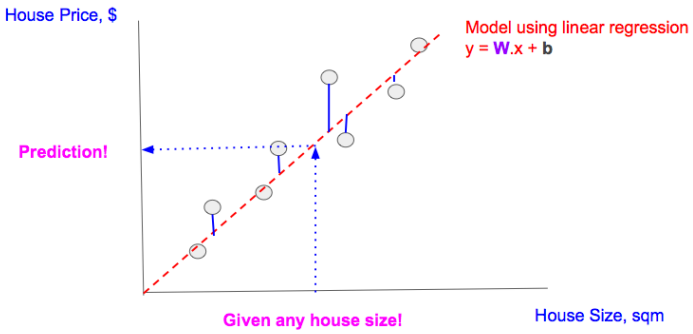


Figure 2.4 | Predicting house prices using Single-feature Linear Regression [1].

2.3.3 Classification

Like regression, classification also involves finding the link between a dependent variable X and an independent variable Y. The only difference is that regression deals with continuous values while classification is used for discrete datasets [21, 16].

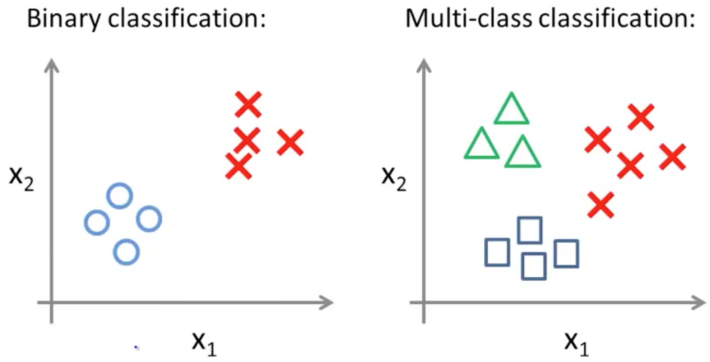


Figure 2.5 | Binary Classification and Multiclass Classification [2].

An example of classification is shown in Figure 2.5. When there are only two choices, it's called binary or binomial classification, when there are more categories, this problem is known as multi-class classification.

2.4 Unsupervised Learning

Unsupervised Learning is a class of Machine Learning techniques to find patterns in data. The data provided to unsupervised algorithms is neither classified nor labeled, which means the algorithm is acting on this information without guidance. The goal of this type of Machine Learning is to group data according to similarities, patterns and differences without any prior knowledge of data [21, 16].

Unsupervised Learning is classified into two categories:

- Dimensionality Reduction
- Clustering

These Unsupervised Learning categories will be inspected in further detail in the coming sections.

2.4.1 Dimensionality Reduction

Dimensionality Reduction is a powerful technique that is widely used in data analytics and data science to help in visualizing data [9]. The main objective of dimensionality reduction is to

[1]Figure from https://towardsdatascience.com.
[2]Figure from https://medium.com.

transform higher-dimensional data into data of smaller dimensions. Some of the most popular types of dimensionality reduction algorithms are Principal Components Analysis (PCA), and Autoencoders [21, 16].

Principal Component Analysis

Principal components analysis (PCA) is a data reduction algorithm that seeks to summarize data via a set of linear combinations of the original data (Pearson, 1901). It is used to transform high-dimensional datasets into a datasets with fewer variables, where the set of resulting variables explains the maximum variance within the datasets [17].

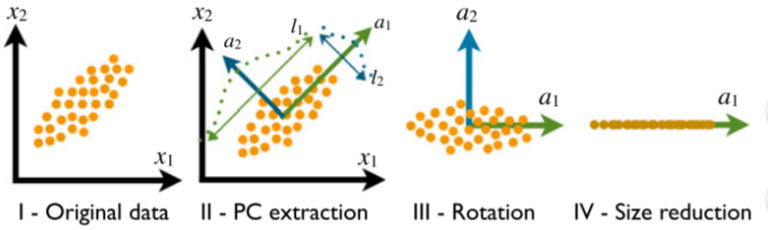


Figure 2.6 | PCA Process [17].

PCA is performed by finding the eigenvectors of the covariance matrix of dataset. These eigenvectors correspond to the directions of the principal components of the original data, their statistical significance is given by their corresponding eigenvalues. In more detail, this technique is shown in Figure 2.6 and can be structured in the following steps [6]:

- Step 1. Collect x_i of an n dimensional data set $X, i = 1, 2, \ldots, m$.
- Step 2. Calculate the mean $\bar{x}$ and substract it from each data point x_i - $\bar{x_i}$.
- Step 3. Calculate the covariance matrix C.

$$C_{ij} = (x_i - \bar{x})(x_i - \bar{x}) \tag{2.2}$$

- Step 4. Determine eigenvalues and eigenvectors of the matrix. C is a real symmetric matrix so a positive real number λ and a nonzero vector α can be found such that

$$C\alpha = \lambda\alpha \tag{2.3}$$

  where λ is called an eigenvalue and α is an eigenvector of C. To find a nonzero α the characteristic equation $|C - \lambda I| = 0$ must be solved. If C is a $n \times n$ matrix of full rank, n eigenvalues can be found $\lambda_1, \lambda_2, \ldots, \lambda_n$. Using $(C - \lambda I) = 0$ all the corresponding eigenvectors can be found.

- Step 5. Sort the eigenvalues and corresponding eigenvectors so that $\lambda_1 \geqslant \lambda_2 \geqslant, \ldots, \geqslant \lambda_n$.
- Step 6. Select the first $d \leqslant n$ eigenvectors and generate the data set in the new representation.

Autoencoders

Another interesting algorithm for data reduction is Autoencoders (AE). Autoencoders are neural networks that aims to copy their inputs to their outputs. They work by compressing the input into a latent-space representation, and then reconstructing the output from this representation. This kind of network is composed of two parts, namely encoder and decoder. This architecture is presented in Figure 2.7. While the encoder aims to compress the original input data into a low-dimensional representation, the decoder tries to reconstruct the original input data based on the low-dimension representation generated by the encoder. The Encoder can be represented by an encoding function $h = f(x)$, where x is the input, and the Decoder can be represented by a decoding function $y = g(h)$. Where y is the output and it wanted to be as close as the original input x [10].

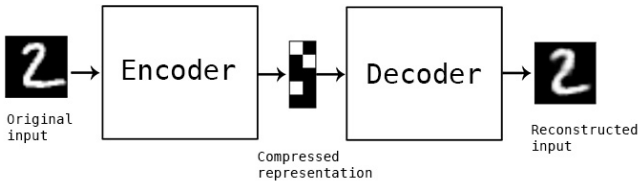


Figure 2.7 | Architecture of an Autoencoder [10].

Since neural networks are capable of learning nonlinear relationships, this can be considered a more powerful (nonlinear) generalization of PCA [12]. As a result, the Autoencoders have been widely used to remove the data noise as well as reducing the data dimension. The difference between these two approaches is illustrated in Figure 2.8.

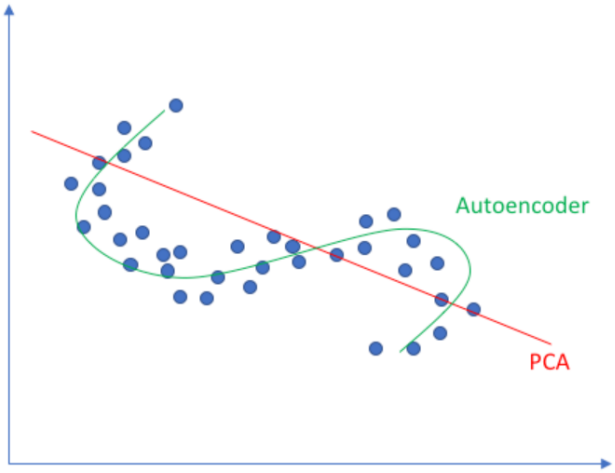


Figure 2.8 | Linear vs nonlinear dimensionality reduction [12].

2.4.2 Clustering

Clustering or cluster analysis is the process of dividing data into groups (clusters) in such a way that objects in the same cluster are more similar to each other than those in other clusters. It is basically a collection of objects based on similarity and dissimilarity between them [21, 16].

K-Means Clustering

K-Means (MacQueen, 1967) is one of the most simplest and popular unsupervised learning algorithms used to solve a clustering problems. The algorithm follows a simple procedure to classify the given data to a fixed clusters k. K-Means clustering aims to find the set of k clusters such that every data point is assigned to the closest center, and the sum of the distances of all such assignments is minimized. The description of K-Means algorithm is represented in Figure 2.9, and summarized in the following steps [1]:

- Step 1. Place randomly k points, equal to the number of clusters, into the space represented by the objects that are being clustered.
- Step 2. Group each dataset point to the group that has the closest centroid, this is done by calculating Euclidean distance.
- Step 3. When all points have been grouped, recalculate the positions of the k centroids.
- Step 4. Repeat steps 2 and 3 iteratively till the centroids no longer move.

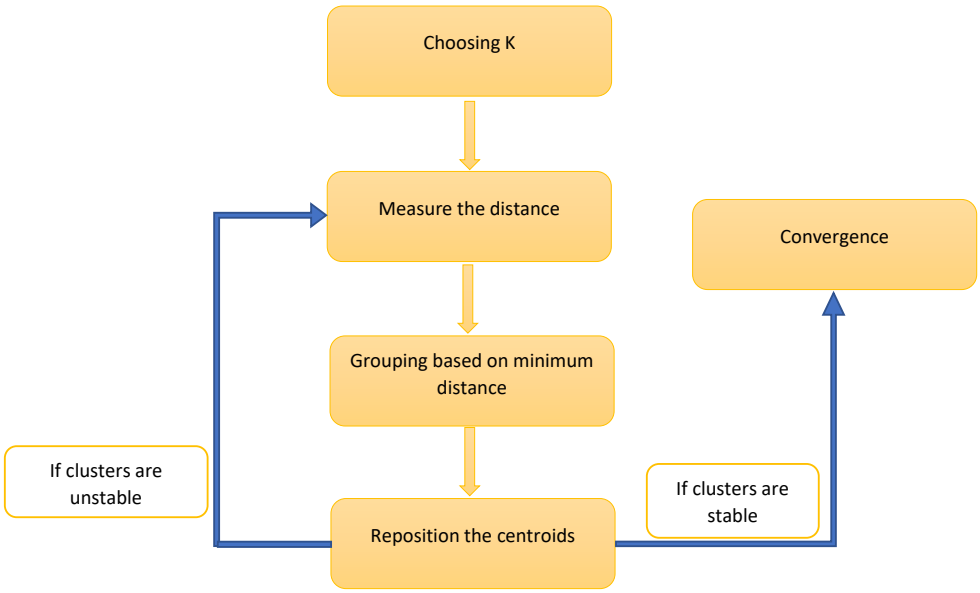


Figure 2.9 | Steps of the K-Means algorithm [1]

Local PCA

In practical applications, PCA method cannot directly deal with space objects with complex structures. That is, the inherent low-dimensional structure in high-dimensional data is complex, and only considering the global structure in PCA will ignore the local characteristics of low-dimensional structures. Therefore, several literatures introduced local PCA as clustering algorithm to keep the maximum information of data [22]. The algorithmic procedure of Local PCA is formally stated as follows (see Figure 2.10):

- Step 1. Place randomly k points, equal to the number of clusters, into the space represented by the objects that are being clustered.
- Step 2. Group each dataset point to the group that has the closest centroid.
- Step 3. Apply PCA on each group to find the eigenvectors.
- Step 4. Compute the orthogonal distance between dataset points and the eigenvectors of each cluster, then assign dataset points to the group that has the minimum orthogonal distance.
- Step 5. When all points have been grouped, recalculate the positions of the k centroids.
- Step 6. Repeat steps 2, 3, 4 and 5 iteratively till the centroids no longer move.

