Summer 2019

Random neural networks for dimensionality reduction and regularized supervised learning

Renjie Hu
University of Iowa

RANDOM NEURAL NETWORKS FOR DIMENSIONALITY REDUCTION

AND

REGULARIZED SUPERVISED LEARNING

by

Renjie Hu

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Industrial Engineering
in the Graduate College of
The University of Iowa

August 2019

Thesis supervisor: Associate Professor Amaury Lendasse

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Renjie Hu

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Industrial Engineering at the August 2019 graduation.

Thesis committee: _____
Amaury Lendasse, Thesis Supervisor

_____
Amany Farag

_____
Yong Chen

_____
Daniel McGehee

_____
Edward Ratner

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor Professor Amaury Lendasse, who has been a tremendous mentor for me. I would like to thank you as a teacher for teaching me little by little about machine learning, Matlab coding, paper writing and so many other things. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on my career have been invaluable. I would also like to thank you as a friend, who is always there for me whenever I'm experiencing difficulties and hardships.

I would also like to thank Professor Yong Chen, Professor Amany Farag, Dr. Edward Ratner and Professor Daniel McGehee, for serving as my committee members. Thank you for letting my defense be an enjoyable moment. Your brilliant comments and constructive suggestions are much appreciated.

Lastly, I would like to give my family, especially my parents, a special thanks. I am appreciative beyond the words for the love and encouragement you give me, which makes me have the courage to pursuit what I really love. Thank you for all the long-lasting companionship, unconditional supports, and constant guidance.

Yours sincerely,

Renjie Hu

ABSTRACT

This dissertation explores Random Neural Networks (RNNs) in several aspects and their applications. First, Novel RNNs have been proposed for dimensionality reduction and visualization. Based on Extreme Learning Machines (ELMs) and Self-Organizing Maps (SOMs) a new method is created to identify the important variables and visualize the data. This technique reduces the curse of dimensionality and improves furthermore the interpretability of the visualization and is tested on real nursing survey datasets. ELM-SOM+ is an autoencoder created to preserves the intrinsic quality of SOM and also brings continuity to the projection using two ELMs. This new methodology shows considerable improvement over SOM on real datasets. Second, as a Supervised Learning method, ELMs has been applied to the hierarchical multiscale method to bridge the the molecular dynamics to continua. The method is tested on simulation data and proven to be efficient for passing the information from one scale to another. Lastly, the regularization of ELMs has been studied and a new regularization algorithm for ELMs is created using a modified Lanczos Algorithm. The Lanczos ELM on average divide computational time by 20 and reduce the Normalized MSE by 14% comparing with regular ELMs.

PUBLIC ABSTRACT

This dissertation explores a few meaningful questions in Machine learning. The first question is: When we have a multi-feature dataset, which are the critical features, that preserve more information of the data than other features. Knowing these critical features will greatly improve the ability to interpret the data. A new method is created to identify the critical features and visualize them, by using Extreme Learning Machines (ELMs) and Self-Organizing Maps (SOMs). This method is tested on a real survey dataset of nurses in Chapter 2.

The second question is: When we have a multi-dimensional dataset, is it possible to reduce the dimensionality of the data but in the meantime preserve as much information as possible? In Chapter 4, A novel autoencoder is developed that conducts dimensionality reduction by preserving the data topology like SOMs but more importantly brings continuity to the projections using two ELMs.

The third question is: When we are training an ELM, how to select the complexity of the network? Usually, different datasets require different complexity of the ELMs. There is no way to know what complexity to choose before the training and validating process. In Chapter 5, a regularized ELM is proposed that could largely speed up the process of validating and allows ELM to have a large amount of hidden neurons without overfitting the problem.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Motivation

In Machine Learning, dimensionality reduction is of great importance for several reasons. Firstly, due to the curse of dimensionality, many machine learning technics can result in overfitting problems, thus, high-dimensional data can be challenging to analyze [75,86,87]. Secondly, the computational load is correlated with the number of the features of the data. Analyzing high-dimensional data can be Computationally intensive [75]. Lastly, the high-dimensional data cannot be visualized directly [75]. "Looking at the data" is crucial for data analysis, because it provides the interpretability that allows us to make some sense of the data before carrying out further analysis.

Generally, when analyzing high-dimensional data, the dimensionality of the data is larger than necessary, especially when the variables are correlated. Another common assumption is that the high-dimensional data is embedded on a lower dimensional manifold [3]; therefor, the data can be transformed onto a lower dimensional space, and the transformed data still preserve the information of the data, nearly without information loss [86]. If the manifold of the data is in 3-D or less than 3-D space, the original data can be precisely visualized by the transformed data in the manifold space. Of course, in reality, the perfect manifold can not always be found, thus the transformation is always associated with information loss, however, the in-

formation loss can be minimized through the searching for the optimal manifold of the data.

Feature selection is a rudimental way to perform dimensionality reduction [45]. The selected variables can only preserve part of the data structure [3]. The domain knowledge is also usually required to perform such selection. In the process of data analysis, feature selection is of great importance. It allows regression or classification models to be robust, by filtering out the redundant or irrelevant data, which generally exists in the training data. This is also thought as the noise reduction process. It is achieved by selecting a subset of "relevant" features, and it builds the models upon those "relevant" features only. As a result, the model becomes easier to learn (the computational load is reduced), the generalization performances are improved and the model can be easily interpreted.

Feature extraction is one of the main dimensionality reduction processes [7], which builds derived values (features) from the original data. The derived features usually come from some type of transformation of the data, projecting the data to another (lower dimensional) space. The transformation can be linear or nonlinear [75].

Linear feature extraction methods works well when data is lying on the linear subspace. Principal Components Analysis (PCA) [114] is a popular linear feature extraction method, which is targeting on maximizing the variance of the data. Multi-Dimensional Scaling (MDS) [79] is preserving the pair-wise distances of the data, which yields to the same results as PCA [86] for linear MDS. Both methods perform poorly when the data is lying on a (curved) nonlinear manifold, which can often be

the case [75].

Methods for nonlinear dimensionality reduction can be further divided into two groups: distance-preserving methods and topology-preserving methods [77]. Distance-preserving methods include Sammon's mapping [123], Curvilinear Component Analysis (CCA) [30], Isomap (IM) [136,137], and Curvilinear Distance Analysis (CDA) [85]. Topology preservation methods are more powerful and, are at the same time, more complex than distance-preserving methods [86], such as Generative Topographic Mapping (GTM) [17], Laplacian Eigenmaps (LE) [14,15], Growing Neural Gas algorithms (GNG) [96], and Self-Organizing Maps (SOM) [78]. Both GTM and SOM use predefined grids and create discrete projections. Neural Gas Algorithm applies a neural network structure and is inspired by SOM. This method aims at finding the optimal data representation (an optimal manifold) [118]. LE creates continuous projections of graphs; however, the performances of the projection are generally poor [86]. The goal of this thesis is to explore the field of dimensionality, finding possible improvements for the existing dimensionality reduction technics and applying the dimensionality reduction technics in real world problems, especially for visualization and missing data imputation. The next section summarizes the different parts of thesis.

