PREDICTING RELIABILITY IN MULTIDISCIPLINARY ENGINEERING SYSTEMS UNDER UNCERTAINTY

A Thesis
Presented to
The Academic Faculty

by

Sungkun Hwang

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
May 2016

PREDICTING RELIABILITY IN MULTIDISCIPLINARY ENGINEERING SYSTEMS UNDER UNCERTAINTY

Approved by:

Dr. Seung-Kyum Choi, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Yan Wang
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Ying Zhang
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 09/25/2015

I dedicate this thesis to God and my family.

ACKNOWLEDGEMENTS

First, I would like to thank my advisor Dr. Seung-Kyum Choi for his support and guidance to me with his knowledge throughout my graduate studies and school experience. I would also like to thank the reading committee, Dr. Yan Wang and Dr. Ying Zhang, for their time to read and provide their knowledge and expertise in the completion of my Master's thesis.

Moreover, I would like to thank my family, especially my father, for their unconditional love, wishes, patience and faith without which I would not have made it this far. They have inculcated me with the virtues of hard work, dedication, perseverance and belief in God since my early days. I exist because of them.

Finally I would like to thank my lab-mates, Recep Gorguluarslan, Thomas Stone, Mahmoud Alzahrani, and Sang-In Park for their help along the way and for their support.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
SUMMARY	xii
CHAPTERS	
1 INTRODUCTION	1
1.1. Multidisciplinary Engineering System	1
1.2. Random Variables	4
1.3. Random Fields	4
1.4. Multivariate Data	5
1.4.1. Introduction of Multivariate Data	5
1.4.2. Multivariate Data Analysis	6
1.4.3. Curse of Dimension of Multivariate Data	7
1.4.4. Uncertainty and Redundancy of Multivariate Data	7
1.4.5. Dependency of Multivariate Data	9
1.5. Dimension Reduction	10
1.5.1. Feature Extraction	12
1.5.2. Feature Selection	13
1.6. Motivation	13
1.7. Research Questions and Hypothesis	14
1.8. Thesis Organization	15

2 BACKGROUND 17
2.1. Representation of Dependency 17
2.1.1. Linear Correlation 17
2.1.2. Rank Correlation 18
2.1.3. Copula 18
2.2. Information Theory 25
2.2.1. Entropy 26
2.2.2. Joint Entropy and Conditional Entropy 27
2.2.3. Mutual Information 28
2.2.4. Relation between Each Terminology 29
2.3. Dimension Reduction 29
2.3.1. Feature Extraction 30
2.3.2. Feature Selection 40
2.4. Neural Network 43
2.4.1. Artificial Neural Network 43
2.4.2. Probabilistic Neural Network 50
3 PROPOSED FRAMEWORK 56
3.1. Objective of Proposed Method 56
3.2. Steps of Proposed Framework 56
3.2.1. Step 1: Generation of Multivariate Data 60
3.2.2. Step 2: Dimension Reduction 60
3.2.3. Step 3: Combination with Neural Networks 68
4 DESIGN EXAMPLES 74
4.1. 3-D Cantilever Beam Example 74
4.1.1. Problem Description 74

4.1.2. Generation of Young's Moduli 76
4.1.3. Dimension Reduction for Young's Moduli 78
4.1.4. Artificial Neural Network for Regression of Deflection Estimation 82
4.2. Solder Joint Analysis Example 84
4.2.1. Problem Description 84
4.2.2. Generation of Random Properties of Solder Ball Joint 86
4.2.3. Dimension Reduction for Properties of Solder Ball Joint 87
4.2.4. Probabilistic Neural Network for Classification 89
4.3. Stretchable Antenna Example 90
4.3.1. Problem Description 90
4.3.2. Generation of Varying Thickness 98
4.3.3. Dimension Reduction of Varying Thickness 99
4.3.4. Artificial Neural Network for Predicting Antenna Deformation 100
4.3.5. Dimension Reduction of Each Coordinate 104
4.3.6. Probabilistic Neural Network for Classification of Antenna Frequency 106
4.3.7. Validation of Efficacy of Proposed Framework 107
5 CONCLUSIONS AND FUTURE WORK 111
5.1. Summary 111
5.2. Conclusion 112
5.3. Contributions 114
5.4. Future Work 116
REFERENCES 119

LIST OF TABLES

	Page
Table 1.1: Comparison between Redundancy and Entropy	9
Table 3.1: Number of Eigenvalues and Percentage of Information	63
Table 3.2: Total Error of Prediction	71
Table 3.3: Probability of Failure	73
Table 4.1: Property of Cantilever Beam	75
Table 4.2: Redundancy of Original Young's Moduli	77
Table 4.3: Number of Eigenvalues and Percentage of Information	79
Table 4.4: Entropy Variation of Young's Moduli	81
Table 4.5: Reconstruction Error	81
Table 4.6: Prediction Error of Displacement	83
Table 4.7: Properties of Solder Joint	86
Table 4.8: Redundancy Reduction	89
Table 4.9: Probability of Failure by PNN and MCS	90
Table 4.10: Properties of Stretchable Patch Antenna	92
Table 4.11: Information of Results from Feature Extraction	100
Table 4.12: Prediction Error of ANN	101
Table 4.13: Redundancy Reduction	105
Table 4.14: Probability of Failure by PNN and MCS	107

LIST OF FIGURES

	Page
Figure 1.1: Model of the Actuation System of Spacecraft	1
Figure 1.2: Crack Analysis of Solder Ball Joint	2
Figure 1.3: Analysis of Stretchable Patch Antenna	3
Figure 1.4: Abstraction of Random Variables	4
Figure 1.5: Deterministic Analysis	5
Figure 1.6: Relationship between Redundancy and Correlation	8
Figure 1.7: The Structure of Dimension Reduction	10
Figure 1.8: Dimension Reduction Techniques	12
Figure 1.9: Dimension Reduction Techniques	13
Figure 2.1: Simulation of Gaussian Copula	20
Figure 2.2: Difference between Correlation Coefficient and Copula	21
Figure 2.3: Family of Copula Densities	24
Figure 2.4: Venn Diagram of Information Theory	29
Figure 2.5: Schematic Model of PCA	32
Figure 2.6: Process of Finding Principal Components	34
Figure 2.7: Scree Plot of PCA	35
Figure 2.8: Schematic Model of Auto-Encoder	37
Figure 2.9: Separating Two Classes Using Means	41
Figure 2.10: Independent Features Test	43
Figure 2.11: Neuron Model	44
Figure 2.12: Single Neural Network	46
Figure 2.13: Activation Functions	47

Figure 2.14: Formation of Artificial Neural Network 47

Figure 2.15: Architecture of Probabilistic Neural Network 52

Figure 2.16: Pattern Layer of PNN 53

Figure 2.17: Summation Layer of PNN 54

Figure 2.18: Limit State Function 55

Figure 3.1: Proposed Framework 59

Figure 3.2: Correlation of First Three Features Generated by Gaussian Copula 60

Figure 3.3: Criterion Deciding either Feature Extraction or Feature Selection 61

Figure 3.4: Scree Plot for Eigenvalues 62

Figure 3.5: Proposed Framework of PCA 63

Figure 3.6: Proposed Framework of AE 64

Figure 3.7: Schematic of Auto-Encoder 65

Figure 3.8: IndFeaT of Original data 67

Figure 3.9: Proposed Framework of IndFeaT 68

Figure 3.10: Proposed Framework for Regression 70

Figure 3.11: Schematic of Artificial Neural Network 70

Figure 3.12: Proposed Framework for Classification 72

Figure 3.13: Schematic of Probabilistic Neural Network 73

Figure 4.1: 3-D Cantilever Beam 75

Figure 4.2: Correlation of First Four Features by Gaussian Copula 76

Figure 4.3: Random Field Realization of Cantilever Beam Elastic Moduli 76

Figure 4.4: Scree Plot of Cantilever Beam 78

Figure 4.5: Schematic of Auto-Encoder 80

Figure 4.6: Redundancy Variation of Young's Moduli 80

Figure 4.7: Schematic of Artificial Neural Network 82

Figure 4.8: Tip Displacement 83

Figure 4.9: Crack Analysis of Solder Joint 85

Figure 4.10: First Four Features by Gaussian Copula 87

Figure 4.11: IndFeaT of 9 Properties 88

Figure 4.12: Schematic of Probabilistic Neural Network 90

Figure 4.13: Geometry of Stretchable Patch Antenna 91

Figure 4.14: Stretchable Antenna with varying thickness 92

Figure 4.15: Schematic of Stretchable Antenna 93

Figure 4.16: Boundary Condition of Stretchable Antenna 94

Figure 4.17: Deformed Geometry of Antenna 95

Figure 4.18: Resonance Frequency of Stretchable Antenna under Deformation 97

Figure 4.19: Shape of Deformed Antenna 101

Figure 4.20: Selected Significant Coordinates 105

Figure 4.21: Resonance Frequency of Deformed Antenna 108

Figure 4.22: Validation of Proposed Method with Resonance Frequency 109

SUMMARY

The future development of engineered products will require a blend of technical knowledge from multiple engineering domains that meet its relevant multidisciplinary design criteria with sufficient accuracy. However, modeling and simulating multidisciplinary engineering systems are challenging due to complexities such as interactions between various input parameters. Moreover, in order to accurately estimate risk and reliability of such complicated systems, critical input parameters and the corresponding uncertainties must be correctly captured and propagated. Multidisciplinary engineering systems often require accurate representations of multivariate phenomena. Thus, it is essential to develop a framework that can handle multivariate phenomena of complex engineering systems under uncertainty.

The proposed research will develop a framework that can accurately capture and model input and output parameters under uncertainties for multidisciplinary systems. Specifically, the Artificial Neural Network (ANN) with Principal Component Analysis (PCA) and the Auto-Encoder (AE) algorithm will be developed to handle this issue. The Independent Features Test (IndFeaT) to select a critical subset of model features will also be utilized when using the Probabilistic Neural Network (PNN). In addition, a copula function will be employed to accurately model input uncertainties. The proposed method permits complicated and multiple properties to be represented effectively and realistically, leading to accurate response predictions. Moreover, even though the general regression or classification method is operated based on original complex data sets that have large amounts of uncertainty and complexity, the proposed method can reduce the high

dimension and uncertainty of the original data to predict a response. This diminishes the error of the prediction and saves simulation costs. To demonstrate the efficacy of the proposed method, experimental results of a cantilever beam test and electro-mechanical systems such as a solder joint and stretchable patch antenna will be estimated.

CHAPTER 1

INTRODUCTION

1.1. Multidisciplinary Engineering System

Repeated physical measurements and experiments increase the chance of predicting accurate results; however, engineers prefer to model and simulate their experiments with computer systems when testing design specifications for reliability. Models and simulations provide advantages in that they can effectively decrease design cost and time, and offer precise feedback that directly affects design decisions. This feedback leads to explicit final decisions and design alternatives. The model also assists in integrating different engineering disciplines (e.g. mechanical and electrical engineering).

Figure 1.1. Model of the Actuation System of Spacecraft [1]

One example of a multidisciplinary engineering system is a spacecraft, as shown in Figure 1.1. Fabrication of such spacecraft requires the designer to have a comprehensive understanding of specific disciplines. As workers from diverse engineering domains are combined into teams, their ability to accurately predict responses and correctly handle input data from differing engineering systems becomes extremely difficult. Uncertainty or redundancy, which result from complex and numerous input variables, from a high dimension of data, or from inadequate professional knowledge of each engineering domain, make it difficult to accurately predict the response of engineering systems. Another example of a system that needs accurate predictions across various engineering domains is shown in Figure 1.2. The crack analysis of a solder ball joint is an example of electro-mechanical engineering. It integrates several disciplines' corresponding material properties and thermal analysis of circuit and solder ball.

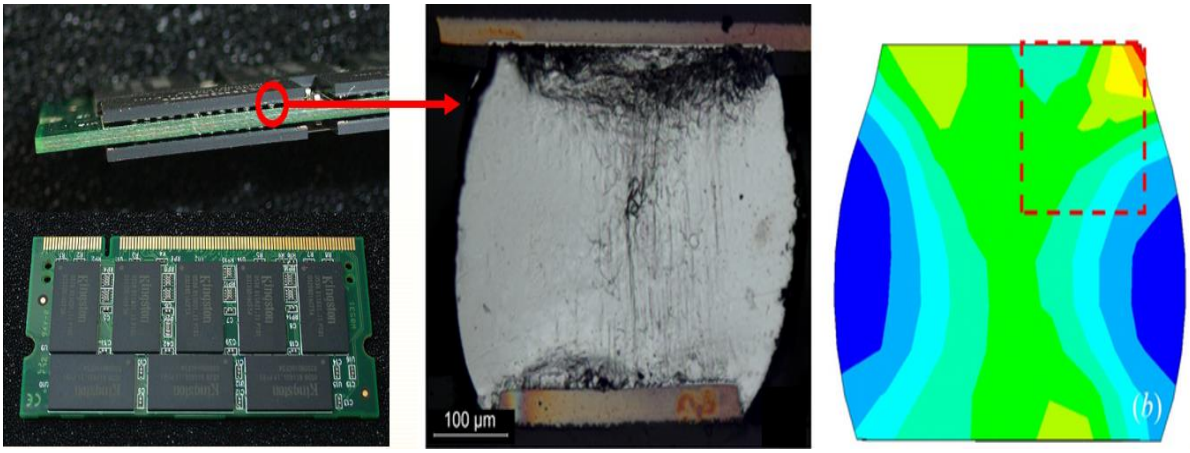


Figure 1.2. Crack Analysis of Solder Ball Joint [2]

Figure 1.3 is also an example of electro-mechanical engineering. Analysis of a stretchable patch antenna enables designers to estimate the allowable resonance frequency of the

antenna when its geometry is contracted and relaxed. Mechanical engineering is necessary to confirm the movement of the geometry, and electronic engineering is necessary to find the reliable resonance frequency.

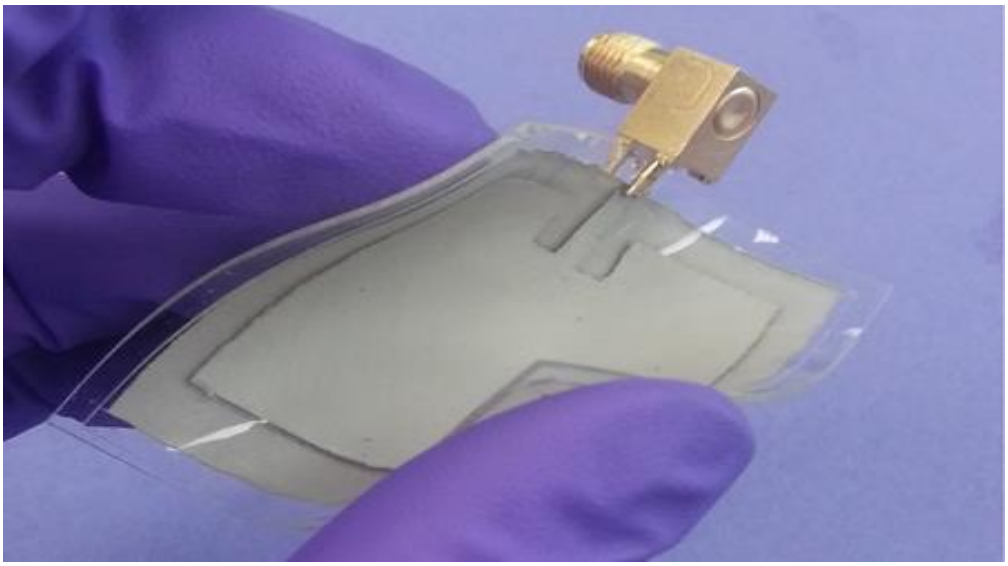


Figure 1.3. Analysis of Stretchable Patch Antenna [3]

Multidisciplinary domains produce large amounts of data with high dimensions and this data must be fairly correlated. This correlation leads to data redundancy. To represent a given property of a system, correlated data must be related, yet this can allow redundant and uncertain data to negatively influence the predicted response of the system. Huge data processing and prediction errors are other examples of issues that would negatively impact a model's accuracy. Despite these issues, precise modeling is required in order for users to draw accurate responses. This research will provide a way to readily and realistically represent all data using a function. Special techniques based on the data will reduce the uncertainty or redundancy caused by the high number of dimensions and will draw to get an accurate response.

1.2. Random Variables

Experiments that cannot predict accurate results can still produce a random variable through random experimentation. The random variable will be described by X and a particular value of the random variable will be denoted by x. Random variable X can take on any value of x in real domains. Random variables have two specific categories: continuous and discrete. Continuous random variables take on an infinite number of possible values in some interval, while discrete random variables take on a finite number of distinct values. Figure 1.4 shows how a sample space S and a random variable X are related. For each sample point s, x is defined as X (s).

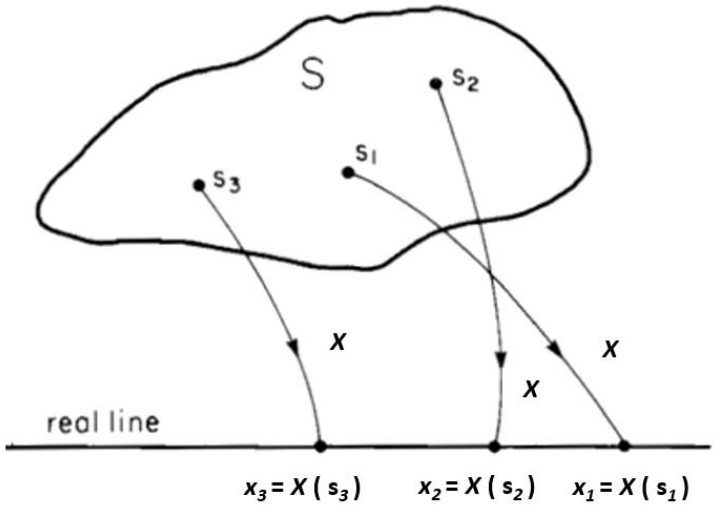


Figure 1.4. Abstraction of Random Variables [4]

1.3. Random Fields

A random field is able to take on values that are multidimensional variables in a stochastic process. Most engineering structural analyses have various distributed properties (e.g. loads, geometry, stiffness properties, manufacturing processes, and operation environments) that are random [5]. Such distributed properties have uncertainty, and this

uncertainty may limit the prediction or modeling response of a design. Thus, not only does uncertainty have to be truncated and comprehensively expressed when giving experiments data, but numerous distributed properties should also be represented realistically. Such precision in expressing input variables helps engineers accurately estimate the statistics of random responses and the reliability of structures [5]. Random fields can be analyzed by random process analysis. A random process can be denoted as the integration of random variables in a given probability space. Due to this definition, engineers consider random fields and random processes to work similarly. However, random fields deal with multidimensional variation, while random processes are based on a single coordinate of time. Through random processes, it is identified that a specific design point is a single function of time. Figure 1.5 shows that the analysis of a random process can represent every design space, since it does not depend on a single design point.

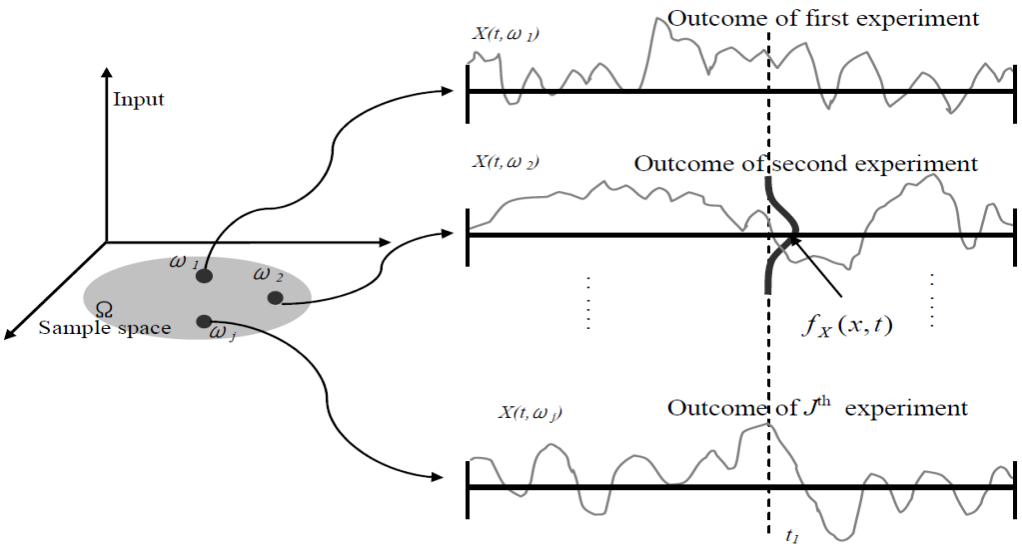


Figure 1.5. Deterministic Analysis [5]

1.4. Multivariate Data

1.4.1. Introduction of Multivariate Data

A set of multivariate data is comprised of observations that are best arranged and handled as a matrix of sample values. The size of the matrix can be denoted by m and n, where m and n are respectively the number of samples and the number of features. In the case of a crack analysis in a solder ball joint, the crack is generated by a set of interconnected factors, including variation of temperature, mismatches in the Coefficient of Thermal Expansion (CTE) between the substrate and chip, material properties, and the size of the solder ball. It is vital to comprehend the interdependence among innumerable features in order for designers to accurately predict and draw the most accurate results. The multivariate data has joint distribution of every feature, where a distribution of any individual feature is said to have a marginal distribution.

1.4.2. Multivariate Data Analysis

Multivariate data analysis suggests that designers must investigate the interrelationships among features, classify samples of data into uniform categories, interpret the results explicitly, and predict the fundamental subsets of a sample. Regression analysis checks the relationship between two sets of variables. One set is comprised of dependent variables while the other linearly related set consists of independent variables. The most important objective of multivariate data analysis is to keep meaningful information after applying analysis to complicated data. However, it is tremendously challenging to use analysis to make precise and purely informative design decisions due to major two issues: uncertainty caused by high dimension and redundancy resulted from excessive correlation of such data. In this research, multivariate data analysis will reduce multidimensional data sets into small sized data sets in order to eliminate these causes. The compact data sets enhance regression or classification accuracy.

1.4.3. Curse of Dimension of Multivariate Data

The curse of dimension is a situation where high dimensions of data hinder the design of a model, the prediction of a response, or the optimization of an objective function. Increasing the number of data dimensions interrupts the problem solving process in terms of finding global optimal points of a set of data [6]. A curse of dimension also results from redundant and highly correlated multivariate data. Such properties can make a set of data more intricate, making the estimation of a response sometimes inaccurate and not meaningful. Complex data dimension can also cause excessive computational operations and overload [7]. A way to cope with the curse of dimension would not only improve the reliability of a model's response, but it would also help clear visualization of complicated data. As the dimension of input data increases, it becomes increasingly difficult to deduce an accurate response. If the number of samples of data is less than the number of features, the prediction will result in inaccurate and unreliable results, because the singular covariance matrix of data will derive an unreasonable output. Even if each set of multivariate data is explicitly estimated, integration of the results may be uninformative due to unpredictable error. Furthermore, the importance of dealing with high dimensional data cannot be overlooked, since reducing the dimension of data makes it simple to describe the interrelationship between sets of data and visualize its arrangement.

1.4.4. Uncertainty and Redundancy of Multivariate Data

Estimating multidimensional data can produce serious loss of information about that data due to uncertainty or redundancy, making it crucial to measure and evaluate these problems. The uncertainty can be calculated by entropy as explained in Equation 1.1. The

entropy is often used as a measure of the uncertainty from the probability distribution of a random variable, x [8]. The entropy of a random variable X is defined by

$$entropy = \sum_{x} p(x) \log_2 \frac{1}{p(x)} \tag{1.1}$$

where p(x) is the probability of each of the occurring variables, and entropy is defined as a measure of information. Entropy can be regarded as the uncertainty of random variable X. On the other hand, redundancy can measure the amount of correlation in data sets. If the data sets have high correlation between one another, they must be highly interdependent. Interdependence is strongly related with redundancy.

$$redundancy = \log_2 N - \sum_{x} p(x) \log_2 \frac{1}{p(x)} \tag{1.2}$$

where N is the total number of events. $\log_2 N$ is the maximum entropy. To measure redundancy of data sets, the opposite relationship of entropy should be employed. Figure 1.6 and Table 1.1 represent the relationship between redundancy and correlation in data sets.

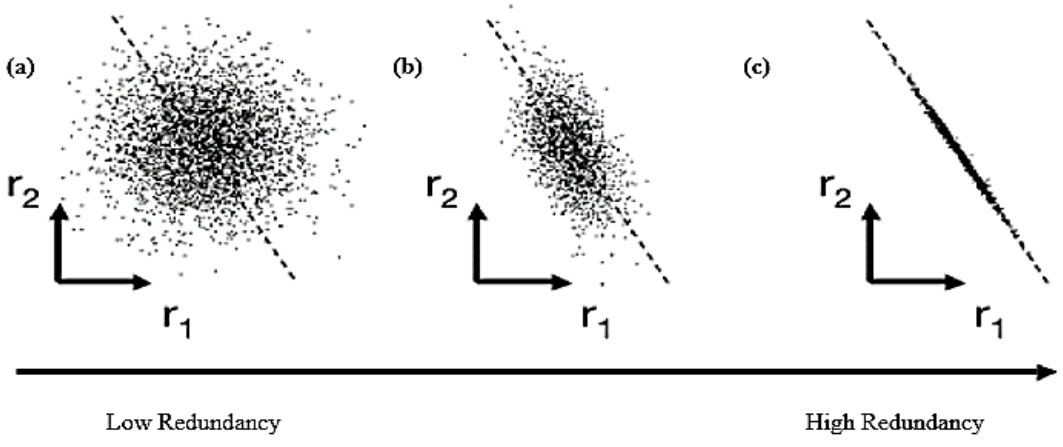


Figure 1.6. Relationship between Redundancy and Correlation [9]

Table 1.1. Comparison between Redundancy and Entropy

		Redundancy	**Entropy**
Correlation	Correlated	High	Low
	Uncorrelated	Low	High
Relationship with		Correlation	Uncertainty
Sensitivity of data size		Low	High

Specific interpretations regarding entropy will be introduced in Chapter 2. In a proposed framework of this research, the calculated value of entropy will evaluate the uncertainty of both raw, high-dimensional data sets and reduced data sets with low levels of complexity, while the redundancy comparison will check the irrelevance of original complex data sets and reduced data sets. If the compact data sets have lower values of entropy or redundancy than the raw data sets, this will indicate that the uncertainty or redundancy of the raw data has been reduced. On the other hand, if the raw data sets' entropy or redundancy is still higher than the values obtained for the compact data sets, much smaller sized data sets with low uncertainty or redundancy will be required.

1.4.5. Dependency of Multivariate Data

In multivariate data analysis, dealing with interdependence of data sets is crucial. Interdependence refers to correlation or dependency. When modeling acceptable correlations, the response of a model can be deduced accurately. However, marginal

distributions of data sets are randomly distributed and thus cannot be considered as an acceptable model. To solve this issue, a copula function will be utilized in the proposed framework.

1.5. Dimension Reduction

Dimension reduction is a method to reduce the number of dimensions of data. Dimension reduction is used to reduce the uncertainty and redundancy of data, to predict responses accurately [10], to prevent serious error, and to simplify data that retains crucial information. Indyk and Motwani (1998) expressed that the curse of dimension can be effectively resolved by dimension reduction [11]. There is a linear relationship between the number of dimensions and samples. Figure 1.7 represents the structure of a dimension reduction.

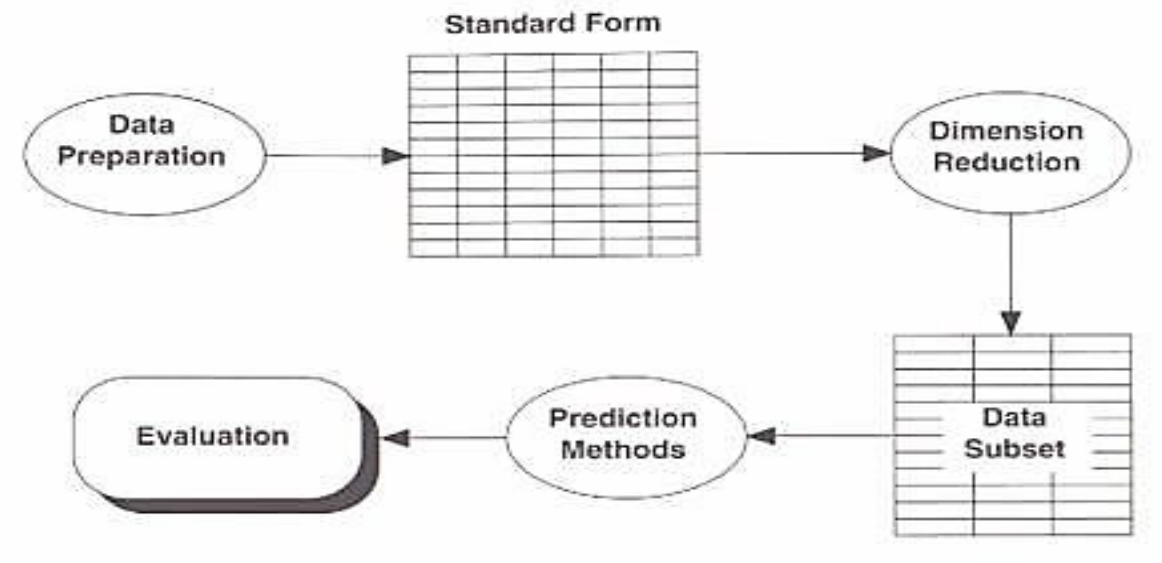


Figure 1.7. The Structure of Dimension Reduction [12]

In some practical cases, valuable information is buried in uninformative data. There are two main reasons leading this problem:

1. Subset(s) of multidimensional data have excessive size.
2. Subset(s) of multidimensional data are highly correlated with each other [13]

For these reasons, redundant and uninformative dimensions must be reduced or eliminated in order to extract useful information from the data. In the 1970's, there was already an endeavor to make a link between dimension reduction and classification [14]; now modern engineers have developed and broadened dimension reduction into multidisciplinary domains (e.g. machine learning, pattern recognition, text mining, classification, etc.). As the use of dimension reduction widened, engineers needed a method to correctly manage and truncate great volumes of multidimensional data. The progressive method that had been developed can be divided into two categories: feature extraction and feature selection. In feature extraction, new features are transformed from the original features. In feature selection, users choose a subset of features from the original. Figure 1.8 expresses various dimension reduction techniques. The fundamental description and properties of these two techniques will be briefly discussed in Sections 1.5.1 and 1.5.2.

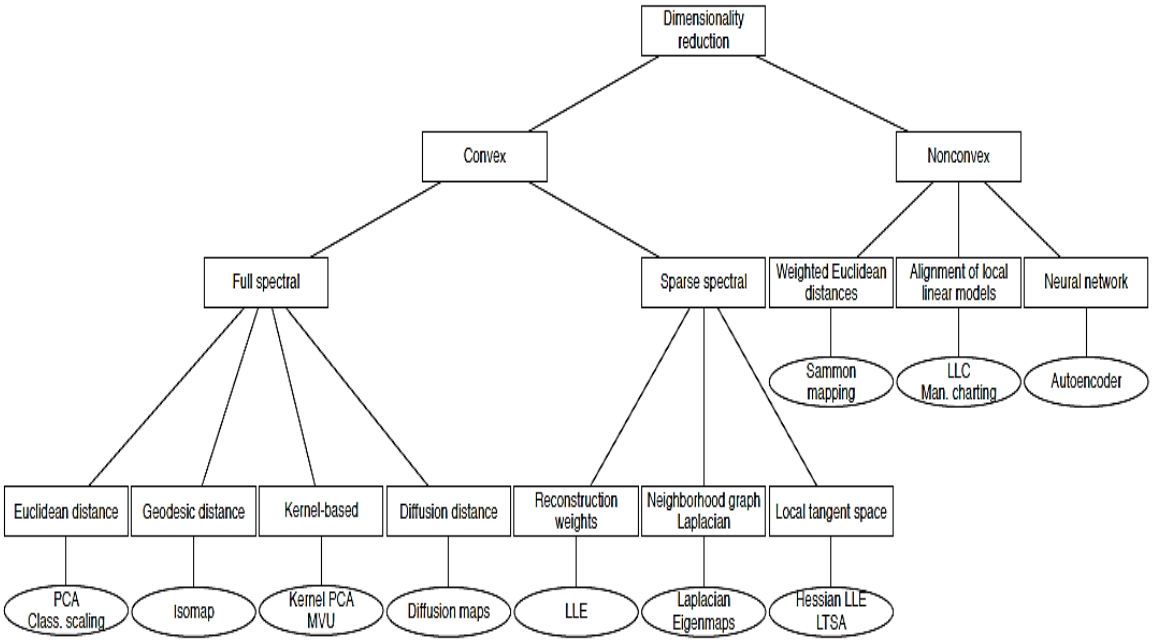
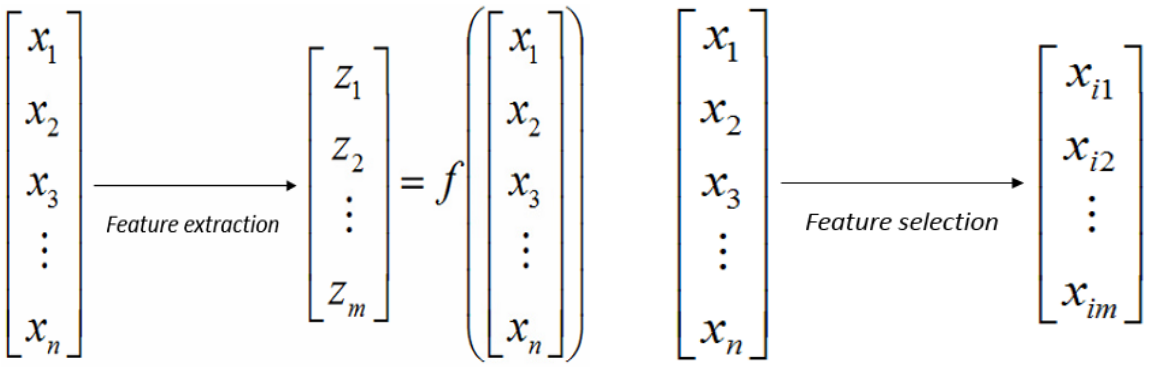


Figure 1.8. Dimension Reduction Techniques [12]

1.5.1. Feature Extraction

Feature extraction generates new features by transmuting them from initial data. For example, according to feature extraction, n dimensions of existing data are changed to m new dimensions ($n > m$). This process is called mapping. In this case, most of the information from the data should also remain after dimension reduction has been performed. Figure 1.9 (a) indicates that x_1, x_2, x_3, x_4, ⋯, and x_n are transformed to z vectors, showing that feature extraction employs dimension reduction to significant eigenvectors of the initial data to take on new features in z vectors. Subsequently, the new features that are reconstructed will have low uncertainty or redundancy. This research will introduce two special techniques of feature extraction: Principal component analysis (PCA) and Auto-encoder (AE).