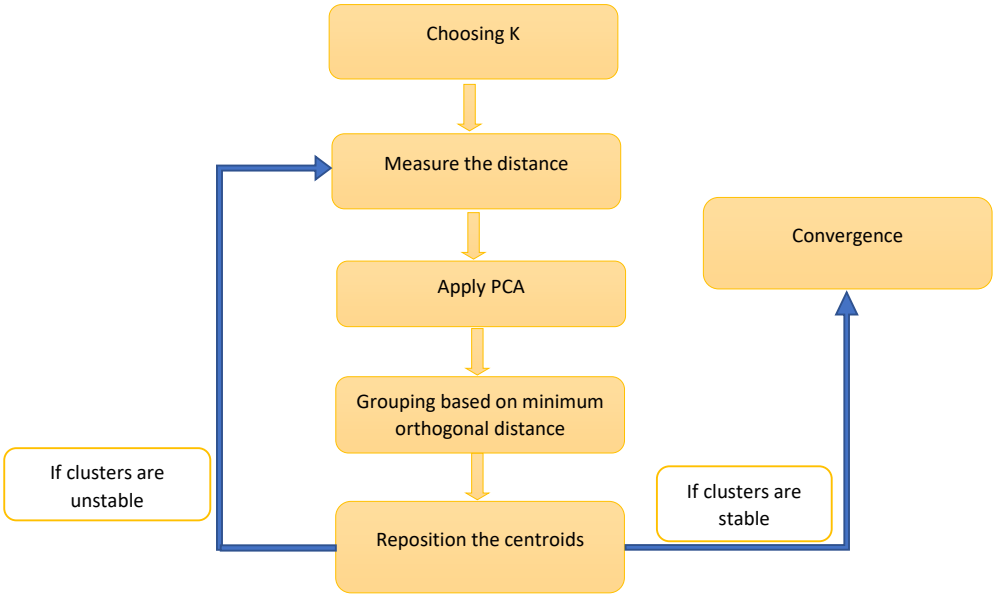


Figure 2.10 | Steps of Local PCA algorithm [22].

2.5 Feature Scaling

Feature scaling (also known as data normalization) is a technique used to standardize the range of features of data in a fixed range. As the range of data values can vary considerably, it becomes a necessary step in data preprocessing while using Machine Learning algorithms. Indeed, most of the time, dataset contains features that vary considerably in magnitudes, units and range. Since most Machine Learning algorithms use the Euclidean distance between two data points in their computations, the features in the dataset that have large scale relative to others become dominating and need to be normalized. Hence, selecting an appropriate scaling method is an essential step in the data analysis and greatly helps to weight all the features equally [17].

The table below summarizes the most common feature scaling methods [3]:

Table 2.1 | Different classes of data pre-treatment methods [3].

Method	Formula	Unit	Goal
Auto scaling	$z_{ij} = \frac{x_{ij} - \bar{x_i}}{s_i}$	(-)	Compare features based on correlations
Range scaling	$z_{ij} = \frac{x_{ij} - \bar{x_i}}{\sqrt{s_i}}$	(-)	Compare features relative to response range
Pareto scaling	$z_{ij} = \frac{x_{ij} - \bar{x_i}}{x_{i_{max}} - x_{i_{min}}}$	O	Reduce the relative importance of large values, but keep data structure partially intact
Vast scaling	$z_{ij} = \frac{x_{ij} - \bar{x_i}}{s_i} \cdot \frac{\bar{x_i}}{s_i}$	(-)	Focus on the features that show small fluctuations

Where $\bar{x}$ is the mean (average) and s_i is the standard deviation from the mean; given by the following equations:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad and \quad s_i = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \bar{x}_i)^2} \tag{2.4}$$

Chapter 3

Literature Review

In this chapter, the main studies found in the literature concerning data reduction and clustering analysis are given. The first study starts from performing data reduction on turbulent flame dataset. Then, clustering algorithm was used for further investigation. Next, study which uses PCA for image compression to explore the effects of reducing the number of principal components of the compressed image is presented.

3.1 Study Using Data Reduction and Clustering Analysis

This study introduces a novel post-processing technique for analyzing high dimensional combustion data. In this technique, t-Distributed Stochastic Neighbor Embedding (t-SNE) was used to reduce the dimensionality of the combustion data. Then, clustering analysis has been performed on this dataset with no prior knowledge.

The turbulent flame dataset used for this study was created using line-imaged measurements of temperature and major species performed for a piloted methane/air jet flame with inhomogeneous inlets [9]. The device used for this study is shown in Figure 3.1

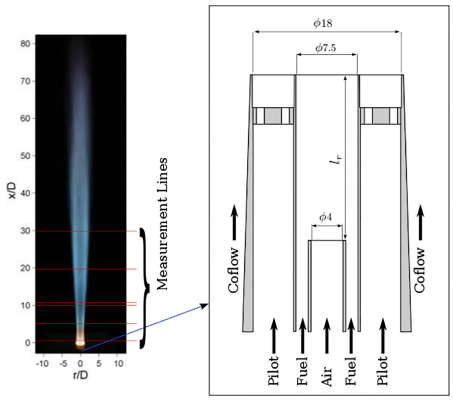


Figure 3.1 (Left:) Flame brush and measurement lines, $x/D = 1, 5, 10, 12, 20$ and 30. (right:) Schematic of the burner with units in millimeter [9].

3.1.1 Data Reduction of Turbulent Flame Dataset

First, t-SNE was applied on the turbulent flame dataset to obtain the two-dimensional manifold. Then, the resulting t-SNE map of the experimental dataset was plotted in Figure 3.2.

The t-SNE classifies the data into 11 distinguishable clusters/regions representing samples with strong similarity. The internal structure of each region is analyzed with the help of clustering algorithm.

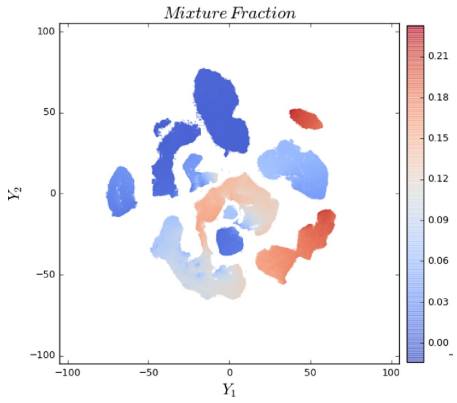


Figure 3.2 | t-SNE map of the experimental data colored with different mixture fractions [9].

3.1.2 Clustering Analysis of Turbulent Flame Dataset

To analyze the internal structure of dataset mentioned above clustering analysis was used [9]. First, the points forming each cluster were extracted. Then, the key variable(s) (feature(s)) most correlated with t-SNE components of these points were identified. Point extraction here is done by using Mean Shift clustering algorithm of Scikit-Learn.

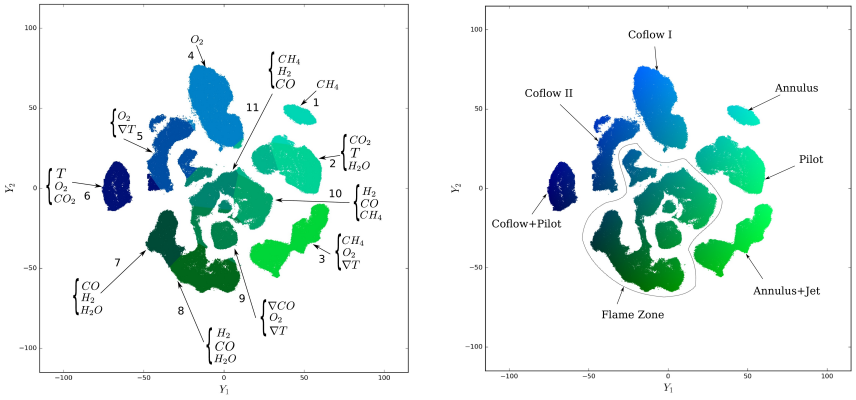
Main Results

The results of the clustering analysis are shown in Figure 3.3. Figure 3.3a shows the most correlated features in each cluster and Figure 3.3b assigns these clusters to the physical space of the device presented in Figure 3.1. This clustering algorithm facilitates post-processing of data and reveals characteristics of each cluster which are not easily recognizable by using only t-SNE map. These characteristics are summarized below.

- Cluster 2 in Figure 3.3a can be marked as pilot stream (in Figure 3.3b) since most of the points in this cluster have high value of temperature and major products but very small amount of fuel and oxygen.

- The regions marked as “coflow I” and “coflow II” in Figure 3.3b can be attributed to the samples with high O_2 and almost zero concentration of fuel due to early mixing with pilot stream. These zones are assigned to cluster 4 and 5 in Figure 3.3a.

- The reaction zone (marked as flame zone in Figure 3.3b) includes samples distributed in several clusters (clusters marked with 9, 10, 8 and 7 in Figure 3.3a) with medium to high value of temperature and CO, and reasonable amount of reactants and major products distributing mainly at moderate radii.



(a) t-SNE map of the experimental data colored with quantized t-SNE colormap. Points in each quantized cluster have meaningful correlation with variable(s) shown in the bracket.

(b) t-SNE map of the experimental data colored with t-SNE colormap. The clusters are projected into the physical space.

Figure 3.3 | Results of clustering analysis [9].

3.2 Study Using PCA for Image Compression

This article [8] has the purpose of describing PCA and applying it on a digital image collected in the clinical routine of a hospital. The study evaluates the PCA performance on digital image from the measurement of the degree of compression and the degree of information loss that the PCA introduces into the compressed image after reducing the number of the principal components. The image is shown in Figure 3.4, it is TIFF (Tagged Image File Format) image of human brain with 512x512 pixels and 262144 units of memory storage.

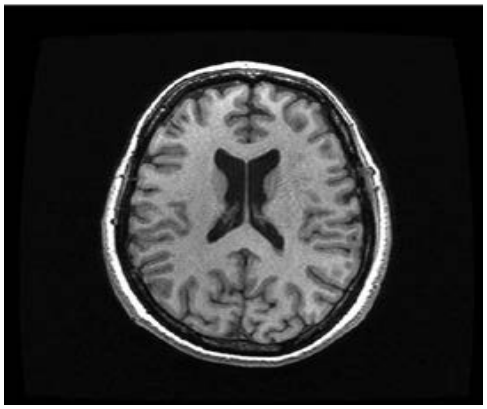


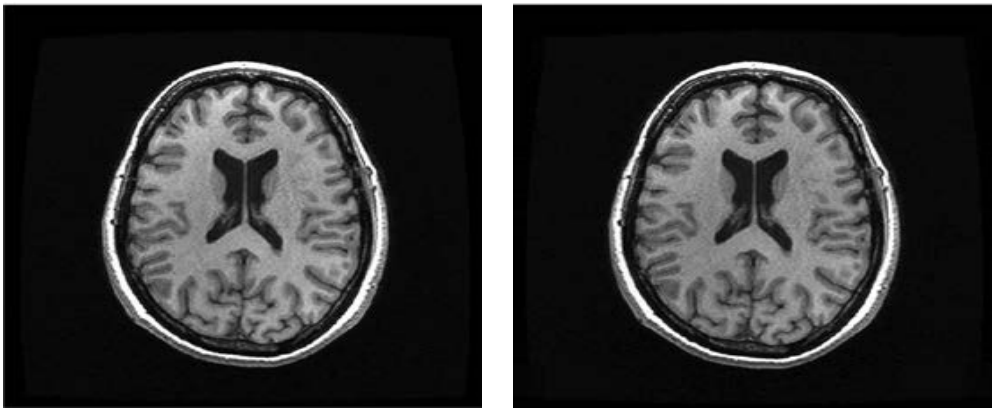
Figure 3.4 | Original image: TIFF with 512x512 pixels [8].