Another point of interest is the regularization problem and the speed up for machine learning algorithms, especially for Artificial Neural Networks (ANNs) [99]. Tow common question related are: 1) how to process large amount of data with reasonable computational time? 2) how to select the structure the complexity and the parameters of ANNs?

Although the significant improvement of the required computational power has been made for many complex algorithms, enabling the solution of large problems, such as SVM, and Deep Learning, the volume of data is growing even faster [6]. Therefore, reducing the computational time for machine learning algorithms is evermore desirable. Extreme Learning Machines (ELMs) [4, 42, 61, 82, 104] is a type of Randomized Neural Networks (RNNs) that is known for its fast training speed and good accuracy. Despite its merits, the performance of ELM is sensitive to the number of neurons. Underfitting can happen when there are not enough neurons, which leads to a poor approximation; while too many neurons often leads to overfitting problems, resulting in poor generalization performance. It is not easy to find the "correct" number of neurons that keeps the balance between a better network performance and simple network topology. Regularization is introduced to deal with this particular dilemma. Many algorithms have been applied to regularize the complexity of ELM, such as L_1 regularization like LASSO [125, 138] or L_2 regularization as known as Ridge Regression or Tikonov regression [105, 115]. Although, these regularization algorithms can significantly reduce the complexity of ELMs, they can't give a direct answer to the "correct" number of neurons, and the performance of ELMs is still largely influenced by the number of neurons it has. Lanczos Algorithm [80,81] originally was introduced to approximate the extreme eigenvalues of symmetric matrices. It is a fast iterative process that is proven to converge quickly [111]. Due to the distinct training process of ELMs, the last step of training ELMs is an Ordinary Least Square problem, which can be solved by the Lanczos Algorithm. Chapter 5 presents a modified Lanczos

Algorithm for ELMs that can speed up the training process, but more importantly does regularization of ELMs and allows ELMs to have a very large number of neurons, while not encountering overfitting problems. In other words, Lanczos ELM can reduce the computational time for the model selection, and just use a large number of neurons to generate the robust outcome without overfitting.

1.2 Overview

Chapter 2 contains an application of data analysis by feature selection and visualization, and shows that such combination is powerful for analyzing the high-dimensional data, providing the interpretability for complicated problems.

In Chapter 3, we created a machine learning supported hierarchical multiscale method to bridge the molecular dynamics to continua.

Chapter 4 is about a novel dimensionality reduction technique, which preserves the intrinsic quality of Self-Organizing Maps (SOM): it is nonlinear and suitable for big data. It also brings continuity to the projection using two Extreme Learning Machine (ELMs) models.

Chapter 5 presents a regularized ELM to give the answers to the speed up of ELM and the model selection problem of ELM. A new iterative method for solving ELM is proposed, which on average is 20% faster than the regular ELM and has 14% lower MSE than the regular ELM.

In Chapter 6, we give a possible directions for further research. In particular, we want to extend the work from Chapter 4 and apply it to missing data imputation.

CHAPTER 2
ELM FEATURE SELECTION AND SOM DATA VISUALIZATION FOR NURSING SURVEY DATASETS

2.1 Introduction for Chapter 2

Medical errors ranked as the eight highest cause of death in the United States [67]. It is estimated that about 44,000 to 98,000 people die annually from medical errors in USA [67]. These numbers are higher than deaths from breast cancer, AIDS, and car accidents combined. Medication errors are the most frequently occurring medical error in healthcare settings [71]. Unfortunately, serious life threatening errors are usually reported, but the majority of other medication errors are not [12].

Medication delivery is a complex multi-stage process that involves several healthcare professionals [19]. Medication errors occur at each step of the medication process [128], with 38% of errors occurring at the administration phase [124]. Nurses spend about 40% of their time administering medications, and by virtue of their role represent the last defense wall that could intercept errors before reaching patients [120]. Most health care organizations rely on nurses to report errors whether they are the cause, witness or collaborator [98].

Medication error reporting is a voluntary process [46]. Reviewing and analyzing medication error reports provide healthcare administrators and safety officers with opportunities for understanding error root causes and subsequently design interventions to prevent subsequent errors [16, 68, 88]. However, having less than 5% of errors reported, makes developing a proper intervention a tough challenge [24].

Fear of blame, punishment, humiliation, retaliation from managers and/or peers were some of the reasons deterring nurses from reporting errors [27, 98]. Mayo and Duncan (2004) [98] argued that all efforts of healthcare administrators, policy makers and scholars to create effective medication errors reporting systems, could fail if nurses remain unwilling to report errors. Therefore, the purpose of this multisite data analysis is to examine interpersonal and organizational factors predicting nurses' willingness to report medication errors.

In this Chapter, we propose a novel combination of Extreme Learning Machines [20, 59, 60, 62] and Self-Organizing Maps [29, 78, 89, 103] to identify which variables lead to the likelihood to report the medical errors. Extreme Learning Machines are accurate by extremely fast prediction models [60], therefore, it is possible with them to test a very large number of possible variables. Self-Organizing Maps are performing nonlinear dimensionality reduction [78] to get an accurate visualization of the data. Combining both techniques reduces the curse of dimensionality [143] and improves furthermore the interpretability of the visualization.

The Chapter is organized as follows: in Section 2.2, we give details about the targeted problem. Section 2.3 is presenting the methodology: data collection, overview of the methodology and details about each methodology components (ELMs and SOMs). Section 2.4 is presenting our experiments, including data preparation, experimental setup, results, visualization results and analysis. Finally, future works are introduced in the Conclusion.

2.2 Problem Description

The surveys that we are analyzing include several types of variables: Interpersonal variables measurements, Organizational variables measurements and Outcome variable measurement. There are described in detail below.

Interpersonal variables measurements:

1) Warmth and belonging climate was measured using the Modified Litwin and Stringer Organizational Climate Questionnaire (M-LSOCQ) [33]. The M-LSOCQ consists of 25 items addressing two main dimensions of unit climate (warmth and belonging; and structure and administrative support). In this study, 11 items measuring the dimension of warmth and belonging were used. Responses use a 4- point Likert scale ranging from 0 (strongly disagree) to 3 (strongly agree). High scores indicate a climate that is characterized by sense of unity and cohesion among team members. This measure included questions such as: "I feel that I am a member of well-functioning team", "People are proud of belonging to this unit", and "A friendly atmosphere prevails among the people in this unit". Some questions were worded positively and some negatively. The validity of the instrument has been established through exploratory factor analysis in a prior study [33]. Cronbach's Alpha for warmth and belonging is reported as 0.91 [33].

2) Organizational trust was measured using Cook and Wall (1980) [73] organizational trust instrument. The organizational trust measure consists of 12 items covering 4 dimensions each dimension consisted of three items. The four dimensions are faith in peers, faith in managers, confidence in peers and, confidence in managers.

Responses use a 4-point Likert scale ranging from (0) definitely disagree to (3) definitely agree. Validity of the instrument was established through factor analysis that supported the four dimensions [73]. This measure included questions such as: "I have a full confident on the skills/of my peers/colleagues", "I feel quite confident that my unit manager will always try to treat me fairly", and " my nurse manager is sincere in his/her attempts to meet our point of views". Some questions were positively worded and some negatively worded. Internal consistency has Cronbach's alpha level ranging from 0.77-0.79 for the four dimensions, and 0.87 for the total scale.