(a) Feature Extraction Process (b) Feature Selection Process

Figure 1.9. Dimension Reduction Techniques

1.5.2. Feature Selection

Feature selection refers to a special case of feature extraction. As shown in Figure 1.9 (b), a feature selection program selects a subset of the original features of the data without any transformation and identifies the most significant subset of *n* features out of the *m* available ($n > m$). Feature selection is required in following situations:

1. Features are expensive to be calculated and obtained.
2. Engineers want to draw informative rules from the classifier.
3. Features are not numeric, such as strings.

The proposed framework in this research will utilize dimension reduction such as feature extraction and feature selection to reduce the high dimension of the initial data obtained from multidisciplinary engineering systems. Compact data sets obtained by the dimension reduction will have little uncertainty, redundancy, or complexity, and will draw accurate responses of the system by saving simulation costs.

1.6. Motivation

Due to the complexity of the multidisciplinary engineering systems, engineers attempt to simplify a model to reduce simulation burdens, such as cost and time, during the design process. However, the concise model is often inadequate for predicting accurate responses. In this research, an electro-mechanical engineering system will be considered in order to demonstrate these difficulties. For example, in the case of an analysis regarding the thermal stress on an electronic chip, the chip is doubtlessly assumed to have constant thickness or uniform temperature variation. However, if the chip has different hot spots from the electronic components, each different thermal distribution should be estimated and represented during modeling. Moreover, even though a system may be minutely modeled, the data of the system might not be easy to handle due to interdependency in the data. This complexity of data makes response prediction unreliable. Based on these problems, this research will develop a framework that can not only present most of the information of a system, but can also reduce the complexity of the problem in order to predict a response correctly. This will be achieved by utilizing a combination of several statistical approaches.

1.7. Research Questions and Hypothesis

Research Question 1

How can we accurately represent correlated random quantities of a complicated multidisciplinary engineering system?

A copula function can be utilized to accurately model complexities of parameters. The copula offers an excellent method for modeling the input complexities in the proposed

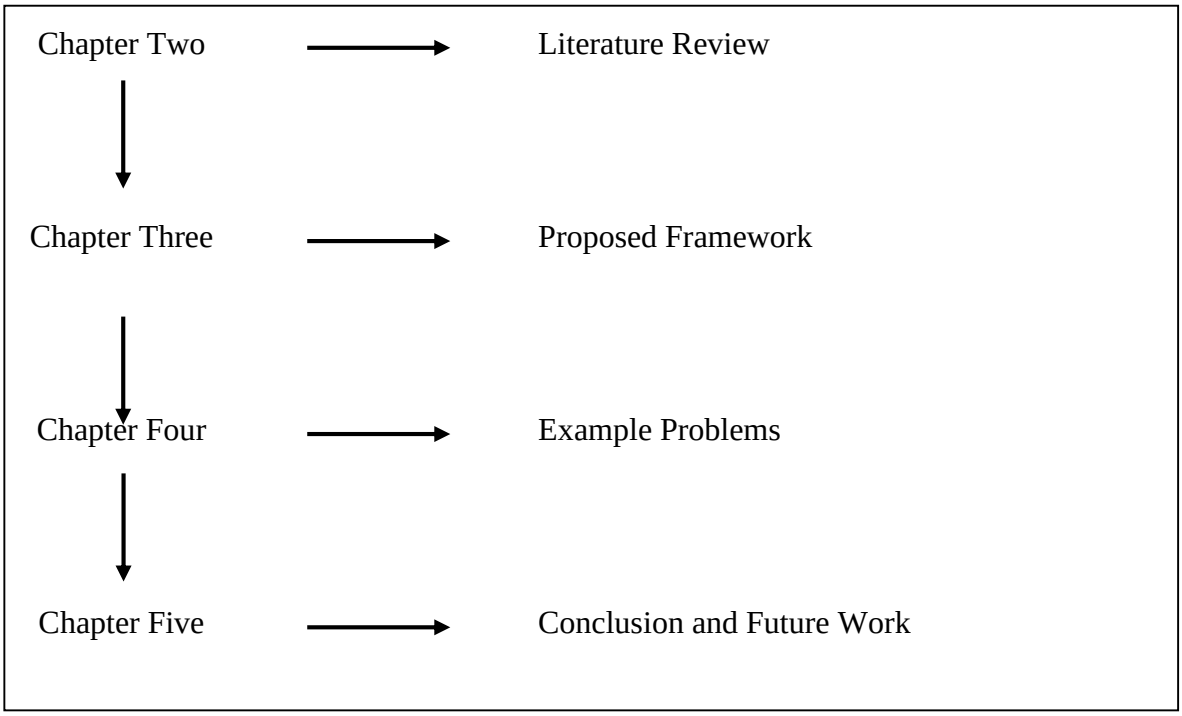
research, as it can combine assorted marginal distributions corresponding to the inputs in order to reproduce a joint distribution.

Research Question 2

How can we accurately predict multidisciplinary engineering systems' behavior while minimizing computational cost?

Multivariate system analysis produces inaccurate results largely because uncertainties or redundancies are created by complexities in data. These inaccuracies can be addressed by dimension reduction techniques. Feature extraction methods such as PCA and AE and feature selection methods can significantly reduce complexities in the analysis process.

1.8. Thesis Organization



Chapter 2 is a review of literature concerning design for complex engineering systems. This will include dimension reduction methods such as feature extraction and feature selection. It also includes a literature review of the copula functions to explain the random field of complicated engineering domains. Chapter 3 will provide details on the development of these methods and will explain how redundancy comparison and the entropy of information are used to determine whether complexities are reduced. Also, an algorithm, implemented in MATLAB, will be presented in this chapter. Chapter 4 will show how the proposed methods can be applied to three specific examples in multidisciplinary engineering domains: the first example shows how feature extraction methods reduce data redundancy and have paramount influence on response prediction by using a cantilever beam example. In the second example, a solder joint analysis will be described to explain feature selection. A stretchable antenna will be presented as the last example as a multidisciplinary engineering domain problem in order to highlight the efficacy of the developed framework.

CHAPTER 2

BACKGROUND

2.1. Representation of Dependency

Identifying the dependency of data sets not having Gaussian distribution is difficult yet essential. Properties of probability can be used to boost accuracy when estimating dependency. According to the probability of the data or random variables, if reasonable dependence in the variables is calculated, the random variables must be dependent. Such dependency of multiple random variables can be effectively obtained by linearly correlating coefficients and rank correlation [15]. Modeling or designing corresponding and accurate representations of the dependency between complicated data sets is also in strong demand. If dependency is insufficiently expressed, it is not feasible to obtain a realistic response. In order to address this issue, a copula function is introduced [16, 17].

2.1.1. Linear Correlation

By using the linear correlation coefficient, the dependency of random variables is assessed in Equation 2.1.

$$\rho(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{2.1}$$

where $\rho(X,Y)$ is a linear correlation coefficient of the random variables *X* and *Y*, $\mathrm{cov}(X,Y)$ denotes the covariance, and σ_X, σ_Y represents the standard deviations. Since $\mathrm{cov}(X,Y) = E(X,Y) - E(X)E(Y)$, the covariance of the random variables can be

expressed. If the random variables are not interdependent, $\rho(X,Y)=0$ because $\mathrm{cov}(X,Y)=0$.

2.1.2. Rank Correlation

Rank correlation depends on the ranks of data by using Kendall's tau (Ƭ) and Spearman's rho (ρ). It is highly recommended for engineers to use these variables because the ranks are not affected by scale estimators [18]. This is a distinctive property that can fit copula to any data. Spearman's ρ using the empirical cumulative distribution function of data can derive the correlation of transformed data. It is independent of marginal distributions. Kendall's Ƭ is also independent of marginal distribution. If there are *n* data, $\frac{n(n-1)}{2}$ pairs of data can be obtained. All pairs can be divided into concordant or discordant data. If the value of Kendall's Ƭ is 1, 0, and -1, it means that the pairs can be synthesized by the perfect correlation copula, independence copula, and perfect anti-correlation copula, respectively.

2.1.3. Copula

A copula is a mathematical function that can disassemble a set of variously sized multivariate distributions into a one dimensional marginal distribution [19]. In other words, a bulk of numerous marginal distributions, or a joint distribution, is divided into individual distribution forms [20]. For particular disciplines, a copula is used reversely to build a set of multivariate distributions, or joint distributions, by combining each of the different marginal distributions that are asymmetric and tail heavy.

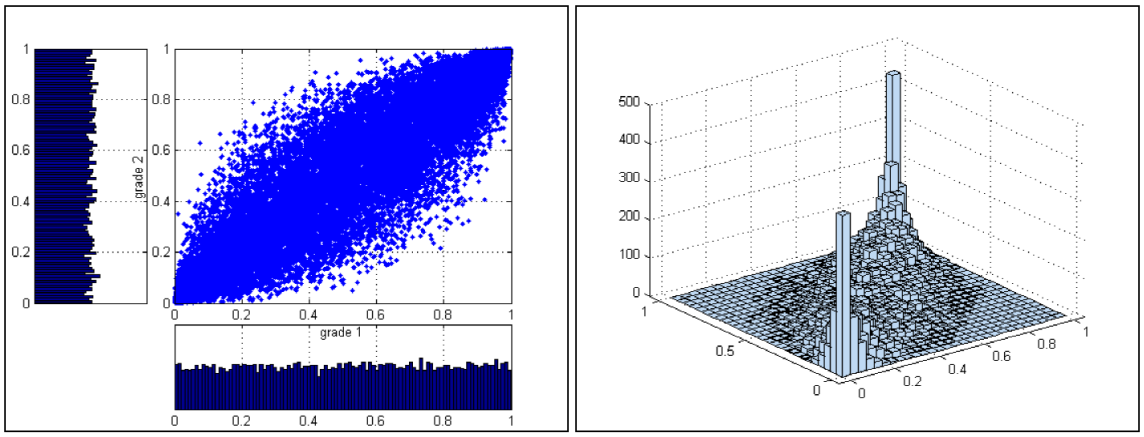
C is a Copula if $C:[0,1]^2 \to [0,1]$ and

(a) $C(0,u)=(v,0)=0$

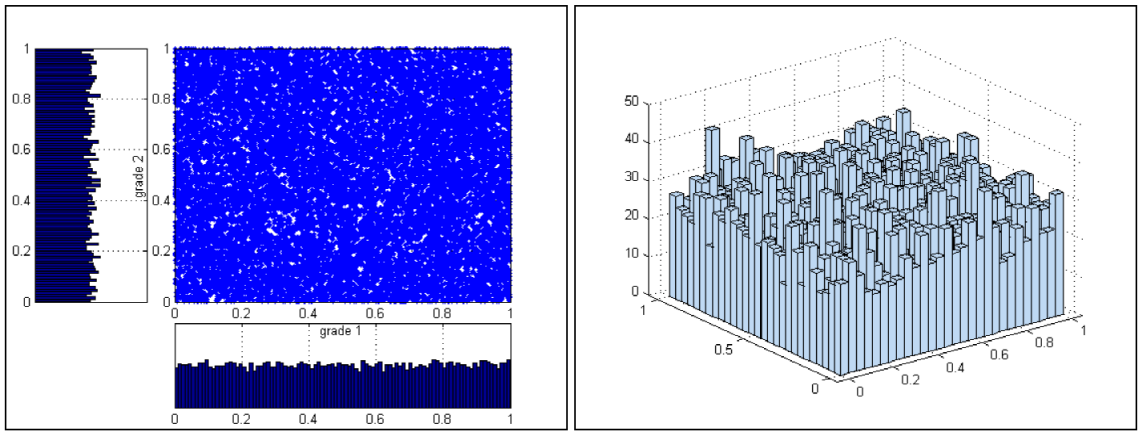
(b) $C(1,u)=(u,1)=u$

(c) $C(v_1,v_2)-C(v_1,u_2)-C(u_1,v_2)+C(u_1,u_2)\geq 0$ for all $(v_1 \leq v_2),(u_1 \leq u_2)$ (2.2)

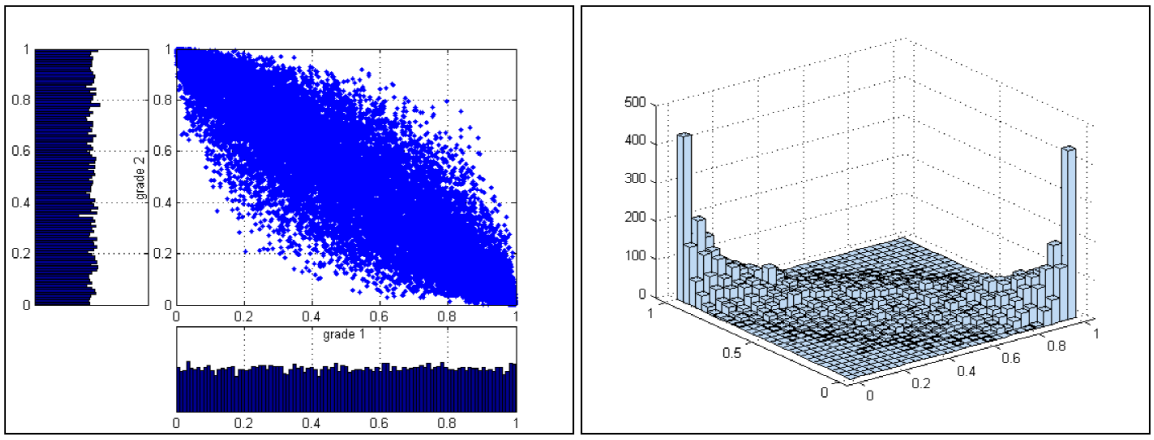
where u and v represent the probability of marginal distributions on $[0,1]^2$ [21]. The property (a) indicates if the probability of marginal distributions regarding the results is zero, the probability of joint distribution should be zero. The property (b) expresses that if the marginal distributions' probability in terms of the results is one, the joint distributions' probability must be one. Property (c) is a two dimensional function in terms of an increasing one dimensional function. Figure 2.1 shows a simulation of Gaussian copula with different correlations. Each figure estimates each relation between marginals and random variables such a grades and pdf of a copula.



(a) Correlation Parameters of 0.9



(b) Correlation Parameters of 0



(c) Correlation Parameters of -0.9

Figure 2.1. Simulation of Gaussian Copula [22]

Compared with the general way to measure dependency which uses a correlation coefficient, using a copula is much more effective due to the fact that correlation coefficients fail to estimate interdependence among non-linear variables and measure the dependence between variables when the standard diviation of one variable becomes very large. Copula, on the other hand, can be used for any relation between variables [23]. Figure 2.2 presents the difference between using a correlation coefficient and copula. Even though the correlation coefficient is good for checking for constant dependency, copula can represent any variable dependencies without the limitation correlation coefficients have.

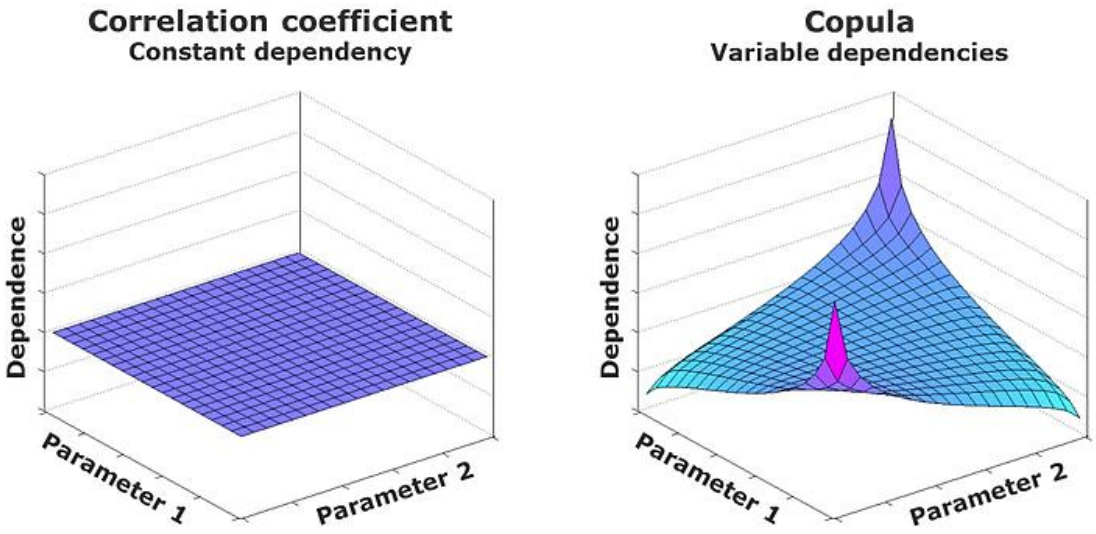


Figure 2.2. Difference between Correlation Coefficient and Copula [23]

Family of Copula

As shown in Figure 2.3, there are many types of copula functions: the Gaussian copula for linear correlation and the student or *t* copula of elliptical copula, the Archimedean copula, the Frank copula, the Clayton copula, and the Gumbel or Logistic copula [16].

1. The Gaussian copula: As the most generally used copula function, the Gaussian copula is constructed from a multivariate normal distribution with probability integral transformation. It has an elliptical symmetric shape.

$$C_\rho(x,y)=\frac{1}{2\pi\sqrt{(1-\rho^2)}}\int_{-\infty}^{\Phi_1^{-1}(x)}\int_{-\infty}^{\Phi_1^{-1}(y)}\exp\left(-\frac{h^2-2\rho hk+k^2}{2(1-\rho^2)}\right)dhdk$$

$$\Phi_1(x)=\frac{1}{2\pi}\int_{-\infty}^{x}\exp\left(-\frac{1}{2}s^2\right)dh \tag{2.3}$$

where ρ represents a linear correlation coefficient, x and y indicate marginal distributions, h and k refer to copula parameters, and Φ represents a standard univariate Gaussian distribution function.

2. The Student or *t* copula: It has radial symmetric shape and the upper and lower tail dependencies are identically characterized. The tail dependency is decided by degrees of freedom and correlation.

$$C_{v,\rho}(x,y)=\int_{-\infty}^{t_v^{-1}(x)}\int_{-\infty}^{t_v^{-1}(y)}\frac{1}{\sqrt{2\pi(1-\rho)}}\left(1+\frac{x^2-2\rho xy+y^2}{v(1-\rho^2)}\right)^{\frac{v+2}{2}}dxdy \tag{2.4}$$

where v expresses degree of freedom and t_v refers to a standard univariate *t* distribution function.

3. The Archimedean copula: The Archimedean copula is an associative class of copulas [24]. This copula is commonly used in practical applications since it can readily manage high multidimensional data sets with only one parameter when estimating and designing dependency in the data sets.

$$C(x,y) = \varphi^{-1}\left(\varphi(x) + \varphi(y)\right) \tag{2.5}$$

where φ represents the generator of copula.

4. The Frank copula (Kendall's Ţ): This copula enables engineers to effectively gauge certain cases with different distribution types. For example, if two marginal distributions have beta and lognormal distribution, the Frank copula can make a joint distribution by using a parameter that decides the level of dependence between the marginal distributions.

$$C(x,y|\alpha) = \frac{-1}{\alpha}\ln\left(1 + \frac{(e^{-\alpha x}-1)(e^{-\alpha y}-1)}{(e^{-\alpha}-1)}\right) \tag{2.6}$$

where α represents the Frank copula parameter.

5. The Clayton copula: This is one of the Archimedean copulas with lower tail dependence.

$$C(x,y|\delta) = \left(x^{-\delta} + y^{-\delta} - 1\right)^{-1/\delta},\ \delta \geq 0 \tag{2.7}$$

where δ indicates the Clayton copula parameter.

6. The Gumbel or Logistic copula: This is also a kind of Archimedean copula but with upper tail dependence.

$$C(x,y|\gamma) = \exp\left(-[(-\ln x)^{\gamma} - (-\ln y)^{\gamma}]^{1/\gamma}\right),\ \gamma \geq 1 \tag{2.8}$$

where γ represents the Gumbel copula parameter.

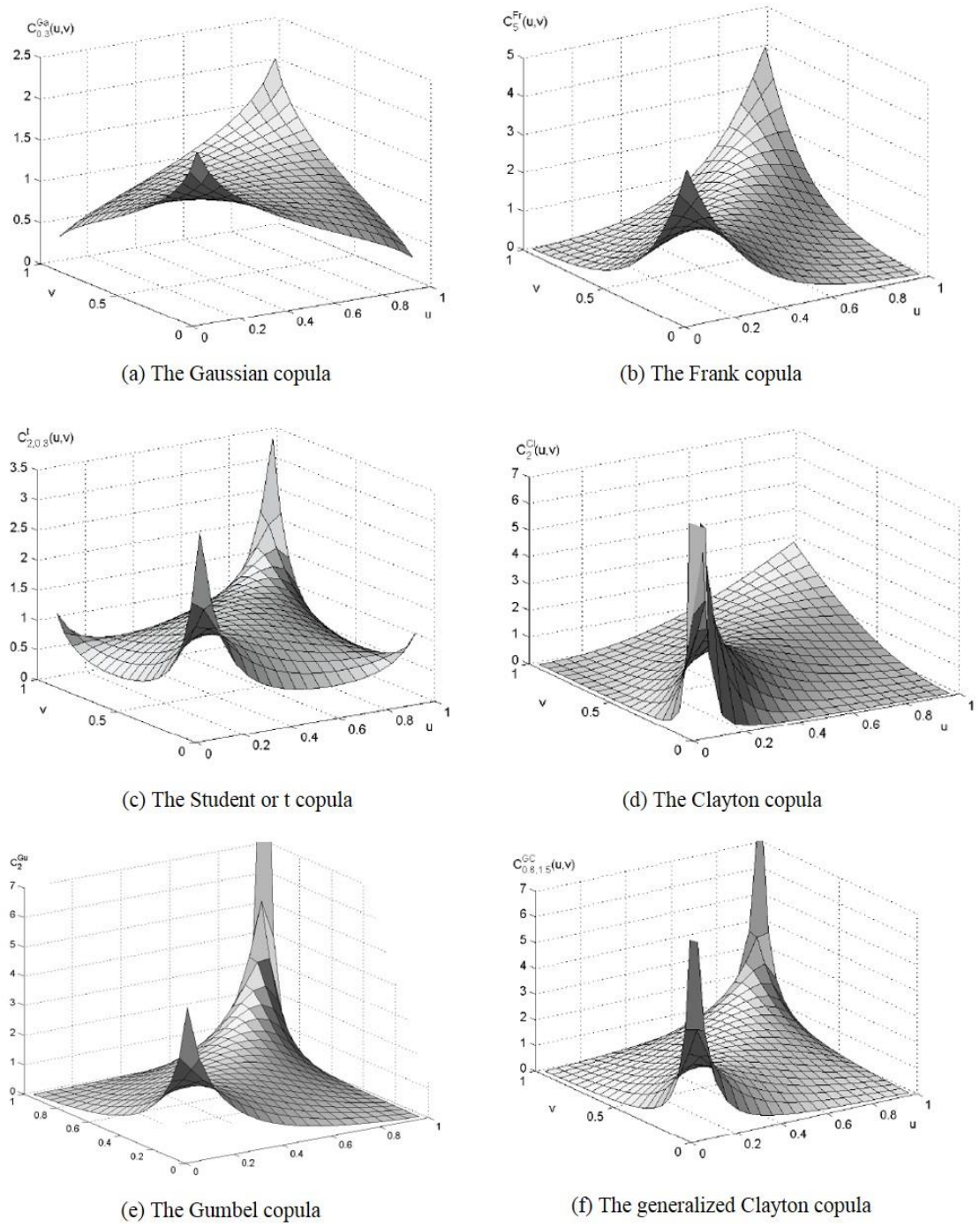


Figure 2.3. Family of Copula Densities [18]: correlation coefficient 0.3 and the t-copula has 2 degrees of freedom

Sklar's Theorem

In 1959, Sklar introduced the definition of copula to decompose a joint distribution with various dimensions into two marginal distribution functions [20]. He defined H_{xy} to be a joint distribution with marginal distributions F and G [17].

$$C:[0,1]^2 \rightarrow [0,1]$$

$$H_{XY}(x,y) = C(F(x),G(y)) \tag{2.9}$$

In Equation 2.9, C refers to a copula and both F and G are distribution functions. The function H is a joint distribution, and the set of F and G are marginal distributions. Two separated marginal distributions can be combined into a joint distribution with any copula function,

$$C(u,v) = H\left(F^{-1}(u),G^{-1}(v)\right) \tag{2.10}$$

A copula function can be obtained by Equation 2.10. The copula is not influenced by any variation of data. For example, without any transformation, a copula function is utilized for not only the joint distribution of x_1, x_2, x_3, x_4, $\cdots$, and x_n, but also that of ln x_1, ln x_2, ln x_3, ln x_4, $\cdots$, and ln x_n [25].

2.2. Information Theory

In 1948, Shannon introduced the information theory to measure information logarithmically in telecommunicating engineering [8]. Information theory was specialized in only telecommunication; however, application of the theory has extended into various engineering domains based on probability distributions. This theory uses a random process

to quantify cumulative information in multivariate probability density functions [26, 27]. This quantity of information can be called entropy. In general, entropy is strongly connected with the rareness, uncertainty, randomness, or redundancy of a random variable [26]. Thus, entropy is often used to measure information or uncertainty of a random variable [28]. The properties mentioned above are extended to express mutual information, which quantifies how correlated two random variables are. In other words, these properties state by what amount the uncertainty of a random variable can be reduced by obtaining information about another variable in the data set.

2.2.1. Entropy

Entropy can be interpreted three ways to obtain different sorts of information. Differing levels of entropy can reflect the amount of information, uncertainty in a random variable or vector, and dispersion in the probability distribution [29].

1. The amount of information: A rarely occurring event yields more information than frequently occurring event.
2. The uncertainty in a random variable or vector: Common or certain events reduce uncertainty, hindering the prediction of response. Therefore, events with uncertainty will have larger amounts of entropy.
3. The dispersion in the probability distribution: Lower dispersion indicates smaller amount of entropy.

Thus, entropy captures uncertainty, randomness, or redundancy of a random variable [8, 26]. Entropy of a discrete random variable X is described as

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x),\ p(x) = \frac{n}{N} \tag{2.11}$$

where *N* refers to number of total events and *n* indicates when *x* occurs. The choice of a logarithmic base corresponds to the unit used when measuring information. For example, if the base is two, the resulting units will be in binary, while if the base is ten, the units may be in decimal. Thus, a logarithmic base can take arbitrary number. However, Shannon defined the unit of entropy as a bit based on signal analysis. A bit has two cases: zero and one. Zero can be denoted as "no" and one as "yes". By using zero and one, the entropy of all events can be effectively and simply estimated. In the case of flipping a coin, bits should equal to one with the base two. The base ten cannot be used because the number of total events is two. Using this logic, the base 2 can estimate the entropy of any event since it is the smallest base. According to equation (1), entropy will be influenced by the value of probability, not the variation of the random variable. Therefore, if the probability is zero, entropy should be zero. Moreover, $H(X)$ is always greater than zero because probability values and $\log_2 \frac{1}{p(x)}$ are always greater than zero [28].

2.2.2. Joint Entropy and Conditional Entropy

Based on random variables, joint entropy can be expressed as

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 p(x,y) \tag{2.12}$$

As shown, if discrete random variables *X* and *Y*, or their joint probability distribution $p(x,y)$ are given, joint entropy can be calculated.

For $(X,Y) \sim p(x,y)$ and a given random variable, entropy should have the conditional property.

$$H(Y|X) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 p(y|x) \tag{2.13}$$

Both joint and conditional entropy are related to entropy by the chain rule.

$$H(X,Y) = H(X) + H(Y|X) \tag{2.14}$$

2.2.3. Mutual Information

Even if two random variables are independent, their dependency cannot be accurately measured by conditional entropy. $H(Y|X)$ indicates that X is greatly influenced by Y or by a small $H(Y)$. Therefore, dependency should be evaluated by mutual information [30]. Mutual information refers to the reduction of randomness or uncertainty of a random variable when the other variable is known.

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y) \tag{2.15}$$

Mutual information can be maximized to minimize joint entropy and be used to include all individual input entropy.

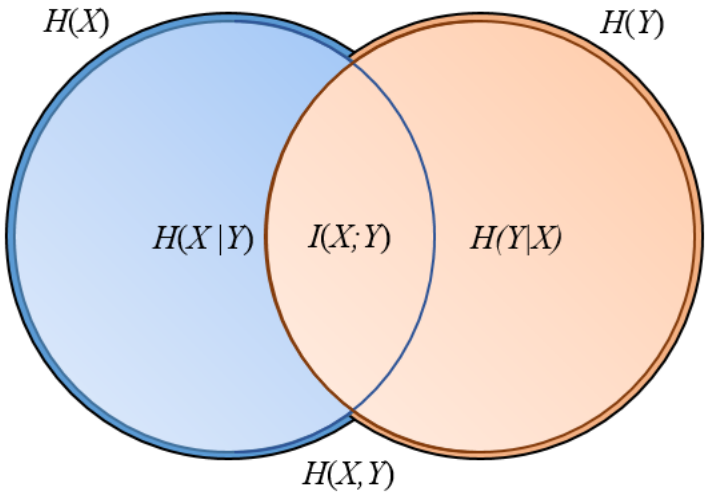


Figure 2.4. Venn Diagram of Information Theory

2.2.4. Relation between Each Terminology

Figure 2.4 shows the relationship between each term. Entropy is a measure of information or uncertainty of a random variable *X* and joint entropy can be expressed as the uncertainty of random variable *X* and *Y*. When a random variable is known, the uncertainty of the other variable is known by conditional entropy. Mutual information states that the amount of random variables known by a random variable *X* corresponds to the uncertainty, randomness, or redundancy of *Y*. If random variables *X* and *Y* are the same, their joint entropy is equal to not only the entropy of *X* or *Y*, but also to their mutual information. On the other hand, if the random variables are independent, their mutual information should be zero, and the joint entropy is the sum of the entropy of *X* and *Y*.

2.3. Dimension Reduction

During the integration of various disciplines, the number of observations examined exponentially increases. For this reason, efficient multivariate data analysis is commonly required to get accurate and realistic responses of systems. However, multivariate data

results in a curse of dimension, as stated in Chapter 1. High dimension has uncountable features in data, and it causes uncertain and inaccurate predictions or overfitting in regression and classification [31]. Moreover, if the features are highly correlated, they may be used to represent the same property, leading to high redundancy and simulation inefficiency [7, 32]. In order to overcome these problems, dimension reduction is commonly used. It can be generally divided into feature extraction and feature selection. In the following subsections, these two categories will be explained in detail.

2.3.1. Feature Extraction

Feature extraction transforms initially high dimensional data to generate new features with a reduced size. As briefly discussed in Chapter 1, feature extraction enables N dimensions of existing data to be changed to have M dimensions. This process is defined as mapping. After dimension reduction, specifically feature extraction, the data should contain most of the information from the original data [33]. Two of the more well-known methods for mapping that will be discussed are the Principal Component Analysis (PCA) and Auto-Encoder (AE).

Introduction of Principal Component Analysis (PCA)

In 1901, Pearson started to conduct research on PCA by focusing on a line or plane to represent data on p dimensions and came up with the geometric optimization problem as a solution [34]. Hotelling stated that there are sets of independent variables in low dimensions that make p variables, and the independent variables are composed of principal components [35, 36]. Based on the theory of Hotelling, the idea that the principal

components' alternative derivations are most likely to estimate the original components was introduced by Girshick [37, 38].

Basic Steps of PCA

In order to extract the most important features, a change of basis should be considered. *X* and *Y* have m (the number of observations, or cases) by n (the number of features) matrices composed by the linear transformation

$$PX = Y \tag{2.16}$$

where *X* refers to an original data set and *Y* refers a re-representation of *X*. As a rotation and scale matrix, *P* transforms *X* into *Y*. Each coefficient of *Y* is a projection onto each row vector of *P*. $P = \{P_1, P_2, \ldots, P_m\}$ denotes new basis vectors to express the columns of *X*. This means that the row vectors of *P* should be the most principal components of *X*. If the basis of *P* is orthonormal, *P* acts as a rotation matrix. As already discussed in "Redundancy", zero covariance should correspond to uncorrelated or independent data. After setting up this assumption, PCA starts to choose and save a normalized direction where *X*'s variance is at its maximum. It repeatedly selects and memorizes directions where the next variance is at its maximum. At this point, the direction is perpendicular to the previous directions due to the orthonormal property [39]. The selected directions' data set is called the principal components.