Main Results

The results show examples of image compression using the PCA formulation. The mean squared error (MSE) and the compression factor ρ defined by:

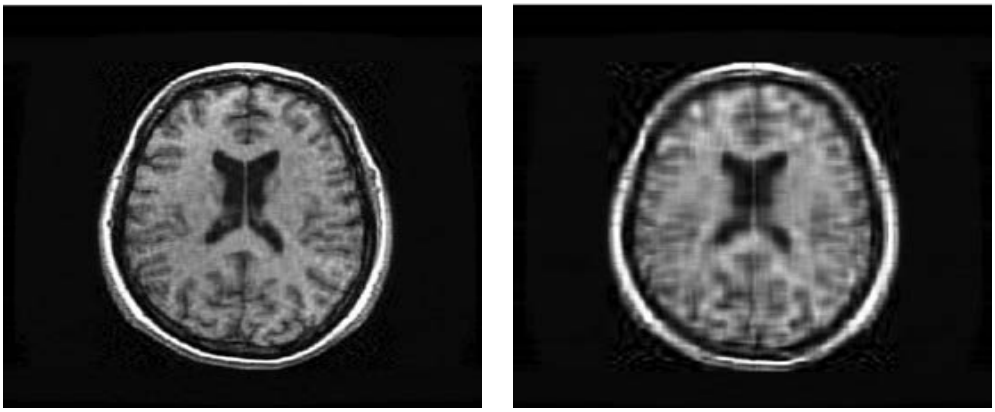
$$\rho = \frac{\text{Unit of memory required to represent the compressed image}}{\text{Unit of memory required to represent the original image}} \tag{3.1}$$

are used to compare and evaluate these results. 4 situations are presented in Figure 3.5.



(a) Recovered image (512x512 pixels) from 512 PCs. Memory necessary = 512x512 = 262144 units of memory. Compression factor $(\rho) = 1$. $MSE = 0$

(b) Recovered image (112x512 pixels) from 112 PCs. Memory necessary = 112x512 = 57344 units of memory. Compression factor $(\rho) = 0.219$. $MSE = 0.213$



(c) Recovered image (32x512 pixels) from 32 PCs. Memory necessary = 32x512 = 16384 units of memory. Compression factor $(\rho) = 0.0625$. $MSE = 0.9375$

(d) Recovered image (12x512 pixels) from 12 PCs. Memory necessary = 12x512 = 6144 units of memory. Compression factor $(\rho) = 0.0234$. $MSE = 0.95050$

Figure 3.5 Recovery of a TIFF image with 512x512 pixels with different number of principal components [8].

- Figure 3.5a shows the recovered image after applying PCA by keeping all the principal

components (512 PCs). The quality of image is the same as the original image and with no loss.

- Figure 3.5b shows the recovered image after applying PCA and keeping 112 components. The image has very good quality and it is very clear. The memory necessary for storage is largely reduced and the error is relatively small.
- When keeping only 32 components (see Figure 3.5a), the digital noise starts to creep into the image. However, the picture's subject is easily discernible.
- Finally, in Figure 3.5d, the compressed matrix distorts the image since it lacks many significant singular values. In this image, one can still discern what the image's subject is, but it's unclear if every picture will have a visible structure such as the one above.

Chapter 4

Introduction to the Case Study

4.1 Motivation

The combustion datasets [9] used for this study were generated from the results of steady-state, quasi-one- dimensional simulations of freely-propagating premixed and counterflow diffusion laminar flames. Simulations were performed by Cantera code [11] for methane-air mixtures using GRI-3.0 mechanism. These datasets are briefly described below.

Freely-propagating premixed flame (FPF) dataset contains 6500 samples extracted from the calculations of 13 freely-propagating premixed flames with equivalence ratio ranging from 0.5 to 1.7 and inlet temperature, Tin of 300 K. Each sample consists of temperature, speed of the flame stream u(m/s), and mole fractions of the 53 species included in GRI-30, leading to a dataset with 55 features.

Freely-propagating premixed flame with varying inlet temperature (TFPF) dataset is the FPF dataset with the addition of three new FPF datasets generated at different inlet temperatures (see Table 4.1) using the same preprocessing method described above. This results in a dataset with 26,000 samples and 55 dimensions.

Table 4.1 | Inlet temperature Tin (K) of TFPF dataset [9].

Tin (K)
300
500
700
900

The majority of recent academic studies on these datasets use Machine Learning techniques such as Dimensionality Reduction and Clustering to analyze and process them. The objective of this thesis is therefore to apply these advanced techniques to be used as a computer aided tool for visualization and internal structure analysis of these multidimensional combustion datasets. Hence the reason why it was selected as a case study for this thesis.

4.2 Methodology

The study of TFPF dataset using Machine Learning tools involves several important steps that are represented in the flowchart of Figure 4.1.

The first step is data normalization performed by using different methods of feature scaling including Auto scaling, Range scaling, Vast scaling, and Pareto scaling. The next step is the projection of the data into 2 and 3 dimensions using data reduction techniques (PCA and

Autoencoders) for visualization purposes. Next, clustering analysis was performed using Local PCA and K-Means algorithms. The main goal is to group the data into several groups. After the elaboration of clustering, Random Forest algorithm was used to select the most important features that hugely impact the performance of the clustering model. Finally, extraction the most relevant information from each cluster is done in the final step.

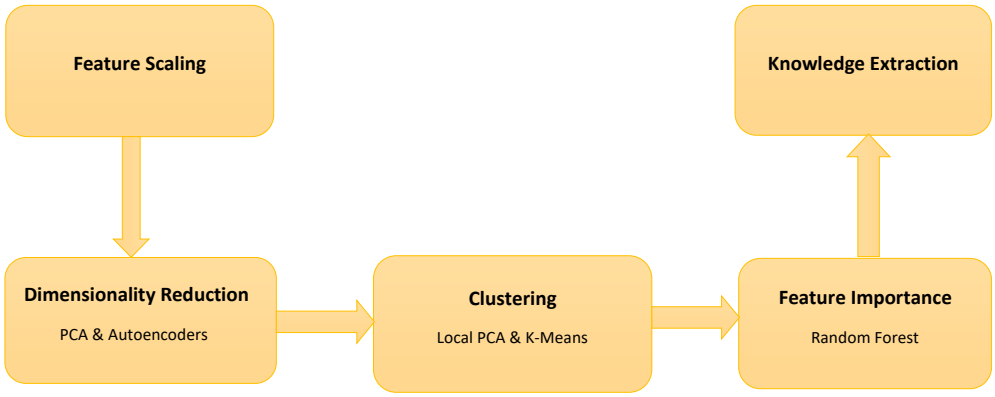


Figure 4.1 | Flow diagram showing the different steps required to analyze TFPF dataset.

Chapter 5

Dimensionality Reduction Analysis

The most common motivations of dimensionality reduction are visualization, compressing the data, and finding a representation that is more informative for further processing. In this section, 2 dimension reduction techniques (PCA and Autoencoders) will be investigated and compared. Furthermore, the effect of scaling techniques is assessed and discussed in detail. The 4 methods introduced in Chapter 2 will be applied on dataset before performing data reduction transformation to investigate how different scaling methods affect the shape of the manifolds and data visualization in the reduced sub-space.

5.1 PCA Analysis

The first technique of data reduction to be studied is Principal Component Analysis. PCA is a fast and flexible method for dimensionality reduction. Its behavior is easiest to visualize by looking at a two-dimensional or three-dimensional dataset.

5.1.1 Choosing a Subset of Principal Components

The typical goal of a PCA is to reduce the dimensionality of the original feature space by projecting it onto a smaller subspace, where the eigenvectors will form the axes. However, the eigenvectors only define the directions of the new axis, since they have all the same unit length[17].

In order to decide which eigenvectors can be dropped without losing too much information for the construction of lower-dimensional subspace, the corresponding eigenvalue need to be inspected. The eigenvectors with the lowest eigenvalues bear the least information about the distribution of the data; those are the ones can be dropped.

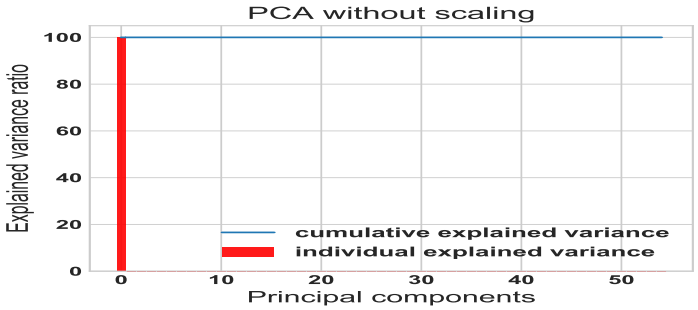
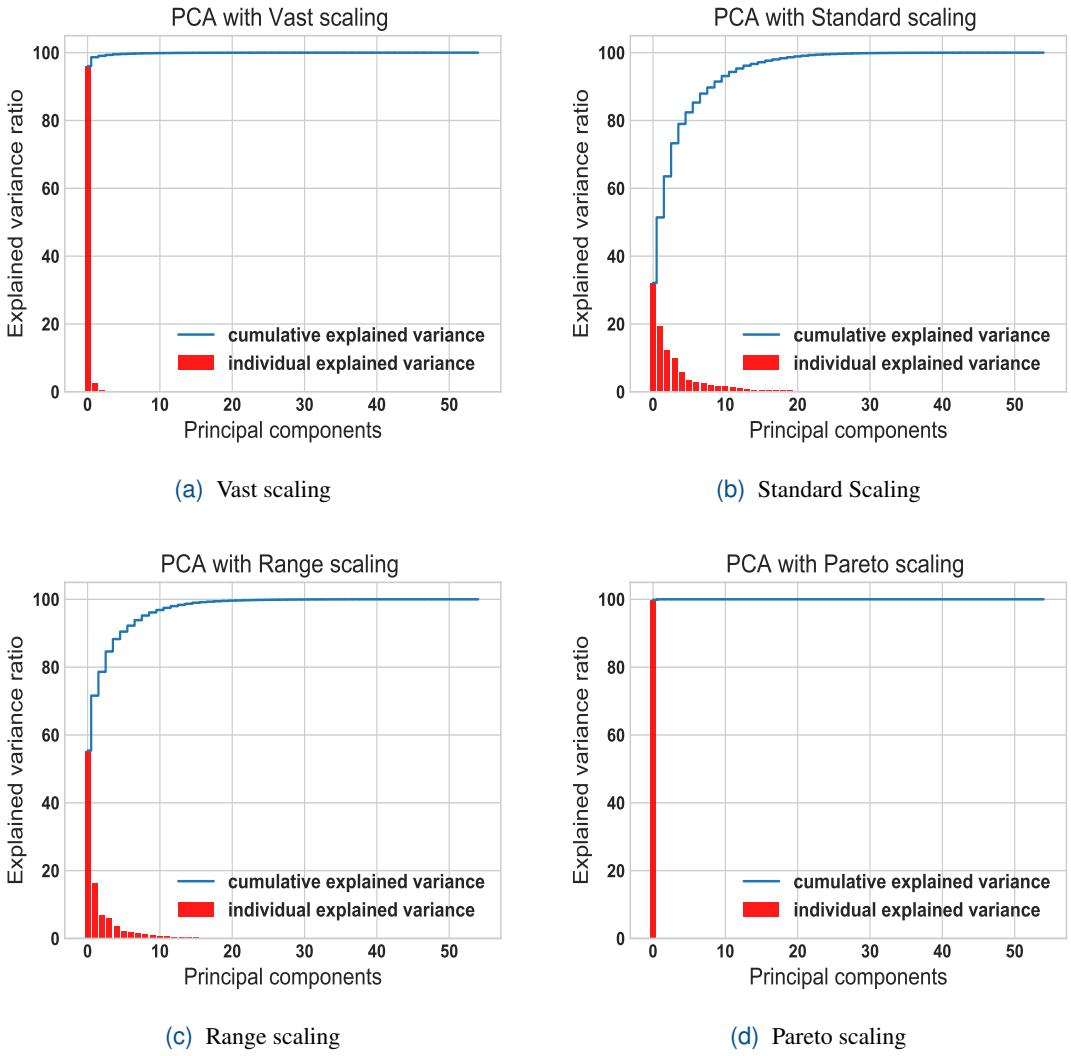


Figure 5.1 Principal components ranked by the amount of variance they capture in the original dataset without scaling.

A common approach used to determine a subset of principal components is the explained variance, which can be calculated from the eigenvalues. The explained variance tells how much information (variance) can be attributed to each of the principal components. Furthermore, the cumulative explained variance ratio gives the ability to estimate how many components are needed to describe the data.



(a) Vast scaling

(b) Standard Scaling

(c) Range scaling

(d) Pareto scaling

Figure 5.2 Principal components ranked by the amount of variance they capture in the original dataset using different scaling methods.