Organizational variables measurements:

1) Nurse manager's leadership style was measured using the Multifactorial Leadership Questionnaire (MLQ-5X rater form) (Bass & Avolio, 2004). The section of the instrument which addresses employee's (nurse's) perceptions to nurse manager's leadership styles (transformational and transactional) will be used. This section of the MLQ-5X consists of 28 items covering the two styles. Transformational leadership style was measured by five subscales of: idealized attributes, idealized behaviors, inspirational motivation, intellectual stimulation, and individualized consideration; each subscale consists of 4 items. Transactional leadership style was measured with two subscales of: contingent reward and management-by-exception (Active); each subscale consists of 4 items. Responses are measured using a 5-point Likert scale ranging from 0 (not at all) to 4 (frequently if not always). Higher scores indicate the nurse manager's tendency toward using a particular leadership style. This measure included questions such as: "my manager Considers the moral and ethical consequences

of decisions"," my manager displays a sense of power and confidence", and " my manager acts in ways that builds my respect". All the items are positively worded. [33] reported internal consistency of this instrument as: 0.95 for transformational, 0.73 for transactional. The construct validity of the instrument has been established through exploratory and confirmatory factor analysis.

2) Safety climate was measured using subscales from Nieva & Sorra (2003) [110] safety climate survey, it is one of the most widely used instruments to measure safety climate. It consists of 12 subscales addressing hospital and unit-based safety climate dimensions. For the proposed study 20 items covering 6 safety climate dimensions of (Manager's actions promoting safety (4 items), organizational learning (3 items), team work within unit (4 items) , communication openness (3 items), feedback and communication about errors (3 items), and non-punitive response to error (3 items)) will be used. Responses use a 5-point Likert scale ranging from 0 (strongly disagree) to 4 (strongly agree). This measure included questions such as: "After we make changes to improve patient safety, we evaluate their effectiveness ", "In this unit, we discuss ways to prevent errors from happening again", and "My unit manager seriously considers staff suggestions for improving patient safety". Some items are positively worded and some were worded negatively. Higher scores indicate better safety climate. The internal consistency reliability for these dimensions range from 0.63 to 0.83 [110].

Outcome variable measurement:

Nurse willingness to report Medication error was measured using three items from the outcome subscale from Nieva & Sorra (2003) safety climate instrument [110]. The original question stem asks nurses to report the frequency of reporting errors at their units. In this study the wording for the question stem was modified to reflect nurse's willingness to report his/her own medication errors if happened. Responses use a 4-point Likert scale ranging from 0 (not likely) to 3 (very likely). Higher scores indicate more willingness to report medication errors. Three items were: "When a mistake is made, but caught and corrected before affecting the patient, how likely are you to report this error", "When a mistake is made, but has no potential harm to the patient, how likely are you to report this error", and "When a mistake is made that could harm the patient, but does not, how likely are you to report this error". Medication error severity gradually increased from the first to the third one. Reliability coefficient for the original question is .84 [110]. The instrument with the rewarded question stem was used in a previous PI research (unpublished data) and showed reliability coefficient of .88 (Attachment- instrument 5).

Completing all the study instruments takes approximately 20 minutes for a subject.

2.3 Methodology

2.3.1 Procedure for Data Collection

This analysis was conducted using data that were collected from three funded projects. Data for the three projects were collected simultaneously after obtaining the required human subject approval. Project 1 is the MNRS and project 2 is the HCGNE. In project 1 and 2, the principle investigator (PI) selected a random sample of 850 Registered Nurses (RN) working in general medical surgical units and nursing homes using one midwestern state nursing registry. The study package containing: the study cover letter, the study survey, and a pre-stamped return envelope addressed to the PI was mailed to the participants' home address. A convenience sample of 75 RN working in five Emergency Departments (ED) affiliated with one midwestern medical center were recruited for the third project, which is CGEAN. The PI attended the monthly staff meeting for the conveniently selected five ED and introduced the study to nursing staff. The PI placed the study packages, contacting the same materials as the first two projects, in the nurses' mailboxes. In order to increase the response rate follow up reminders over a three week period was mailed to each participant (project 1and 2), and was placed in nurses' mail boxes in the third project. For all three project a $10 compensation check was mailed to each participant after receiving the completed survey. The PI has achieved a 44% and 47 %response rate when using a similar procedure in previous studies of hospital nurses.

2.3.2 Methodology Overview

We propose a novel combination of Extreme Learning Machines [20, 59, 60, 62] and Self-Organizing Maps [29, 78, 89, 103] to identify which variables lead to the likelihood to report the medical errors. Extreme Learning Machines are accurate by extremely fast prediction models [60], therefore, it is possible with them to test a very large number of possible variables. Self-Organizing Maps are performing nonlinear dimensionality reduction [78] to get an accurate visualization of the data. Combining both techniques reduces the curse of dimensionality [143] and improve furthermore the interpretability of the visualization.

Interpretability is one of the essential goal of data analysis. However, it is difficult for human to understand the data in high dimension, especially for data that have a nonlinear relationship [45]. Visualization is a great approach to bring the interpretability for data in such a way that, humans can visually examine the relationship of data [75]. Since visualization can bring comprehensive insight for the problems, it should be carried out whenever possible. Although, visualization is recommended, it is not easy to obtain a "good" visualization, when the number of features of data is large.

In this Chapter, a Two-Phase Visualization technique is proposed, using Extreme Learning Machine to perform feature selection first, Self-Organization Map to perform visualization secondly. This technique is a non-linear approach for both feature selection and visualization, and will reveal the nonlinear relationship between the features and the target(s).

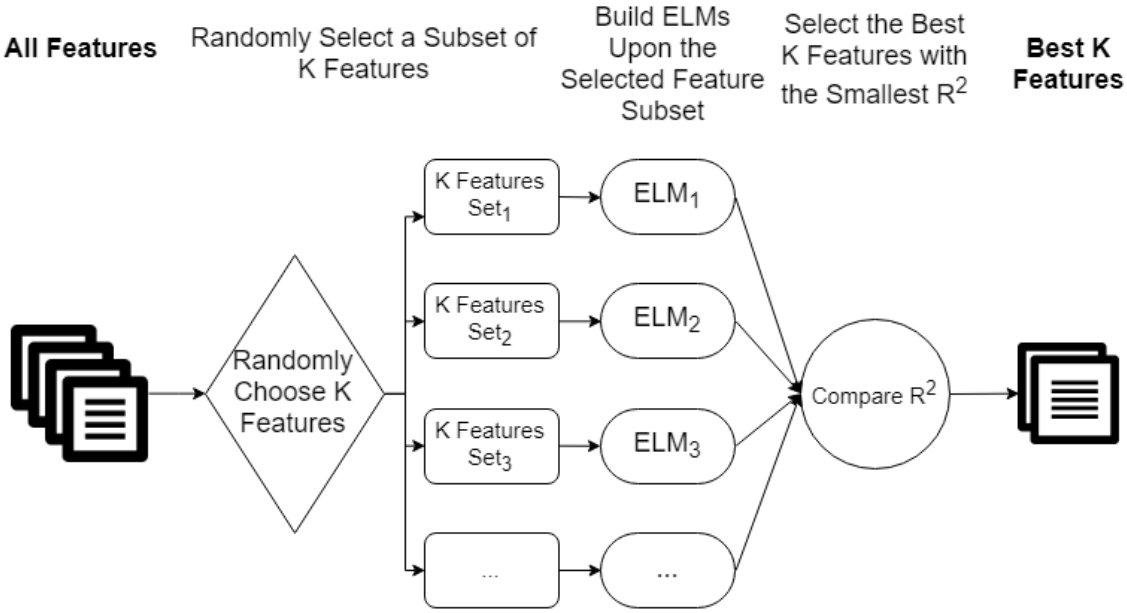


Figure 2.1: Phase I: ELM-WSF

Many ELMs are built in the fist phase to evaluate the relationship between the different subset of features and the target variables. R^2 value is used as the criteria to measure such evaluation. Feature sets with large R^2 values are selected and used for the visualization in the second phase.