Analysis of PCA

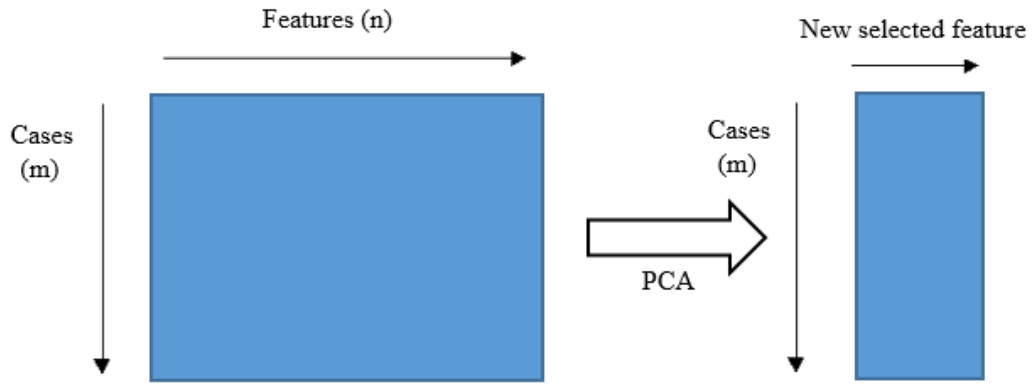


Figure 2.5. Schematic Model of PCA

To reduce the dimension of the data, PCA uses eigenvector decomposition. According to Equation 2.17, a covariance matrix can be obtained by a *P* matrix with the orthonormal property

$$C_Y = \frac{1}{n-1} YY^T \quad where\, Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \tag{2.17}$$

In terms of orthonormal matrix *P* from Equation 2.16, the covariance matrix is represented as

$$C_Y = \frac{1}{n-1} YY^T = \frac{1}{n-1} (PX)(PX)^T = \frac{1}{n-1} PAP^T \tag{2.18}$$

where in $A = XX^T$, *A* is symmetric. *A* can be also denoted by a diagonal matrix, *D*, and *A*'s eigenvectors matrix, *E*, as EDE^T. At this point, since matrix *A* is comprised of eigenvectors

from the rows of the P matrix, $P = E^T$, $A = P^T DP$. Due to the property of orthonormal matrixes, the inverse of P should be equal to its transpose. Finally, a covariance matrix C_Y is measured as follows,

$$C_Y = \frac{1}{n-1} PAP^T = \frac{1}{n-1} P(P^T DP)P^T = \frac{1}{n-1} D \tag{2.19}$$

Therefore, X's principal components should be the eigenvectors of $PC_x = \frac{1}{n-1} XX^T$, and each diagonal variable of C_Y will be the variance of X. Principal components can be also determined by singular value decomposition (SVD). From X_i of n dimension, a covariance matrix is estimated as follows,

$$C = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(X_i - \overline{X})^T}{(n-1)} \tag{2.20}$$

$$C = U\Lambda U^T = \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix}^T \tag{2.21}$$

in matrix form, U represents the matrix of orthogonal eigenvector, u_i, and Λ refers to the matrix of eigenvalue, λ_i. Equation 2.21 estimates the eigenvalue of a covariance matrix. Figure 2.6 depicts how principal components in terms of large values of variance can be determined.

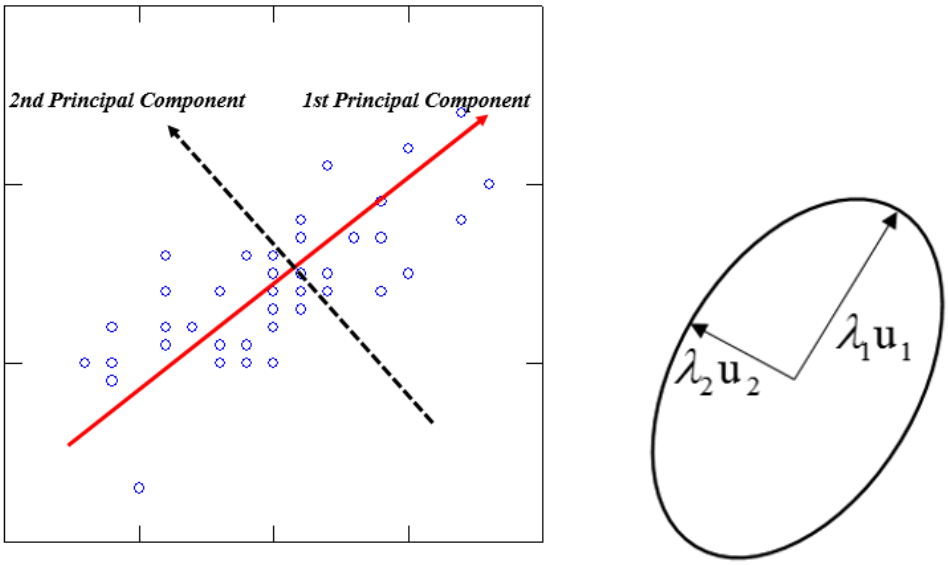
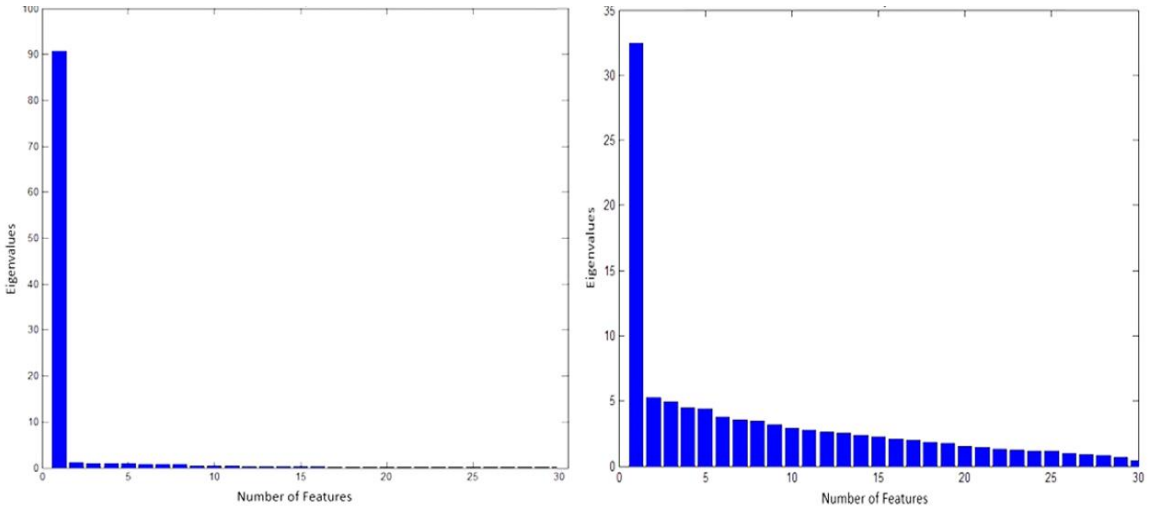


Figure 2.6. Process of Finding Principal Components

Between *n* eigenvalues, the first largest eigenvalues are selected ($\lambda_1, \cdots, \lambda_n$ where $n > m$). With the selected eigenvalues, an eigenvector $W = [u_1 \quad u_2 \quad \ldots \quad u_n]$ can be used to calculate $Y = W^T X$. After getting all principal components, it is important to decide how many components should be selected. There are many of methods to do so: cumulative percentage of total variation, partial correlation, Horn's procedure, or Kaiser's rule [9]. However, the scree plot is the easiest. The scree plot presents the eigenvalues corresponding to the principal components in descending order. The *X* axis refers to the number of features while the *Y* denotes the magnitude of the eigenvalues for the principal components. The rate of decline is fast at first and then levels off. The scree plot is shaped like a curved bend and expresses the maximum number of principal components to extract. Figure 2.7 shows that 1 and 21 eigenvalues are selected to represent 91% of original data. This means that 30 features can be truncated to 1- or 21-features with a 9% loss of information of the original data.



(a) 1 Eigenvalues (b) 21 eigenvalues

Figure 2.7. Scree Plot

Limitations of PCA

PCA uses eigenvectors from a covariance matrix. The accuracy of PCA can only be enhanced when the data sets have a Gaussian distribution. This is because in non-Gaussian distributed data, the largest variance cannot reflect a significant axis [9]. Moreover, PCA only considers the directions with the largest variance to be the most important components, but there is no guarantee that the directions sorting the principal components are the most accurate [40]. PCA also uses the orthogonal transformation matrix and linear combinations about the original data, but its properties are not useful when analyzing non-linear data sets [9]. Finally, PCA is only valuable if the original data sets are correlated. If the sets are uncorrelated or independent, PCA is not efficient in extracting the principal components. In cases of non-Gaussian distribution or non-linear analysis and uncorrelated data sets, auto-encoder will be used in this study.

Introduction of Auto-Encoder (AE)

Auto-Encoder (AE) is a special type of artificial neural network (ANN). ANN imitates the human neuron system so to predict values or patterns of an output based on those recognized from the input. ANN will specifically be discussed in Section 2.4.1. The purpose of AE is reduce dimension by transforming inputs into outputs. In 1980, Hinton and the PDP group first indicated a concept for AE. This concept was extended to unsupervised learning by Hebb and Oja [41, 42]. In the 2000's, AE describing deep architecture is introduced with various notions such as Restricted Boltzmann Machines (RBMS), stacked AE, and trained bottom up [27, 32, 43, 44]. Based on these works, AE is used for analysis on complicated nature phenomenon that have multiple dimension and non-linear data.

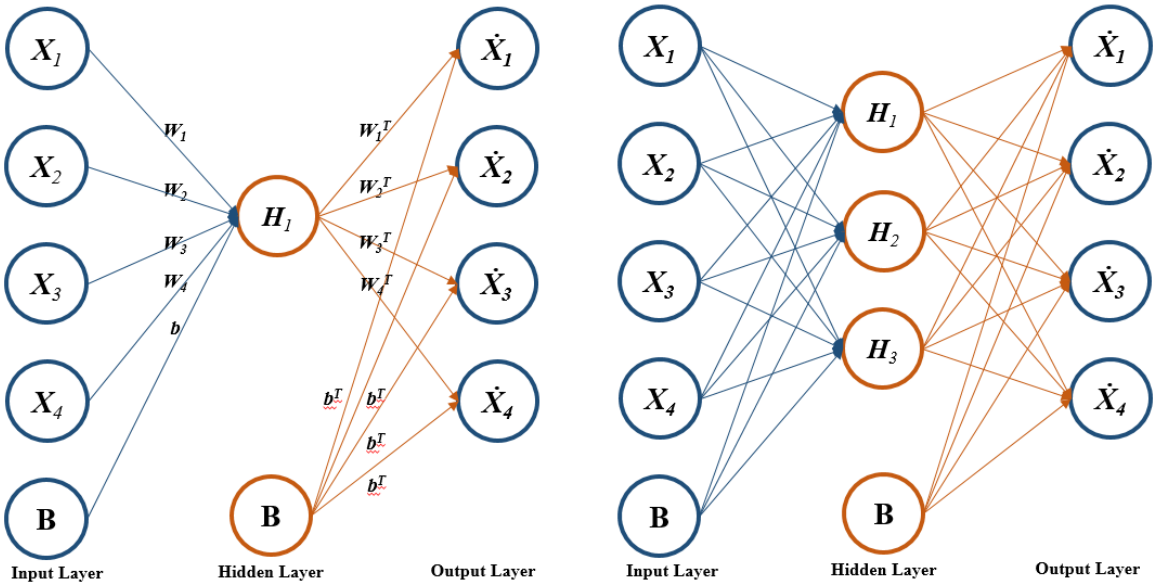
Supervised Learning vs Unsupervised Learning

AE is based on an unsupervised learning algorithm, even though AE has a certain case of artificial neural network that has the supervised learning property. Unsupervised learning plays a significant role in reducing high dimensions of data [32]. In this section, the definition of supervised and unsupervised learning algorithm is explained.

- Supervised learning: Given input and target data, the algorithm finds specific patterns in the original input data. These patterns make a connection between the input and target values. Moreover, the patterns are used to predict or classify target values [45].
- Unsupervised learning: This is only based on input data since no target data is given. It discovers a certain relationship from the input data to find intrinsic structures. It can also extract significant information from data without being

controlled by the attributes of the particular data. Thus, unsupervised learning can create new lower dimensions that include the most important information. In other words, unsupervised learning can be used to truncate high dimensions of data [32]. Clustering, probability distribution estimation, and using association in data are also other examples of unsupervised learning.

Steps of Auto-Encoder



(a) Single Auto-Encoder (b) Formation of Auto-Encoder

Figure 2.8. Schematic Model of Auto-Encoder

AE is composed of neural networks with multiple layers. Since AE is a technique that uses an unsupervised learning algorithm, the target data of the network should be regarded as input data. Thus, Figure 2.8 shows that AE has three layers: the input, hidden, and output layer. The circles with B in the bottom of each layer indicate both bias units and the intercept term. $\dot{X}$ represents the reconstruction of X. The purpose of training neural

networks is to minimize the difference between units of an input and an output layer. In other words, the training aims to reduce the mean squared error of reconstruction. In Figure 2.8, the number of units on a hidden layer (3) is smaller than that of other layers (4). Due to the hidden layers, multilayer AEs are presented with a bottleneck shape. Since the reduced number of hidden units is used and the low dimensional representations are obtained by extracting the unit values, AE can have a property in terms of dimensionality reduction. AE shares the weight values and biases between the previous and next layers [46]. As shown in Figure 2.8 (a), each arrow contains weight and bias units. AE is based on a feed-forward condition. Thus, hidden units and output units can be calculated as:

$$H = \sum_{i=1}^{n} W_i x_i + b \tag{2.22}$$

$$Y = \sum_{i=1}^{n} W_i^T H_i + b^T \tag{2.23}$$

where W, x, b denote weights, input units, and bias, respectively. In Equation 2.23, each W^T and b^T represents the transpose of W and b. The squared-error cost function can calculate the reconstruction error, RE, by:

$$RE = \frac{1}{2}\left\| Y - X \right\|^2 \tag{2.24}$$

where Y and X denote units of output and input, respectively. AE can be divided by de-noising, stacked, and sparse AE:

1. De-nosing AE: Commonly used to avoid overfitting by adding noise to the input data when the system is trained.

2. Stacked AE: Necessary to represent a deep network. Each layer of stacked AEs is trained one by one and utilized as the input for the following layer.
3. Sparse AE: Allows the network to find intrinsic structures instead of learning how to map between input and output or adding noise. Furthermore, by applying parsimony constraints to the hidden unit, it can be used as a technique of dimension reduction.

Spares Auto-Encoder (SAE)

AE has a distinguished condition, such as parsimony or sparsity conditions, in terms of reducing the number of hidden units. It is directly related with dimension reduction. AE originally considers output data to be equal to input data, $Y \approx x$. From this relation, it is apparent that AE tries to learn an approximation of the identity function. By applying certain constraints and reducing the number of hidden layers of the identity function, AE takes on a bottleneck shape. This shape indicates that the original dimensions of the data are truncated. The certain constraint is a sparsity constraint. A single neuron will be active when its output value is close to one. On the other hand, for a zero output value, the neuron will be inactive. The sparsity constraint aims to make all neurons inactive. The average activation of hidden layer is defined as follows,

$$\tilde{\rho}_j = \frac{1}{n}\sum_{i=1}^{n}\left[a_j^{(k)}(x^{(i)})\right] \tag{2.25}$$

where $a_j^{(k)}$ refers to the activation of hidden unit *j* and hidden layer *k*. When the activation values of hidden layers are close to zero, ρ can be regarded as a sparsity parameter. The relationship between the average of activation values, $\tilde{\rho}_j$, and the sparsity parameter is

$$\tilde{\rho}_j = \rho \tag{2.26}$$

To get more accurate results, it is essential to add an additional penalty term. In this case, *s* represents the number of hidden units, and the penalty term is represented by

$$P = \sum_{j=1}^{s} \rho \log \frac{\rho}{\tilde{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\tilde{\rho}_j} \tag{2.27}$$

By definition of Kullback-Leibler (KL) divergence, Equation 2.27 is replaced by $KL(\rho \| \tilde{\rho}_j)$ [27]. KL divergence seeks to find the minima of the penalty term to get $\tilde{\rho}_j = \rho$.

2.3.2. Feature Selection

While feature extraction focuses on transforming an original data to take on new, reduced data, feature selection focuses on finding a subset of the original high dimension data such that the small sized subset only takes meaningful features. Given a set of original data, *n*, the number of selected subsets, *m* should be finite and still informative. This process of transformation is represented as a dimension reduction algorithm. If the selected subset does not have enough information from the original data, the subset's significance will be diminished. There are many feature selection techniques using mutual information, single variable classifiers, or even genetic algorithm [47]. However these techniques are costly. Furthermore, analysis of the distributions often have unreliable results when the information for all of the data's distributions are not fully given. Although an analysis with the mean and variance is not considered to be the best solution in this situation, it results

in accurate results simply and quickly. Figure 2.9 indicates how the analysis using mean and variance is explained.

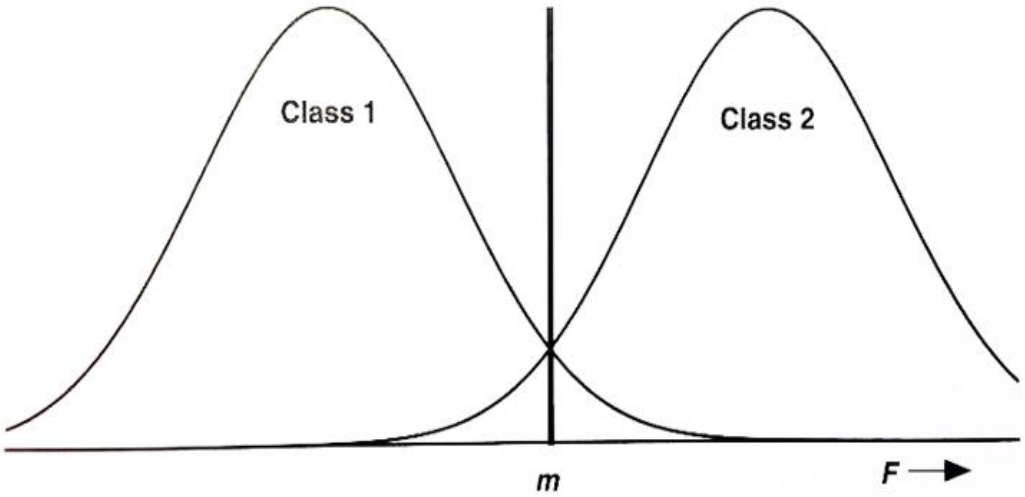


Figure 2.9. Seperating Two Classes Using Means [48]

It is easy to analyze the importance of the two classes by their mean. The two distributions have largely different means and variance between each other. Thus, classes 1 and 2 are regarded to be relatively independent and both classes will be selected as significant data sets. In short, if the means are far apart, the features' interest will be increased; it is distinguished between the two classes. A well-known feature selection technique, Independent Features Test, will now be introduced.

Introduction of Independent Features Test (IndFeaT)

In 1998, Weiss and lndurkhya suggested a simple hypothesis test to determine which predictor has an informative connection with the target data [48]. This technique was meant to quickly and effectively abandon uninformative features. IndFeaT assumes that the target data is categorical, or that all features of data belong to one of two classes.

Weiss made the threshold to be the significance value for each class. Through this value, selected features can be estimated to see if each feature is informative or not. Equation 2.28 solves for the feature scoring used in this method.

$$IF(F) = \frac{\left|mean(A) - mean(B)\right|}{\sqrt{\frac{\text{var}(A)}{n_1} + \frac{\text{var}(B)}{n_2}}} \tag{2.28}$$

where *IF* denotes informative features' scoring, and each *A* and *B* represents a data set corresponding to feature (*F*)'s values. n_1 and n_2 refer to the number of features in the classes. Feature scoring can be estimated by the summing all the scores. If a single score is of class *A*, the other classes are estimated as class *B*. Thus, Equation 2.29 defines feature selection performed to be:

$$IF(F) = \frac{\left|mean(A) - mean(B)\right|}{\sqrt{\frac{\text{var}(A)}{n_1} + \frac{\text{var}(B)}{n_2}}} > sig \tag{2.29}$$

where *sig* represents significance value. IndFeaT will select some informative features by comparing them to the significance value. Weiss suggested informative and important features to be higher than the significance value since higher values can choose more reliable features. In Figure 2.10, a data set has 12 features. After applying IndFeaT, the features can be selected by comparing them to the significance value. In this example, the value of 2 is used as the selection criterion. The features that are under the significance value are discarded and the rest of the 10 features are regarded to be informative to the subset. In general, Weiss suggests that features are informative at the significance value of 2 or higher [48].

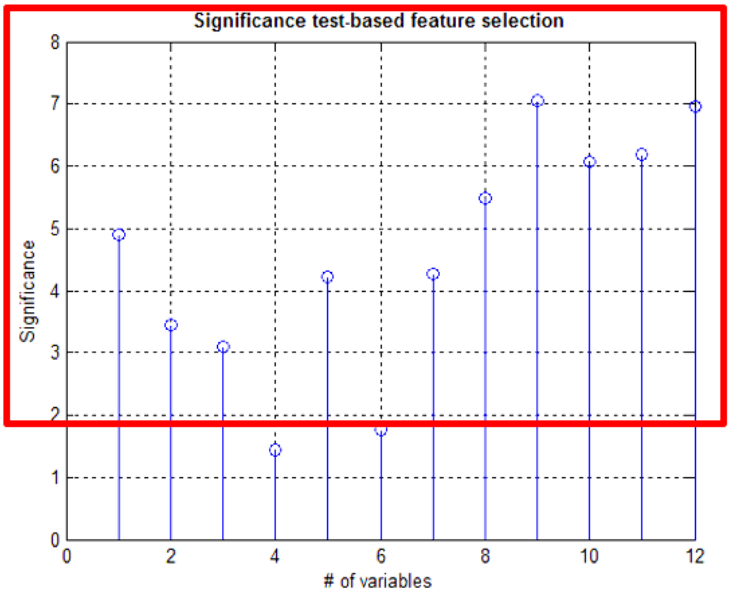


Figure2.10. Independent Features Test

2.4. Neural Network

2.4.1. Artificial Neural Network

In 1943, McCulloch and Pitts introduced a computational model of neural networks based on neurology [49]. This research is focused on biological processes in the brain and applications of the neural network with artificial intelligence [41]. In the 1950s, Farley and Clark first used computational calculators based on Hebbian learning to model biological behavior [50]. After 8 years, Perceptron, a three layer system with an input, output nodes, and association layer, is introduced by Rosenblatt. Perceptron can learn to associate a known input to a random output data. The output is also weighted and connected, and the function is fired when the sum of inputs exceeds a certain threshold [51]. Werbos, in 1974, created the back-propagation learning method. The method has three layers of neurons (e.g., input, output, and hidden layers) [52]. The Artificial Neural

Network is a system designed based on the human brain. As shown in Figure 2.11, a brain neuron collects input signals through dendrites. It releases electrical spikes that split into various branches. At each branch, a synapse alters the activity of each branch. If the amount exciting activity surpasses that of inhibitory, a neuron will fire. By mimicking the property of the neuron system, artificial neural networks can be considered as a black box, as they readily predict an output pattern based on a recognized input one. After training, the neural network can define the formats to identify similarities when presented with new inputs, such as incomplete information versus noisy data.

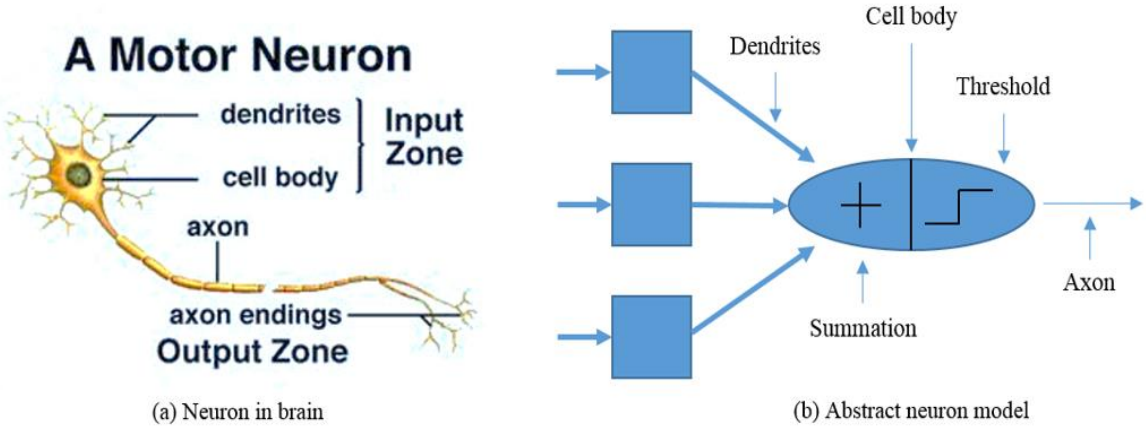


Figure 2.11. Neuron Model [53]

Neural networks applications have four prime categories:

1. Clustering: The ANN can find the relationship between patterns and combines the patterns by similarity in a cluster.
2. Classification/Pattern recognition: The ANN can allocate an input pattern to one of various classes. The classes contain algorithmic implementations.

3. Function approximation: The ANN can take linear or non-linear function approximation that can be used in fitting functions.
4. Prediction: The ANN can be used to predict future data based on given input data. The prediction significantly influences decision support systems.

Types of neural network can be classified by:

1. Applications: clustering, classification, function approximation, and prediction
2. Connection type: static (feedforward), dynamic (feedback)
3. Learning methods: supervised and unsupervised learning
4. Topology: single or multilayer, recurrent, and self-organized

Steps of Artificial Neural Network

Figure 2.12 shows that a single computational neuron is called a unit. It receives input variables, $X = \{x_1, x_2 \cdots, x_n\}$ from other units. Each input has a weight W that can represent synaptic learning. b refers to bias, and outputs should be

$$Y = f\left(\sum_{i=1}^{n} W_i x_i + b\right) \tag{2.30}$$

where f, W, b, denote an activation function, weights, and bias for a single neuron, respectively. $f_1^{(2)}$ indicates that an activation function for first neuron is in the second hidden layer.

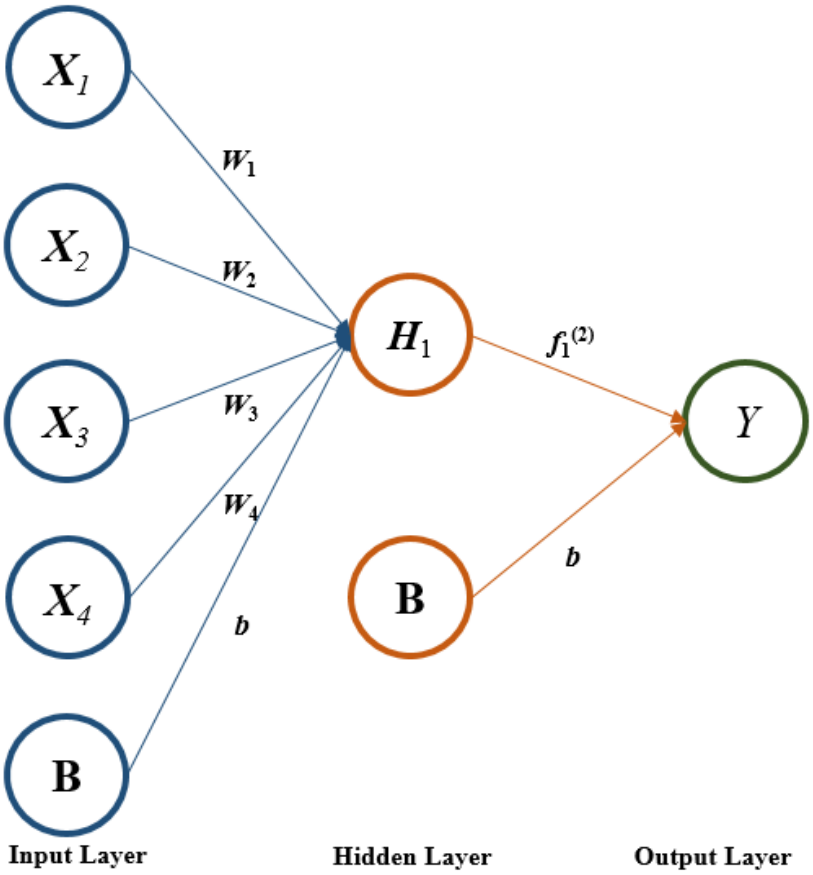
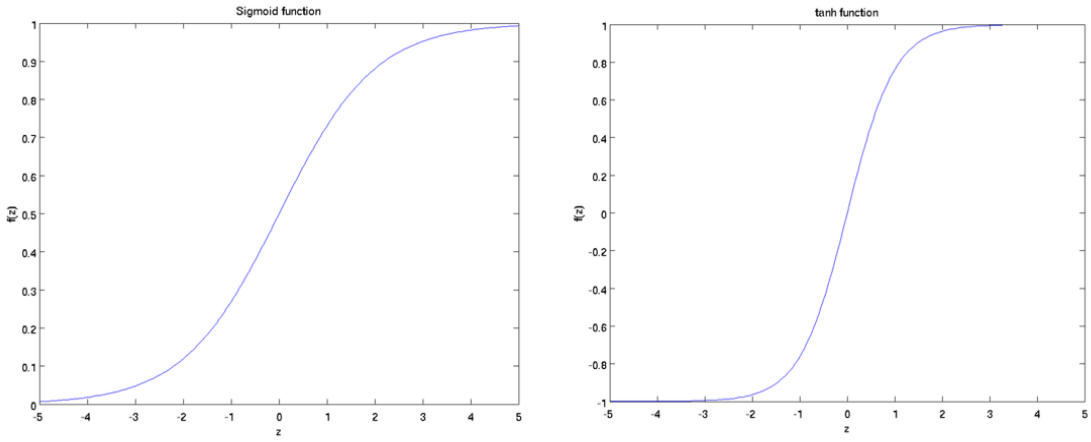


Figure 2.12. Single Neural Network

As shown in Figure 2.13, the most general activation function, *f*, is a sigmoid function. The sigmoid function's slope represents the closeness to the threshold point. Its output range is from 0 to 1. As a revised version of the sigmoid function, $f(H) = \frac{1}{1+\exp(-H)}$ and the tanh function, $f(H) = \frac{e^{H} - e^{-H}}{e^{H} + e^{-H}}$, are commonly used since the output range is from -1 to 1 where *H* indicates hidden neurons.



(a) Sigmoid Function **(b) Tanh Function**

Figure 2.13. Activation Functions

Formation of Artificial Neural Network

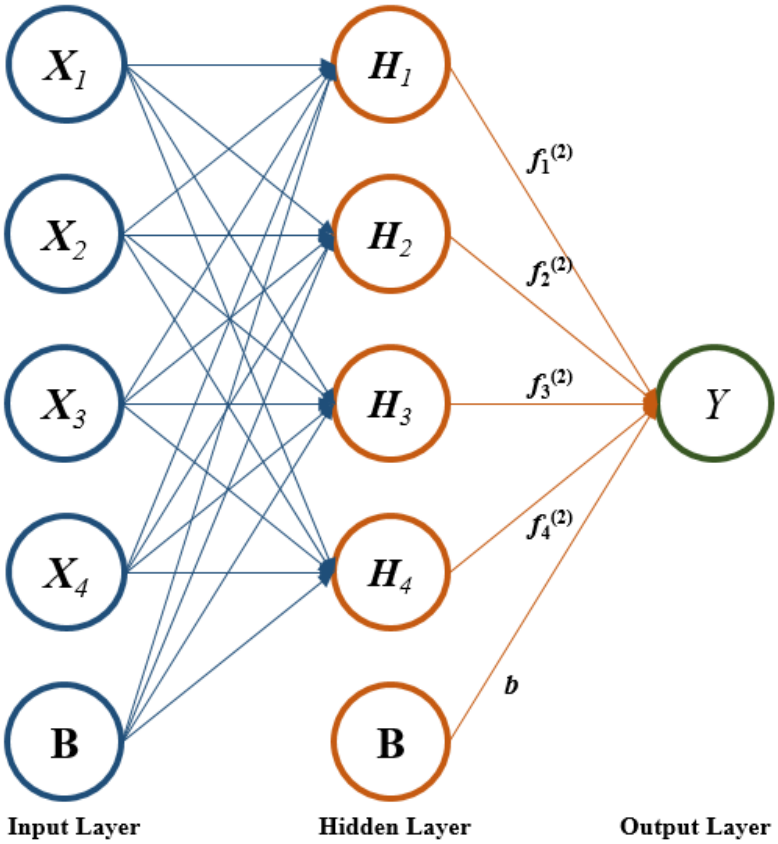


Figure 2.14. Formation of Artificial Neural Network

Figure 2.14 depicts that the circle with B in the bottom of each layer indicates both bias units and the intercept term. In Figure 2.14, the input layer contains 4 input units, while the output layer has 1 unit. As discussed above, a neural network is characterized by sets of W and b. $W_{ij}^{(l)}$ and b indicate a weight and bias between unit j and layer l. Activation, the output value, can be denoted by $f_j^{(l)}$. In the case of unit j and layer l, the total weighted sum of the inputs can be expressed as:

$$Wt_i^{(l)} = \sum_{j=1}^{n} W_{ij}^{(l)} x_j^{l-1} + b_i^{(l-1)} \tag{2.31}$$

In general, if layer 1 is the input layer and layer n_l is the output layer, each layer l is highly related with l+1. By using the concept of a feedforward neural network, all activations in each layer can be estimated in turn.