The plot shown in Figure 5.1 is plot of the explained variance (eigenvalue magnitudes sorted in descending order) and cumulative explained variance ratio as function of the number of principal components after applying PCA transformations on the TFPF dataset without scaling. It clearly shows that most of the variance (99 % of the variance to be precise) can be explained by the first

principal component alone. This is because the variance scale is huge in the training set and data have high non-homogeneous units. If the PCA is applied on such dataset, the resulting weight for features with high variance will also be significant. As a result, principal components will be biased towards features with high variance, leading to false results. Therefore, it is imperative to standardize the dataset before applying PCA.

Making sure that each feature has approximately the same scale can be a crucial pre-processing step because most algorithms are very sensitive to the scaling of the data. Unfortunately, there is no best way to scale variables before running Principal Component Analysis. Data pretreatment is problem dependent [3]. Below are 4 methods of features scaling that have been investigated. First, for each method, scaling was performed on dataset. Then, PCA transformation was applied on those scaled datasets. Finally, the explained variance ratio and cumulative variance ratio were plotted in Figure 5.2.

It can be seen that, in Pareto scaling (Figure 5.2d), most of variance is attributed to the first component. That can be explained by the fact that Pareto scaling method reduce only the relation importance of large values but keep data structure partially intact and close to the original dataset. For Vast Scaling, shown in Figure 5.2a, the first principal component contains 95% of the information and 99% of the information are retained by the first 3 principal components. Auto scaling (Standard scaling) and Range scaling, on the other hand, show different behavior. For Auto scaling (Figure 5.2b), the first component beers about 32% of information while in Range scaling (Figure 5.2c) it is about 57% of variance for the first component. Here it can be noticed that the two-dimensional projection loses a lot of information both for Auto scaling and Range scaling and that 10 components are needed to retain more 90% of the variance for Auto Scaling and 5 components are required in the case of Range scaling method.

5.1.2 Reconstruction Error of PCA

To evaluate the data reconstruction errors, the data were compressed and then decompressed by means of PCA. The obtained results are shown in Figure 5.3 and Figure 5.4. Furthermore, the difference of the reconstruction error of the first and second component is displayed in Table 5.1.

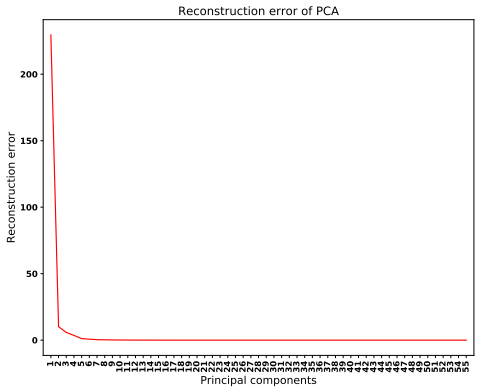
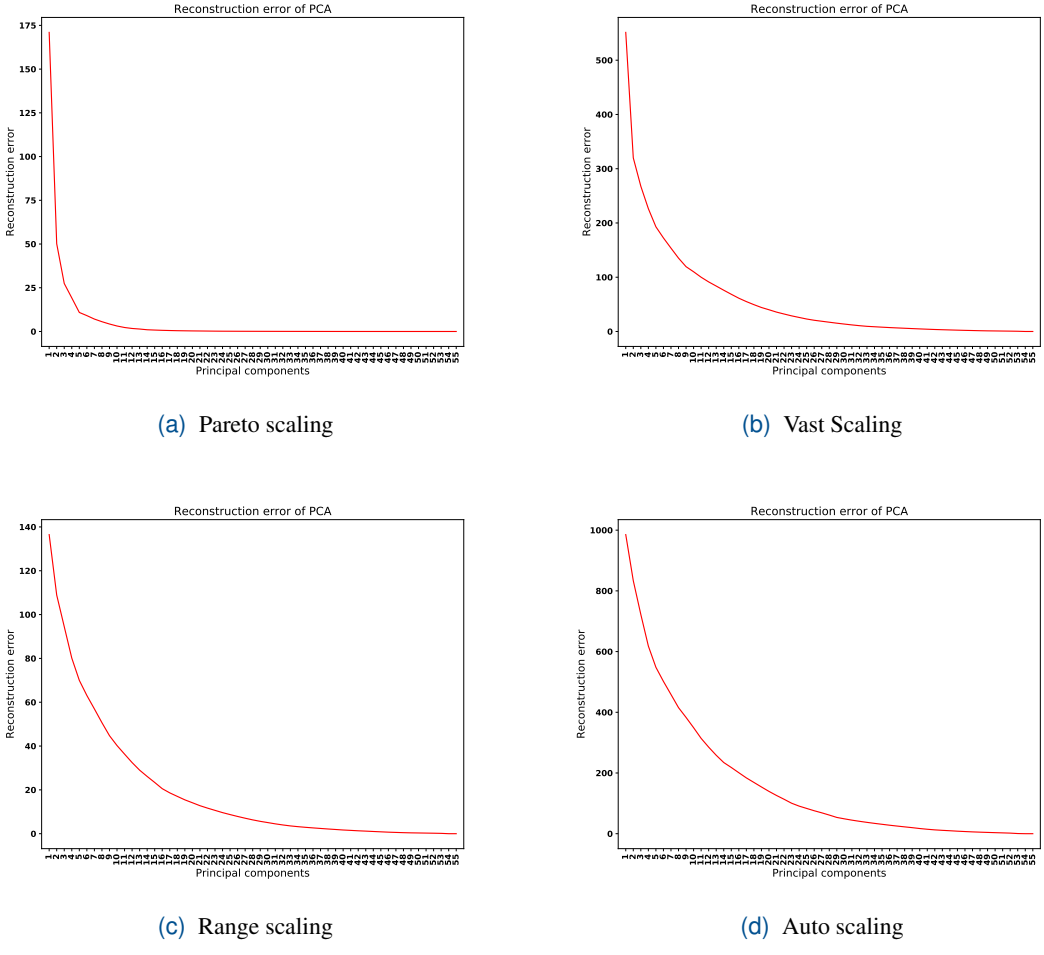


Figure 5.3 | Reconstruction error of PCA without scaling

The table shows that performing PCA without scaling results in high reconstruction error when keeping only one component (compared to second component). Thus, even performing PCA without scaling needs one principal component to recover up to 99% of the variance of the selected dataset, retaining only one principal component lead to very poor reconstruction of the dataset. This is also valid in case of using Pareto scaling. On the other hand, the other scaling methods such us Vast, Range and Auto scaling have smaller reconstruction error difference. It is also imperative to note that the reconstruction error with respect to the original dataset is always zero when all PCs are retained.



(a) Pareto scaling

(b) Vast Scaling

(c) Range scaling

(d) Auto scaling

Figure 5.4 | Reconstruction error of PCA with different scaling methods.

Table 5.1 Comparison of the reconstruction error of the first component and the second component.

Scaling method	Reconstruction error of the first component	Reconstruction error of the second component
Without scaling	229.536	10,096
Pareto	170.996	50,158
Range	136.498	108,907
Vast	551.283	319,997
Auto	985.364	833,526

5.1.3 Projection on the New Feature Space

By applying PCA to TFPF dataset without scaling and retaining only 2 components, Figure 5.5 is obtained. The Figure shows the TFPF dataset plotted using the first principal component and the second principle component. The inlet temperature Tin (K) is added as data labels.

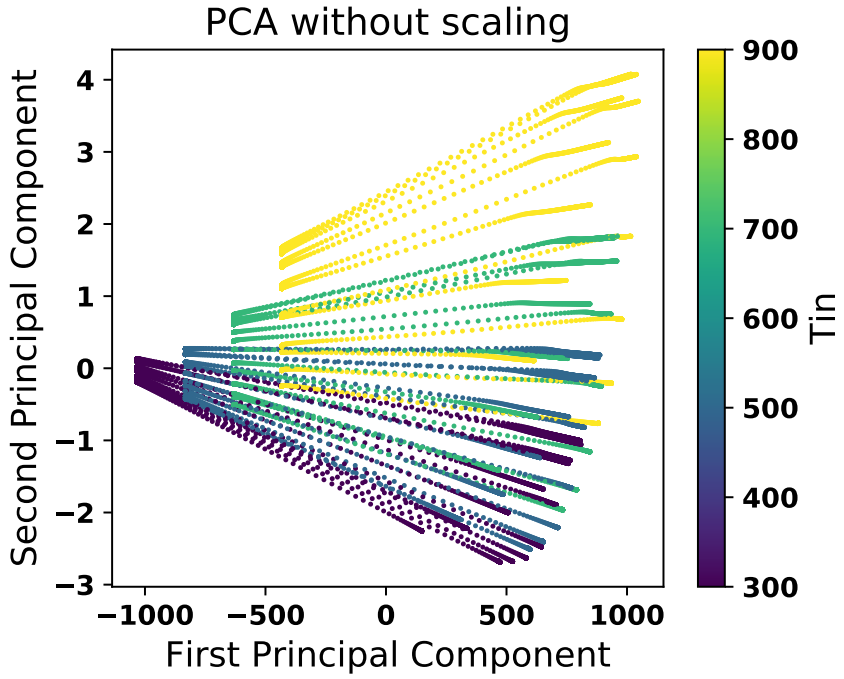
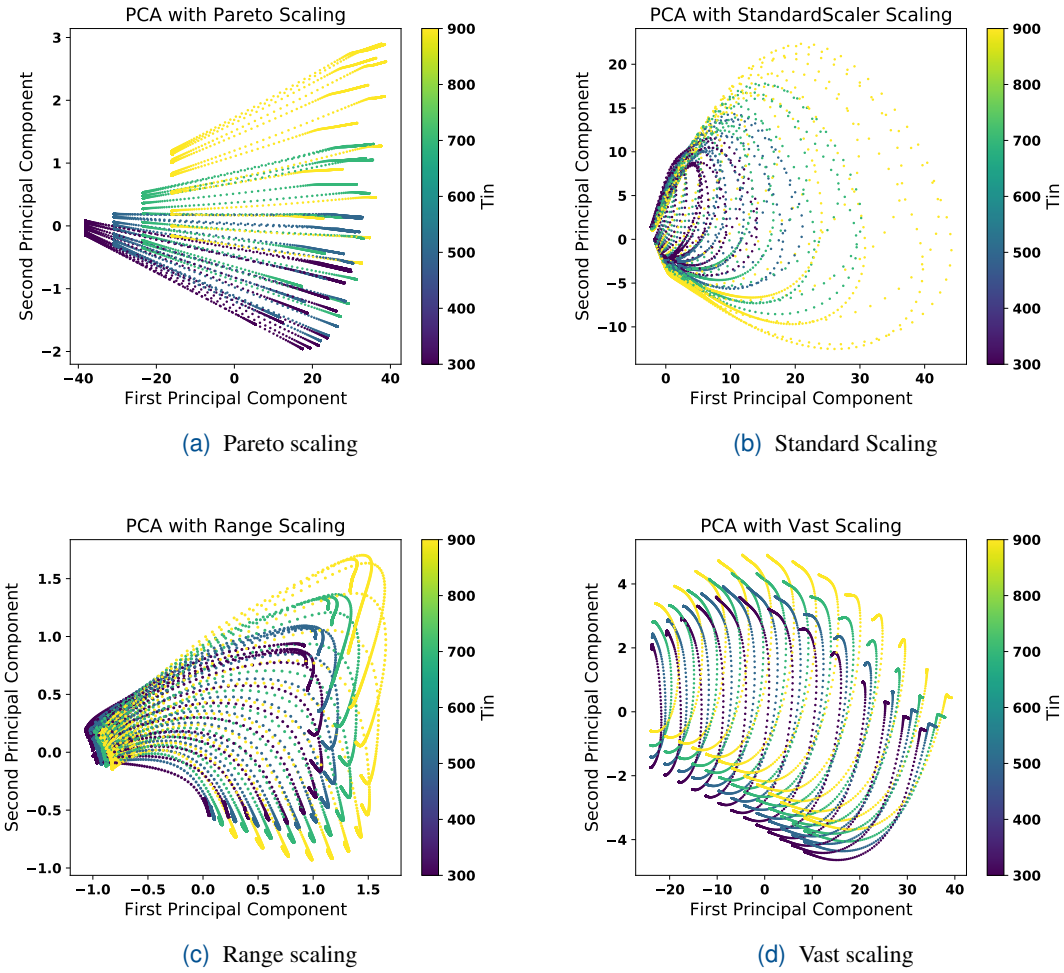


Figure 5.5 PCA maps of PC1 vs PC2 of the TFPF dataset colored with Tin without scaling.