2.3.3 Details

2.3.3.1 Feature Selection

In the process of data analysis, Feature Selection (FS) is of great importance. It allows the regression or classification models to be robust, by filtering out the redundant or irrelevant data, which generally exists in the training data. This is also thought as the noise reduction process. It is achieved by selecting a subset of

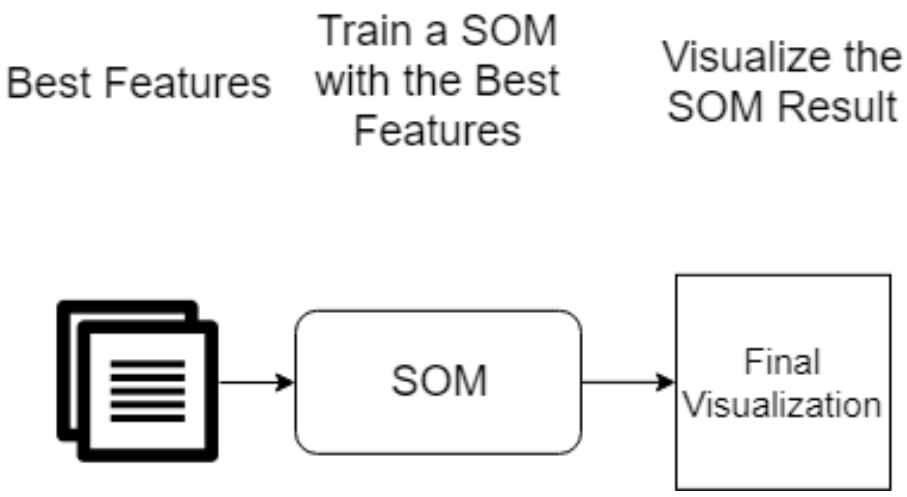


Figure 2.2: Phase II: SOM Visualization

"relevant" features, and build the models upon those "relevant" features only. As a result, the model becomes easier to learn (the computational load is reduced), the generalization performances are improved and the model can be easily interpreted.

The FS process can be described as follow: for a dataset X, whose feature set is denoted as F, that has p features, we find a subset of features S, that contains p' features, where $S \in F$ and $p' < p$. In theory, the feature set S should be selected in such a way that the model built with these features gives the minimum generalization error [45].

Besides the benefit of improving the generalization performances, FS also assist for a better data visualization [75], simplifying the models and making them easier to interpret by users or practitioners [77].

FS algorithms can be broken up into three categories: the filter algorithms, the wrapper algorithms, and the embedded algorithms [45]. The filter methods utilize the

characteristics of the training data and selects a subset of features without involving the final model [116]. In contrast, the wrapper method involves the learning model and targets on improving the generalization performance of the final model [101]. Although the wrapper method is more computationally expensive than the filter method, the generalization performance of the former approach is better than the later approach [148]. The embedded method is the hybrid of the filter and wrapper methods [31]. In our methodology, we use the wrapper approach with ELM as the training model. The detail is in the Section 2.3.3.

ELM Wrapper Feature Selection Paradigm

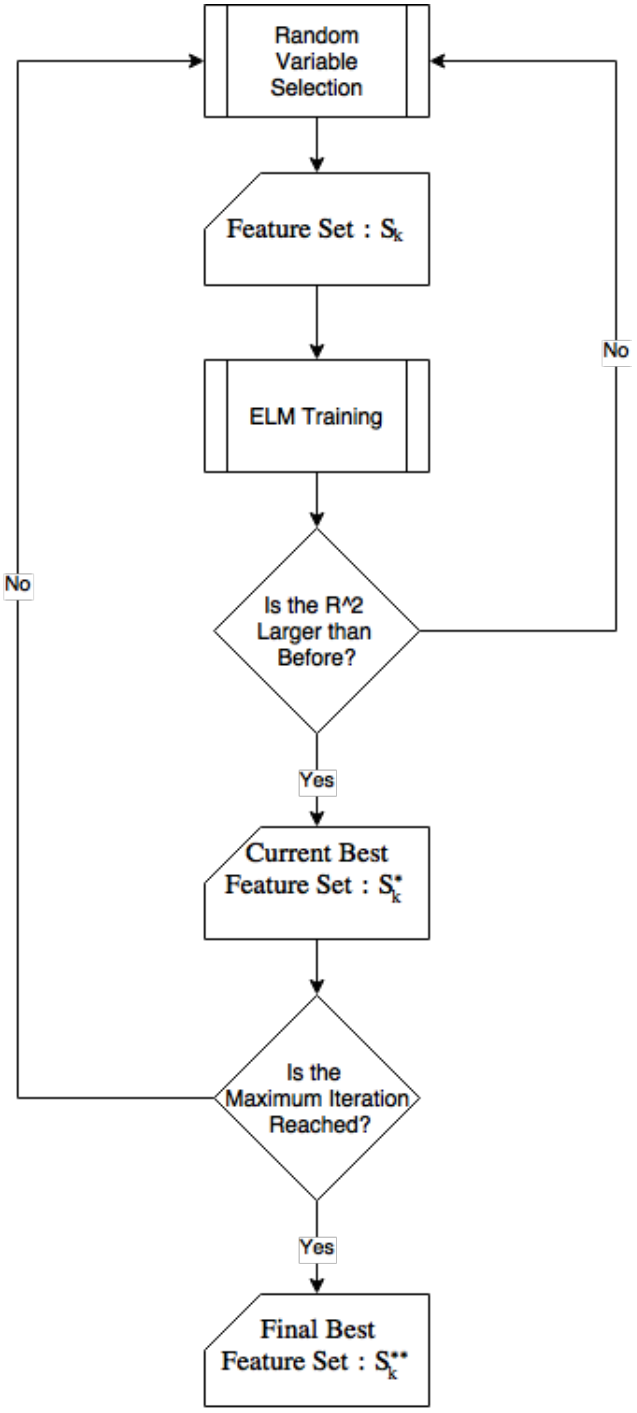


Figure 2.3: ELM Wrapper Feature Selection

ELM wrapper Feature Selection (ELM-WFS) is the proposed method for feature selection in the first phase. The main frame of this method is the wrapping approach feature selection. The learning model is ELM. The searching algorithm is exhaustive search and greedy hill climbing. The evaluating function is the R^2 value of the model [38]. The optimality criteria is using the predefined number of the iterations.