Backpropagation

In order to train a neural network, it is essential to use batch gradient descent [54]. Since an optimization algorithm is being used, the cost function should be minimized. An equation of the squared error cost function, r, can be expressed by $\frac{1}{2}\|Y - y\|^2$, where Y and y indicate the output values from ANN and real value of the output. When considering a training set of p cases, total cost function is defined as follows

$$C = p^{-1} \sum_{i=1}^{p} \left(r^2\right) + \frac{\lambda}{2} \sum_{l=1}^{n_l - 1} \sum_{l=1}^{s_l} \sum_{l=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \tag{2.32}$$

where λ denotes the weight decay parameter. n_l and s_l represent the number of output layers and hidden layers. The first term regards the total weighted sum of the cost function, while the second term regards weight decay to minimize the weights' magnitude. This function is commonly utilized for regression and classification. Minimizing the cost function is required by using batch gradient descent. With a learning rate α, each W and b are updated by an iteration of gradient decent.

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} C \tag{2.33}$$

$$b_i^{(l)} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} C \tag{2.34}$$

In order to calculate partial derivatives, the backpropagation method is required. If (x,y) is set as training data, a feedforward algorithm starts to estimate all activation values for all layers, including the output layer.

Deciding Number of Hidden units (rules of thumb)

The most important part of a neural network is the number of hidden units that should be used in a neural network system. Numerous data miners and engineers have developed a standard to find the proper number of hidden units called the "rules of thumb", but even with this rule, it is still challenging to obtain high accuracy. An exact number of hidden units can be only deduced by iterative simulations, resulting in a waste of simulation costs and hours. Thus, some general but reasonable standards should be considered. In 1992, Blum introduced an approximate rule of thumb that the size of hidden layers should be between the input and output layer size [55]. In the end of 1990s, another rule of thumb

established was that there should never be more than twice the number of hidden units as that of input units [56]. Boger and Guterman also asserted that engineers should have as many hidden units as needed to keep 70 – 90% of the variance of the set of input data [57]. In modern commercial fields, two special standards are broadly established. The first standard is that $\frac{2}{3}(N_i + N_o)$, and the other standard is that $N_h = \frac{N_s}{(\alpha \cdot (N_i + N_o))}$, where N_h, N_s, N_i, and N_o refer to number of hidden units, samples in training data set, input units, and output units, respectively. α is an arbitrary scaling factor between 2 and 10. After finding a reasonable number of hidden layers, the next step is to explore the amount of training needed. Baum and Haussler introduced a guideline to guarantee the convergence of neural network systems by using a criterion [58],

$$N \geq \frac{W}{\varepsilon} \tag{2.35}$$

where N is the necessary number of trainings, ε is an allowable error, and W is the weight function. Thus if 10% error is permitted, the amount of training is ten times as many as that of weight functions.

2.4.2. Probabilistic Neural Network

As a pattern classifier, Probabilistic Neural Network (PNN) was introduced to handle the weakness of general backpropagation neural network [59-61]. In the PNN algorithm, a response of input patterns is similar to that of training patterns near input data space [45]. Probability density functions of different classes [59] are estimated by the Bayes decision rule and Parzen nonparametric estimator [62]. PNN can be an

implementation of a statistical algorithm considered by the kernel discriminant analysis. As a multilayered feedforward neural network, PNN defines an input pattern by processing an input data from one layer to the next without feedback loops. PNN can be quickly trained because the magnitude's order is faster than backpropagation. PNN also takes a free parameter, a smoothing factor, to add or remove training samples without network retraining. PNN does not have local minima problems. On the other hand, it requires a large amount of computation memory since a node or neuron at each training step should be separately estimated.

Based on Bayes decision rule and Parzen nonparametric estimator, PNN states that the decision rule can reduce the expected risk of misclassification in pattern classification [63]. If the probability density function (pdf) of different categories *A* and *B* are given, a data set *X* belongs to class *A* when $f_A(X) > f_B(X)$ and all $A \neq B$, where $f_A(X)$, $f_B(X)$ represent the pdf for class *A* and *B*, respectively. Using the Bayesian optimal decision rule, PNN's classification decision becomes as follows:

$$h_A c_A f_A(X) \rangle h_B c_B f_B(X) \quad all\ A \neq B \tag{2.36}$$

where h_A and h_B denote the priori probability of occurrence of patterns from each class, and l_A and l_B denote a loss function or misclassification cost associated with the decision that a data set *X* belongs to each class. The accuracy of the decision boundaries relies on an estimation of the pdf for each class. The Gaussian kernel can express a multivariate estimate of class-conditional pdf of each class by

$$f_k(X) = \frac{1}{2\pi^{p/2}\sigma^p} \frac{1}{m} \sum_{i=1}^{m} e\left[\frac{-(X - X_{Tki})^T (X - X_{Tki})}{2\sigma^2} \right] \tag{2.37}$$

where k, $f_k(X)$, p, σ, m, i, X_{Tki} represent classes, the summation of the multivariate Gaussian distributions centered at each the training sample, dimension of measurement space, the smoothing parameter, the total number of training patterns, the number of patterns, and the i^{th} training pattern from class k, respectively. The multivariate estimate can classify the data set X after being trained with training data set X_T. The smoothing parameter impacts on the locations of the training data set. When the parameter increases, the degree of interpolation between the data set also increases. As show in Figure 2.15, PNN for classifying the data set X into two classes A and B is composed of four layers: the input, pattern, summation, and output layers.

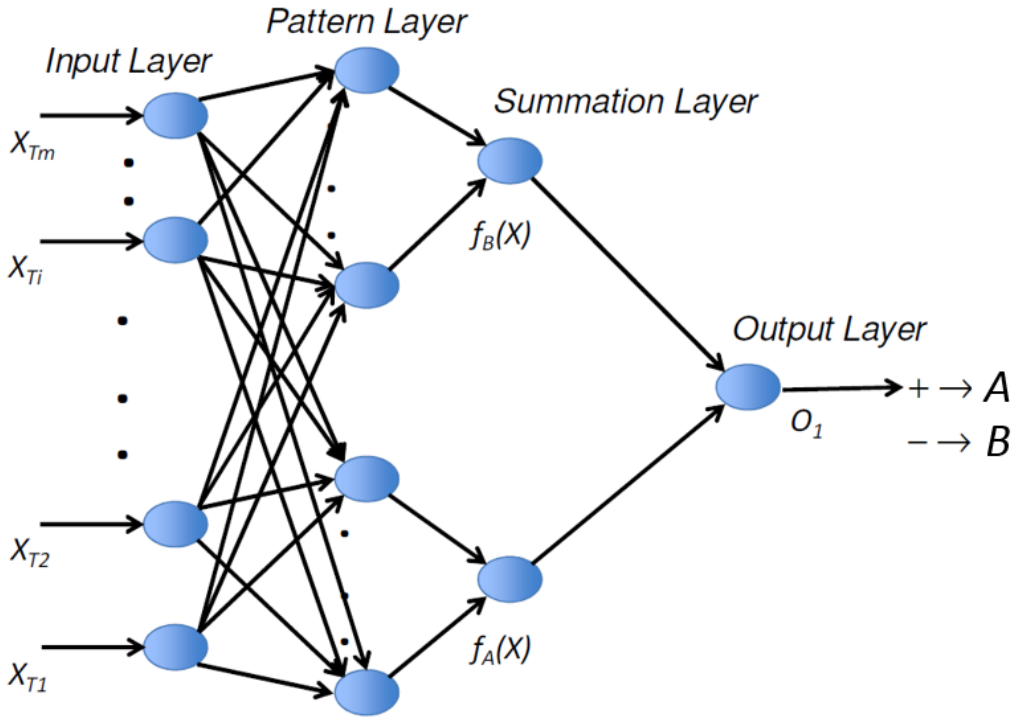


Figure 2.15. Architecture of Probabilistic Neural Network [64]

It can be assumed that the training data set X_T has n data points, and each contains m dimensions. Then PNN can have m, n, and 2 neurons in the input, pattern, and summation

layers, respectively. Figure 2.16 shows how pattern layers are comprised. For the input data set, X is normalized to its unit length and corresponds to the input unit neurons. In the second layer, with a weight function that can be operated by using the dot product with the input data set, each pattern unit is combined to the input data set. The dot product is expressed as $Z_i = X \cdot W_i$ where i, Z_i, and W_i represent the pattern number, pattern units, and weight functions, respectively. While general neural networks employ a backpropagation algorithm [65], PNN utilizes an exponential function as a transfer function. The non-linear transfer function is expressed as

$$g_i(X) = e\left[\frac{-(W_i - X_T)^T (W_i - X_T)}{2\sigma^2}\right] \tag{2.38}$$

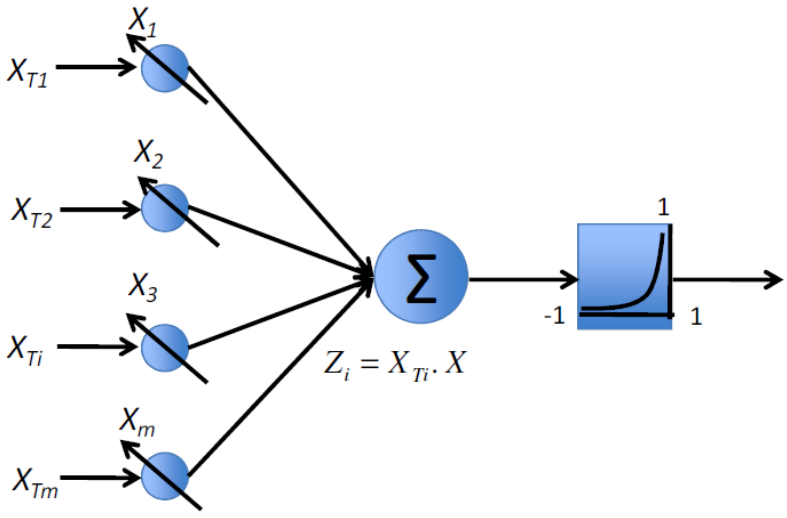


Figure 2.16. Pattern Layer of PNN [64]

In the third layer as shown in Figure 2.17, the response of each pattern unit is connected with the proper summation unit neuron. After assigning the weight functions to

each pattern unit that is similar to each input data set of the training pattern, the network is ready to be trained. A single pattern unit is needed for each training pattern. The training uses the feedforward operation, and the accuracy of the training is controlled by the smoothing parameter. In the last step classes can be divided depending if the class conditional pdf's subtraction is greater than zero.

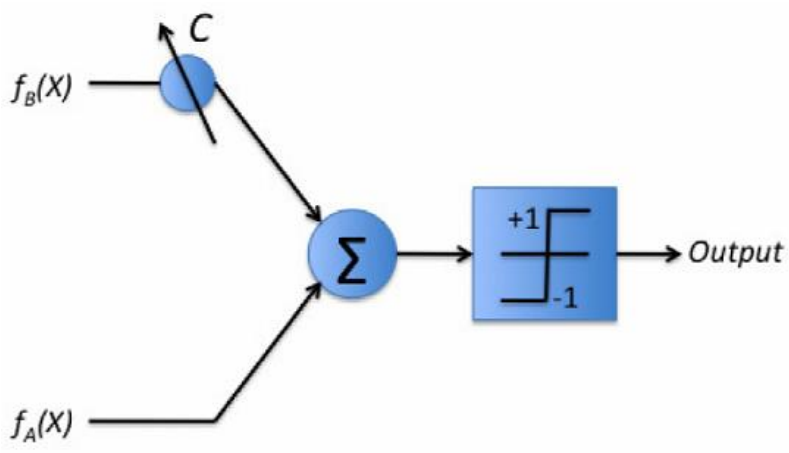


Figure 2.17. Summation Layer of PNN [64]

Limit state function estimation via PNN

PNN can be applied to estimate the limit state function of a given system [66]. As shown in Figure 2.18, the limit state function denotes a negative value when the model is under failure conditions, while having a positive value under stable conditions.

$$g(X) = R(X) - S(X) \tag{2.39}$$

$$P_f = P[g(.) < 0] \tag{2.40}$$

where R is the resistance and S is the load of a certain model. Both parameters are functions of a random variable X. In case of P_f, $g(.) = 0$ refers to the classified surface. If $g(.) < 0$, $g(.)$ represents the class B. If $g(.) > 0$, it represents the class A. Thus, there are two classes depending upon $g(.)$.

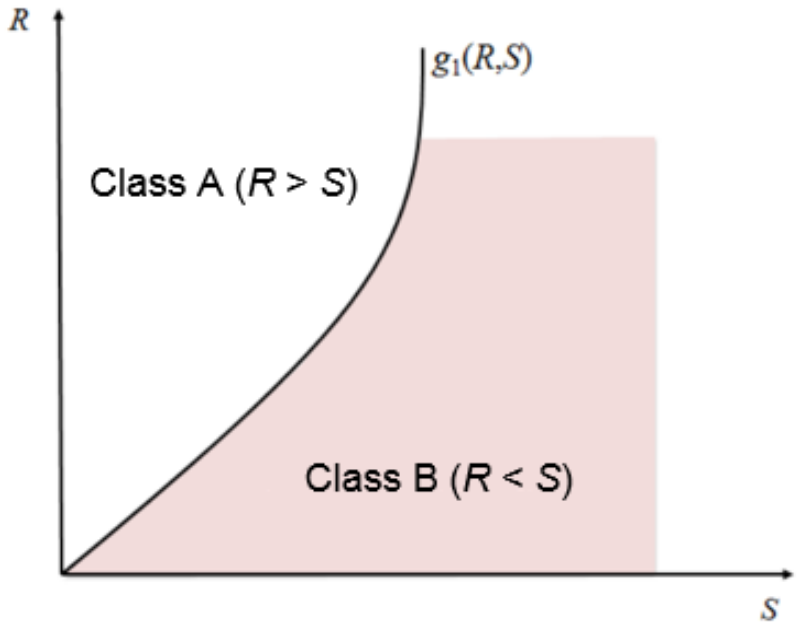


Figure 2.18 Limit State Function

CHAPTER 3
PROPOSED FRAMEWORK

The goal of this research is to accurately predict a system's response using high dimensional input data obtained from multidisciplinary engineering systems. To achieve this goal, a unified framework has been developed by using copula, dimension reduction, and regression and classification from neural network methods. This chapter will explain the proposed framework and provide a detailed overview of the process for each step of the framework with a simple example.

3.1. Objective of Proposed Framework

The aim of this research is to enhance predictions made from multidisciplinary engineering systems under uncertainty. A framework has been developed to help accomplish this goal. To present all features of a system accurately and realistically, sophisticated data sets should be obtained without any loss of information. Moreover, to get a precise analysis of a system, each component with its own marginal distribution must be estimated accurately. This way, the framework will first provide intricate marginal distributions and combine them into a joint distribution. However, the complicated data includes redundancy or uncertainty, and such attributes will negatively affect the accuracy of prediction or classification of a system. Thus, the proposed method is introduced to resolve these problems regarding the representation of sophisticated data and reduction of the attributes. The proposed method is based on the research questions of this thesis.

3.2. Steps of Proposed Framework

The proposed framework has three prime parts: generation of multivariate data, dimension reduction, and representation of multivariate system behavior by neural networks.

1. Generation of multivariate data: How the copula function resolves problems with data with various random distributions will be covered in this paper. Highly correlated and uncorrelated data will be generated, since the proposed framework aims to express how redundancy resulting from correlation between data sets from multidisciplinary engineering systems works. The proposed framework will also generate random data sets to show how uncertainty caused by high dimensional data works. Such problems can be minimized to take independent data sets. By using the copula function, the generated variables will be multidimensional and will also have realistic properties like a real model.

2. Dimension reduction: In this step, two special techniques will be used to reduce the size of the multivariate data. The first technique, feature extraction, is divided into Principal Component Analysis (PCA) and Auto-Encoder (AE); these transform the original, larger sized data, making it smaller. The reduced data will retain most of the information from the original data. The second technique is feature selection. Feature selection will be used to choose a subset of the original data by employing the Independent Features Test (IndFeaT). This test will use an application of the mean and variance. Equation 3.1 and 3.2 refer to redundancy and entropy.

$$redundancy = \log_2 N - \sum_x p(x) \log_2 \frac{1}{p(x)} \tag{3.1}$$

where N is the total number of events and $p(x)$ is the probability of each of the occurring variables. $\log_2 N$ refers to the maximum entropy.

$$entropy = \sum_x p(x) \log_2 \frac{1}{p(x)} \tag{3.2}$$

After reducing the dimension of the data, the redundancy comparison will check how much redundancy are discarded and how much the data set becomes independent. If the estimated value is still high compared to the original data's value, all three techniques will be operated again, further minimizing the size of the data.

3. Representation of multivariate system behavior: Based on the reduced and still informative data from Steps 1 and 2, a neural network will predict and classify the responses of a system wanted for design. Both ANN and PNN will be utilized for regression and classification, respectively.

By following these three steps, engineers will be readily able to represent and estimate complex data sets of a system and to predict any response of the system accurately using only a small number of data sets; the reduced sets can be regarded as missing data or distorted data [67]. This leads to saving simulation costs and time. Figure 3.1 shows a flowchart of the described framework.

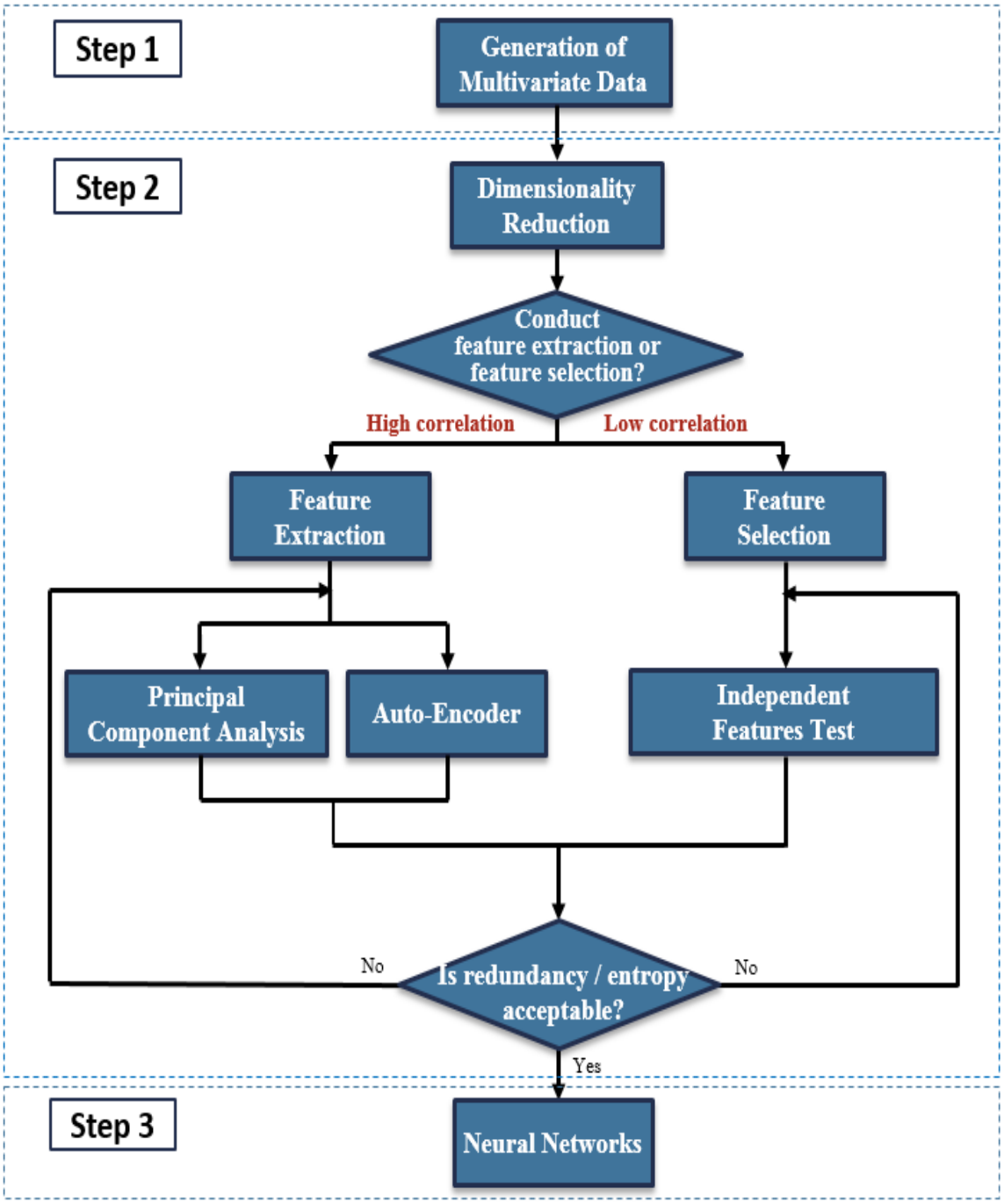
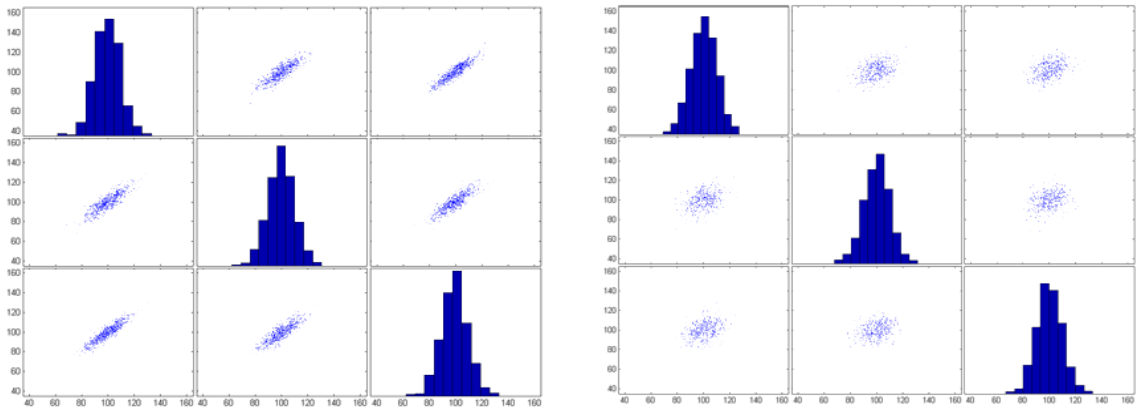


Figure 3.1. Proposed Framework

3.2.1. Generation of Multivariate Data

In Step 1, as shown in Figure 3.1, multivariate data using the copula must first be generated. The multivariate data generated by the copula can be regarded as parameters obtained in multidisciplinary engineering systems. To promote a better understanding of the framework, a data set containing 1000 simulations and 15 features is used as an example; for correlated data, a matrix of correlation parameters with a mean of 0.7 and COV of 0.2 will be used. For uncorrelated data, correlation parameters' matrix will have a mean of 0.3 and COV of 0.2. Figure 3.2 shows how the copula estimates information of marginal distributions.



(a) Correlated Data (b) Uncorrelated Data

Figure 3.2. Correlation of First Three Features Generated by Gaussian Copula

3.2.2. Dimension Reduction

In step 2 of the framework, the user should decide whether feature selection or feature extraction can be applied to the given problem. In order to decide this, a certain criteria is given with a notion of redundancy [33, 67, 68]; a prime problem of complicated multivariate data is redundancy that is related with correlation. If each variable is highly

correlated, it results in high irrelevance and low prediction accuracy. Feature extraction is able to reduce redundancy of data well, since it focuses on retaining both the independency within the original data, while feature selection has more consideration for the reduction of data size. After implementing feature extraction on the raw data, the correlation of data should be efficiently lower. Thus, engineers can choose either feature extraction or feature selection based on whether or not redundancy in the data sets should be removed. Specifically, if redundancy value of raw data exceeds 3 meaning that the raw data has high correlation, feature extraction will be employed to lessen redundancy. On the other hand, if redundancy value of raw data is lower than 3 indicating low correlation of the raw data, feature selection will reduce data size. Data obtained by feature selection can effectively reduce uncertainty, since the uncertainty is strongly related with data size. Thus, the amount of dimension reduction can be decided based on the necessity of a low computational cost and reduction of uncertainty. Figure 3.3 shows which dimension reduction method can be employed based on the decisions from the criterion.

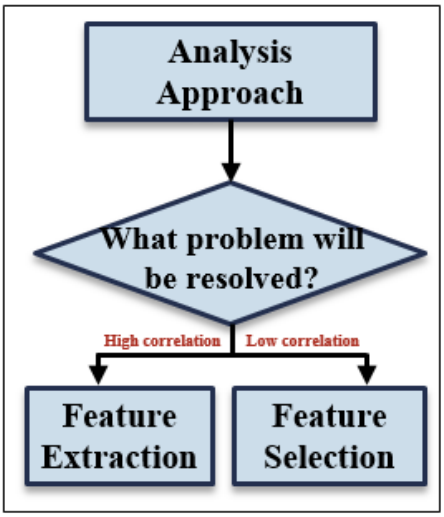


Figure 3.3. Criterion Deciding either Feature Extraction or Feature Selection

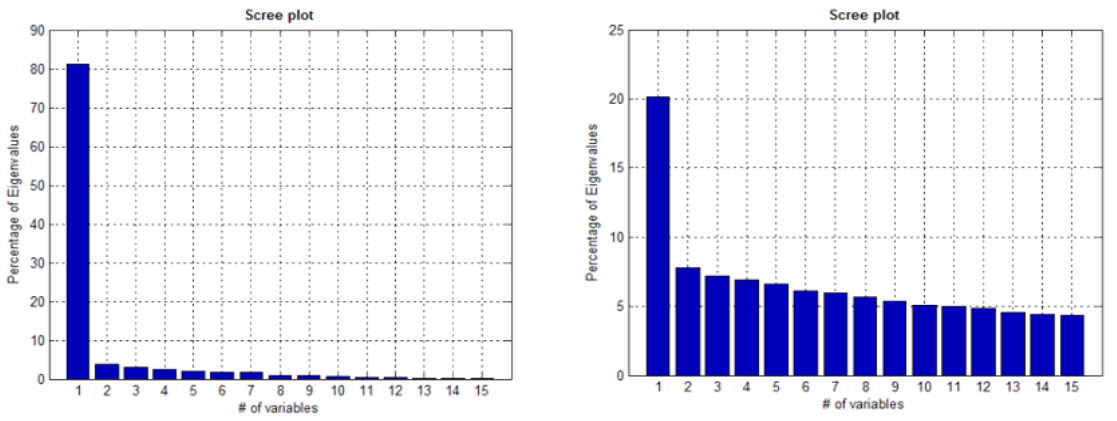
After feature extraction is operated, redundancy comparison will be employed to both raw data and reduced data. Based on redundancy comparison, if redundancy value of raw data is still higher than that of reduced data, feature extraction should be operated again to reduce redundancy more. In other case, based on entropy comparison, feature selection will be utilized with same procedure of feature extraction.

Feature Extraction

The next step has the feature extraction technique transform correlated and uncorrelated original data into smaller sized data sets by extracting significant eigenvalues.

Principal Component Analysis

In order to select the informative eigenvalues, PCA employs a scree plot for the correlated and uncorrelated data. According to Table 3.1 and Figure 3.4, the correlated data has 5 principal eigenvalues while the uncorrelated data has 13 principal eigenvalues. This shows that the original data's dimension, 15, will be reduced to 5 and 13. Each case still contains most of the information (over 91%) from the original data.



(a) Eigenvalues of Correlated Data (b) Eigenvalues of Uncorrelated Data

Figure 3.4. Scree Plot for Eigenvalues

Table. 3.1. Number of Eigenvalues and Percentage of Information

Scree Plot	Correlated Data	Uncorrelated Data
Number of Eigenvalues	5	13
Percentage of Information	92.64 %	91.24 %

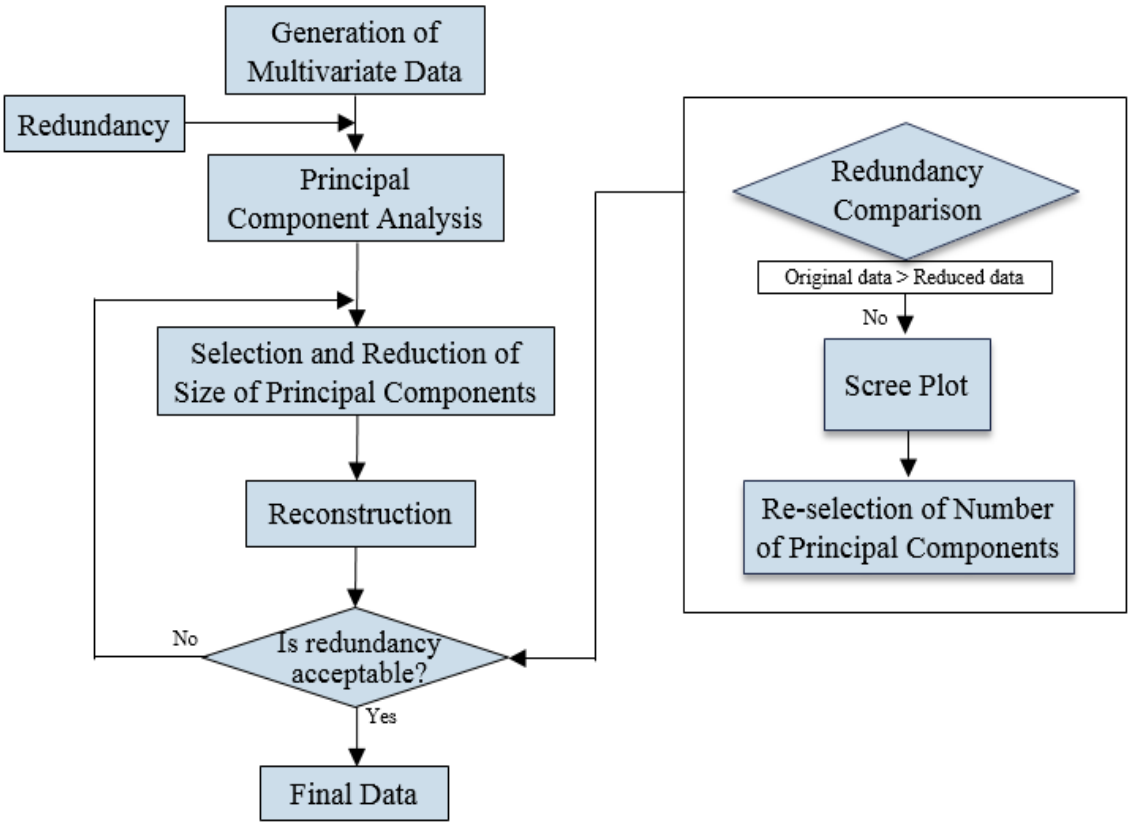


Figure 3.5. Proposed Framework of PCA

As shown in Figure 3.5, multivariate data will be first generated with a huge dimension. In the second step PCA will be employed to get compact, new data. Redundancy comparison will be used to check the correlation reduction.

Auto-Encoder

While even AE has been broadly used for non-linear and uncorrelated data, PCA is meaningful for linear analysis with correlated data. Thus, if PCA is not appropriate for the given problem, AE is considered to conduct feature extraction in this framework. With a process similar to PCA, AE also takes hidden neurons that are considered as principal components. A redundancy comparison also examines the efficacy of AE in terms of dimension reduction as shown in Figure 3.6.

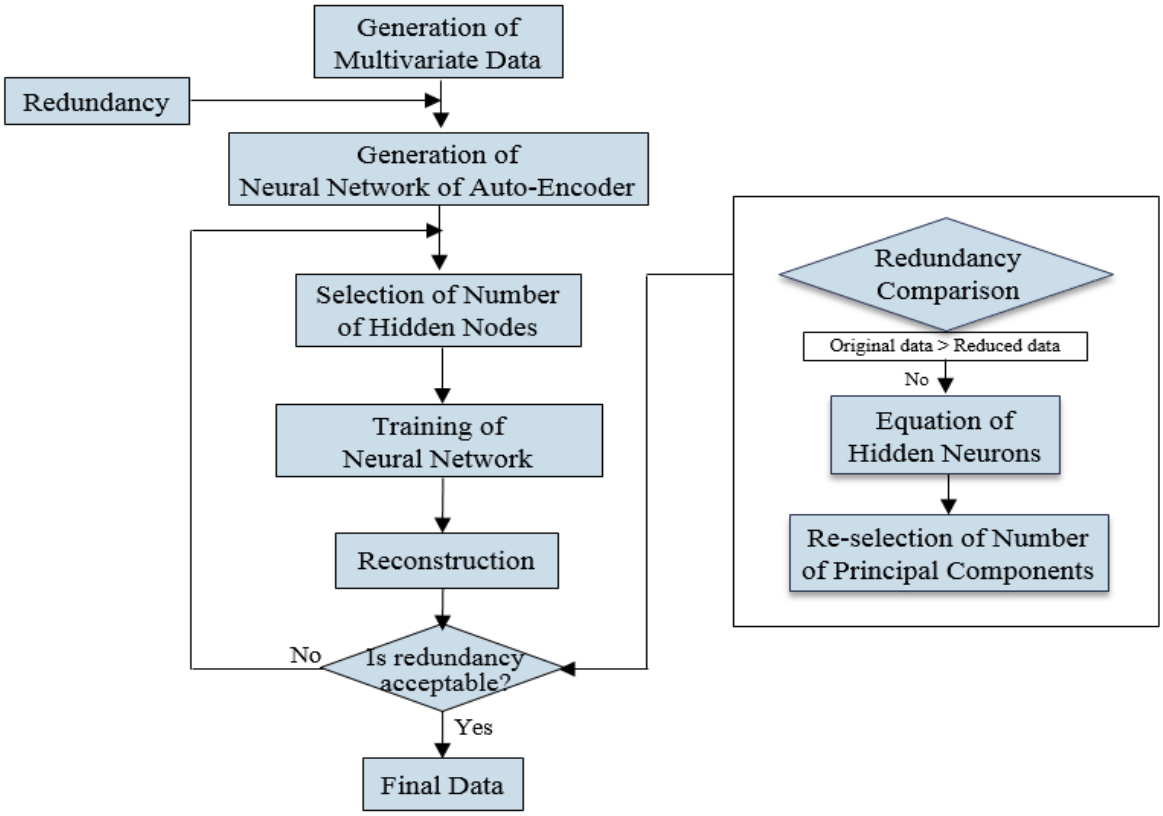


Figure 3.6. Proposed Framework of AE

Figure 3.7 explains how AE works in the proposed method. In regards to the number of hidden neurons, unlike PCA, AE cannot simply determine how many eigenvalues should be selected, and there is no specific standard for choosing the number

of eigenvalues. Engineering fields, however, commonly use $N_h = \frac{N_s}{(\alpha \cdot (N_i + N_o))}$, where α, N_h, N_s, N_i, and N_o refer to an arbitrary scaling factor, number of hidden units, samples in training data set, input units, and output units, respectively. For this reason, this thesis's AE technique will be operated with the above equation to decide the best number of hidden units.

$$N_h = \frac{N_s}{(\alpha \cdot (N_i + N_o))} = \frac{1000}{(3 \cdot (15+15))} = 11.1$$

Based on this equation, the matrix (1000 x 15) generated by the copula function is estimated. Twelve hidden units are selected to take a little bit more information. If one method takes many more eigenvalues than the other method, its complexities will be high. In the AE technique, the backpropagation algorithm are used to minimize the gradient of the difference between input and output data, respectively. AE can originally get the reduced number of hidden units by using the sparsity algorithm, but the equation mentioned above is readily employed as well when designing the neural network. After deciding hidden units, AE is modeled as shown in Figure 3.7. The initial 15 input data of AE is reduced into 12 dimensional data. This process is called "encoding", while the case where minimized data returns to 15 output data is called "decoding" or "reconstruction".