Performing PCA without scaling makes the separation between dataset observations much harder to see. Certainly, there are 4 different maps (colors) corresponding to the 4 different inlet temperatures Tin. However, the lines/trajectories are still very dense and overcome each other particularly at low inlet temperature, which makes their separation difficult.



(a) Pareto scaling

(b) Standard Scaling

(c) Range scaling

(d) Vast scaling

Figure 5.6 PCA maps of PC1 vs PC2 of the TFPF dataset colored with Tin and by using different scaling methods.

Figure 5.6 shows the TFPF dataset plotted using the first principal component and the second principal component using different methods of scaling.

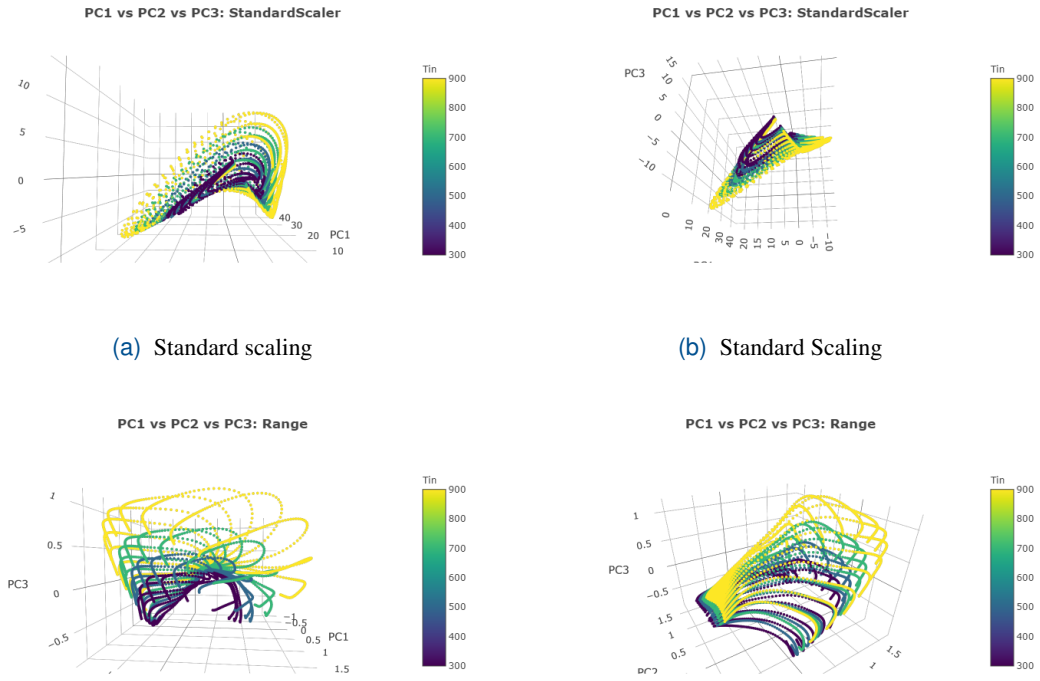
The figure 5.6a obtained using Pareto scaling is largely similar to the one without scaling. As mentioned before, this is because the Pareto scaling effect only features with large values but keep data structure partially intact and close to the original dataset.

With Auto scaling some improvements can be seen (Figure 5.6b, particularly when the value of the first component score is greater than 10 where the trajectories are very distinct. The dense zone can be explained by the fact that the trajectories form circles and the points at the end of the trajectories tend to rejoin the starting points.

Range and Vast scaling (Figure 5.6c and 5.6d respectively), on the other hand, give good results. First, the trajectories are very distinct and separated from each other. The dataset is

categorized into 13 distinguishable trajectories, each trajectory corresponds to different specific equivalence ratio. Furthermore, the 13 distinct trajectories form 4 maps for each inlet temperature. The structure of data which consists of 4 FPF datasets with different inlet temperature $T_{in}(k)$ leading to 52 trajectories is now clearly seen. In addition, it can be noted that for all Figures, these trajectories are classified into different zones where the trajectories are either very dense or clearly separated. These zones will be studied in more details with the help of clustering algorithms.

Figures 5.7 and 5.8 were obtained in the same way as before but plotted using the fist 3 principal components. The 3D figures allow these maps to be viewed from different angles. Furthermore, the Figures clearly illustrates the existence of the four maps that that are separated from each other.



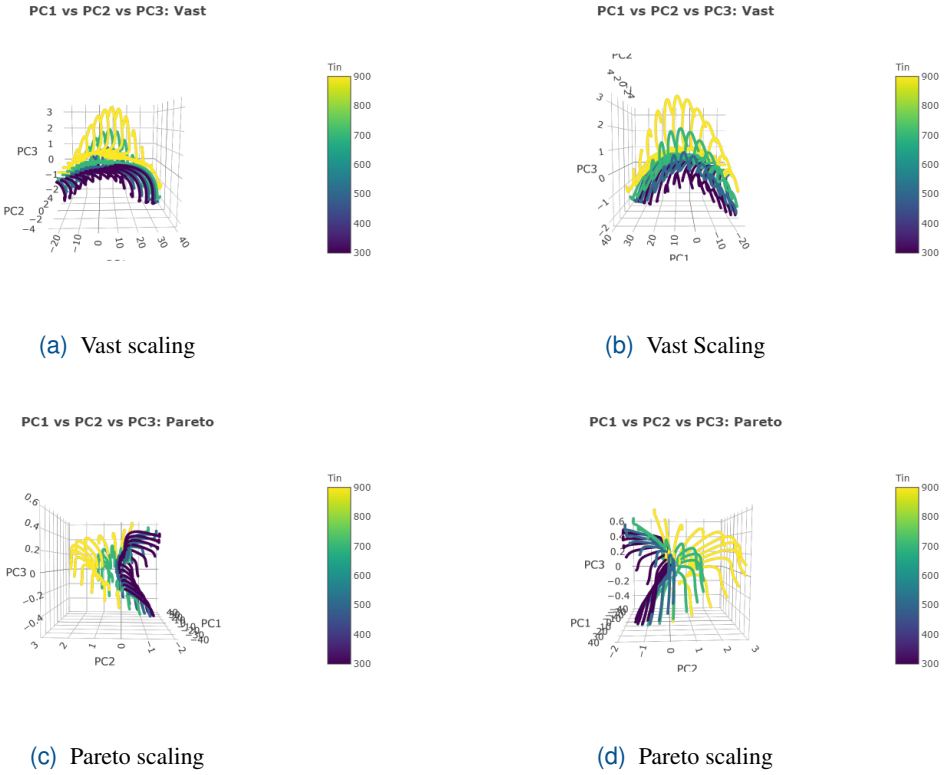
(a) Standard scaling

(b) Standard Scaling

(c) Range scaling

(d) Range scaling

Figure 5.7 PCA maps of PC1 vs PC2 vs PC3 of the TFPF dataset colored with Tin (from different sides).



(a) Vast scaling

(b) Vast Scaling

(c) Pareto scaling

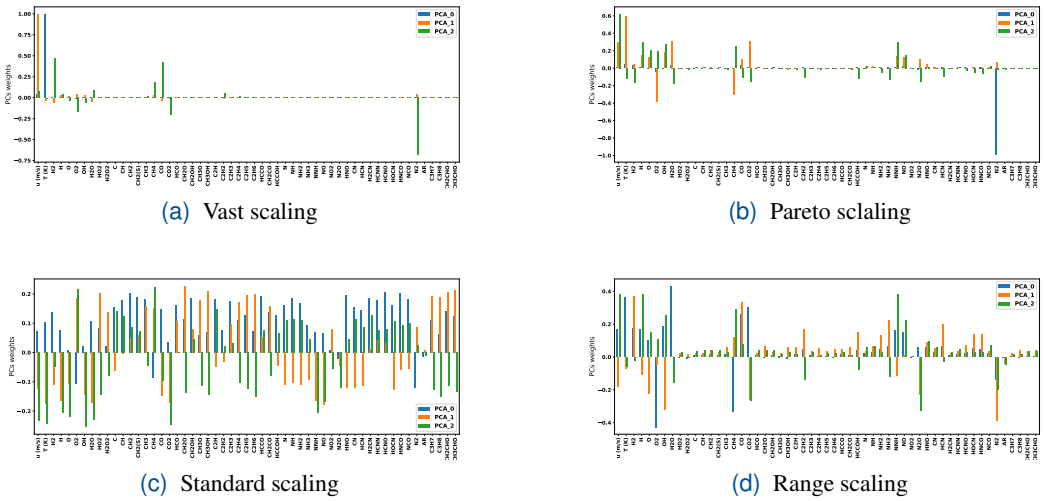
(d) Pareto scaling

Figure 5.8 PCA maps of PC1 vs PC2 vs PC3 of the TFPF dataset colored with Tin (from different sides).

5.1.4 Results Interpretation

From the results obtained so far, clear separated trajectories and regions in the data manifold were observed thanks to PCA maps using only two PCs. However, the interpretation of these components is not so easy to understand. Unfortunately, the dimensionality reduction step might come at the cost of having a reduced set of components that are harder to interpret.

With PCA, the found components are linear combinations of original features (variables). The most important features associated with the first 3 principal components are represented in Figure5.9. In particular, the Figures 5.9a, 5.9b, 5.9c and 5.9d graphically indicate the weight of the original variables on the first 3 components using different scaling methods.



(a) Vast scaling

(b) Pareto sclaling

(c) Standard scaling

(d) Range scaling

Figure 5.9 | Contributions of the variables to the first 3 principal components.

Variables that are correlated with first 3 PCs are the most important in explaining the variability in the dataset. On the other hand, variables that do not correlated with any PC or correlated with the last PCs are variables with low contribution and might be removed to simplify the overall analysis. From Figure 5.9, one can note that Besides Auto scaling which assigns an equivalent weight to almost all variables, other scaling methods put higher weight only on specific variables such as the speed of the flame stream u(m/s), the temperature T (K), CH4, CH3, O2, CO2 CO, H2O and OH. These important variables identified with this analysis will be verified and confirmed in the last chapter.

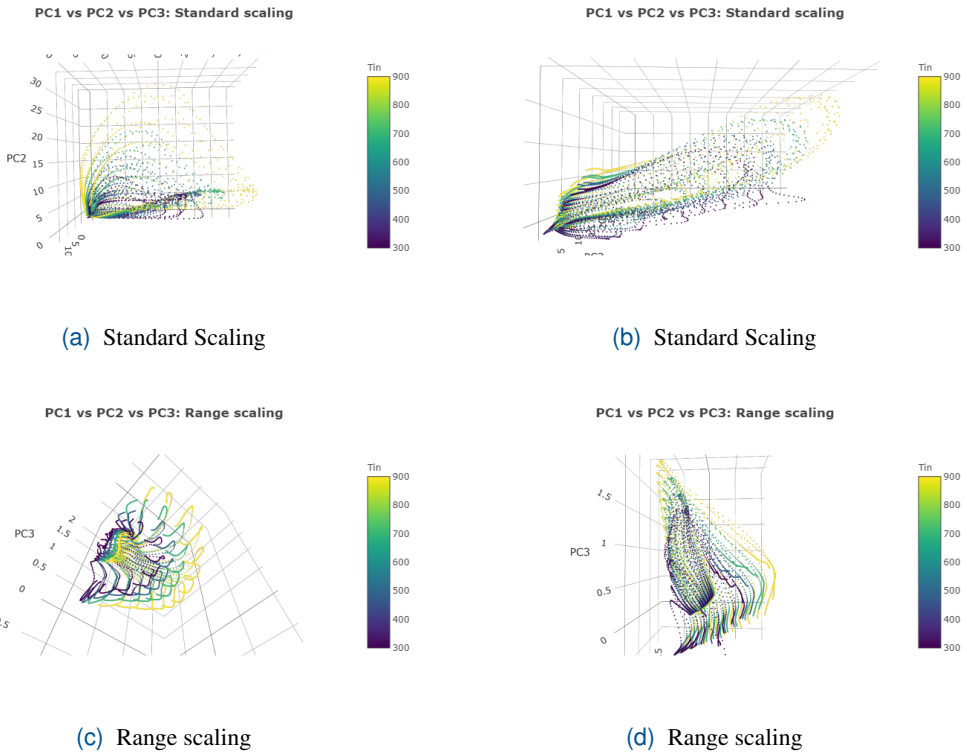
5.2 Autoencoders Results

PCA is often used for data visualization of high-dimensional datasets. The Autoencoders are another nonlinear algorithm to achieve the same purpose in the context of Deep Learning.