ELM-WFS initialized by selecting a subset of features $\mathbf{S}_0$, from a given dataset $\mathbf{X}$ with p features. Then, ELM is build upon $(\mathbf{S}_0, \mathbf{Y})$, where $\mathbf{Y}$ is the corresponding target variable. The performance of this model is evaluated by R^2. A new random search is then started in the feature space, generating a new subset of features: $\mathbf{S}_1$. The new model is build upon $(\mathbf{S}_1, \mathbf{Y})$ and its performances are evaluated. If the performance of the new model with the new feature set $\mathbf{S}_1$ is found better than the old model with the feature set $\mathbf{S}_0$, $\mathbf{S}_1$ is selected over $\mathbf{S}_0$. The search continues and better feature sets are selected, until a predefined stopping criteria is reached.

ELM is a very fast machine learning model, which can speed up the training process. In order to achieve both a better R^2 and the model interpretability, exhaustive search is applied to find a model with as high accuracy as possible, meanwhile, as simple as possible. The R^2 measures the regression accuracy, and allows a comparison with other feature selection method.

step 1. Initialization. From the feature space, a subset of k features: $\mathbf{S}_k$ is selected randomly, with the $k = 1$ at the beginning.

step 2. Building ELM. An ELM is built upon the selected model, with prede-

fined number of hidden neurons. The input is the data with the selected features: $\mathbf{S}_k$; the output is the regression value of the data. In our case, the input is the selected set of questions from the survey data, and the output is regression value of one of the error report question.

step 3. Computing the R^2. With the regression value from the ELM model, we could evaluate the model by compute the R^2 value between the prediction and the true value.

step 4. Updating the $\mathbf{S}_k^*$. If the R^2 from step 3 is larger than the R^2 from the previous model, the current $\mathbf{S}_k$ becomes the best set of features: $\mathbf{S}_k^*$; otherwise, $\mathbf{S}_k$ stays the same.

step 5. Random Feature Selection. Randomly select new k features: $\mathbf{S}_k$ from the feature space.

step 6. Optimality Criteria Checking. If the maximum iteration number is reached, then $\mathbf{S}_k^*$ becomes $\mathbf{S}_k^{**}$, which denotes the final best k-variables. k is increased by one and the method start from step 1 again. If the iteration is not at the maximum, repeat from step 2. to step 6 again.

2.3.3.2 Extreme Learning Machine

Extreme Learning Machine (ELM) in [20,106] as important emergent machine learning techniques, are proposed for training Single-hidden Layer Feed-forward Neural Networks (SLFNs) [54,57–60].

In contrast with the traditional Feedforward Neural Networks (FNNs), which

generally are trained by the well-known backpropagation (BP) algorithms, in ELM, the wight for the hidden layer are randomly initiated and then fixed without iteratively tuning. Then commonly used activation functions are applied on the hidden neurons. The only parameters learned in ELM are the weights between hidden layer and the output layer. In this way, the parameters of the hidden neurons can be independent of the training data, which makes it possible for ELM to attain the near optimal generalization bound of traditional FNN. Theoretical studies as in [54, 57, 59] has shown that ELM has the universal approximation and classification properties.

The unique training process of ELM provides a huge leverage for the learning speed. A non-iterative solution of ELM provides a speedup of 5 orders of magnitude compared to Multilayer Perceptron ([122], MLP) or 6 orders of magnitude compared to Support Vector Machines ([25], SVM).

The Extreme Learning Machine [4,42,61,104,130] is introduced as a generalized Single-Layer Feed-forward Network (SLFN) [57–60]. This type of Network is capable of solving classification, regression and clustering problems. According to Huang et al. in [61], ELM has good generalized performance in most cases and the learning speed is thousands of times faster than conventional neural networks [42, 52].

ELM belongs to the family of Randomized Neural Networks (RNNs). Unlike traditional neural networks and learning algorithms, the ELM algorithm shows that hidden nodes can be randomly generated. Thus, the weights from the first layer can be independent from the training data. Because there is no dependence between the input and output weights, ELM has a non-iterative linear ordinary least square

solution for the output weights, unlike the conventional Back-propagation training procedure [50]. On top of the distinct properties of ELM, Huang et al in [57, 59] stated that ELM has the universal approximation capability, indicating that ELM can universally approximate any continuous target functions in any compact subset X of the Euclidean space $\mathbb{R}^n$ [162].

The rest part of this Section gives a brief explanation of the original ELM. In order to keep a uniform meaning for notations throughout the Chapter, some of the original notations for ELM have been modified.

Figure 2.4 shows a typical structure of ELM, which contains three layers: the input layer, the hidden layer, and the output layer. Input layer weights ($\boldsymbol{w}$) and biases ($\boldsymbol{b}$) are randomly generated and don't involve in the further training anymore. $\boldsymbol{X} \in \mathbb{R}^{m \times d}, \boldsymbol{X} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_m})^T$ is the input data, with sample size m, and feature size d. Through the first layer, $\boldsymbol{\theta}$ is mapped to N-dimensional ELM random feature space. After the nonlinear transformation $\boldsymbol{f}$, the hidden layer output is:

$$h_i(\boldsymbol{x}) = f(\boldsymbol{x^T w_i} + b_i), \ i \in [1, N]. \tag{2.1}$$

f is also called the activation function. Many nonlinear function can be applied here, such as a sigmoid function. Other activation functions are listed in [57, 59]. The last layer is the ELM functional output:

$$f_{ELM}(\boldsymbol{x}) = \sum_{i=1}^{N} \theta_i h_i(\boldsymbol{x}) = \boldsymbol{h(x)^T \theta} = \hat{\boldsymbol{t}}, \tag{2.2}$$