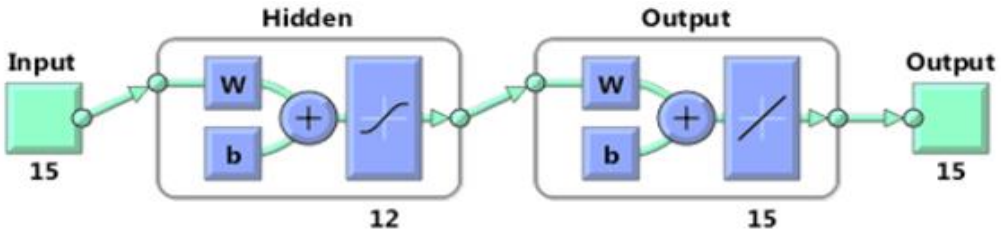


Figure 3.7. Schematic of Auto-Encoder

Comparison PCA with AE in terms of Reconstruction of Original Data

In order to check the irrelevance between the original and new data, both will be estimated by redundancy comparison again. If the new data is still informative and contains the most significant eigenvalues, its reconstructed data can cover and represent the original data similarly. Without irrelevance, PCA can get the reconstructed data by a row feature matrix based on the chosen eigenvectors and the mean-adjusted data matrix. AE already includes a reconstruction step during the process of decoding the significant eigenvalues. The accuracy of the two techniques is estimated by the absolute value.

$$Error = mean\left(\sum\left(\frac{|reconstructed\ \ data - original\ \ data|}{|original\ \ data|}\times 100\ (\%)\right)\right) \tag{3.1}$$

Feature Selection

Feature selection is the next process to be conducted during dimension reduction. The feature selection procedure chooses a subset that is composed of significant variables. For feature selection, the proposed method makes an assumption that dimension reduction is applied to variation of random variables. The variation refers to difference between each random variable and its mean. Thus, selected informative variations are used in the new subset, while insignificant variations are regarded as zero when predicting a response. In the next step, ANN and PNN are utilized for regression and classification. A subset gained by feature selection will be used to predict a response through ANN.

Independent Features Test (IndFeaT)

This technique sets up not only the significance value of “2”, but also sets up the categorical target values to select important variables. The significance value is calculated by Equation 2.29. Weiss stated that the significance value of 2 is generally accepted by his experiments [48]. Higher values may be used for more reliable features. Figure 3.8 shows how IndFeaT is applied to find important variables in data set. For the data set, IndFeaT chooses all variables except 9, 14, and 15 features. If an estimated value of a certain variable is greater than the significant value, this means that the estimate is away from the mean, and can be denoted as an independent variable. If a variable has an independent property from the data set, its properties are uncorrelated with the data set and can be regarded as an independent variable. As shown in Figure 3.9, a subset of multivariate data generated by copula will be estimated by IndFeaT to find the most informative features. Entropy will also check if IndFeaT reduces the uncertainty of data.

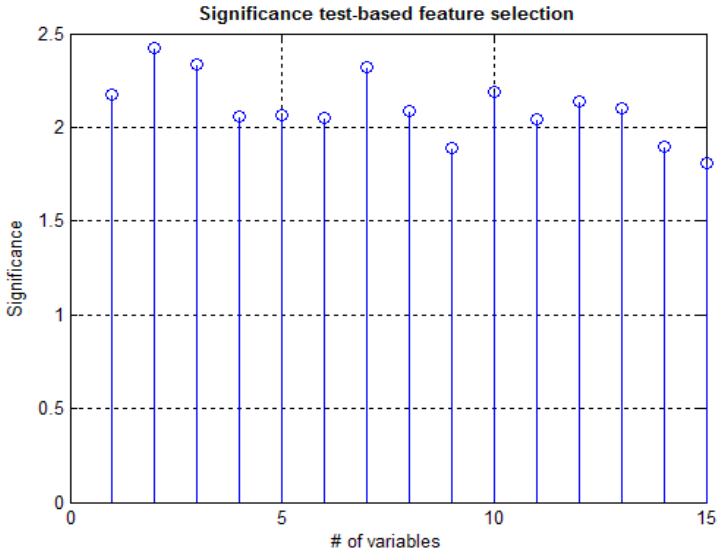


Figure 3.8. IndFeaT of Original data

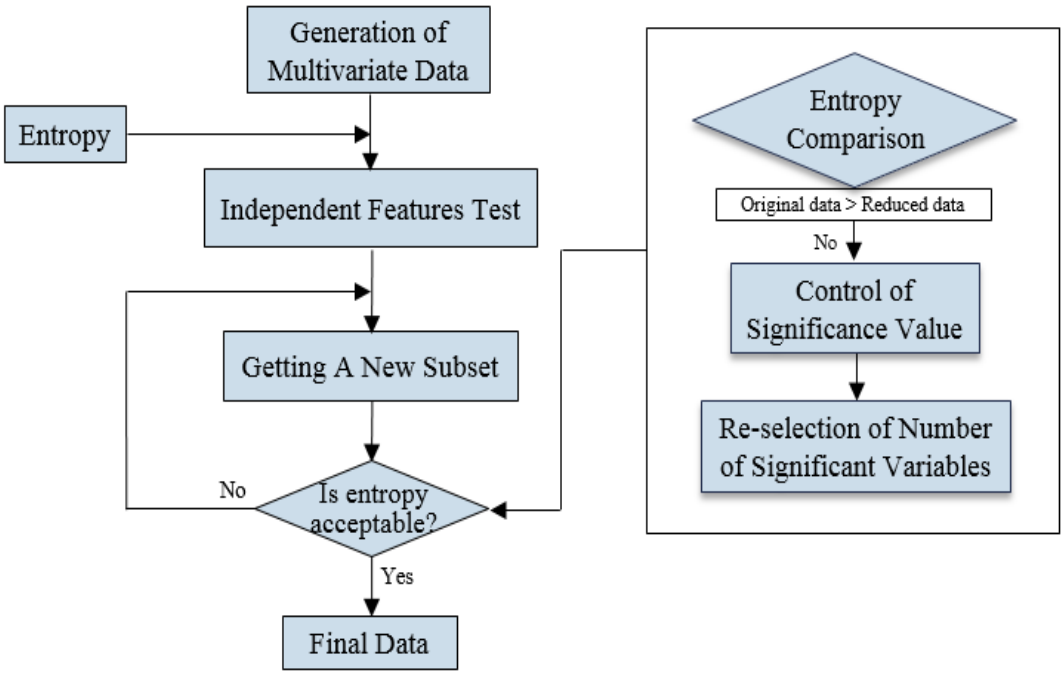


Figure 3.9. Proposed Framework of IndFeaT

Entropy

In the last step of the framework regarding dimension reduction, the entropy value, used to validate the efficacy of the reduction of dimension, is checked. The entropy of both the original multivariate data and the new data set obtained by feature selection are compared, since entropy is related to the uncertainty of data. As the size of the data increases, uncertainty also increases. If the entropy of the new data sets is not dropped, IndFeaT will use a significance value greater than 2. After redundancy or entropy value drops when it is compared to the value of the original data, the proposed framework can go on to the next step, neural networks.

3.2.3. Combination with neural networks

After successfully conducting feature extraction and feature selection, the final step is to apply the data to neural networks in order to construct a behavioral model of multidisciplinary engineering systems. This research suggests two frameworks: feature extraction with regression of ANN and feature selection with classification of PNN. Classification of ANN is not considered due to the computational complexity of high dimensional input data and its slow learning algorithm. Thus, if a given example seeks optimal data representation, feature extraction will be utilized. Feature selection can be used when classification analysis is required

Artificial Neural Network for Regression

Reduced dimensional data obtained from feature extraction will be used as sample input data for a regression neural network. The network will predict a single output value. An artificial neural network is for regression and classification, but in this research a regression method will be utilized. First, in order to train the neural network, the original data and new single output variables are required. After the training is finished, a sample set of input data reduced by PCA and AE will be employed. If the sample sets have lower redundancy, meaning that the irrelevance of the data is lower, the prediction of the neural network will be highly accurate. Moreover, even though the sample data set's size is small, if the data retains most of the information from the original data by taking all principal components, the prediction will be more accurate. Figure 3.10 explains the proposed framework for regression.

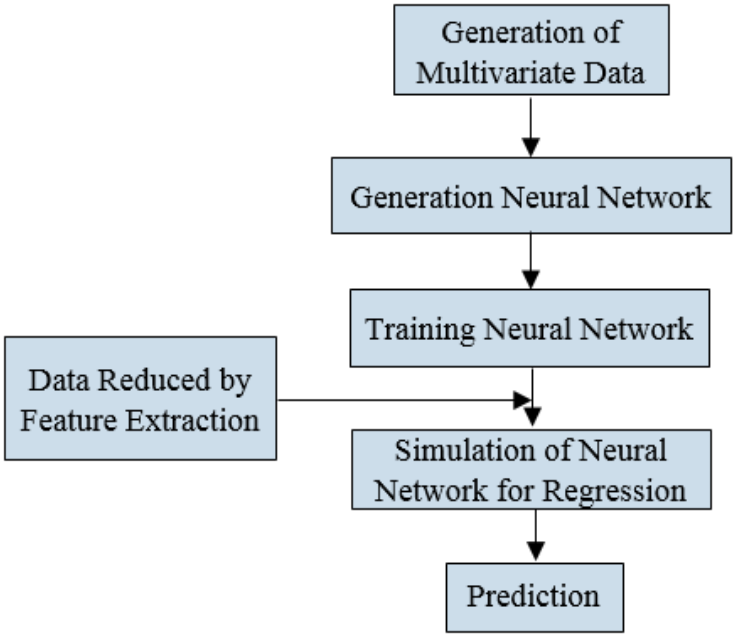


Figure 3.10. Proposed Framework for Regression

With the numerical example discussed in steps 1, and 2, this proposed framework is simply explained. The example uses two cases, correlated and uncorrelated data. Each data set has a Gaussian distribution. Figure 3.11 shows how the input data is used in a artifical neural network for regression. The number of input, hidden, and output nerons are 15, 15, and 1, respectively.

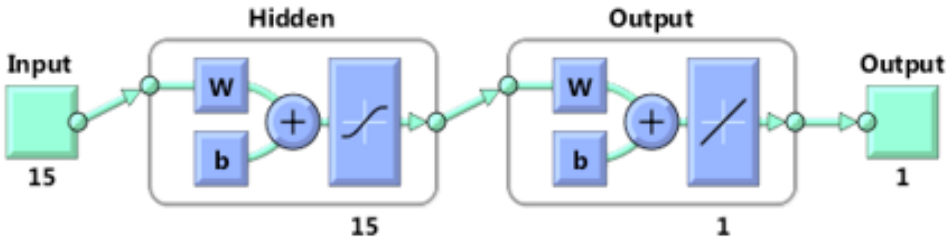


Figure 3.11. Schematic of Artifical Neural Network

Table 3.2. Total Error of Prediction

	Total Error of Prediction (%)	
	Correlated Original Data	**Uncorrelated Original Data**
Original Data + ANN	3.72	2.06
PCA + ANN	2.38	1.71
AE + ANN	2.97	1.48

As shown in Table 3.2, for this simple example, the original data + ANN shows that after training the neural network with the original data, the data is used again as an input for the ANN to check the accuracy of the neural network. For correlated data, PCA + NN has the smallest error. However in the case of uncorrelated data, its error was increased. The result of AE + ANN for any case is realiable due to the lower error of prediction compared to the original data + ANN. This means that PCA and AE effectively reduced the redundancy of the complicated original data since the accuracy of prediction is reliable.

Probabilistic Neural Network for Classification

For classification, a Probabilistic Neural Network (PNN) will be implemented. Similar to the progress of regression with artificial neural network, input data sets for the neural network can be taken by feature selection, specifically, IndFeaT. Figure 3.12 depicts the proposed framework for classification with PNN.

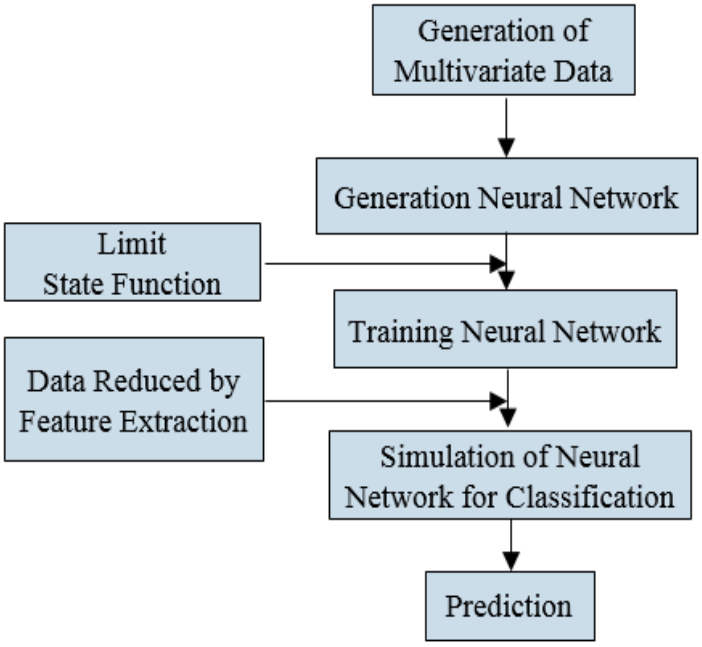


Figure 3.12. Proposed Framework for Classification

To make two output categories, the limit state function in Equation 2.39 will be used. As stated, $g = R - S$, where R is data from the simulation and S is the experimental value. R is a set of means from random variables and S represents generated random variables. $g(.) < 0$ represents the failure region, otherwise, $g(.) > 0$ denotes the safe region. After estimating categories, the results of the classification can be expressed by the probability of failure: $P_f = \frac{N_{fail}}{N_{total}}$ where N_{total} and N_{fail} represent the number of total simulations and failures of classification, respectively. Figure 3.13 indicates that PNN works for classification with 15 input neurons and 2 output classes.

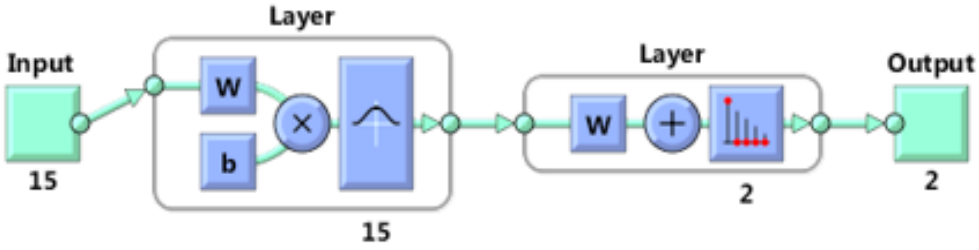


Figure 3.13. Schematic of Probabilistic Neural Network

The P_f from the proposed method is compared with the P_f from the Monte Carlo Simulation (MCS). MCS needs many iterative experiments to get the most accurate approximate value. As many experiments (or cases) are conducted, the value of P_f is diminished. MCS can estimate P_f by using the total number of experiments and the number of failure modes calculated by the limit state function $P_{f\_MCS}[g<0]=\frac{N_f}{N}$, where N represents total number of experiments and N_f represents the number of failure modes. As shown in Table 3.3, the probability of classification failure with IndFeaT and MCS are nearly similar. Thus, the classification process has been accurately operated.

Table 3.3. Probability of Failure

	Original Data with 1,000 Samples	**MCS with 10,000 Simulations**	***Pf* difference between original data and data with MCS**
Pf	0.0936	0.0938	0.21 %

CHAPTER 4

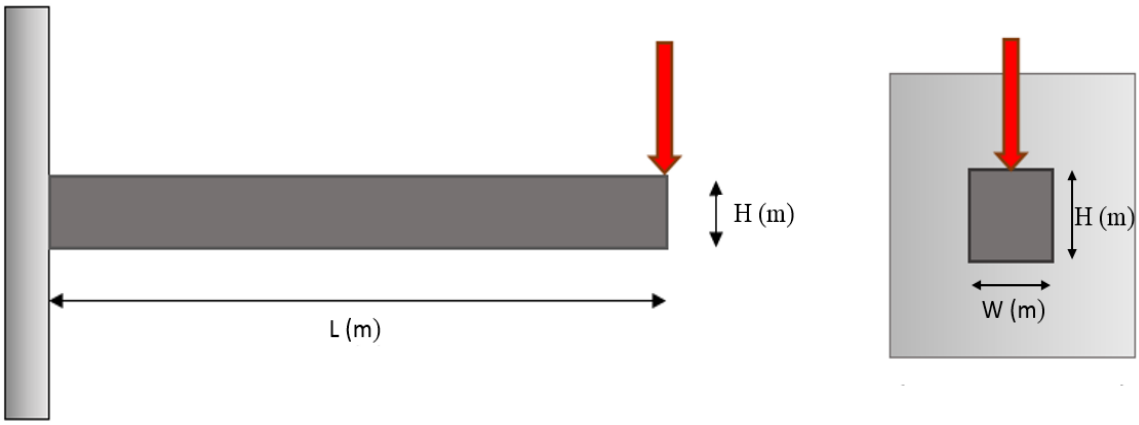
DESIGN EXAMPLES

This chapter will demonstrate the efficacy of the proposed method with three examples. The first example will show how the proposed method is applied to a 3-D cantilever beam and contributes to accurate prediction of the system's response behavior. The second example will discuss how the proposed method can produce good results in classification problems such as the fatigue life analysis of a solder joint example. The last example is a stretchable patch antenna problem that can be considered as a typical example of a multidisciplinary system. This example will show the advantage of the method in designing antenna substrates realistically, and predicting the most reasonable frequency in varying thickness and displacement.

4.1. 3-D Cantilever Beam

4.1.1. Problem Description

In this example, a 3-D cantilever beam shown in Figure 4.1 is considered. A point load is applied to a node of the beam in the orthogonal direction. This beam has a length of 4m, a height of 0.1m, and a width of 0.1m, as shown in Figure 4.1. In this problem, Young's moduli, E, of each element is considered as a random variable. The variable is generated with $\mu_E = 2.05 \times 10^{11}$ pa and COV = 0.1. The design parameters and their properties are summarized in Table 4.1.



(a) Cantilever Beam on x-y Axes (b) Cantilever Beam on y-z Axes

Figure 4.1. 3-D Cantilever Beam

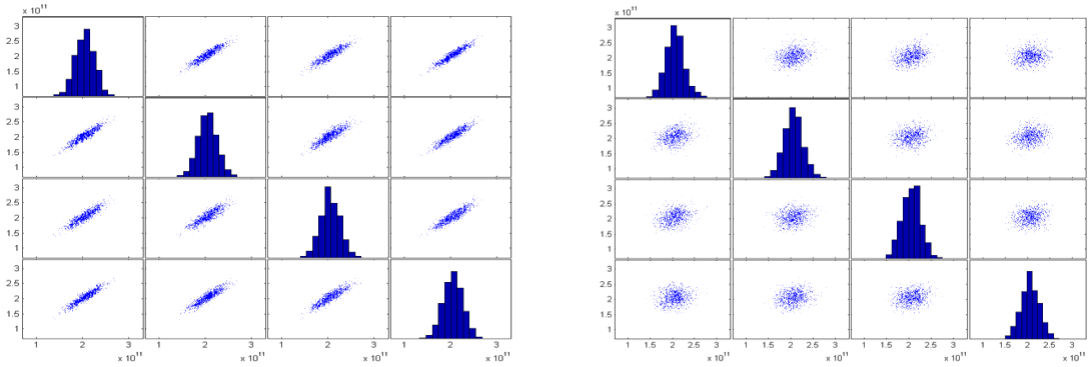
The set of Young's moduli is composed as a random field. The random field is represented by the Gaussian copula function in this example. Thus, the beam can be divided into 30 unit elements, and each element has a different Young's moduli. Moreover, each variable is assumed to be highly correlated to a matrix of correlation parameters with a mean of 0.7 and COV of 0.2. For the case of an uncorrelated example, the correlation parameters' matrix has a mean of 0.3 and COV of 0.2. Thus, this example will progress with two data sets, correlated and uncorrelated, with Gaussian distribution.

Table 4.1. Property of Cantilever Beam

Length (m)	4
Height (m)	0.1
Width (m)	0.1
Loading magnitude (N)	1000 in z direction
Young's moduli (pa)	$\mu_E = 2.05\times10^{11}$, $COV_E = 0.1$
Number of elements	30

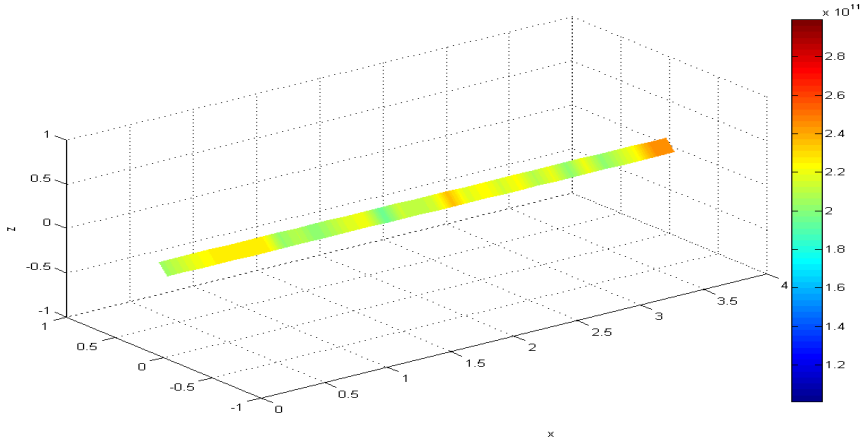
4.1.2. Generation of Young's Moduli

A cantilever beam is considered to have 30 Young's moduli. 30 random variables make a Gaussian random field generated by the copula function. Each value is highly correlated and uncorrelated. Figure 4.2 expresses how each variable can be demonstrated by the Gaussian copula. Each data set has distinguished marginal distributions, and all relationships between the marginal distributions are shown by the copula. Figure 4.3 shows each Young's modulus using color scales for random field realization.



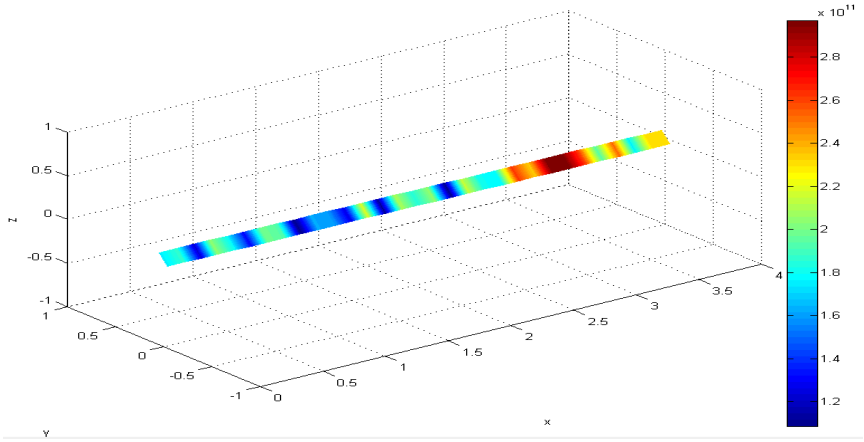
(a) Correlated Young's Moduli (b) Uncorrelated Young's Moduli

Figure 4.2. Correlation of First Four Features by Gaussian Copula



(a) Random Field Realization of Correlated Cantilever Beam Elastic Moduli

Figure 4.3. Random Field Realization of Cantilever Beam Elastic Moduli



(b) Random Field Realization of Uncorrelated Cantilever Beam Elastic Moduli

Figure 4.3. Random Field Realization of Cantilever Beam Elastic Moduli (Continued)

After making two data sets, 1000 (Simulations) by 30 (features) matrices for Young's moduli, the original data sets' redundancy is checked. Table 4.2 shows the redundancy of each Young's moduli for the correlated and uncorrelated cases.

Table 4.2. Redundancy of Original Young's Moduli

	Redundancy
Correlated Young's Moduli	7.6828
Uncorrelated Young's Moduli	3.4950

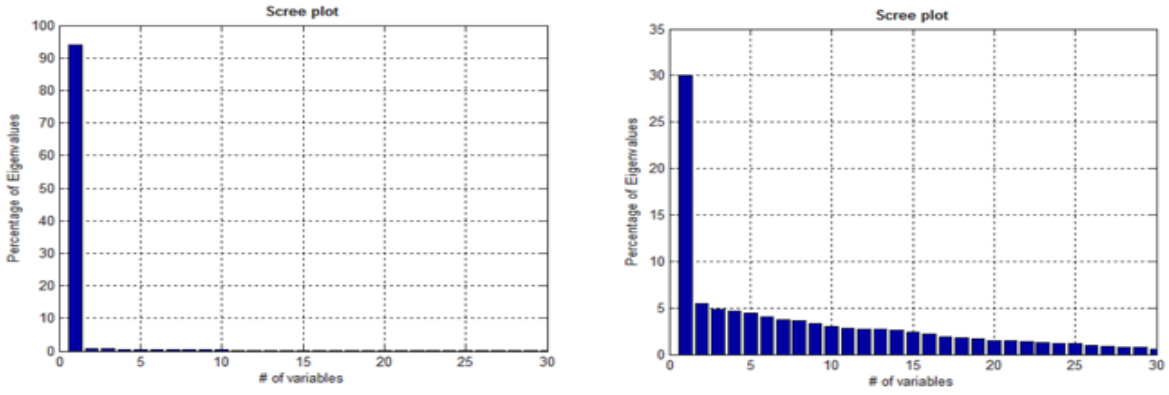
By Equation 3.1, the correlated data set has a value of 7.6828 for redundancy, while the uncorrelated data set's redundancy is 3.4950. The correlated data set has much more redundancy than the uncorrelated one. Since each redundancy value is higher than 3, feature extraction will reduce redundancy of correlated and uncorrelated data. Although

dimension reduction will be applied to both the correlated and the uncorrelated Young's moduli, the correlated moduli may be required to further lessen redundancy.

4.1.3. Dimension Reduction for Young's Moduli

After describing the randomness of the input data, the next step is to decide the process of feature extraction and feature selection as shown in Figure 3.1. Based on the criterion to distinguish feature extraction and feature selection, this problem is chosen by data analysis of feature extraction since this example does not individually estimate each Young's modulus. Instead, it requires an analysis to represent the optimal tip displacement based on the correlated or uncorrelated Young's moduli consisting of a random field. Also, a fast learning algorithm is not as important as accurate predictions. Thus, in this example, both PCA and AE are considered to conduct feature extraction. Feature extraction is employed to reduce high dimensional data. After diminishing the dimension of the data, redundancy comparison will be utilized how much redundancy have decreased compared to the original data.

Principal Component Analysis



(a) Eigenvalues of Correlated Young's Moduli (b) Eigenvalues of Uncorrelated Young's Moduli

Figure 4.4. Scree Plot of Cantilever Beam

Table 4.3. Number of Eigenvalues and Percentage of Information

Scree plot	Correlated data	Uncorrelated data
Number of Eigenvalues	4	25
Percentage of Information	96.09 %	96.11 %

According to Figure 4.4 and Table 4.3, PCA selected 4 and 25 eigenvalues of correlated and uncorrelated data sets to contain 96.09% and 96.11% of information, respectively. The dimension of the original data sets can be reduced to 4 for correlated data and 25 for uncorrelated data without a large loss of information. Based on the data sets, a redundancy comparison will check for reduction of redundancy. As shown in Figure 4.6 and Table 4.4, PCA led to the correlated data's redundancy reduction of 86.52 % of the original data, while the reduction in uncorrelated data is 22.39 %. Efficiency of PCA regarding the redundancy reduction will be compared to that of AE in the next step.

Auto-Encoder

First, using the equation for deciding on the total number of hidden neurons, $N_h = \frac{N_s}{(alpha \cdot (N_i + N_o))} = \frac{1000}{(1 \cdot (30 + 30))}$ = 16.6, AE is first designed. AE has 30 dimensions for the input and output data and 17 dimensions for hidden neurons. If AE selects 17 hidden neurons, 98.8 % and 84.6 % of information from the original data are taken because hidden neurons can be regarded as eigenvalues in linear analysis. A schematic of AE is shown in Figure 4.5. With 30 input data points, AE reduces its

dimension to 17, indicating that 17 new data points are significant eigenvectors that can reduce the redundancy of the original input data.

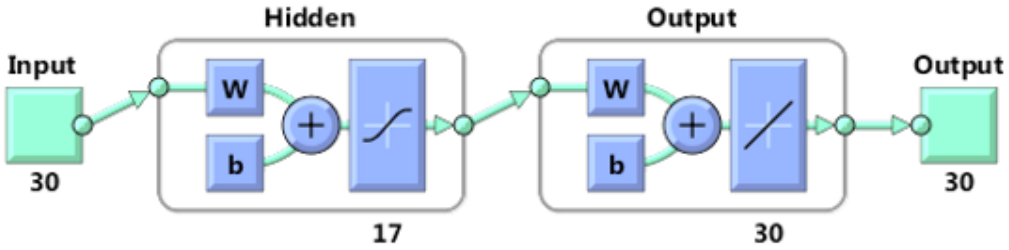


Figure 4.5 Schematic of Auto-Encoder

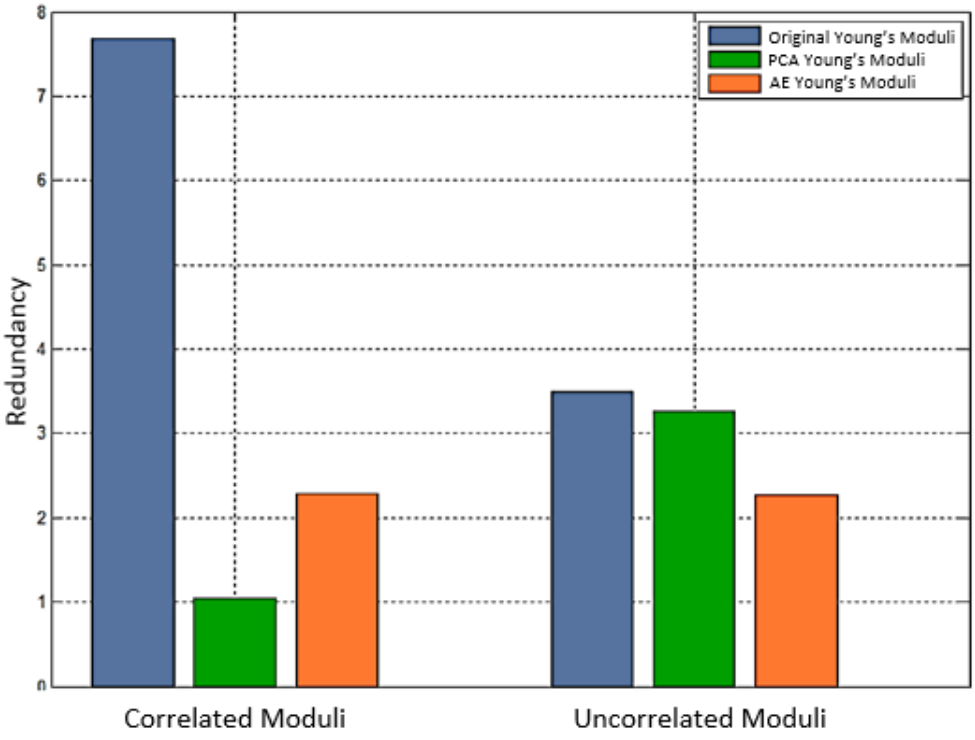


Figure 4.6. Redundancy Variation of Young's Moduli

Table 4.4. Redundancy Variation of Young's Moduli

	Correlated Young's Moduli (Redundancy Reduction %)	Uncorrelated Young's Moduli (Redundancy Reduction %)
Redundancy of Young's Moduli	7.6828	3.4950
Redundancy of Young's Moduli with PCA	1.0403 (86.52%)	3.2711 (22.39%)
Redundancy of Young's Moduli with AE	2.2806 (70.31%)	2.2678 (35.11%)

From AE, the redundancy reduction of the original data is estimated in Figure 4.6 and Table. 4.4. The reduction of redundancy is 70.31 % for correlated data and 35.11% for uncorrelated data. Unlike PCA, which cannot effectively reduce redundancy in uncorrelated cases, AE has a much higher reduction rate. However, the performance of AE in correlated data is slightly unsatisfactory compared to that of PCA. Since AE is operated with 17 hidden neurons that are more than 4 neurons that PCA took. As the number of hidden neurons increases, efficiency of redundancy deduction decreases. These properties are also checked by the reconstruction error, as shown in Table 4.5. PCA efficiently reconstructs correlated data with small error while being discredited when rebuilding uncorrelated data. AE is reliable for reconstructing both data sets. Reconstruction error can be calculated by the absolute value.

$$Error = mean\left(\sum\left(\frac{|reconstructed\ \ data - original\ \ data|}{|original\ \ data|}\times 100\ (\%)\right)\right) \tag{4.1}$$

Table 4.5. Reconstruction Error

Reconstruction Error (%)	Correlated data	Uncorrelated data
Original data VS data with PCA	1.53%	4.74%
Original data VS data with AE	1.87%	1.56%

4.1.4. Artificial Neural Network for Regression of Deflection Estimation

An ANN is designed with 30 dimensions of input data, 21 dimensions of hidden neurnons, and 1 dimension of output data. The same network properties of AE are used for the ANN. Figure 4.7 represents how the ANN is comprised.