As with any neural network there is a lot of flexibility in how Autoencoders can be constructed such as the number of hidden layers, the number of nodes in each layer and the type of activation function. In this illustration, the following architecture $55 \rightarrow 22 \rightarrow 11 \rightarrow 3 \rightarrow 11 \rightarrow 22 \rightarrow 55$ for Autoencoders is being used. In other words, 55 nodes is used in the input layer, 22 nodes in the first hidden layer, 11 nodes in the second layer and finally, the last hidden layer consists of 3 nodes. Once all the hidden layers are trained, backpropagation algorithm (BP) is used to minimize the cost function and update the weights with labeled training set to achieve fine-tuning [14]. By using this Architecture Figures 5.10 and 5.11 were obtained.

The graphs, visualized from different sides, are significantly similar to those obtained using PCA including the separation of data, distinguishable trajectories and density of regions particularly for Auto, Range and Vast scaling. Those results are illustrated in Figures 5.11a, 5.10a and 5.10c. On the other hand, Pareto scaling (Figure 5.11c and 5.11d) does not perform well since most trajectories are placed one on top of the other forming a single map with no distinguishable

trajectories. The similarity of the results of PCA and Autoencoders algorithms can be explained by the fact that the original dataset is highly correlated and most features are linear with each other.



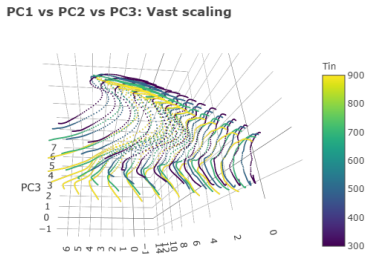
(a) Standard Scaling

(b) Standard Scaling

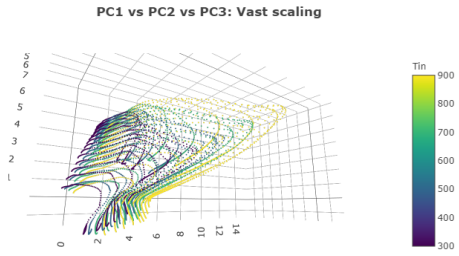
(c) Range scaling

(d) Range scaling

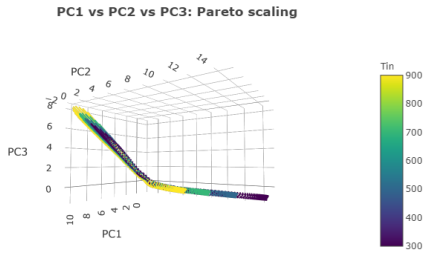
Figure 5.10 Autoencoders maps of PC1 vs PC2 vs PC3 of the TFPF dataset colored with Tin (from different sides)



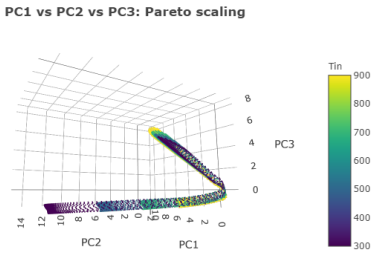
(a) Vast scaling



(b) Vast Scaling



(c) Pareto scaling



(d) Pareto scaling

Figure 5.11 Autoencoders maps of PC1 vs PC2 vs PC3 of the TFPF dataset colored with Tin (from different sides).

Chapter 6

Clustering analysis

In this chapter, both K-Means and Local PCA clustering are carried out. The primary objective of the proposed clustering algorithms is to provide a general exploratory analysis. The FPF dataset at $Tin = 300K$ is used since its simplicity facilitates the evaluation of the results and the explanation of the whole process.

6.1 Choosing the Number of Clusters

Choosing a number of clusters k is not necessarily straightforward. Especially when the dataset is large and there are no assumptions about the data. A large number k can lead to an overly fragmented partitioning of the data. This will prevent interesting patterns from being discovered in the data. On the other hand, a too small number of clusters will potentially lead to too generalist clusters containing a lot of data. The difficulty will therefore lie in choosing a number of clusters k that will highlight interesting patterns within the data. Unfortunately, there is no automated process to find the right number of clusters.

The most common method to choose the number of clusters is to launch clustering algorithm with different values of k and calculate the variance of the different clusters. Variance is the sum of the distances between each centroid of a cluster and the different observations included in the same cluster. Thus, we try to find a number of k so that the selected clusters minimize the distance between their centers (centroids) and observations in the same cluster [2].

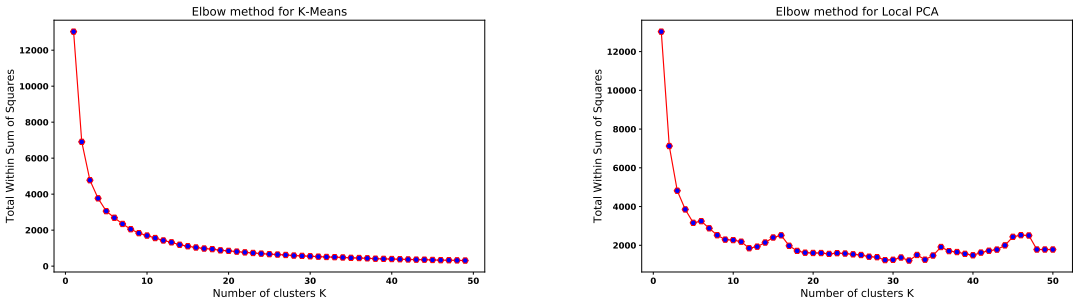
The variance of the clusters is calculated as follows:

$$V = \sum_{j} \sum_{x_i \to c_j} D(c_j, x_i)^2 \tag{6.1}$$

With:

- c_j: The center of the cluster (the centroid)
- x_i: The ith observation in the cluster with centroid c_j
- $D(c_j, x_i)$: The distance (Euclidean or other) between the center of the cluster and the point x_i

Generally, by putting in a graph the different numbers of k clusters according to variance, Figure 6.1 was obtained.



(a) K-Means Clustering (b) Local PCA Clustering

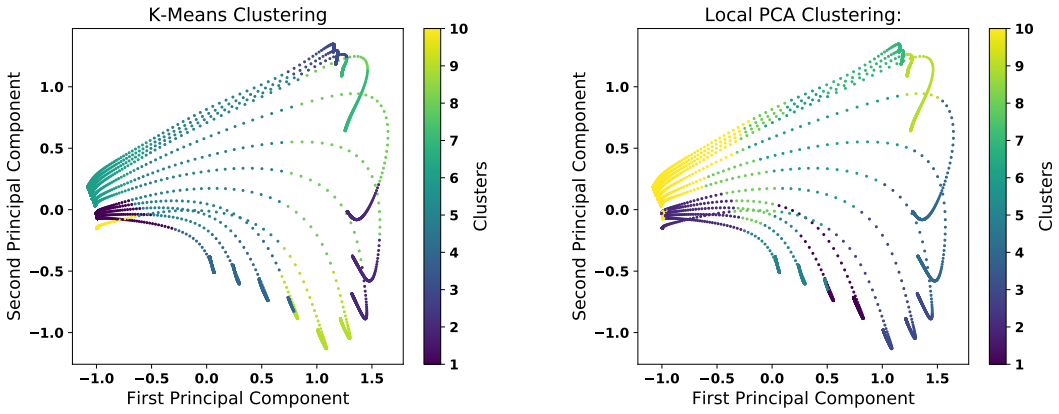
Figure 6.1 | The elbow method for K-Means and Local PCA algorithms.

The optimal number of clusters is the point representing the elbow. The elbow point is the point of the number of clusters from which the variance no longer decreases significantly. Indeed, the fall of the variance curve (distortion) between 1 and 4 clusters is significantly greater than that between 5 and 50 clusters for both algorithms. Here the elbow can be represented by K being between 5 and 10. This is the optimal number of clusters. Finally, the choice, based on this graph, between 5 or 10 clusters, remains a little vague. The choice will depend on dataset and what one is looking to accomplish. The difference between clustering with 5 and 10 clusters will be highlighted in the next section.

6.2 K-Means and Local PCA Analysis

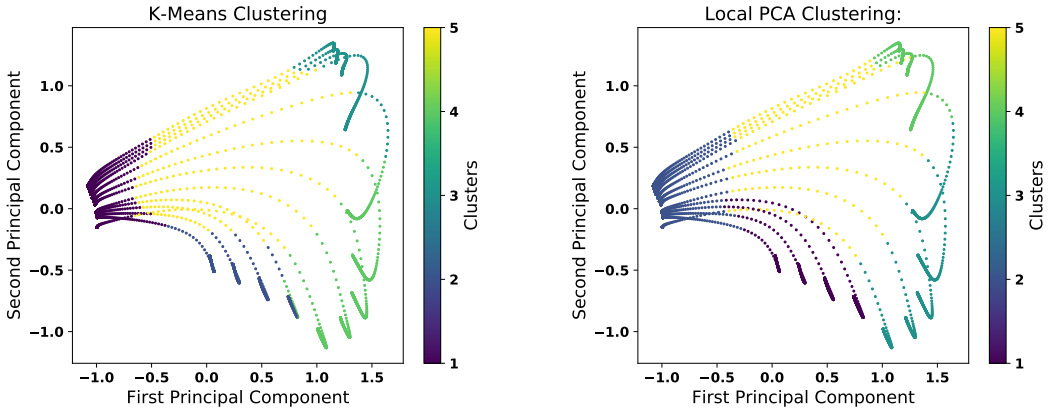
The resulting PCA maps obtained by using K-Means and Local PCA clustering algorithms on the FPF dataset at $Tin = 300K$ are shown in Figure 6.2. The scaling method used is Range method. For this illustration, 10 clusters have been chosen for the both algorithms. The clusters partition is largely similar to each other. Moreover, both algorithms group the dataset based on the shape of the trajectories, density of each zone and the variation of the specific equivalence ratio.

It can be seen from both maps that the first zone (dense area) is grouped into 2 clusters. This is because the specific ratio is varying and effecting the quantities of other chemical species like CH4, CH3 and O2. In the median zone, the points begin to separate and the trajectories are more distinct. At the end of trajectories, very quickly, the points start to accumulate again. It is also imperative to note that the partitions are significantly influenced by the different values of the specific ratio of the fuel/O2 mixture. Hence, reducing the number of clusters from 10 to 5 clusters can lead to low impact of the later on dataset partitioning. The effect of reducing the number of clusters by choosing only 5 clusters is illustrated in Figure 6.3. Both maps are almost identical to each other. This time, the dense region is represented by one cluster (cluster 1 in K-Means and clusters 2 in Local PCA). The majority of observations in the middle zone are also assigned to one cluster (cluster 5) except for few trajectories that are very short in length. Finally, the other clusters are attributed to the region where points tend to condense again. This region is still affected by the specific equivalence ratio.



(a) K-Means Clustering (b) Local PCA Clustering

Figure 6.2 | PCA maps of the FPF dataset colored with clusters number: 10 clusters are used.