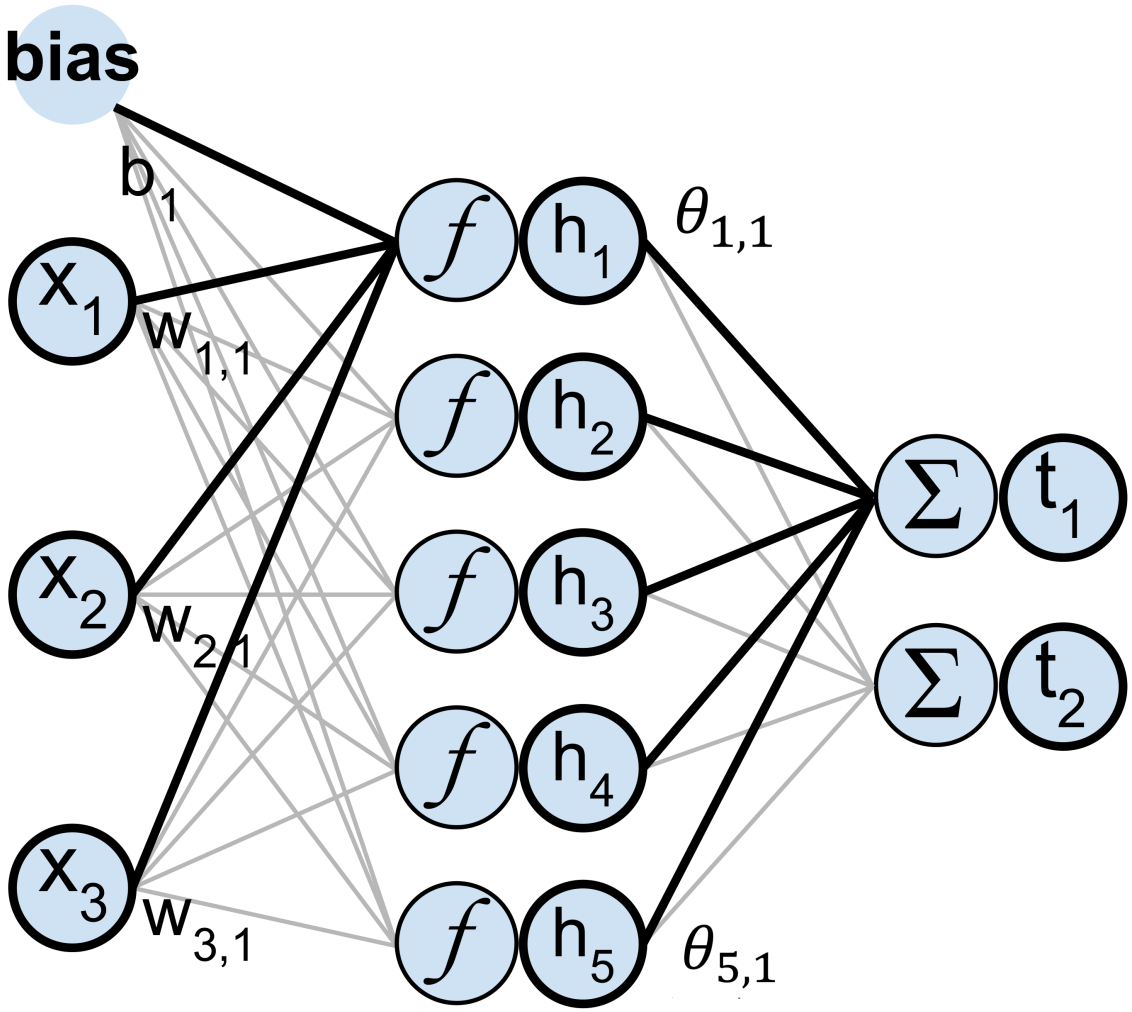


Figure 2.4: ELM Structure

Where, $\boldsymbol{h(x)} = (h_1(\boldsymbol{x}), \ldots, h_N(\boldsymbol{x}))^T$, $\boldsymbol{\theta}$ is the output weights $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)^T$ and $\hat{\boldsymbol{t}}$ is the approximation of $\boldsymbol{t}$ — the true target value (i.e. labels, or regression values) of $\boldsymbol{x}$.

The last step for training an ELM is to determine the output layer coefficients: $\boldsymbol{\theta}$. If $\boldsymbol{T} = (\boldsymbol{t_1}, \ldots, \boldsymbol{t_m})^T$ is the corresponding target matrix of the input matrix $\boldsymbol{X}$, $\boldsymbol{\theta}$ should satisfy the following equation:

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \|f_{ELM}(\boldsymbol{X}) - \boldsymbol{T}\|^2, \tag{2.3}$$

in which, ELM function: $f_{ELM}(\boldsymbol{X}) = \hat{\boldsymbol{T}}$ is an approximation of the true target matrix $\boldsymbol{T}$.

To simplify the problem, introduce $\mathbf{H} \in \mathbb{R}^{m \times N}$:

$$\mathbf{H} = \begin{pmatrix} h_1(\boldsymbol{x_1}) & \ldots & h_N(\boldsymbol{x_1}) \\ \ldots & \ddots & \ldots \\ h_1(\boldsymbol{x_m}) & \ldots & h_N(\boldsymbol{x_m}) \end{pmatrix}, \tag{2.4}$$

and the minimization problem in equation 2.3 can be rewritten as:

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{H}\boldsymbol{\theta} - \boldsymbol{T}\|^2. \tag{2.5}$$

Solving the above problem finishes the ELM training process. Typically, solving this problem is the most computational intense step in the ELM Algorithm.

Practically, the implementations of the pseudoinverse include a small regularization term $\mathbf{H}^\dagger = (\mathbf{H}^T\mathbf{H} + \alpha\mathbf{I})\mathbf{H}^T$.

2.3.3.3 Self-Organizing Maps for Visualization

SOM is a popular nonlinear dimensionality reduction tool that uses a predefined 2-D grid to capture the topology of the data in the high dimension [3] (see Figure 2.5).

Besides the two-dimensional map representation, each point on the grid will attain a weight, or prototype, which is basically its d-dimensional representation in the original d-dimensional data space.

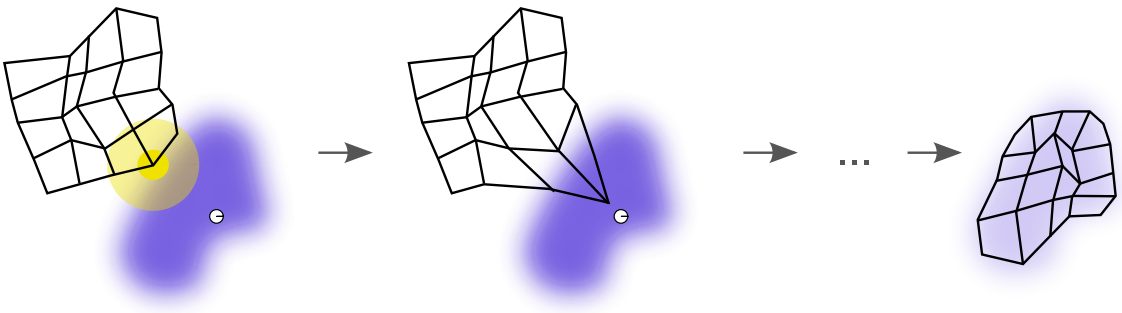


Figure 2.5: An illustration of the training of a self-organizing map. The blue blob is the distribution of the training data, and the small white disc is the current training datum drawn from that distribution. At first (left) the SOM nodes are arbitrarily positioned in the data space. The node (highlighted in yellow) which is nearest to the training datum is selected. It is moved towards the training datum, as (to a lesser extent) are its neighbors on the grid. After many iterations, the grid tends to approximate the data distribution (right). [147].

The grid, which consists of a rectangle including the points located on a rectangular lattice, is accompanied with randomly initialized weights for each point. Finally, after a considerable number of iterations, these weights will be updated to the points' positions in the original d-dimensional data space. In the iterative algorithm, units (or prototypes) $\boldsymbol{c}_s$, for $s = [1, \ldots, N]$, in which N is the number of points on a 2-D grid, are updated with the following rule:

$$\boldsymbol{c}_s \leftarrow \boldsymbol{c}_s + \alpha\sigma_\lambda(r, s)(\boldsymbol{x}_i - \boldsymbol{c}_s) \tag{2.6}$$

where $\boldsymbol{x}_i$ is the ith data point, α is a learning rate between 0 and 1, and σ_λ which is called the neighborhood function returns zeros for non-neighbors, and ones for other non-zero values for valid neighbors. In addition, d is a distance function and $r = \underset{s}{\operatorname{argmin}}\{d(\boldsymbol{x}_i, \boldsymbol{c}_s)\}$.