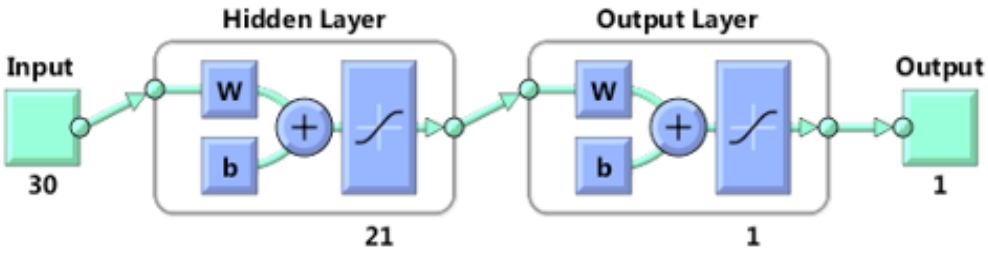
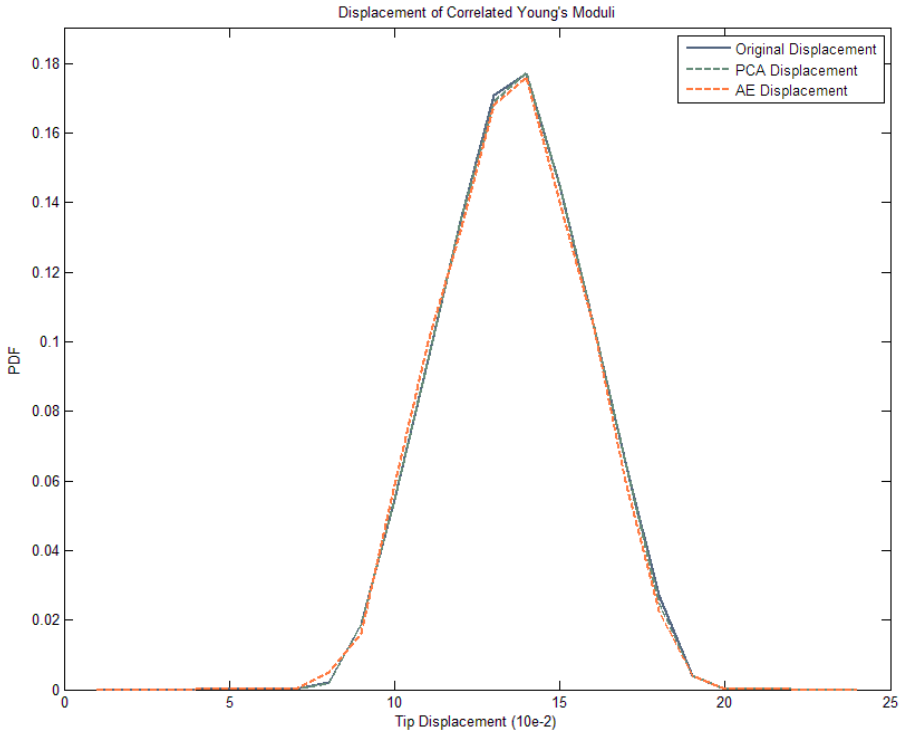


Figure 4.7. Schematic of Artificial Neural Network

After generating the reduced Young's moduli through PCA and AE, the designed ANN is used to provide a statistical estimation of the tip displacement. To demonstrate the accuracy of the proposed method, the tip displacement of new data sets obtained by PCA and AE are compared to those of the original data by using the probability density estimation in Figure 4.8. In the uncorrelated data, the displacement of the new data sets provides results almost identical to those of the original displacement. However, with correlated data, AE has slightly lower accuracy than PCA. Even though AE failed to yield a response similar to the original displacement, the result of AE with 3.91% prediction error may be acceptable, since the method and this example are based on linear analysis. If complicated non-linear analysis, such as considering multiple loading conditions, is used, PCA's results may not be reliable. Prediction error is estimated by the absolute value by Equation. 4.1.

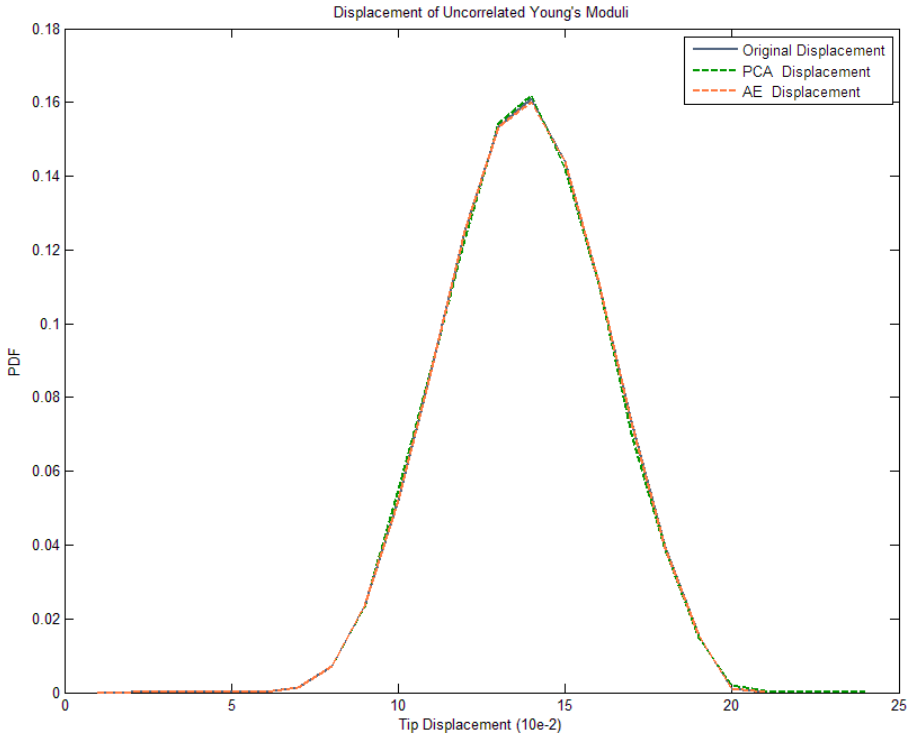
Table 4.6. Prediction Error of Displacement

	Correlated Original Data		Uncorrelated Original Data	
	Displacement (0.0128)	**Prediction Error (%)**	**Displacement (0.0125)**	**Prediction Error (%)**
Original Data with ANN	0.0135	5.47	0.0128	2.40
Data with PCA and ANN	0.0132	3.12	0.0128	2.40
Data with AE and ANN	0.0133	3.91	0.0127	1.57



(a) Tip Displacement of Correlated Young's Moduli

Figure 4.8. Tip Displacement



(b) Tip Displacement of Uncorrelated Young's Moduli

Figure 4.8. Tip Displacement (Continued)

4.2. Solder Joint Analysis Example

4.2.1. Problem Description

In this example, the proposed method is applied to the crack analysis of a solder joint, as shown in Figure 4.9. The Ball Grid Array (BGA) method is commonly used in packaging engineering and plays a crucial role in connecting a chip with a substrate by using a solder ball. However, this method is exposed to destruction due to propagation of a crack in the solder ball. The propagation is generally due to thermal stress. Based on the thermal stress equation, variance of temperature, coefficient of thermal expansion (CTE), and properties of the chip and solder ball are assumed to be prime causes of the total number of cycles that generate failure. The number of cycles is called fatigue life. If a BGA's fatigue life obtained by experiments is smaller than a value of fatigue life attained

by simulation, the BGA cannot be maintained anymore due to the propagation of the crack. This example, therefore, shows what variables such as CTE, temperature, or properties of BGA are influential to trigger a crack. After finding the most significant variables, IndFeaT of feature selection will reduce the dimension of the original data consisting of all variables to get data with low value of entropy. This is because that 9 significant variables are considered that they are independent and uncorrelated each other. In the last step, a Probabilistic Neural Network (PNN) will obtain two classes, safe and unsafe, by using the concept of fatigue life and the limit state function. Then, the reduced new data will be used to predict accurate classes. To validate this process, the classifications P_f and MCS's P_f are compared. If both are close together, accuracy of this example is reliable.

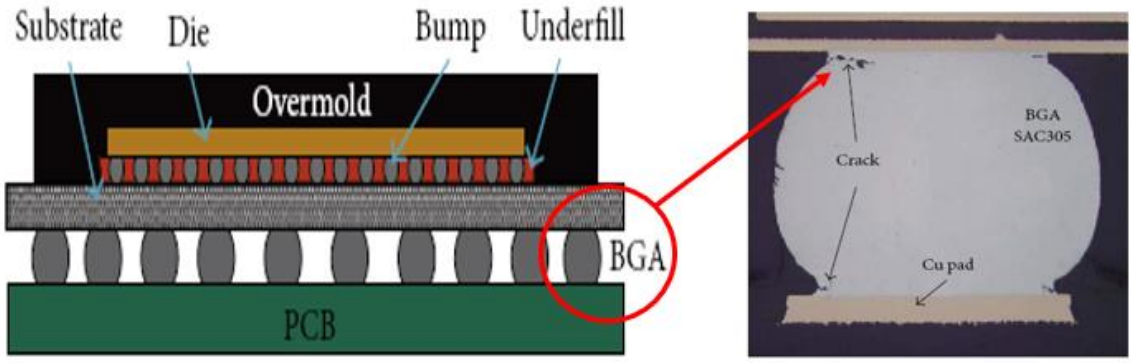


Figure 4.9 Crack Analysis of Solder Joint [69]

This problem is based on the Coffin_Manson fatigue model [70]. The equation of fatigue life and each property is listed in Table 4.7:

$$N_f = 0.5\left(\frac{\Delta\gamma}{2\varepsilon'_f}\right)^{1/c} \tag{4.2}$$

$$c = -0.442 \times (-6\times10^{-4} T_m) \times 1.74\times10^{-2} \ln\left(1+\frac{360}{t_D}\right) \tag{4.3}$$

$$\Delta\gamma = F \cdot \frac{L_D}{h} \cdot \Delta\alpha \cdot \Delta T \tag{4.4}$$

Table 4.7. Properties of Solder Joint

Property	Definition	Mean and COV
N_f	The fatigue life	
ε'_f	The fatigue ductility coefficient	(0.325, 0.01)
T_m *(°C)*	The Mean cyclic solder joint temperature	(100, 0.05)
t_D *(min)*	The half-cycle dwell time	(15, 0.05)
F	The Empirical non-ideal factor	(1,0.01)
L_D *(m)*	The Distance of the solder joint from the Neutral Point (DNP),	(0.0176, 0.1)
h (m)	The Solder joint height	(0.00075, 0.1)
$\Delta\alpha$ *(ppm/°C)*	The Absolute difference in coefficients of thermal expansion of solder joint and substrate	(4.5, 0.2)
ΔT *(°C)*	The Cyclic temperature swing	(165, 0.2)
c	The Fatigue ductility exponent	(-0.411, 0.01)

4.2.2. Generation of Random Properties of Solder Ball Joint

A solder joint example has 9 important properties. It means that the 9 properties are highly independent and uncorrelated each other. Thus, the Gaussian copula function generates a random data set with correlation parameters with a mean of 0.2 and COV of 0.1 by using these properties. Figure 4.10 shows how the first four properties share interdependency. In an analysis of a solder joint, variation of CTE and temperature, dwell

time, and shape of the solder ball are considered to be important variables. Based on this fact, each data set with different COV is generated.

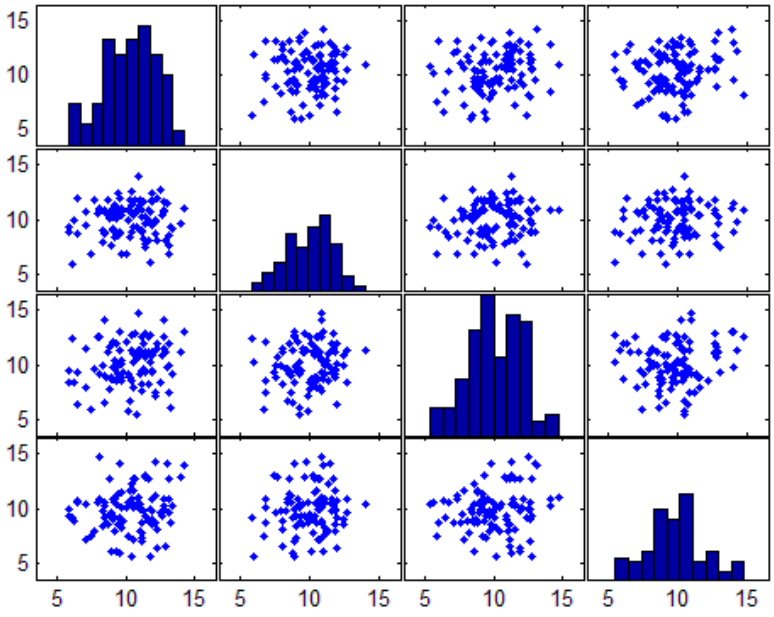


Figure 4.10. First Four Features by Gaussian Copula

4.2.3. Dimension Reduction for Properties of Solder Ball Joint

After assigning 9 properties, we should decide to use either feature extraction or feature selection for dimension reduction. In this example, each of the 9 properties impacting fatigue life will be estimated to check each significance value by reducing the uncertainty of data sets, since redundancy value of the properties is 1.8645 that is smaller than 3. Thus, feature selection will be employed. The selected properties can be used to make a new subset that has a fast learning algorithm and low level of computational complexity. In the last step, the method used by the subset to improve accuracy of classification will be confirmed. This example will focus on feature selection for dimension reduction. IndFeaT is allowed to take the most informative properties based on significance

value. In properties' relationships, IndFeaT selects 2, 3, 5, 6, 7, and 8 properties. It is assumed that the variation of CTE and temperature, dwell time, and shape of solder ball are already assigned as significant variables as shown in Figure 4.11. To check the uncertainty of the data set, redundancy comparison is used in Table 4.8. Redundancy reduction is estimated by the absolute value (Equation 4.1). IndFeaT leads to 29.10% of uncertainty reduction. Thus, the current process shows that 9 properties' dimensions are effectively reduced by IndFeaT. Within the generated random properties, the data sets' significance is lower than the selected data sets' importance. Thus, in real simulations, such as the finite element method, if the properties selected by IndFeaT are controlled and revised first, the probability to reduce simulation cost and time and obtain more accurate results will increase.

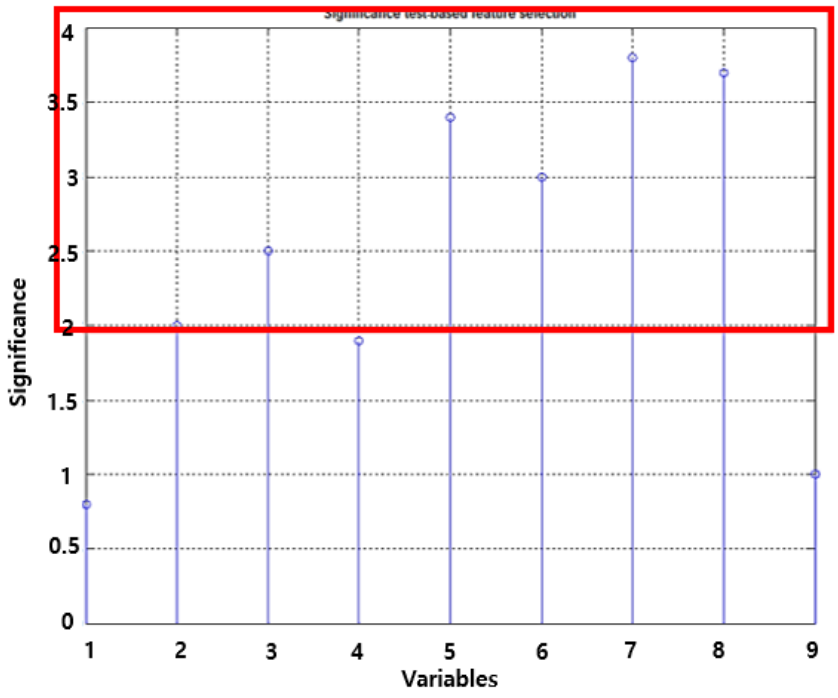


Figure 4.11. IndFeaT of 9 Properties

Table 4.8. Redundancy Reduction

	Redundancy	Redundancy Reduction (%)
Original Data	1.8645	29.10%
Original Data with IndFeaT	1.3219	

4.2.4. Probabilistic Neural Network for Classification

In this process, PNN is required for classification based on the target. To assign two classes, the limit state function and experimental results make the target data. Fatigue life from the simulation has a value of 1279 (mean: 1279, COV: 0.07) and an experimental value of 1436 [71]. By applying the limit state function, $g = R - S$ **(**where R is an experimental value and S is data from the simulation), two classes are generated; if g is greater than zero, the class is "class A", otherwise its class is "class B". Class B refers that solder joint system will be discarded due to propagation of crack. Figure 4.12 depicts PNN with the solder joint example. PNN does not require a training algorithm, so this process is remarkably simple when compared to that of ANN. In Table 4.9, the P_f is estimated by the limit state function, while the results in the second row are the probability of failure by PNN with IndFeaT. To check the accuracy of this process, MCS is used to estimate P_f. If MCS has a large number of samples, P_f of MCS will be decreased. The reduced P_f is more accurate. In this example, a PNN with 1,000 samples and MCS with 10,000 samples are compared. In conclusion, reducing size of data the data set leads to lower classification accuracy, since P_f of IndFeaT and PNN is bigger than P_f of PNN. However, the values of P_f of IndFeaT and MCS are similar, indicating that this process performed well. It is

confirmed that feature selection diminishes not only the uncertainty of data, but also the probability of failure for fatigue life.

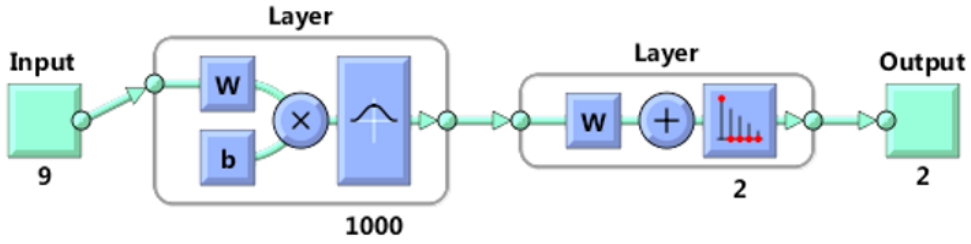


Figure 4.12. Schematic of Probabilistic Neural Network

Table 4.9. Probability of Failure by PNN and MCS

	PNN with 1,000 samples	**MCS with 10,000 samples**	***Difference of P_f between PNN and MCS (%)***
Probability of Failure of Original Data	0.0251	0.0256	1.99%
Probability of Failure of Reduced Data by IndFeaT	0.0276	0.0281	1.81%
Difference of P_f between original and reduced data (%)	9.96%	9.38%	

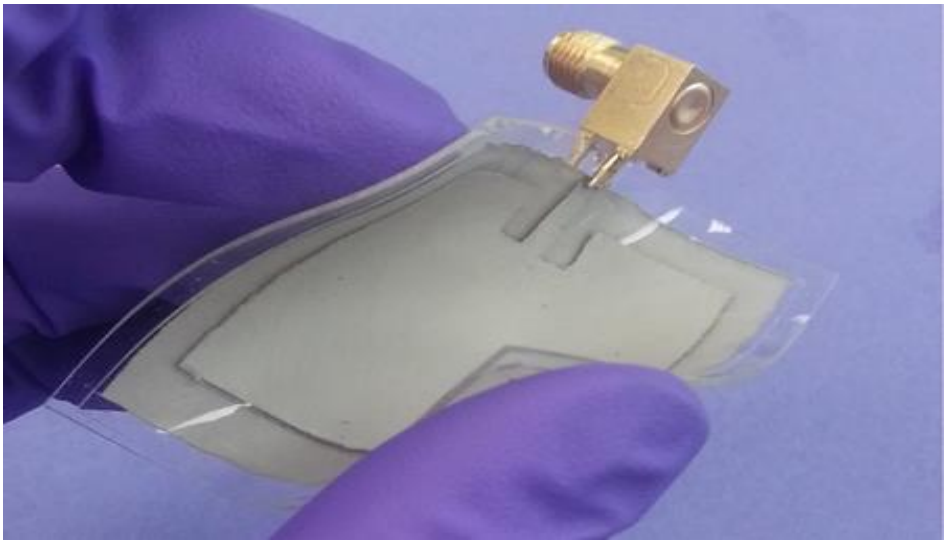
P_f of IndFeaT is bigger than P_f of the original due to elimination of features. However, P_f of IndFeaT and MCS are almost the same. Efficacy is guaranteed.

4.3. Stretchable Antenna Example

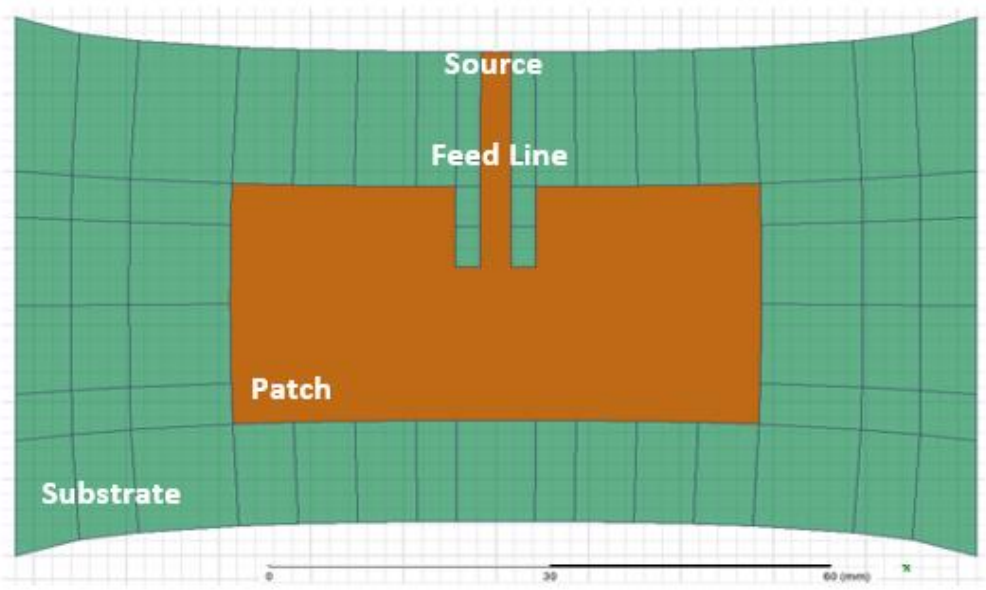
4.3.1. Problem Description

Stretchable patch antennas are composed of a substrate, patch, feed line, ground, and a source. Fabricating a substrate with constant thickness for stretchable antennas is uncommon because the substrate should be fabricated manually by engineers [72]. Therefore, a substrate has to be modeled taking into consideration its varying thickness.

Information regarding the geometry, varying thickness, and properties of the substrate is depicted in Figure 4.13 - 4.14, and Table 4.10.



(a) Experimental Frequency Analysis of Stretchable Patch Antenna [3]



(b) FEM Simulation of Stretchable Patch Antenna under Deformation

Figure 4.13. Geometry of Stetchable Patrch Antenna

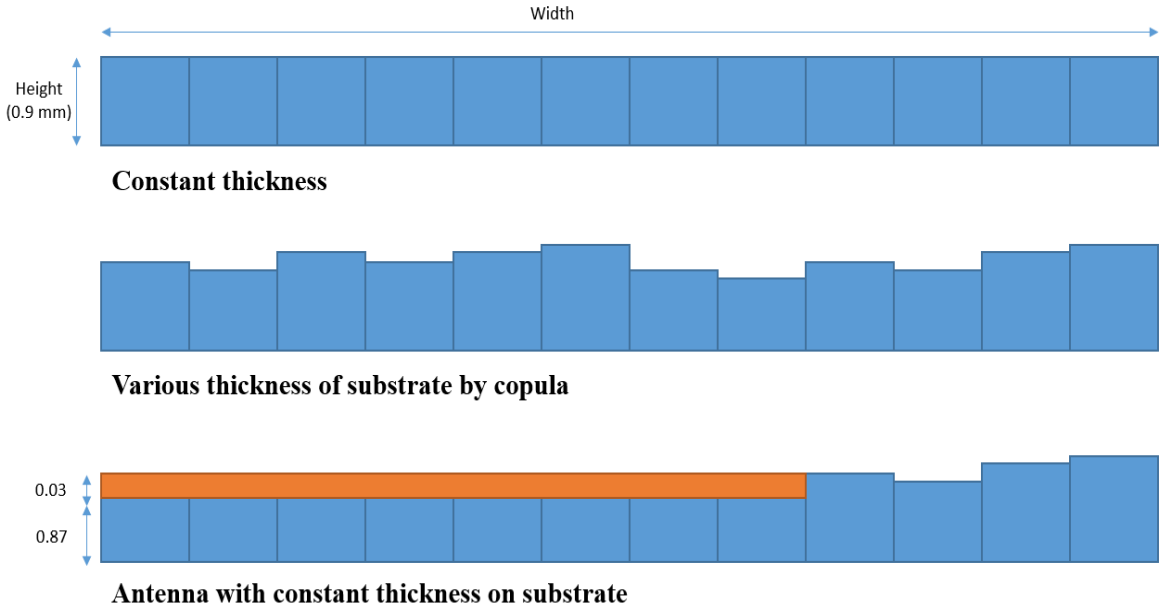


Figure 4.14. Stretchable Antenna with Varying Thickness

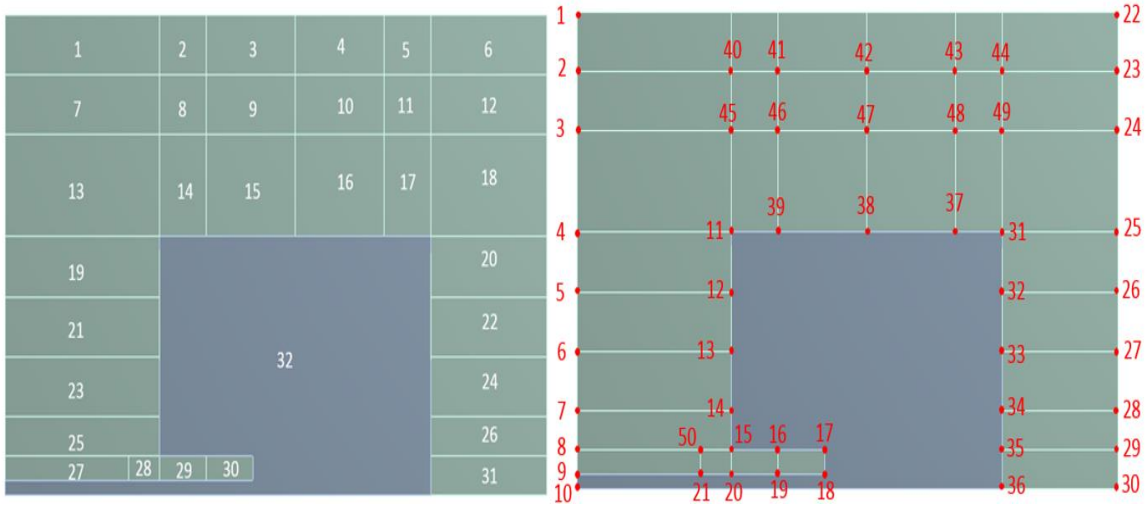
Table 4.10. Properties of Stretchable Patch Antenna

	Young's Modulus (Mpa)	Thickness (mm)	Width (mm)	Length (mm)
Substrate	1.32	$\mu = 0.09$, $COV = 0.1$	70	80
Patch	0.0124	0.03	35	43
Feed Line	0.0124	0.03	32	2.5
Ground	1.32	0.05	70	80
Source (surface)			$\mu = 0.09$, $COV = 0.1$	2.5

	Permittivity	Conductivity (S/cm)	Dielectric Loss Tangent	Magnetic Loss Tangent (kg/m^3)
Substrate	3	0	0.01	0.0001
Patch	1	1.51e4	0.01	0.001
Feed Line	1	1.51e4	0.01	0.001
Ground	1	1.51e4	0.01	0.001
Air	1	0	0	0

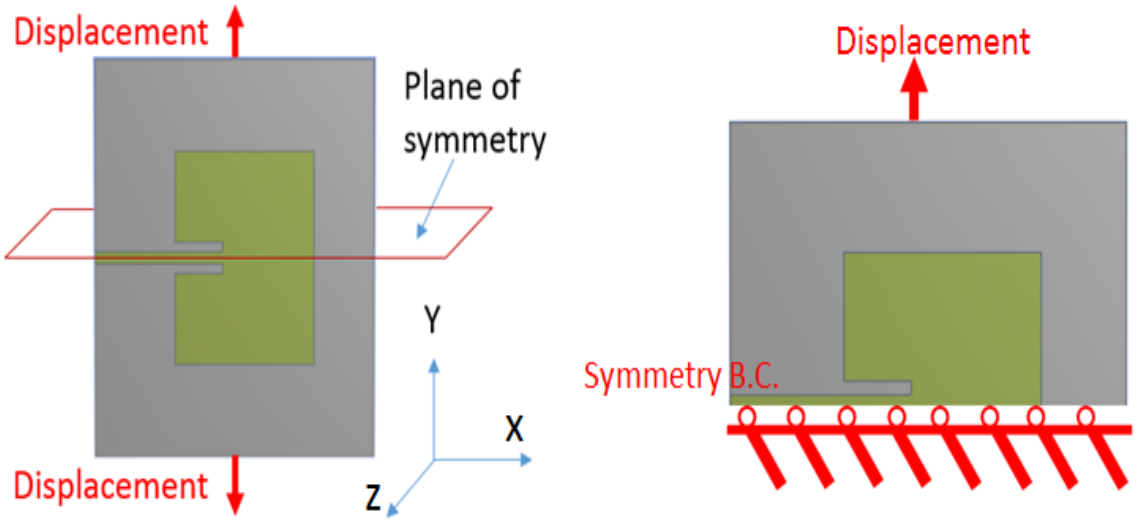
The most important goal of the analysis of the stretchable patch antenna with varying thickness is to confirm that the antenna can maintain a reliable frequency range when contracted and relaxed. There are two main points to check for an allowable frequency range.

- Mechanical behavior: The antenna is pulled to get information about deformations. The deformation is calculated through a tensile test. Specifically, both ends of the antenna are put under tension. The antenna is assumed to be symmetric because both ends of the antenna deform symmetrically from the center. This symmetry reduces the size of the finite element model, which, in turn, reduces the time and cost of the finite element analysis. The deformed shape of the antenna is represented as the *x* and *y* coordinates.



(a) 31 Parts with Different Thickness (b) 50 Coordinates

Figure 4.15. Schematic of Stretchable Antenna



(a) Tensile Test Boundary Condition (b) Tensile Test Symmetry Boundary Condition

Figure 4.16. Boundary Condition of Stretchable Antenna

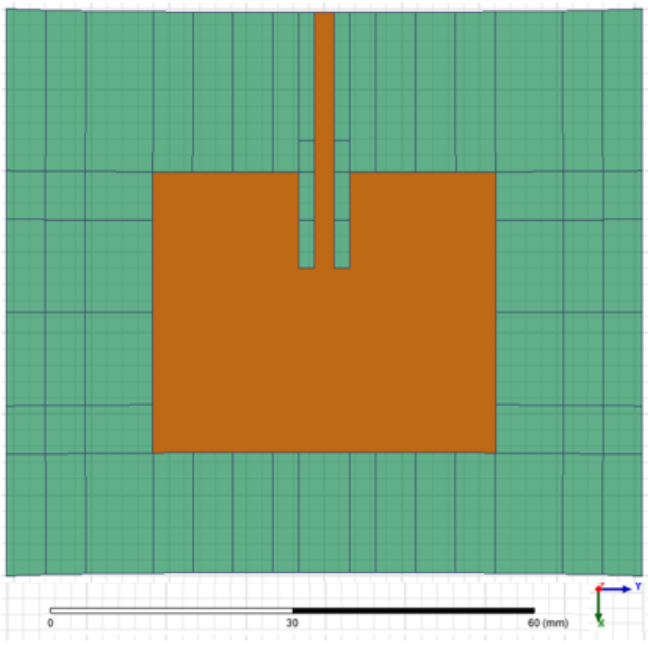
In other words, symmetry is assumed for the boundary condition of the stretchable patch antenna, when

1) The geometry is symmetric

2) Boundary conditions (force and constraints) are symmetric.