(a) K-Means Clustering (b) Local PCA Clustering

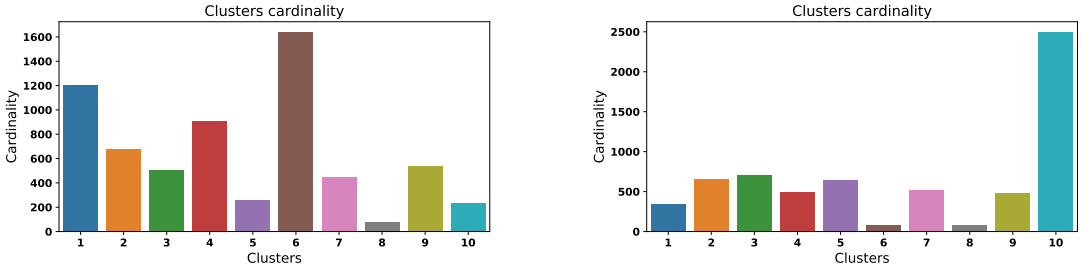
Figure 6.3 | PCA maps of the FPF dataset colored with clusters number: 5 clusters are used.

Clusters Cardinality

Figure 6.4 presents the cardinality (population) of each cluster for the both algorithms using 10 clusters. One can see that, with Local PCA algorithm (Figure 6.4b) cluster number 10 has the biggest population (more than 70% of the whole population) and assigned to the densest zone shown in Figure 6.2. In addition, clusters 6 and 8 have the lowest population and correspond to the middle region. The remaining clusters have an almost equitable number of points and belong to the last region.

For K-Means clustering (see Figure 6.4a), cluster 1 and 6 have the highest population and they

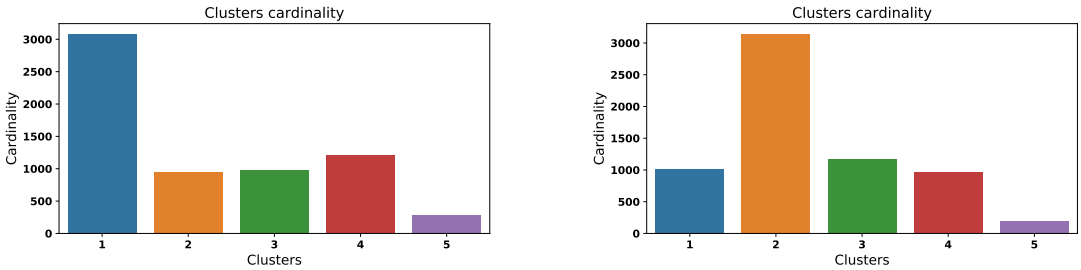
belong to densest zone. Clusters 5 and 8 have the lowest population and correspond to the middle region.



(a) K-Means clustering (b) Local PCA clustering Scaling

Figure 6.4 | Cluster cardinality using 10 clusters. Scaling method: Range scaling

By using only 5 clusters (see Figure 6.5) the results are quite similar to each other. The cluster that represents the dense zone has the biggest population for both algorithms. The cluster that represents the middle zone (cluster 5) have lowest population. The other clusters that represent the end of the trajectories have an almost equitable population and contain about 1,000 observations for each cluster.



(a) K-Means Clustering (b) Local PCA Clustering

Figure 6.5 | Cluster cardinality using 5 clusters. Scaling method: Range scaling.

To highlight the characteristics of each cluster and each zone, the most important features that have a high impacts on the formulation of the clusters were identified using a technique called feature selection (feature importance) performed by the help of Random Forest classifier. This technique is the objective study of the coming sections.

6.3 Feature Importance

Feature importance, also known as feature selection, will basically explain which features are more important in the model. The concept is straightforward: the importance of a feature is measured by calculating the increase in the model's prediction error after permuting the feature. A

feature is important if shuffling its values increases the model error, because in this case the model relied on the feature for the prediction. A feature is unimportant if shuffling its values leaves the model error unchanged, because in this case the model ignored the feature for the prediction [15]. The permutation feature importance measurement was introduced by Breiman (2001) [5] for Random Forests. Random Forest is one the most popular Machine Learning method thanks to its good accuracy, robustness and ease of use. It also provides a straightforward method for feature selection called mean decrease impurity.

Random Forest consists of several decision trees. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure based on which the (locally) optimal condition is chosen is called impurity (information gain/entropy). Thus, when training a tree, one can compute how much each feature decreases the weighted impurity in a tree. For a Random Forest, the impurity decrease from each feature can be averaged and the features are ranked according to this measure [19].

Since the FPF dataset is now labeled which means that each input is assigned to corresponding cluster, classification can be applied using Random Forest classifier on FPF dataset as input data and clusters indexes as target variable. The mean decrease impurity method was computed directly from the Random Forest and used to select the most relevant variables with the mentioned dataset. The results are presented in Figures 6.6 and 6.7

As one can see from both Figures, features like H2O, T (K), O2, CO2, CO and OH are estimated to be the most informative ones for the Random Forest. One can be also noticed that features like CN, HNO, NH, NNH, NCO, AR, C3H7 and H2CN are less important and can be dropped. This confirms the results found so far with PCA analysis.

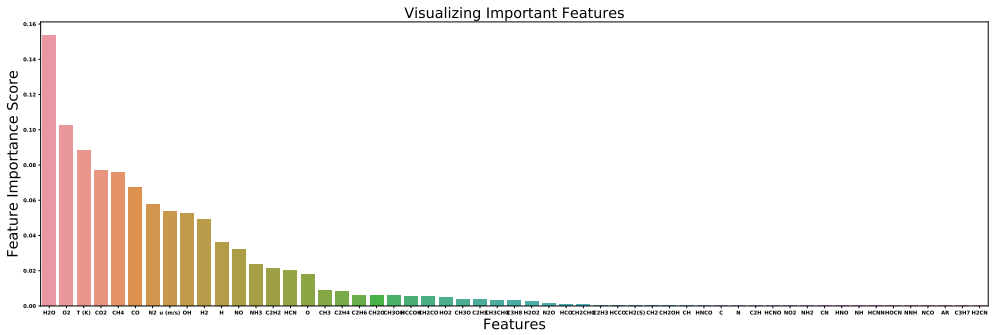


Figure 6.6 | The most important variables in the case of Local PCA clustering

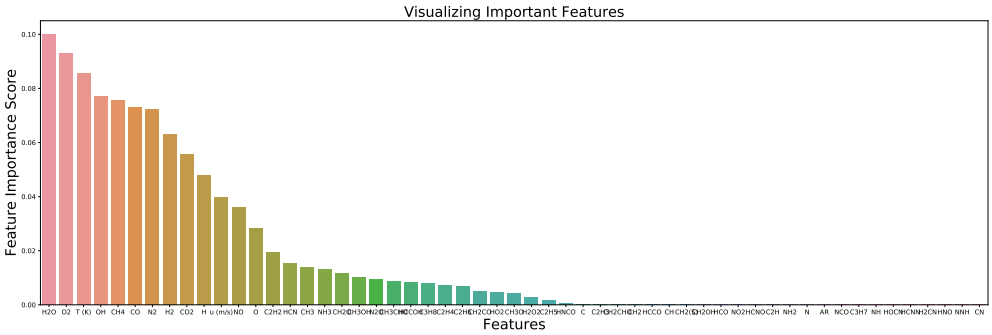


Figure 6.7 | The most important variables in the case of K-Means clustering

6.4 Cluster Interpretation

The last step of clustering process deals with the representation of the clusters. The ultimate goal of this section is to provide meaningful insights from the original dataset.

Table 6.1 | Knowledge Extraction using Local PCA clustering

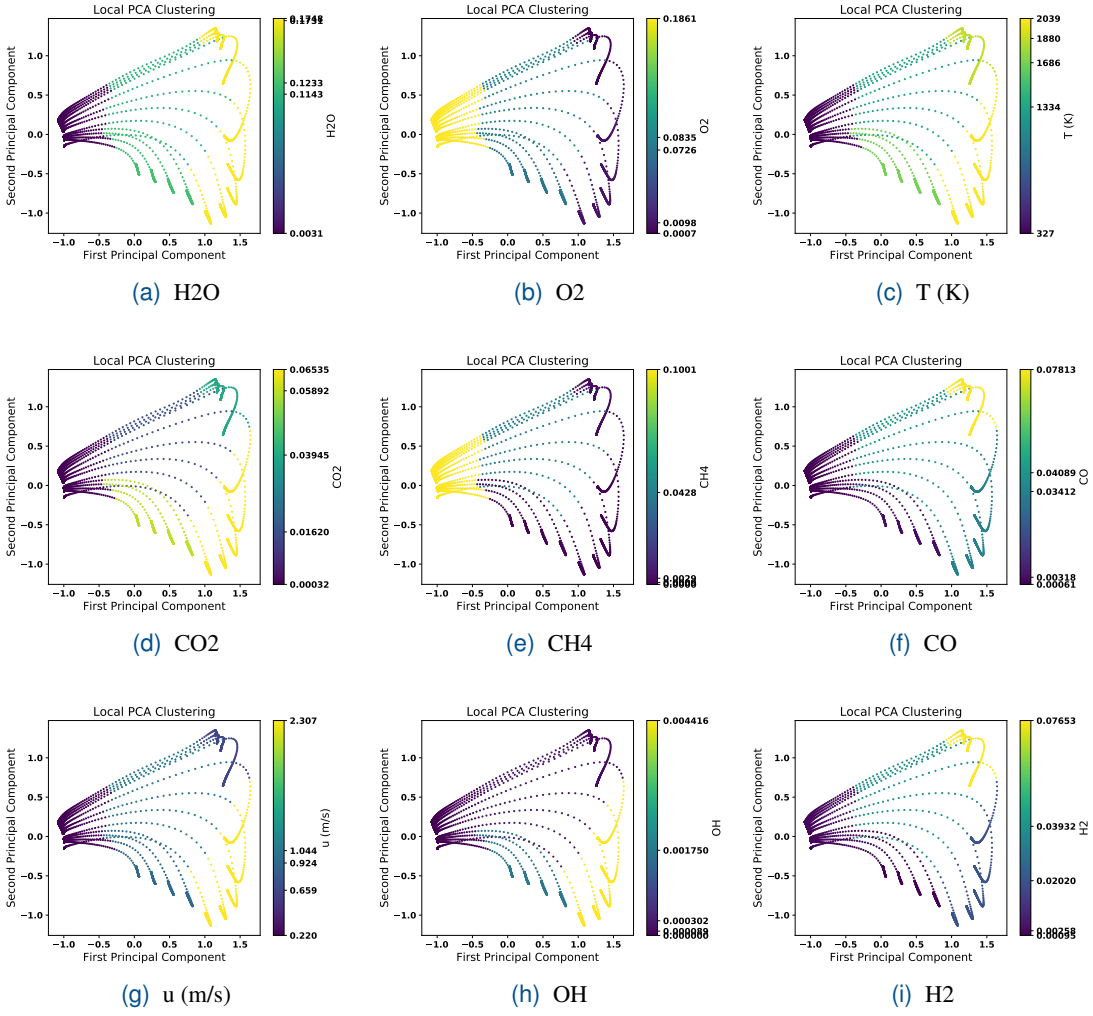
Clusters	H2O	O2	T (K)	CO2	CH4	CO	u (m/s)	OH	H2	Zone
2	0.0031	0.1861	327	0.00032	0.1	0.0006	0.220	0.00	0.00095	Preheating
5	0.12	0.08	1334	0.039	0.04	0.04	1.00	0.002	0.04	Reaction
1, 3 & 4	0.18	0.0007	2039	0.065	0.0004	0.08	2.307	0.0045	0.075	Post-Combustion

Table 6.2 | Knowledge Extraction using K-Means clustering.