After the projection, according to SOM algorithm, each point $\boldsymbol{c}_s$, $s = [1, \ldots, N]$ on the 2-D grid is a representative of a group of points in d-dimensional data space. Basically, $\boldsymbol{c}_s$ is the Best Matching Unit (BMU) of a group of points in original data space.

Therefore, Self-Organizing Maps are performing a discrete nonlinear dimensionality reductions.

In order to understand the visualization, colors are used to transform the SOM into a heat map that helps understanding the importance of a given variable. Using several heat maps help analyzing the data as illustrated in the next Section.

2.4 Experiments

In this section, the proposed Two-Phases Visualization is tested using the nursing dataset. The original survey for this dataset are listed in the appendices. In total, 144 questions are asked in the survey. This includes all the unique questions from the three different surveys. 380 subjects have participated in these surveys. 165 of them took the HCGNE survey; 75 took the CGEAN survey; 144 took the MNRS survey. Although the design and the format of theses surveys are slightly different, most questions are the same (Uncommon questions are omitted in the experiment, only common ones are used).

2.4.1 Data Preparation

Each survey data is collected in a separate ".csv" file. The features in the dataset are corresponding to the questions from the survey, and the values of the features are the subjects' answers to the questions. The name of the features (for main questions) are coded in two parts: "the abbreviation of the survey section name" + "the question number". For example: Feature "*LSHPQ1*" means "question 1" in the section of "Nurse manager's leadership style", representing the question: "My unite manager provides me with assistance in exchange for my efforts". Table 2.1 shows the correspondences among the Feature Names, the Measurements, and the survey sections. Other Feature Names are the abbreviations of miscellaneous survey questions, also showing in the Table 2.1. The last three Feature Names are the outcomes variables, which are also the target variables in our experiments. The

detailed survey questions can be found in the appendices.

Table 2.1: Feature Names Reference Table

Feature Names	**Measurements**	**Survey Sections**
LSHP	Nurse manager's leadership style	Unit nurse manager
WARMCLIM	Warmth and belonging climate	Unit work environment
SAFCLIM	Safety climate	Additional aspects of work environment
WARMCLIM	Warmth and belonging climate	Unit work environment
ORGTRUST	Organizational trust	Interpersonal relationships
NGEDU		Nursing education degree
YSOFRNEXP		Total Years of Experience
EXPCURRUNIT		Years worked in the current unit
EXPCURRMNG		Years worked with the current manager
WORKHRS		How many hours work per week
SHIFTWRK		Work shift
EMPSTATUS		Employee status
HOSPSIZE		Hospital size
REPSYSTFAMIL		Hospital reporting system
COMPLERREP		Hospital reporting system
REPSYSTYP		Hospital reporting system
EXPCURRUNIT		Hospital reporting system
ERREPQ1	Outcome variable	Will you report this error
ERREPQ2	Outcome variable	Will you report this error
ERREPQ3	Outcome variable	Will you report this error

All category variables are converted to numerical values using label encoder.

In the experiment, all Safety Climate features are omitted, because the HCGNE survey has a different design than the other two surveys on this part. Combining the two designs will create too many missing values in the dataset.

All the rows with missing values has also been removed.

After clean-up the above data, the rest of 68 features and 328 samples are used in the experiment.

In the experiment, the notation $\mathbf{Y}_i \in \mathbb{R}^{328\times 1}$ denotes the target variable $ERREPQi$, where $i = 1, 2, 3$. $\mathbf{X} \in \mathbb{R}^{328\times 68}$ denotes the total feature set.

2.4.2 Experimental Setup

The three outcome questions are:

ERREPQ1: When a mistake is made, but caught and corrected before affecting the patient, how likely are you to report this error?

ERREPQ2: When a mistake is made, but has no potential harm to the patient, how likely are you to report this error?

ERREPQ3: When a mistake is made that could harm the patient, but does not, how likely are you to report this error?

Due to the distinct nature of these three questions, one subject can give very different answers to these questions. It is intuitive to analyze three questions separately. Thus, the Two-Phases Visualization has been applied on $(\mathbf{X}, \mathbf{Y}_1)$, $(\mathbf{X}, \mathbf{Y}_2)$, and $(\mathbf{X}, \mathbf{Y}_3)$ separately.

For each output variable $\mathbf{Y}_i$, ELM Wrapper Feature Selection is applied first. 20 subset features $\mathbf{S}_k^{**} \in \mathbb{R}^k$, where $k = 1, 2, ...20$ are selected. For each k, $R^2_{(\mathbf{S}_k^{**}, \mathbf{Y}_i)} > R^2_{(\mathbf{S}_k, \mathbf{Y}_i)}$, for any subset of k features $\mathbf{S}_k$, where $R^2_{(\mathbf{S}_k, \mathbf{Y}_i)}$ is evaluated as:

$$R^2_{(\mathbf{S}_k, \mathbf{Y}_i)} = 1 - \frac{MSE(\mathbf{S}_k, \mathbf{Y}_i)}{Var(\mathbf{Y}_i)}, \tag{2.7}$$

$$MSE(\mathbf{S}_k, \mathbf{Y}_i) = \frac{1}{N}(\mathbf{Y}_i - \hat{\mathbf{Y}}_i)(\mathbf{Y}_i - \hat{\mathbf{Y}}_i)^T, \tag{2.8}$$

$$\hat{\mathbf{Y}}_i = ELM(\mathbf{S}_k, \mathbf{Y}_i). \tag{2.9}$$

One **optimal** feature set for visualization is then chosen from the 20 candidate, with k^* number of features. The selection criteria for the **optimal** feature set is based on the R^2 values of the k candidates: on the one hand the R^2 value should be as large as possible, on the other hand the number of features, k should be as less as possible. In general, we choose the last k, that gives the biggest raise in R^2.

SOM Visualization is applied next on the selected best K^* features.

2.4.3 Results

2.4.3.1 ERREPQ1

ERREPQ1: When a mistake is made, but caught and corrected before affecting the patient, how likely are you to report this error?

Feature Selection Results

The best feature sets $\mathbf{S}_k^{**}$, where $k = 1, 2, ...20$, for $\mathbf{Y}_1$ are selected by ELM-WFS. The following table lists selected feature names for $k = 1$ to $k = 5$.

Table 2.2: Selected Features for ERREPQ1

	Feature Names				
$\mathbf{S}_1^{**}$	EXPCURRUNIT				
$\mathbf{S}_2^{**}$	YSOFRNEXP	WORKHRS			
$\mathbf{S}_3^{**}$	EXPCURRUNIT	SHIFTWRK	LSHPQ20		
$\mathbf{S}_4^{**}$	EXPCURRUNIT	WARMCLIMQ7N	ORGTRUSTQ5	ORGTRUSTQ10	
$\mathbf{S}_5^{**}$	EXPCURRUNIT	SHIFTWRK	LSHPQ20	ORGTRUSTQ12N	FAMILIAREXTENTQ4

The R^2 values for the best features are showing in Figure 2.6.