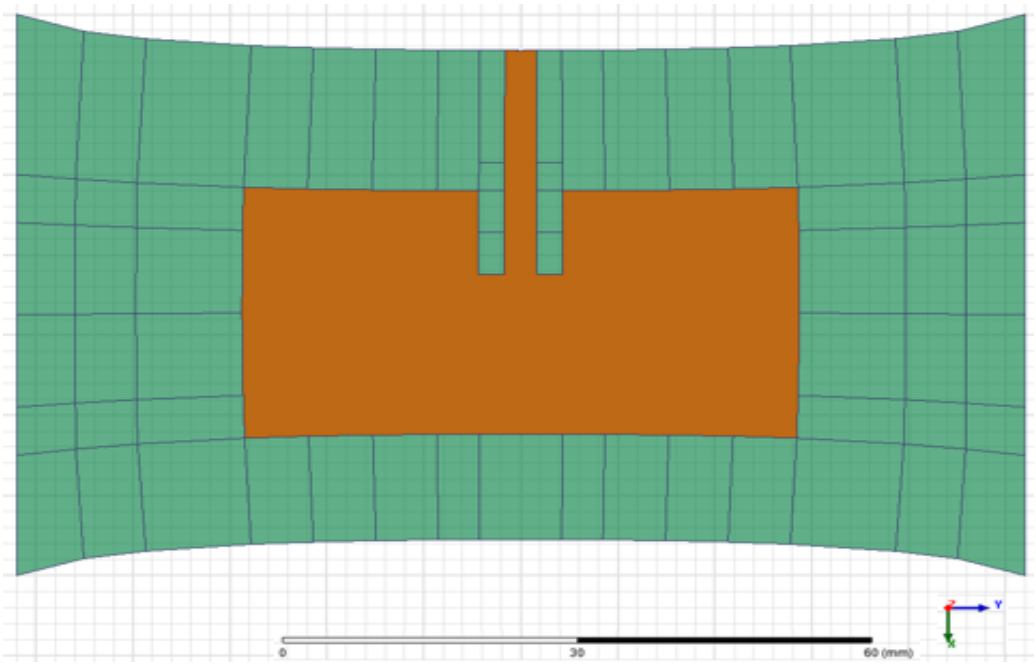
In this example, the antenna thickness is in the *Z* direction (out of plane direction). The geometry of the antenna is symmetric along the *X* axis in the middle. Since a tensile test is done, the loading condition on the antenna is also symmetric (two equivalent loads or displacements at each end are applied). Since these two conditions are satisfied, half of the antenna can be assumed to have symmetrical boundary conditions. The *XZ* plane in the middle along the *X* axis is called "plane of symmetry". The general rule for symmetrical boundary conditions is that the displacement vector component perpendicular to the plane of symmetry is zero and the rotational vector components parallel to the plane of symmetry are zero. Let U_x , U_y , and U_z denote the displacement components in the *X*, *Y*, and *Z*

directions on the plane of symmetry, respectively (i.e. on the bottom edge). Then, the displacement perpendicular to the plane of symmetry is $U_y = 0$, while U_x and U_z are free (i.e. displacement in X and Z directions are allowed). Let R_x, R_y, and R_z denote the rotational components about the X, Y, and Z directions on the plane of symmetry, respectively (i.e. on the bottom edge). Then the rotational components parallel to the plane of symmetry are R_x=0 and R_z=0, while R_y is free. Figure 4.17 shows a deformed antenna with 1mm and 12mm thickness.



(a) 1mm Displacement

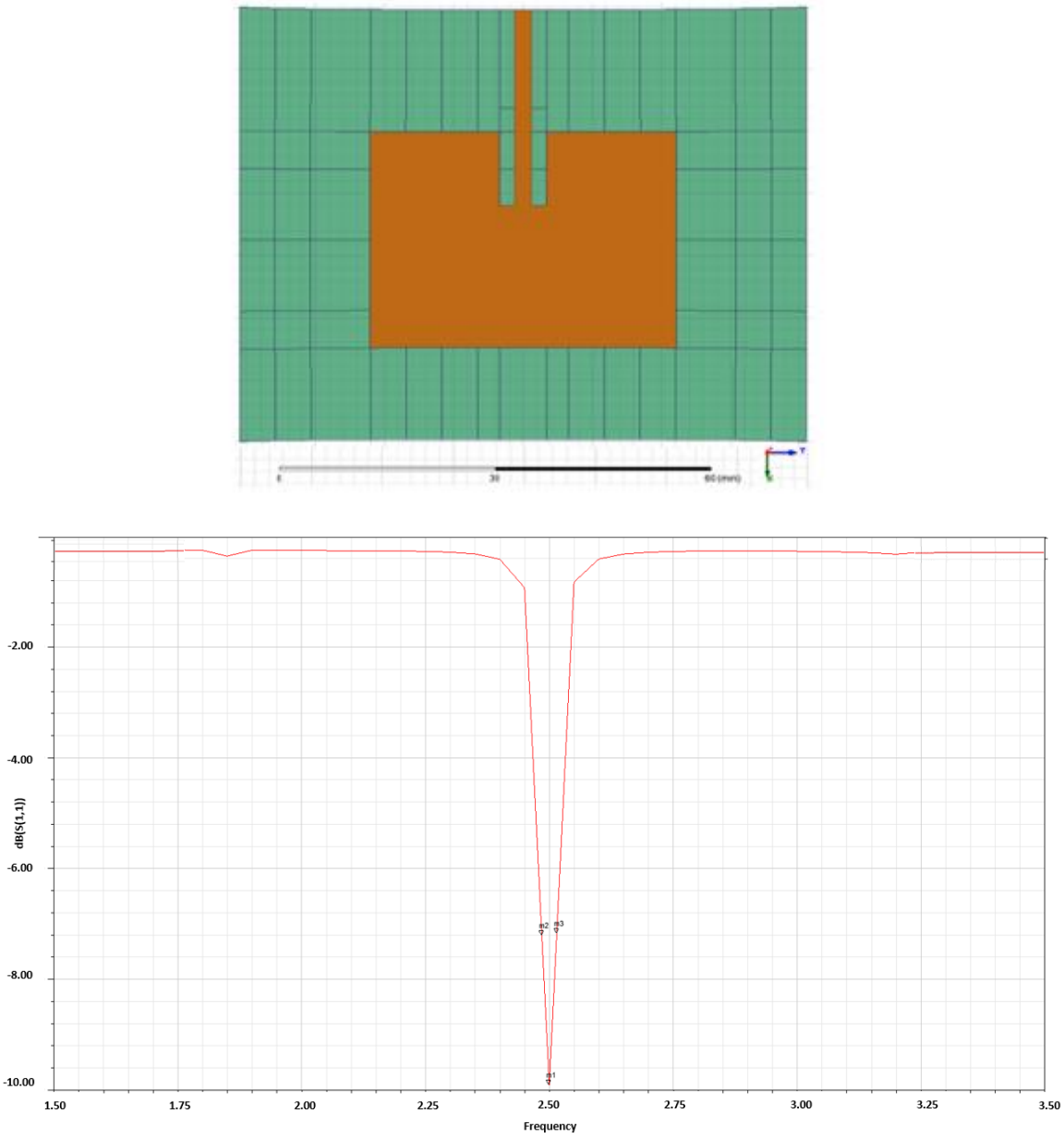
Figure 4.17. Deformed Geometry of Antenna



(b) 12mm Displacement

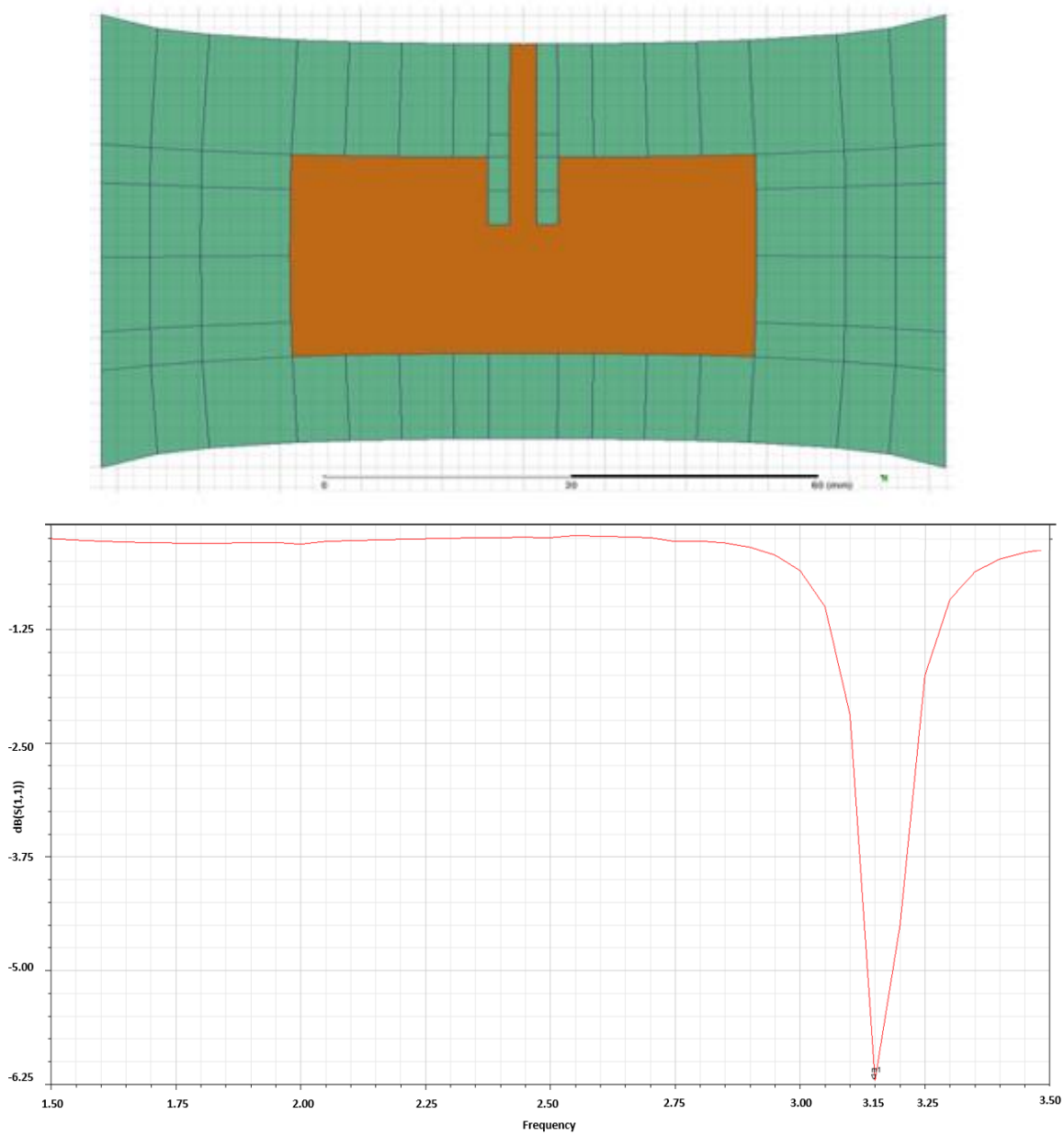
Figure 4.17. Deformed Geometry of Antenna (Continued)

- Electrical analysis: This process investigates if the deformed antenna affects the variation of resonance frequency. The deformation on the antenna will be accepted as an allowable deformation if the frequency of the antenna stays in a reliable range (3dB frequency). On the other hand, if the deformed antenna is out of the suitable range, the antenna model will be rejected. After validating the first step, frequency will be calculated by HFSS software using the coordinates of the deformed model. Figure 4.18 show that how electrical analysis with HFSS can investigate if deformed antenna affects the variation of resonance frequency.



(a) Stretchable Patch Antenna under 1mm Deformation and Its Resonance Frequency

Figure 4.18. Resonance Frequency of Stretchable Antenna under Deformation



(b) Stretchable Patch Antenna under 12mm Deformation and Its Resonance Frequency

Figure 4.18. Resonance Frequency of Stretchable Antenna under Deformation (Continued)

4.3.2. Generation of Varying Thickness

The Gaussian copula is only used to generate random thicknesses with a matrix of correlation parameters with a mean of 0.5 and COV of 0.2, because the thickness of the patch antenna and substrate are not always highly correlated or uncorrelated. Thirty-two

different thicknesses of the substrate are generated, and a patch with a constant thickness of 0.03mm is modeled. The redundancy of the original thickness is calculated, as shown in Table 4.11. One hundred twenty-one different values of displacement are tested on the end of the antenna, with a range of 0mm to 12mm. The displacement is calculated by the equation below:

$$strain = \frac{\Delta L_D}{L_o} = \frac{12mm}{40} = 30\%$$

where ΔL_D represents the applied displacement and L_o represents original length of the substrate. This example has a range of 0% (0mm displacement) to 30% (12mm displacement). The X and Y coordinates are separately estimated for regression, since dimension reduction and ANN take a single variable as output data. This example focuses on estimating the coordinates when displacement is applied to the antenna with varying thickness. The input for dimension reduction and ANN represents the thickness, while the output is assumed as each coordinate. One hundred twenty-one values of displacement are used to more accurately train the original 32 values of thickness.

4.3.3. Dimension Reduction of Varying Thickness

Based on Figure 4.15, each part of the antenna has a different thickness, and the dimension of the thicknesses has to be truncated. In this example, an antenna has 32 parts; 31 parts are for the substrate and 1 is for the patch. X and Y coordinates can be obtained from the deformed antenna. For dimension reduction, this example will use feature extraction because varying thicknesses should represent information as coordinates, rather

than having each thickness be estimated individually by reducing redundancy. Thus, feature extraction, such as PCA and AE, reduces the redundancy of a set of complex thicknesses. Redundancy comparison confirms if the irrelevance between the original data and truncated data is effectively diminished. Table 4.11 reveals that the redundancy of the thickness is minimized from 3.432 to 1.708 (PCA) and 1.926 (AE). In order to maintain 90% of the information from the thickness, PCA selected 14 eigenvalues and AE took 20 hidden neurons. The reconstruction error by AE is smaller than that by PCA, since AE takes more hidden neurons; in linear analysis, the hidden neurons of AE are regarded as eigenvalues. Reconstruction error and Redundancy error are calculated by Equation 4.1.

Table 4.11. Information of Results from Feature Extraction

	Number of Eigenvalues	Reconstruction Error (%)	Redundancy (Redundancy Error %)
Original Thickness	0	0	3.432
Original Thickness with PCA	14	2.51	1.708 (50%)
Original Thickness with AE	20	2.37	1.926 (43.91%)

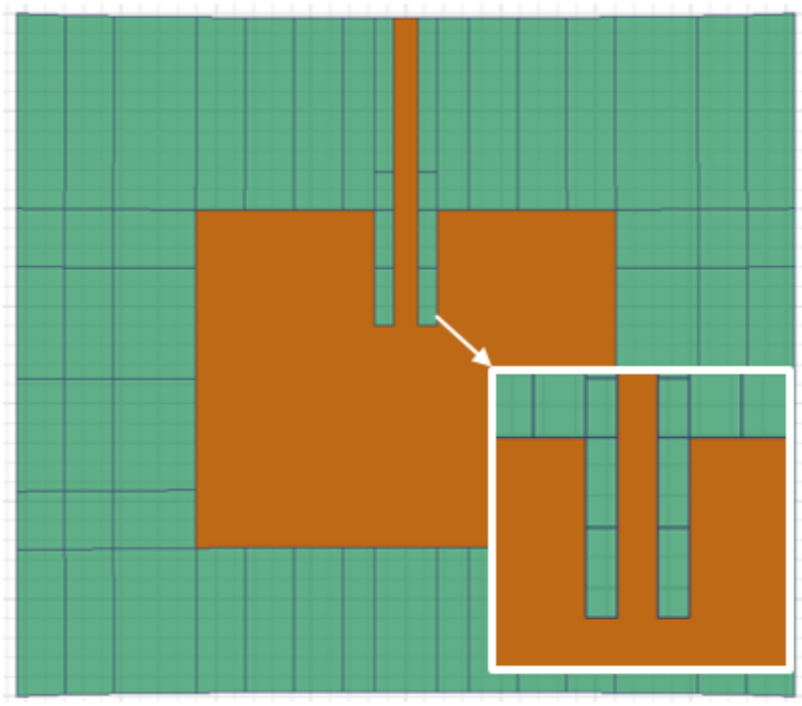
4.3.4. Artificial Neural Network for Predicting Antenna Deformation

In the next step, ANN will be utilized using information gathered from the varying thicknesses, the strain as input data, and the coordinates of the responses from the model. Table 4.12 presents the prediction error of ANN. For *X* coordinates, ANN has an error of 3.27%. The thickness used in PCA takes an error of 6.28%, while that of AE has an error of 5.41%. For the *Y* coordinate, considering the ANN's own error of 5.32%, the prediction error of PCA and AE are acceptable. The prediction error of the ANN was calculated by

Equation 4.1. Therefore, dimension reduction has been effectively performed. The antenna is modeled from the predicted coordinates as shown in Figure 4.19.

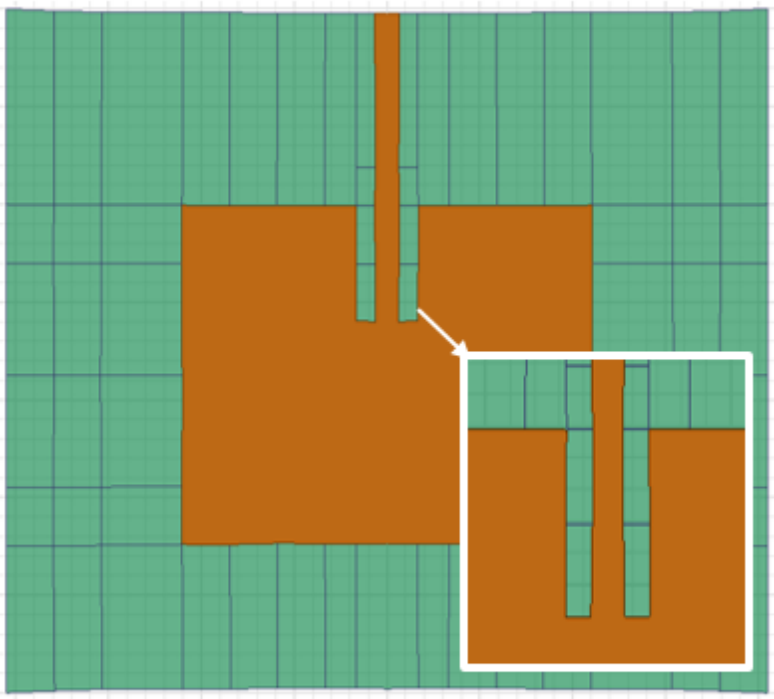
Table 4.12. Prediction Error of ANN

Prediction Error of ANN (%)	X Coordinates	Y Coordinates
Original Thickness	6.28	7.43
Original Thickness with PCA	3.27	5.32
Original Thickness with AE	5.41	7.17

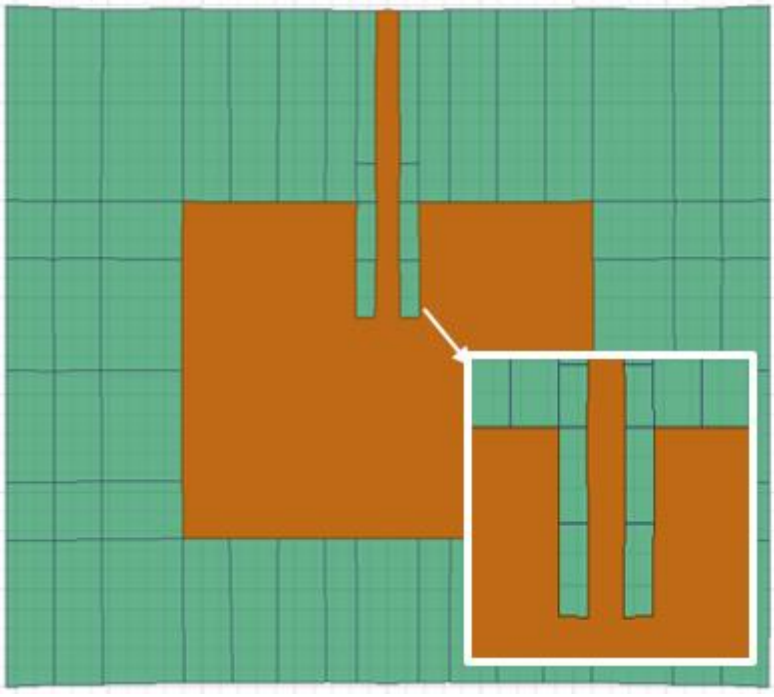


(a) 1mm deformed antenna

Figure 4.19. Shape of Deformed Antenna

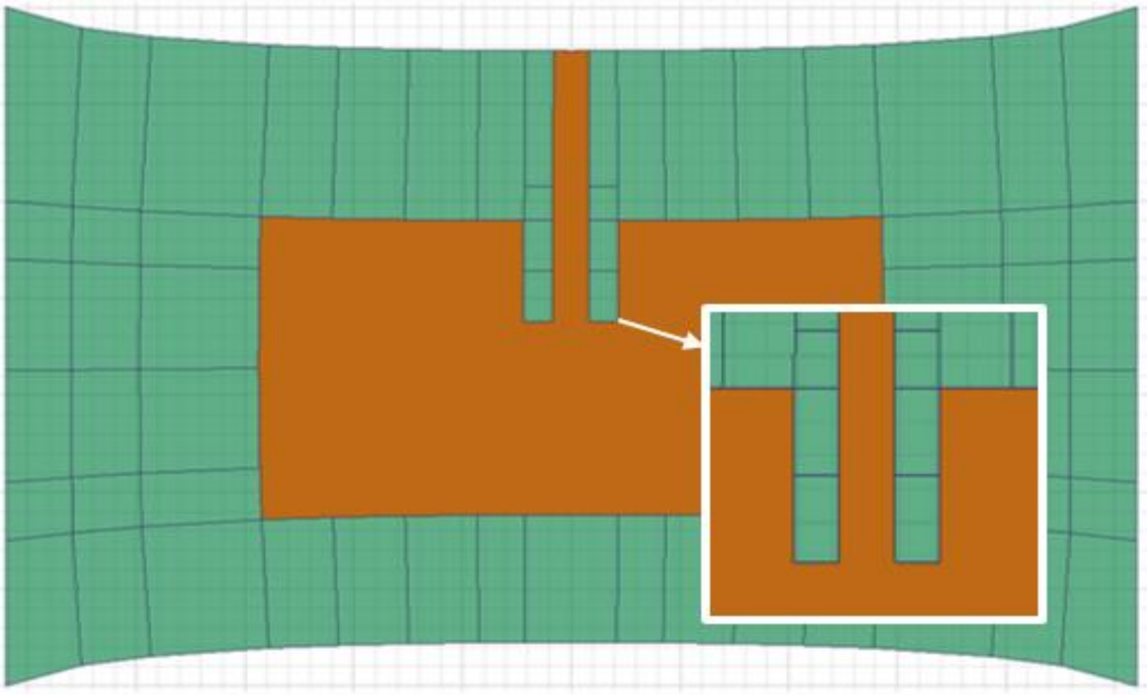


(b) 1mm deformed antenna by AE

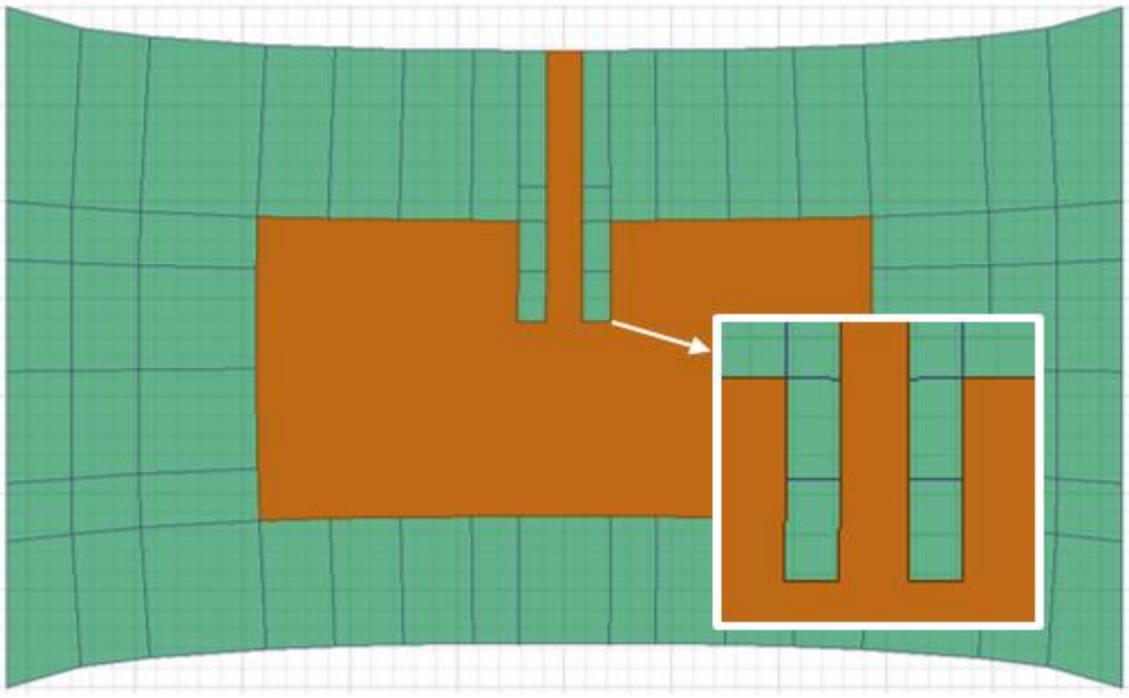


(c) 1mm deformed antenna by PCA

Figure 4.19. Shape of Deformed Antenna (Continued)

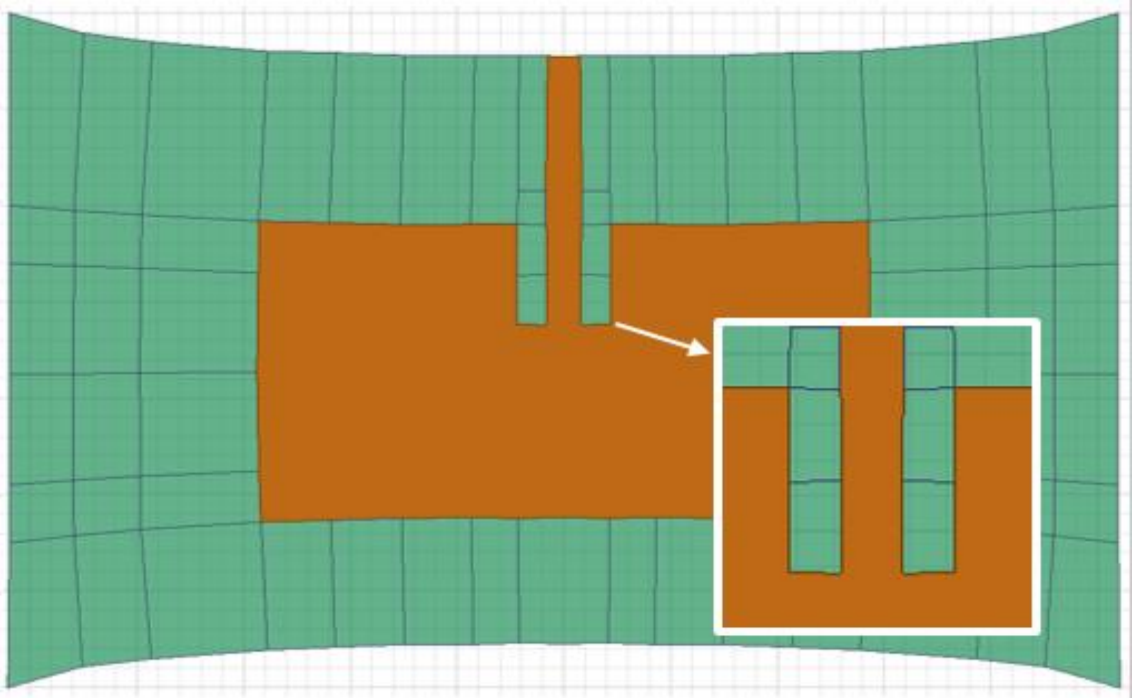


(d) 12mm deformed antenna



(e) 12mm deformed antenna by AE

Figure 4.19. Shape of Deformed Antenna (Continued)



(f) 12mm deformed antenna by PCA

Figure 4.19. Shape of Deformed Antenna (Continued)

4.3.5. Dimension Reduction of Each Coordinate

The reconstructed antenna's resonance frequency can be estimated by the coordinates predicted by feature extraction and ANN. For the second process, each of the 50 coordinates is used as input data, and 121 different frequencies represent the output data. This data set is a high dimensional data set, so the dimension has to be reduced. For dimension reduction, feature selection is considered, since this process should estimate each significant coordinate in order to re-design an antenna based on the significant coordinates. Moreover, with the selected coordinates, frequency will be gauged to see if it can be classified into a reliable bandwidth. IndFeaT makes a new subset by selecting significant coordinates exceeding the "Significance value" of 3 to obtain the highest accuracy. Even though the significance value of 2 can be selected, it keeps most of the

original coordinates. Thus, to take the most important 5 coordinates, the value of 3 had to be chosen, after applying the IndFeaT method, 4 meaningful coordinates are newly selected as a reduced data set. Entropy estimates whether uncertainty is minimized. In Table 4.13, the value of redundancy after IndFeaT is smaller than that of the original coordinates indicating uncertainty has been reduced. Redundancy reduction can be estimated by Equation 3.1. Figure 4.20 shows which significant coordinates were selected. In this figure, a blue circle represents a meaningful coordinate.

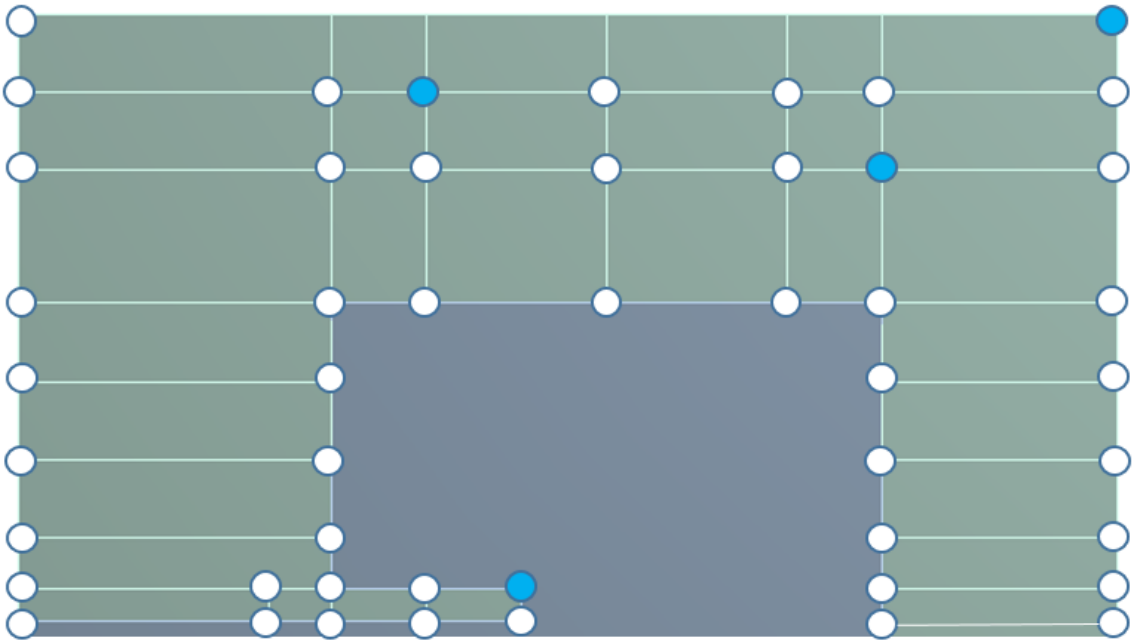


Figure 4.20. Selected Significant Coordinates

Table 4.13. Redundancy Reduction

	Redundancy	**Redundancy Reduction (%)**
Original Coordinates	12.5689	13.9567%
Original Coordinates with IndFeaT	10.8147	

4.3.6. Probabilistic Neural Network for Classification of Antenna Frequency

In the next step, a reliable range for the resonance frequency is estimated using the new coordinates. The range is expressed using 3dB frequency based on the non-deformed antenna. In Figure 4.21, the non-deformed antenna has a resonance frequency of 2.5 GHZ. According to the resonance frequency, a reliable range is 2.4849 GHZ (m_2) to 2.5151 GHZ (m_3). For classification the limit state function is estimated. The limit state function is defined as $g(.)= r - s$ (where r represents 2.5 GHZ and s represents the simulation frequency). Resonance frequency should stay in a frequency of 3dB. If g has negative value, the stretchable patch antenna system will be in class B. In this problem, if the system is in class B, it should be discarded or re-designed due to unstable resonance frequency.