Clusters	H2O	O2	T (K)	CO2	CH4	CO	u (m/s)	OH	H2	Zone
1	0.0002	0.1869	318	0.00020	0.1007	0.0004	0.213	0.00	0.00062	Preheating
5	0.1	0.1039	1187	0.015	0.045	0.033	0.95	0.0016	0.0299	Reaction
2, 3 & 4	0.1738	0.0010	2032	0.65	0.0008	0.07	2.2	0.0044	0.076	Post-Combustion

By labeling FPF dataset belonging to each cluster with the mean score of the most important features found in Figures 6.6 and 6.7, general affinity of each cluster can be identified (See Tables 6.26.1). For example, cluster 1 found using K-Means clustering (cluster 2 in Local PCA clustering) that corresponds to the dense region with high cardinality value can be marked as Preheating zone since most points/observations in this cluster have high CH4 and O2 values and lower values of T (K), H2O CO2, CO and OH. deploying similar arguments, other clusters in the map can be marked as indicated in Tables 6.2 and 6.1. Cluster 5 found using both clustering algorithms shows significant decrease of both CH4 and O2 values since the reactants are starting to burn while products such as H2O, CO2, CO and OH begin to form. In addition, the speed of flame stream and temperature values begin to rise. The region belongs to this cluster is marked as Reaction zone. Clusters 2, 3 and 4 in K-Means (1, 3, & 4 in Local PCA) are identical to

each other but they are affected by the specific ratio. They constitute of high values of T (K), H2O CO2, CO and OH but absence of CH4, CH3 and O2. These clusters can be marked as Post-Combustion zone. These results are shown in Figures 6.8 and 6.9 and summarized in Tables 6.2 and 6.1.

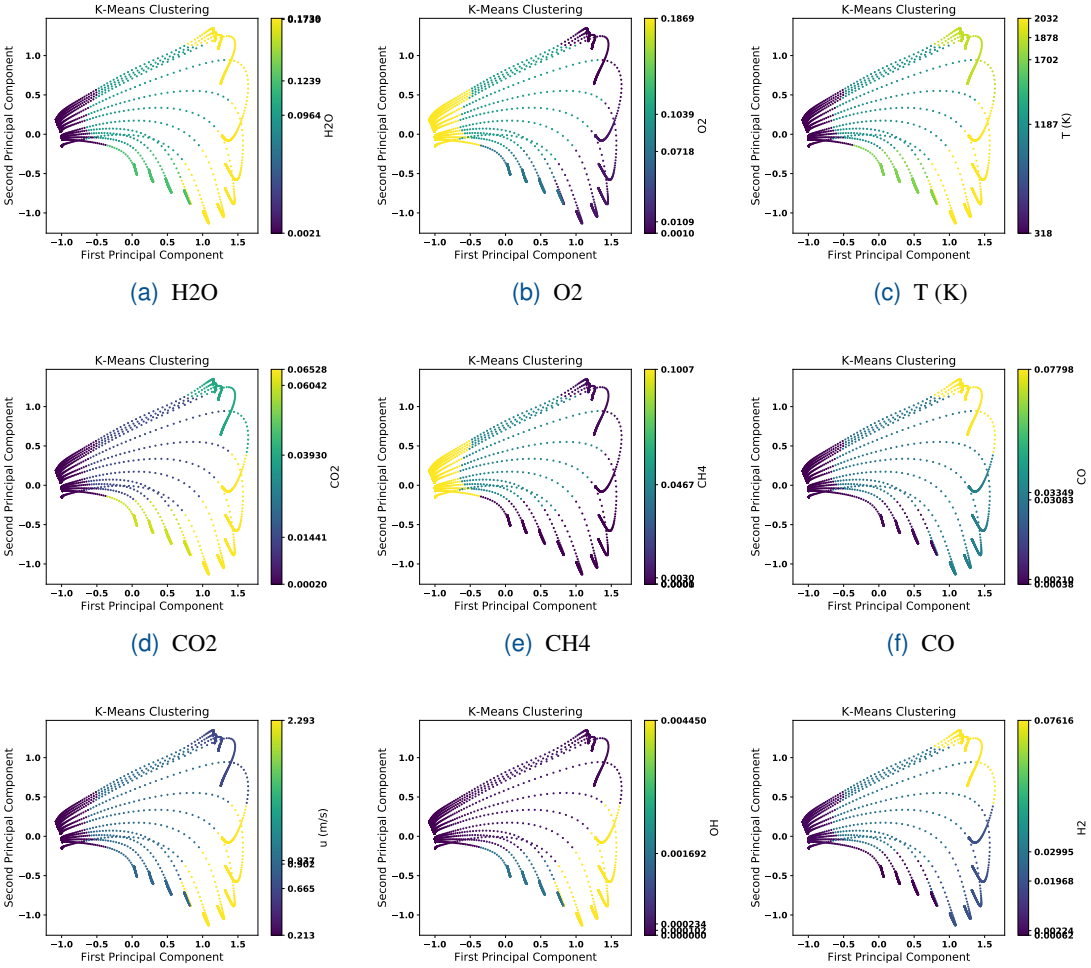


(a) H2O (b) O2 (c) T (K)

(d) CO2 (e) CH4 (f) CO

(g) u (m/s) (h) OH (i) H2

Figure 6.8 PCA maps of PC1 vs PC2 of the TFPF dataset colored with the most important variables as labels for knowledge extraction using Local PCA clustering.



(a) H2O (b) O2 (c) T (K)

(d) CO2 (e) CH4 (f) CO

(g) u (m/s) (h) OH (i) H2

Figure 6.9 PCA maps of PC1 vs PC2 of the FPF dataset colored with the most important variables as labels for knowledge extraction using K-Means clustering.

Chapter 7

Conclusion

In the first part of this thesis, a review of Machine Learning techniques such as Dimensionality Reduction and Clustering was made. Combustion data generated from premixed and non-premixed laminar flame simulations of a series of well documented piloted flames with inhomogeneous inlets was selected as case study to investigate the performance of Principal Component Analysis and Autoencoders algorithms. Different scaling methods were applied before implementing those algorithms.

The projection of the original data into lower supspace using PCA and Autoencoders was performed and compared for each scaling method. The results show that the structure of original dataset (TFPF) which consists of 4 FPF datasets with different inlet temperature Tin (K) forms 4 maps. Each map corresponds to one FPF dataset and classified into 13 distinguishable trajectories, each trajectory corresponds to different specific equivalence ratio.

In order to investigate the internal structure of these datasets, clustering analysis was performed using both K-Means and Local PCA algorithms. To avoid the effect of change in the inlet temperature Tin (K) on the clustering analysis performance, the FPF dataset at $Tin = 300K$ was used. First, an optimal number of clusters was selected using the elbow method. Then, Random Forest classifier was used to select the most important features that have a high impacts on the formulation of the clusters. Finally, general affinity of each cluster was identified by labeling the dataset belonging to each cluster with the average score of these selected features.

By the help of this clustering analysis, the combustion process which takes place in 3 stages was identified. The first stage is Preheeting marked with a high value of CH4 and O2 and the absence of H2O and CO2. The second stage is Reaction stage, which shows a significant decrease of both CH4 and O2 values since the reactants start to burn while products such as H2O, CO2, CO and OH begin to form. The last stage is Post-combustion, this stage is marked by a high values of H2O and CO2 and absence of CH4 and O2.

Bibliography

[1] Lambros S. Athanasiou, Dimitrios I. Fotiadis, and Lampros K. Michalis. "5 - Plaque Characterization Methods Using Optical Coherence Tomography". In: *Atherosclerotic Plaque Characterization Methods Based on Coronary Imaging*. Ed. by Lambros S. Athanasiou, Dimitrios I. Fotiadis, and Lampros K. Michalis. Oxford: Academic Press, 2017, pp. 95–113. ISBN: 978-0-12-804734-7. DOI: https://doi.org/10.1016/B978-0-12-804734-7.00005-1. URL: http://www.sciencedirect.com/science/article/pii/B9780128047347000051.

[2] Younes Benzaki. *Tout ce que vous voulez savoir sur l'algorithme K-Means*. Date Published: April 12, 2018. URL: https://mrmint.fr/algorithme-k-means.

[3] Robert van den Berg et al. "Van den Berg RA, Hoefsloot HCJ, Westerhuis JA, Smilde AK, Van der Werf MJ.. Centering, scaling, and transformations: improving the biological information content of metabolomics data. BMC Genomics 7: 142-157". In: *BMC genomics* 7 (Feb. 2006), p. 142. DOI: 10.1186/1471-2164-7-142.

[4] Niket Borade and Ratnadeep Deshmukh. "Comparative Study of Principal Component Analysis and Independent Component Analysis". In: *International Journal of Computer Applications* 92 (Apr. 2014), pp. 45–49.

[5] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[6] Rafael Calvo, Matthew Partridge, and Marwan Jabri. "A Comparative Study of Principal Component Analysis Techniques". In: (Oct. 1998).

[7] Liton Chandra Paul, Abdulla Suman, and Nahid Sultan. "Methodological analysis of principal component analysis (PCA) method". In: *International Journal of Computational Engineering & Management* 16 (Mar. 2013), pp. 32–38.

[8] Rafael Espirito Santo. "Principal Component Analysis applied to digital image compression". In: *Einstein (São Paulo, Brazil)* 10 (June 2012), pp. 135–9. DOI: 10.1590/S1679-45082012000200004.

[9] Ehsan Fooladgar and Christophe Duwig. "A new post-processing technique for analyzing high-dimensional combustion data". In: *Combustion and Flame* 191 (2018), pp. 226–238. ISSN: 0010-2180. DOI: https://doi.org/10.1016/j.combustflame.2018.01.014. URL: http://www.sciencedirect.com/science/article/pii/S001021801830018X.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[11] David G Goodwin, Harry K Moffat, and Raymond L Speth. "Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes". In: *Caltech, Pasadena, CA* (2009).

[12] Jeremy Jordan. *Introduction to autoencoders*. Date Published: March 19, 2018. URL: https://www.jeremyjordan.me/autoencoders/.

[13] Prateek Joshi. *Artificial intelligence with python*. Packt Publishing Ltd, 2017.

[14] Guifang Liu, Huaiqian Bao, and Baokun Han. "A Stacked Autoencoder-Based Deep Neural Network for Achieving Gearbox Fault Diagnosis". In: 2018.

[15] Christoph Molnar. *Interpretable Machine Learning*. Date Published: July 16, 2019. URL: https://christophm.github.io/interpretable-ml-book/.

[16] Andreas C Müller, Sarah Guido, et al. *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.", 2016.

[17] Alessandro Parente and James Sutherland. "Principal component analysis of turbulent combustion data: Data pre-processing and manifold sensitivity". In: *Combustion and Flame* 160 (Feb. 2013), pp. 340–350. DOI: 10.1016/j.combustflame.2012.09.016.

[18] A. Parente et al. "Investigation of the MILD combustion regime via Principal Component Analysis". In: *Proceedings of the Combustion Institute* 33.2 (2011), pp. 3333–3341. ISSN: 1540-7489. DOI: https://doi.org/10.1016/j.proci.2010.05.108. URL: http://www.sciencedirect.com/science/article/pii/S1540748910002725.

[19] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. "Robust Feature Selection Using Ensemble Feature Selection Techniques". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Walter Daelemans, Bart Goethals, and Katharina Morik. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 313–325.

[20] The Royal Society. *Machine learning: the power and promise of computers that learn by example*. Last updated: July 20, 2017. URL: https://www.appg-ai.org/library/machine-learning-power-promise-computers-learn-example/.

[21] Jake VanderPlas. *Python Data Science Handbook: Essential Tools for Working with Data*. 1st. O'Reilly Media, Inc., 2016. ISBN: 1491912057.

[22] Lin Wu, Xiaofeng Zhu, and Tao Tong. "Global and local clustering with kNN and local PCA". In: *Multimedia Tools and Applications* 77.22 (Nov. 2018), pp. 29727–29738. ISSN: 1573-7721. DOI: 10.1007/s11042-018-6488-1. URL: https://doi.org/10.1007/s11042-018-6488-1.

[23] Chun Sang Yoo et al. "A DNS study of ignition characteristics of a lean iso-octane/air mixture under HCCI and SACI conditions". In: *Proceedings of the Combustion Institute* 34.2 (2013), pp. 2985–2993. ISSN: 1540-7489. DOI: https://doi.org/10.1016/j.proci.2012.05.019. URL: http://www.sciencedirect.com/science/article/pii/S154074891200020X.

[24] Jianbo Yu. "Local and global principal component analysis for process monitoring". In: *Journal of Process Control* 22.7 (2012), pp. 1358–1373. ISSN: 0959-1524. DOI: https://doi.org/10.1016/j.jprocont.2012.06.008. URL: http://www.sciencedirect.com/science/article/pii/S0959152412001497.