$g(.) \geq |2.5151 - s| = 0.0302$: Class A

$g(.) \leq |2.5151 - s| = 0.0302$: Class B

Based on the limit state function, the Monte Carlo Simulation (MCS) is conducted to validate the efficacy of the proposed method. MCS has low probability of failure (P_f) as the number of samples increases. This example uses 121 and 10,000 samples for high accuracy and low probability of failure. MCS is operated with 10,000 samples, since the range of frequency is evenly subdivided. Comparing P_f of the of PNN classification with that of MCS, the results reveal that the classification is accurate because the P_f value obtained by classification is nearly equal to that obtained by MCS. The reduced coordinates have a P_f value of 0.3016, which indicates a 7.37% increase in the P_f value from the original coordinates. The difference is acceptable. Moreover, P_f of the reduced coordinates is close

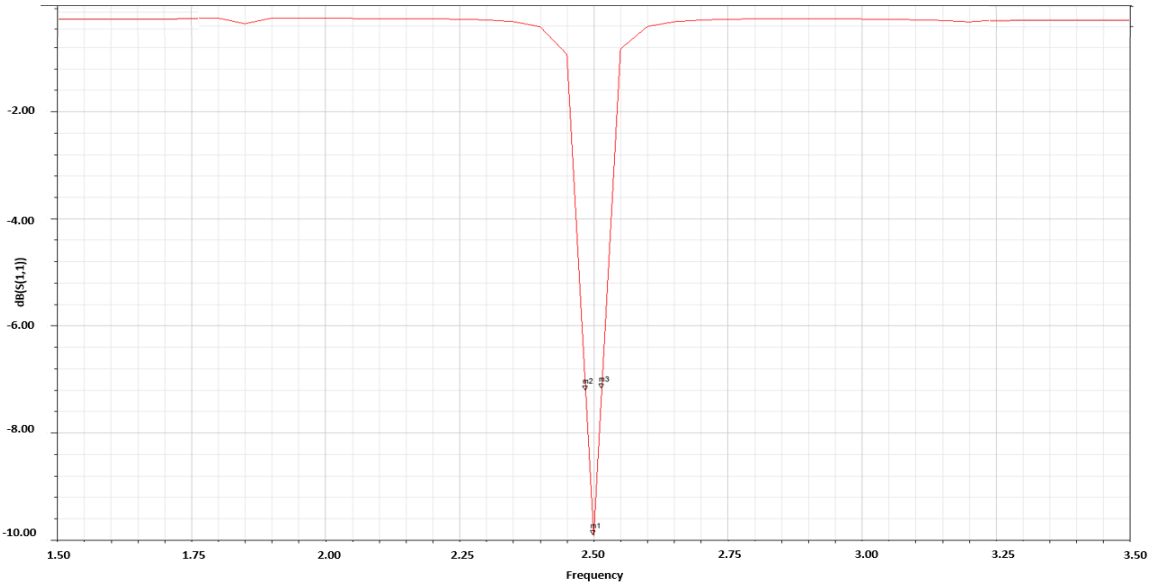
to that of MCS, indicating that the feature selection and classification have been accurately performed. P_f by PNN and MCS are listed in Table 4.14. Each value is calculated by Equation 4.1.

Table 4.14. Probability of Failure by PNN and MCS

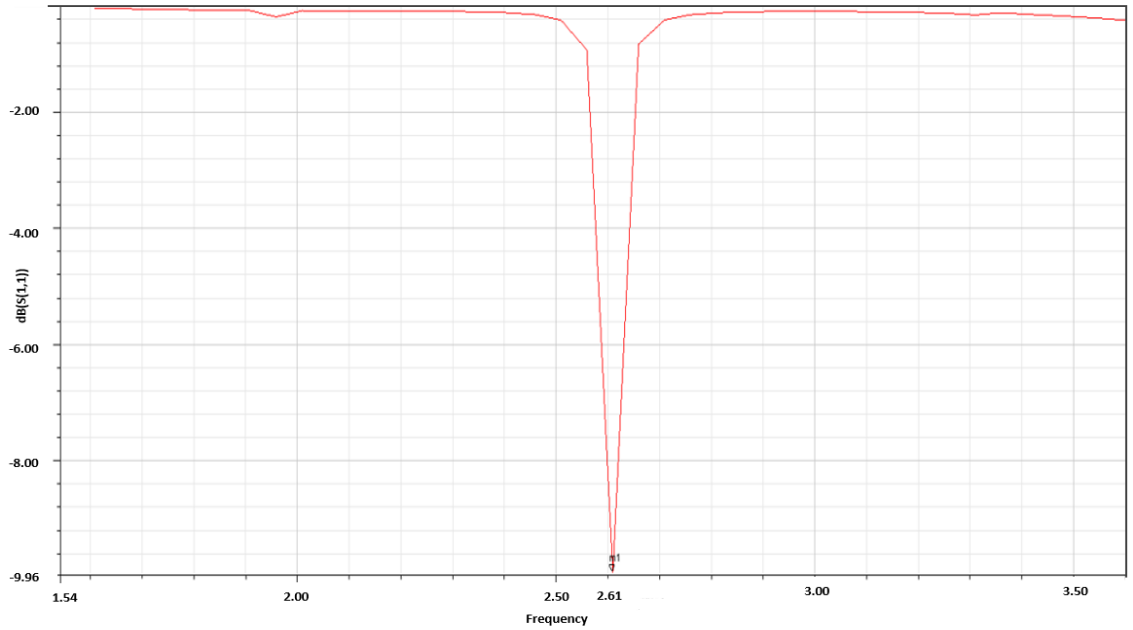
	PNN with 121 samples	**MCS with 10,000 samples**	***Difference of P_f between PNN and MCS (%)***
Probability of Failure of Original Data	0.2809	0.3035	8.05%
Probability of Failure of Reduced Data by IndFeaT	0.3016	0.3247	7.66%
Difference of P_f between original and reduced data (%)	7.37%	6.98%	

4.3.7. Validation of Efficacy of Proposed Framework

First, each resonance frequency from the deformed antenna is estimated. The antenna remains acceptable until it undergoes a displacement of 3.1mm while maintaining 2.51 GHZ. However, variation in frequency is detected starting from a displacement of 3.2mm. The maximum displacement of 12mm has a resonance frequency of 3.2 GHZ. Hence, the allowable maximum displacement should be 3.1mm. In the last step, resonance frequency of the original coordinates and new coordinates from feature extraction and feature selection are compared. In Figure 4.22, feature selection's frequency has relatively lower accuracy. This is because feature selection directly removes uninformative coordinates, while feature extraction only keeps some principal components and reconstructs the coordinates with the same dimensions as the original coordinates. Without reducing the dimension of the coordinates, feature extraction can reduce redundancy. In conclusion, the proposed method accurately predicted resonance frequency.

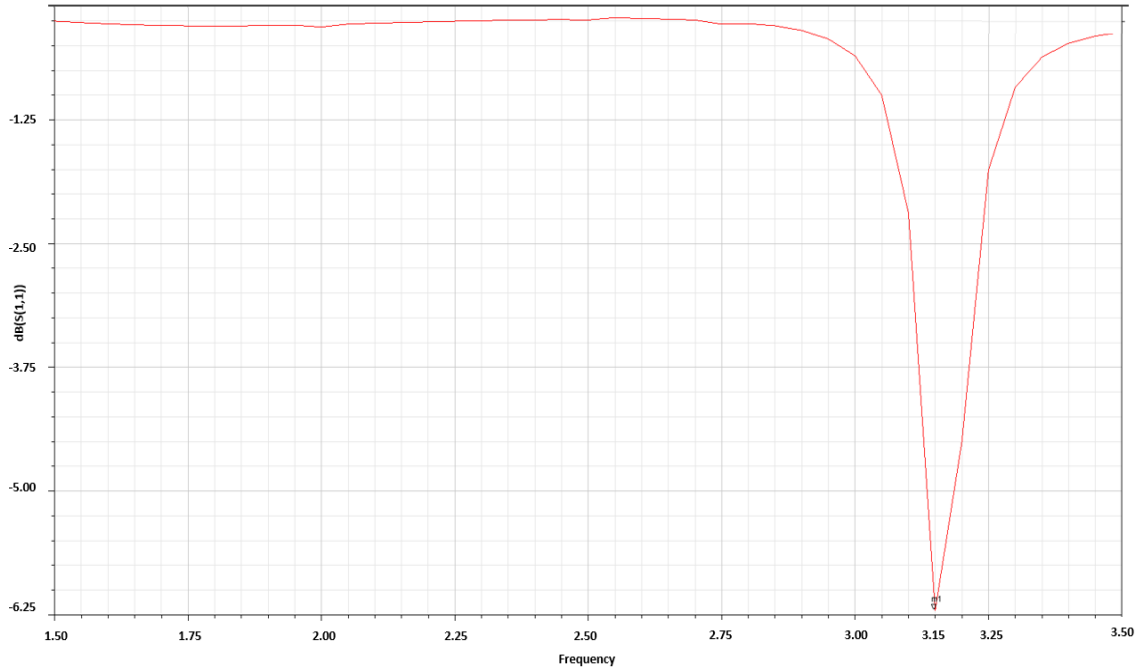


(a) Resonance Frequency of Deformed Antenna with 0mm Displacement



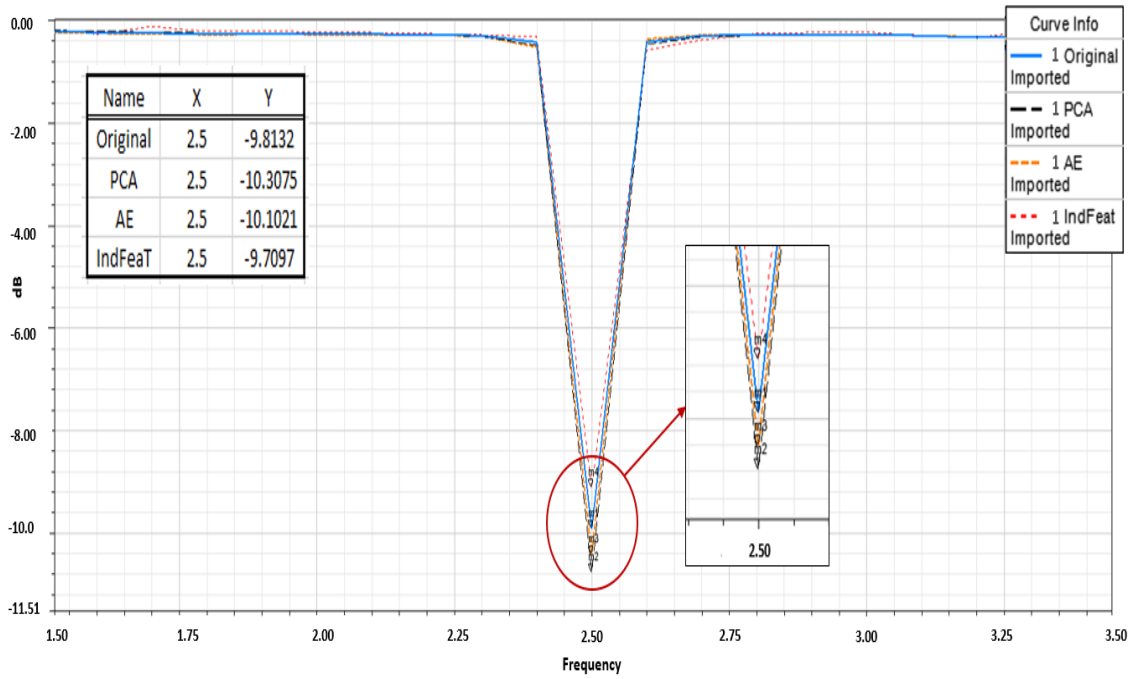
(b) Resonance Frequency of Deformed Antenna with 3.2mm Displacement

Figure 4.21. Resonance Frequency of Deformed Antenna



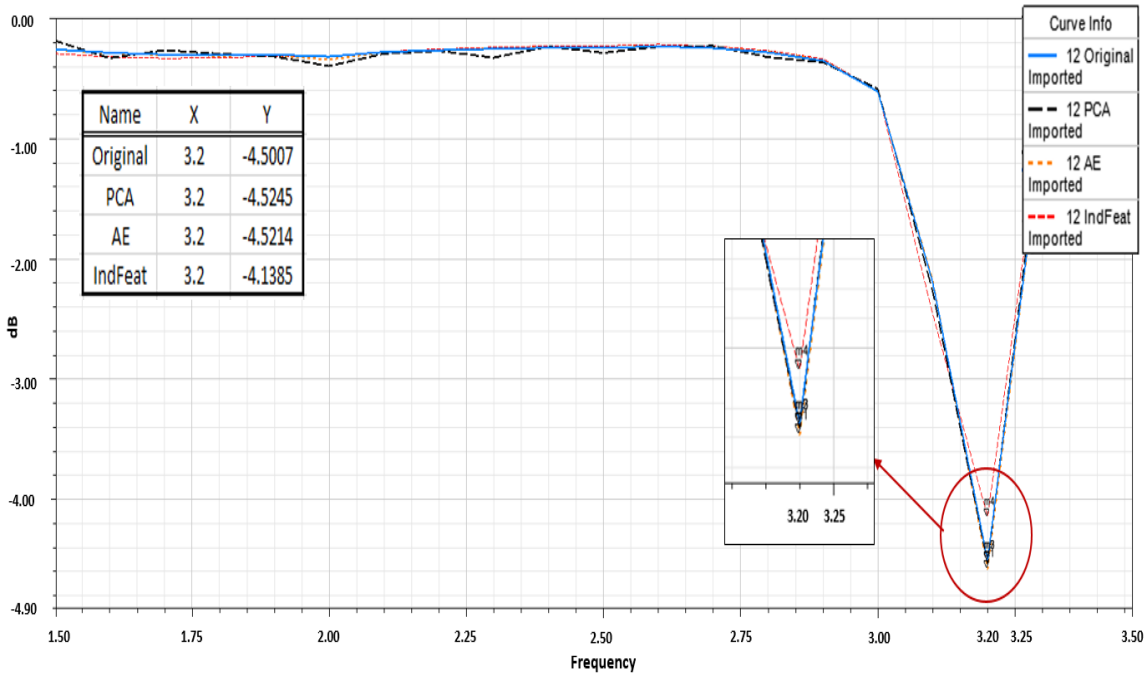
(c) Resonance Frequency of Deformed Antenna with 12mm Displacement

Figure 4.21. Resonance Frequency of Deformed Antenna (Continued)



(a) Validation of Proposed Method with Resonance Frequency for 1mm Displacement

Figure 4.22. Validation of Proposed Method with Resonance Frequency



(b) Validation of Proposed Method with Resonance Frequency for 12mm Displacement

Figure 4.22. Validation of Proposed Method with Resonance Frequency (Continued)

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1. Summary

An integration of technical knowledge from multidisciplinary engineering domains is essential for developing future engineering products. Accordingly, product development that reflects significant multidisciplinary design criteria with reliable accuracy is required. However, modeling and simulating multidisciplinary engineering systems are challenging due to the complexities in interactions between various input parameters and other complex behaviors. These complexities may result from uncertainty or redundancy in the input parameters, and are found to negatively affect predictive accuracy of complicated systems. Accurate estimation of risk and reliability of such complicated systems is not possible without correctly capturing critical input parameters and propagating corresponding uncertainties. Moreover, accurate representations of multivariate phenomena are required by multidisciplinary engineering systems. Thus, it is urgently needed to develop a framework which can handle multivariate phenomena of complex engineering systems under uncertainty.

For these reasons, the proposed research developed a framework that can accurately capture and model input parameters and predict responses. Specifically, to effectively demonstrate interdependency of various input parameters, the copula function is used. By using copula, every marginal distribution of input parameters are divided and represented into a joint distribution. Modeling based on the copula function will be more realistic and will draw accurate predictions of the response. However, many input parameters are

sophisticated and often fairly correlated. The problems are prime causes of inaccurate response predictions. The problems are connected with uncertainty and redundancy. For these reasons, the proposed method applies dimension reduction techniques such as the Principal Component Analysis, Auto-encoder, and Independent Features Test to reduce these concerns. Reduced input data sets play a crucial role in predicting precise responses without large losses of information. Artificial Neural Network and Probabilistic Neural Network are used to forecast and classify responses of a complicated system accurately.

In Chapter 2, literature backrounds of terms of the proposed method are reviewed, including the copula fucntion, dimension reduction method, feature extraction, and feature selection. The concepts of entopry and neural network are also disucssed. Chapter 3 provides the framework and functionality of the proposed method with a simple numerical example. The algorithm is operated by MATLAB. Chapter 4 explains how the method can be used to three specific examples. The first example is an application of the method in a 3-D cantilever beam. The next example is a solder joint analysis in terms of feature selection with classification. For the last example, a stretchable antenna is used to show entire process of the method.

5.2. Conclusion

Based on research questions, the proposed method shows great potential for designing a system and predicting its response in multidisciplinary engineering domains.

Research Question 1

How can we accurately represent the correlated random quantity of a complicated multidisciplinary engineering system?

The copula function was utilized to realistically design a system with an understanding of marginal distributions. In first example, correlated and uncorrelated 30 Young's moduli were randomly generated by the Gaussian copula function. A joint distribution combined the total interdependency of the moduli and was represented as a data set. In the second example, 9 conditions consisting the fatigue life of a solder joint were separated to assign each special properties by copula. For the last example, copula played a vital role in representing varying thicknesses of stretchable antenna. It is impossible to make a constant substrate since this process is hand operated by engineers. Thus the Gaussian copula function was employed to generate different thicknesses to consider the interdependency of each thickness.

Research Question 2

How can the prediction or classification procedure be made computationally efficient for a multidisciplinary engineering system?

The proposed method resolved complexities leading to uncertainty or redundancy by adapting the dimesionality reduction method. Reduced random variables had lower values of complexities so that the variables drew accurate predictions and classifications of the response. Uncertatinty reduction was also checked by entropy. In the first examaple, feature extraction such as PCA and AE effectively reduced the input data's complexity without large loss of information by using eigenvalues. Dimensionailty reduction was applied to minimize the principal components' size, not original data's dimension. Redundancy comparison confirmed irrelevance was reduced for correlated and uncorrelated data. In the next step, the reduced new data sets were used as another input to

predict an accurate response by ANN. Prediction error indicated that the reduced data with low redundancy led to reliable results. In the second example, the feature selection method, IndFeaT, figured out significant properties regarding fatigue life of a solder joint. A new subset was also estimated by entropy, and the result indicated IndFeaT decreased the complexity of the original data sets. To validate if the new data sets can draw precise claissfication or not, PNN was employed. The value of the probability of faliure by PNN and MCS confirmed that the subset successfully derived an accurate classification. For the last exmaple, the dimesion of varying thicknesses with low redundancy was obtained by feature extraction. ANN was applied to predict accurate coordinates of the deformed shape of the antenna. Redundancy comparison and prediction rate also confirmed whether the process worked accurately. IndFeaT selected significant information from the coordinates to make a reduced dimensional subset. Entropy checked if uncertainty was reduced. Based on the subset, the antenna's reliable frequency range was effectively classified by PNN.

5.3. Contributions

In this research various important challenges have been stated. The specific contributions and their explanations are listed below:

1. Copula: The major contribution of the proposed framework is that it can represent most information from complex input data by using a copula function. While general statistical methods cannot accurately represent features of over two data sets, the framework with copula permits complicated and various features to be represented effectively and realistically. This advantage can lead to more accurate predictions of responses than that of a simplified model.

2. Dimension reduction: After representing the inputs' complexities, the dimension of the inputs became huge. It led the estimation and prediction of the complex behavior of the given engineering system to have large computational costs. Hence, the proposed framework suggests dimension reduction of input data sets to handle these problems.

    - Criteria to decide to use either feature extraction or feature selection: Experienced engineers typically might determine what dimension reduction algorithm should be utilized for a certain problem. However, inexperienced engineers are unfit to figure out the algorithms of dimension reduction. In this case, the proposed criterion used before running dimension reduction is required because it effectively suggests a clear distinction between feature extraction and feature selection. This results in accurate guidance and prediction of responses of complicated systems.

    - Feature extraction: Based on eigenvectors, feature extraction generates new data with reduced dimension and complexity. The small sized data shows a good performance as an input data for predicting accurate responses of engineering systems. Specifically, AE resolved two limitations of PCA: uncorrelated and non-linear data analysis.

    - Feature selection: By selecting a new subset of original data, feature selection showed how each significant feature in reduced data could be estimated. An analysis of each important feature led to accurate predictions of responses of given systems. Moreover, feature selection improved the speed of the learning algorithm compared to feature extraction and reduced computational cost.

3. Entropy / redundancy comparison: As a kind of criterion regarding uncertainty or irrelevance, **e**ntropy or redundancy comparisons between original data and reduced data expressed how dimension reduction worked well. A common method that can show the reduction of redundancy or uncertainty uses a matrix with correlation parameters. However, as the dimension of data increases, uncertainty or redundancy analysis by a correlation matrix is impractical. Entropy or redundancy comparison can be still employed in this situation since it provides information of uncertainty or irrelevance as a scalar value.

4. Neural Networks: Combining the feature extraction with ANN accurately represented an optimal response of the engineering systems. Unlike a general regression algorithm, ANN could enhance the accuracy of its prediction of response by controlling various parameters. Moreover, feature selection with PNN showed a better prediction and estimation for classification of reliability of a complex engineering system. Compared to ANN, PNN could reduce computational time since it has no feedback loss, leading to a faster learning rate.

5.4. Future Work

Even though some limitations of the proposed framework and examples have been stated, there are some suggestions to enhance accuracy of the framework and examples:

1. Analysis with data having non-Gaussian distribution:

Even though the copula function can generate many types of distributions, this thesis is only focused on Gaussian distribution. Each different copula function has

distinct tail properties, and PCA cannot work accurately for non-Gaussian distributions. Thus, the proposed framework will be applied to data sets with non-Gaussian distribution to check if this method can produce good accuracy in other distributions as well.

2. Non-linear analysis:

The proposed method and all examples are based on linear analysis. In future work, cantilever beams and stretchable antennas with multiple loading conditions will be estimated. The results will be also represented by the finite element method.

3. Different types of Auto-Encoder:

In this method, AE used a backpropagation algorithm to diminish the gradient. Only one AE, sparse AE, is employed. However, it is unlikely that AE with the backpropagation algorithm can work accurately for non-linear analysis or in a case where the data's uncertainty is not fairly reduced. In future work, Stacked AE or the Restricted Boltzmann machine will be examined to resolve the weakness of using the backpropagation algorithm.

4. Physical Experiment of Antenna Example:

The initial design of the presented antenna will be physically fabricated and tested. Then the performance of the physical data will be compared to that of the simulation results from the current research. Moreover, in the example of the stretchable patch antenna, the most informative coordinates are recombined by the feature selection method. Significant coordinates must have a vital influence on the deformation of the antenna. In future work, the antenna's shape near the significant

coordinates will be thicker and stronger so that the antenna can endure larger displacements. Moreover, to get better performance, an antenna should have S11 gain under 10dB. The example in this research has 9.96dB. Thus, in future work, the geometry of antenna will be regulated to get better S11 gain.

REFERENCES

1. NASA, *Autonomous Orbit Control Technology* 2014(NASA).

2. Li, J., H. Xu, T.T. Mattila, J.K. Kivilahti, T. Laurila, and M. Paulasto-Kröckel, *Simulation of dynamic recrystallization in solder interconnections during thermal cycling.* Computational Materials Science, 2010. **50**(2): p. 690-697.

3. Song, L., A.C. Myers, J.J. Adams, and Y. Zhu, *Stretchable and reversibly deformable radio frequency antennas based on silver nanowires.* ACS Appl Mater Interfaces, 2014. **6**(6): p. 4248-53.

4. Gardner, W.A., *Introduction to random processes: with applications to signals and systems*. 1986: Macmillan Pub. Co.

5. Choi, S.-K., R. Grandhi, and R.A. Canfield, *Reliability-based structural design*. 2006: Springer Science & Business Media.

6. Bellmam, R., *Dynamic ProgrammingPrinceton University Press*. 1957, Princeton.

7. Fan, J. and Y. Fan, *High dimensional classification using features annealed independence rules.* Annals of statistics, 2008. **36**(6): p. 2605.

8. Shannon, C., *A mathematical theory of communication, bell System technical Journal 27: 379-423 and 623–656.* Mathematical Reviews (MathSciNet): MR10, 133e, 1948.

9. Shlens, J., *A tutorial on principal component analysis.* arXiv preprint arXiv:1404.1100, 2014.

10. Blum, A.L. and P. Langley, *Selection of relevant features and examples in machine learning.* Artificial intelligence, 1997. **97**(1): p. 245-271.

11. Indyk, P. and R. Motwani. *Approximate nearest neighbors: towards removing the curse of dimension*. in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 1998. ACM.

12. van der Maaten, L.J., E.O. Postma, and H.J. van den Herik, *Dimension reduction: A comparative review.* Journal of Machine Learning Research, 2009. **10**(1-41): p. 66-71.

13. Fayyad, U. and R. Uthurusamy, *Evolving data into mining solutions for insights.* Communications of the ACM, 2002. **45**(8): p. 28-31.

14. Roesner, E.H., *Ars antiqua: organum, conductus, motet*. Vol. 5. 2009: Ashgate Pub Co.

15. Clemen, R.T. and T. Reilly, *Correlations and copulas for decision and risk analysis.* Management Science, 1999. **45**(2): p. 208-224.

16. Nelsen, R.B., *An introduction to copulas*. Vol. 139. 2013: Springer Science & Business Media.

17. Sklar, A., *Random variables, distribution functions, and copulas: a personal look backward and forward.* Lecture notes-monograph series, 1996: p. 1-14.

18. Schmidt, T., *Coping with copulas.* Copulas-From theory to application in finance, 2007: p. 3-34.

19. Cherubini, U., E. Luciano, and W. Vecchiato, *Copula methods in finance*. 2004: John Wiley & Sons.

20. Sklar, M., *Fonctions de répartition à n dimensions et leurs marges*. 1959: Université Paris 8.

21. Marcantoni, E., *Collateralized Debt Obligations: A Moment Matching Pricing Technique Based on Copula Functions*. 2014: Springer Science & Business Media.

22. Stein, M., *Copula Opinion Pooling in Asset Allocation.* 2008.

23. Höhndorfa, L., J. Sembiringa, and F. Holzapfela, *Copulas applied to Flight Data Analysis.* Probabilistic Safety Assessment and Management PSAM, 2014. **12**.

24. Venter, G.G. *Tails of copulas*. in *Proceedings of the Casualty Actuarial Society*. 2002.

25. Lin, G.D., X. Dou, S. Kuriki, and J.-S. Huang, *Recent developments on the construction of bivariate distributions with fixed marginals.* Journal of Statistical Distributions and Applications, 2014. **1**(1): p. 14.

26. Cover, T.M. and J.A. Thomas, *Elements of information theory*. 2012: John Wiley & Sons.

27. Kullback, S., *Information theory and statistics*. 1968: Courier Corporation.

28. Yentes, J.M., N. Hunt, K.K. Schmid, J.P. Kaipust, D. McGrath, and N. Stergiou, *The appropriate use of approximate entropy and sample entropy with short data sets.* Annals of biomedical engineering, 2013. **41**(2): p. 349-365.

29. Misra, N., H. Singh, and E. Demchuk, *Estimation of the entropy of a multivariate normal distribution.* Journal of multivariate analysis, 2005. **92**(2): p. 324-342.

30. Correa, C.D. and P. Lindstrom, *The mutual information diagram for uncertainty visualization.* International Journal for Uncertainty Quantification, 2013. **3**(3).

31. Spruyt, V. *The Curse of Dimension in classification*. 2014; Available from: http://www.visiondummy.com/2014/04/curse-dimension-affect-classification/.

32. Lee, J. and M. Verleysen. *Unsupervised dimension reduction: overview and recent advances*. in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. 2010. IEEE.

33. Liu, H. and H. Motoda, *Feature extraction, construction and selection: A data mining perspective*. 1998: Springer Science & Business Media.

34. Person, K., *On Lines and Planes of Closest Fit to System of Points in Space. Philiosophical Magazine, 2, 559-572*. 1901.

35. Hotelling, H., *Analysis of a complex of statistical variables into principal components.* Journal of educational psychology, 1933. **24**(6): p. 417.

36. Hotelling, H., *Relations between two sets of variates.* Biometrika, 1936: p. 321-377.

37. Girshick, M., *On the sampling theory of roots of determinantal equations.* The Annals of Mathematical Statistics, 1939. **10**(3): p. 203-224.

38. Girshick, M., *Principal components.* Journal of the American Statistical Association, 1936. **31**(195): p. 519-528.

39. Baldi, P. and K. Hornik, *Neural networks and principal component analysis: Learning from examples without local minima.* Neural networks, 1989. **2**(1): p. 53-58.

40. Malava, A., *Principal Component Analysis on Term Structure of Interest Rates.* Helsinki University of Technology Department of Engineering Physics and Mathematics Working Paper, 1999.

41. Hebb, D.O., *The organization of behavior: A neuropsychological theory*. 2005: Psychology Press.

42. Oja, E., *Simplified neuron model as a principal component analyzer.* Journal of mathematical biology, 1982. **15**(3): p. 267-273.

43. Bengio, Y. and Y. LeCun, *Scaling learning algorithms towards AI.* Large-scale kernel machines, 2007. **34**(5).

44. Erhan, D., Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, *Why does unsupervised pre-training help deep learning?* The Journal of Machine Learning Research, 2010. **11**: p. 625-660.

45. Mush, D. and B. Horne, *Progress in supervised neural networks: what's new since Lippman.* IEEE signal processing magazine, 1993: p. 8-39.

46. Sperduti, A. *Linear autoencoder networks for structured data*. in *Ninth International Workshop on Neural-Symbolic Learning and Reasoning*. 2013.

47. Wang, L. and X. Fu, *Data mining with computational intelligence*. 2006: Springer Science & Business Media.

48. Weiss, S.M. and N. Indurkhya, *Predictive data mining: a practical guide*. 1998: Morgan Kaufmann.

49. McCulloch, W.S. and W. Pitts, *A logical calculus of the ideas immanent in nervous activity.* The bulletin of mathematical biophysics, 1943. **5**(4): p. 115-133.

50. Farley, B. and W. Clark, *Simulation of self-organizing systems by digital computer.* Transactions of the IRE Professional Group on Information Theory, 1954. **4**(4): p. 76-84.

51. Rosenblatt, F., *The perceptron: a probabilistic model for information storage and organization in the brain.* Psychological review, 1958. **65**(6): p. 386.

52. Werbos, P., *Beyond regression: New tools for prediction and analysis in the behavioral sciences.* 1974.

53. *Image: A motor neuron," ed online*. Available from: http://www.shutterstock.com/s/neurotransmitter/search-vectors.html.

54. Rojas, R., *Neural networks: a systematic introduction*. 2013: Springer Science & Business Media.

55. Blum, A., *Neural networks in C++: an object-oriented framework for building connectionist systems*. 1992: John Wiley & Sons, Inc.

56. Swingler, K., *Applying neural networks: a practical guide*. 1996: Morgan Kaufmann.

57. Boger, Z. and H. Guterman. *Knowledge extraction from artificial neural network models*. in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*. 1997. IEEE.

58. Baum, E.B. and D. Haussler, *What size net gives valid generalization?* Neural computation, 1989. **1**(1): p. 151-160.

59. Specht, D.F., *Probabilistic neural networks.* Neural networks, 1990. **3**(1): p. 109-118.

60. Specht, D.F. *Probabilistic neural networks and general regression neural networks*. in *Fuzzy logic and neural network handbook*. 1996. McGraw-Hill, Inc.

61. Specht, D.F. *Probabilistic neural networks for classification, mapping, or associative memory*. in *Neural Networks, 1988., IEEE International Conference on*. 1988. IEEE.

62. Parzen, E., *On estimation of a probability density function and mode.* The annals of mathematical statistics, 1962: p. 1065-1076.

63. Richard, M.D. and R.P. Lippmann, *Neural network classifiers estimate Bayesian a posteriori probabilities.* Neural computation, 1991. **3**(4): p. 461-483.

64. Patel, J., *Enhanced classification approach with semi-supervised learning for reliability-based system design.* 2012.

65. Hastie, T., R. Tibshirani, J. Friedman, and J. Franklin, *The elements of statistical learning: data mining, inference and prediction.* The Mathematical Intelligencer, 2005. **27**(2): p. 83-85.

66. Patel, J. and S.-K. Choi, *Classification approach for reliability-based topology optimization using probabilistic neural networks.* Structural and Multidisciplinary Optimization, 2012. **45**(4): p. 529-543.

67. Mohamed, A.K., F.V. Nelwamondo, and T. Marwala. *Estimating missing data using neural network techniques, principal component analysis and genetic algorithms*. in *Proceedings of the Eighteenth Annual Symposium of the Pattern Recognition Association of South Africa*. 2007.

68. Pudil, P. and J. Novovičová, *Novel methods for feature subset selection with respect to problem knowledge*, in *Feature Extraction, Construction and Selection*. 1998, Springer. p. 101-116.

69. Mi, J., Y.-F. Li, Y.-J. Yang, W. Peng, and H.-Z. Huang, *Thermal cycling life prediction of Sn-3.0 Ag-0.5 Cu solder joint using type-I censored data.* The Scientific World Journal, 2014. **2014**.

70. Manson, S.S., *Behavior of materials under conditions of thermal stress*. 1953: NACA TN 2933.

71. Darveaux, R., J. Heckman, A. Syed, and A. Mawer, *Solder joint fatigue life of fine pitch BGAs–impact of design and material choices.* Microelectronics Reliability, 2000. **40**(7): p. 1117-1127.

72. Chahat, N., M. Zhadobov, and R. Sauleau, *Antennas for body centric wireless communications at millimeter wave frequencies.* Progress in Compact Antennas, 2014.