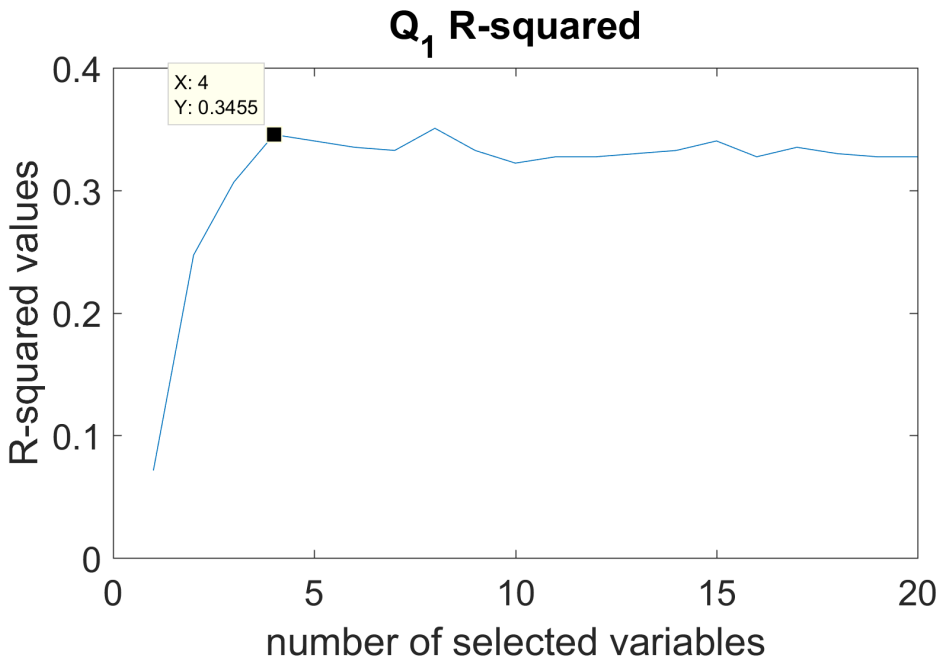


Figure 2.6: R Squared Values for ERREPQ1

Since $\mathbf{S}_4^{**}$ gives the highest R^2 for $ERREPQ_1$, the **Optimal** feature set for visualization is $\mathbf{S}_4^{**}$, which are:

- EXPCURRUNIT: Years of experience in the current unit.
- ORGTRUSTQ5: I can rely on my peers/colleagues to lend me hand (help me) if I needed it.
- ORGTRUSTQ10: Most of my peers/colleagues efficiently do their work even if the unit manager is not around.
- WARMCLIMQ7N: People in this unit really do not trust each other.

Visualization Results

SOM is built upon the optimal feature set and the outcome variable 1: $(\mathbf{S}_4^{**}, \mathbf{Y}_1)$.

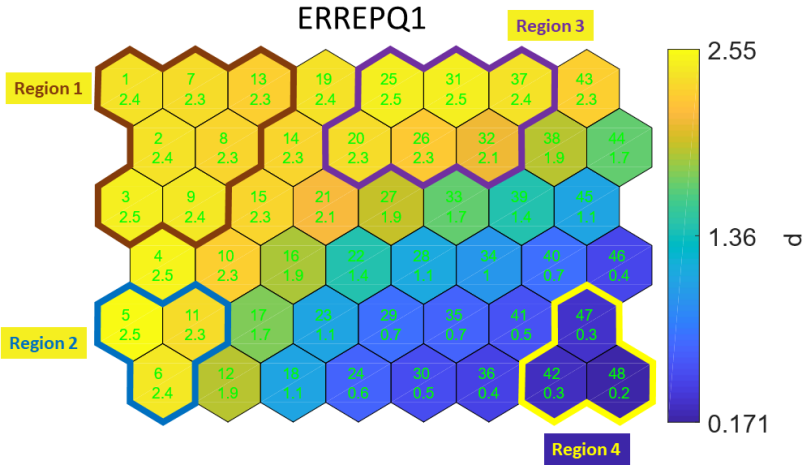


Figure 2.7: When a mistake is made, but caught and corrected before affecting the patient, how likely are you to report this error? 0: Not Likely at All; 1: Somewhat Not Likely; 2: Somewhat Likely; 3: Very Likely.

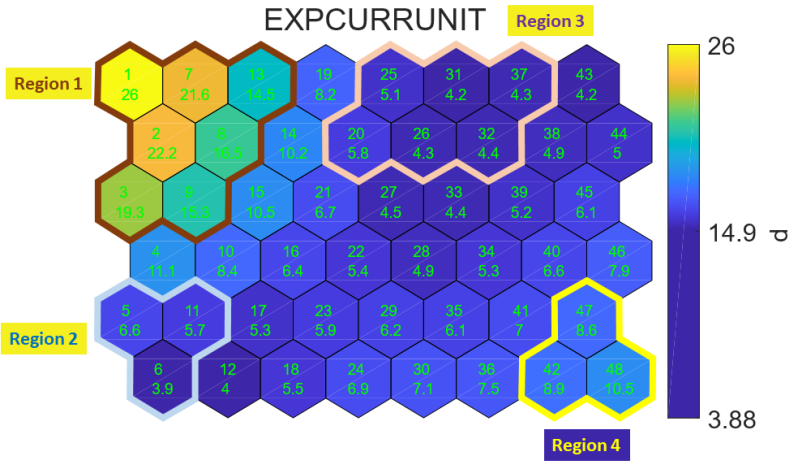


Figure 2.8: How long you have been working in your current unit?

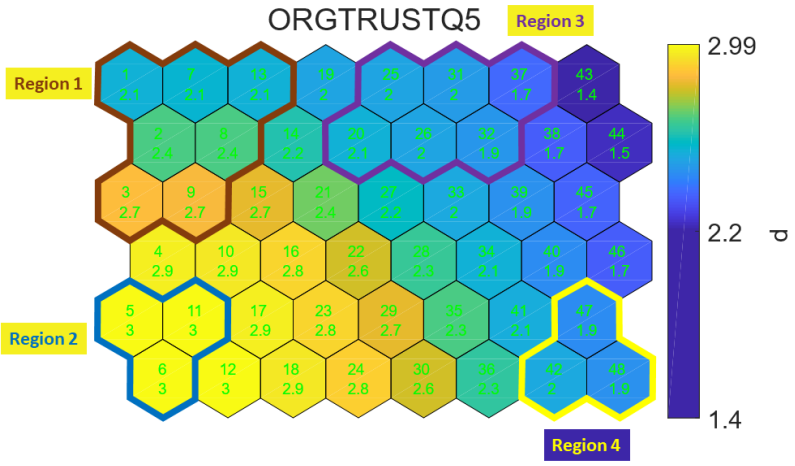


Figure 2.9: I can rely on my peers/colleagues to lend me hand (help me) if I needed it. 0: Definitely Disagree; 1: Inclined to Disagree; 2: Inclined to Agree; 3: Definitely Agree.

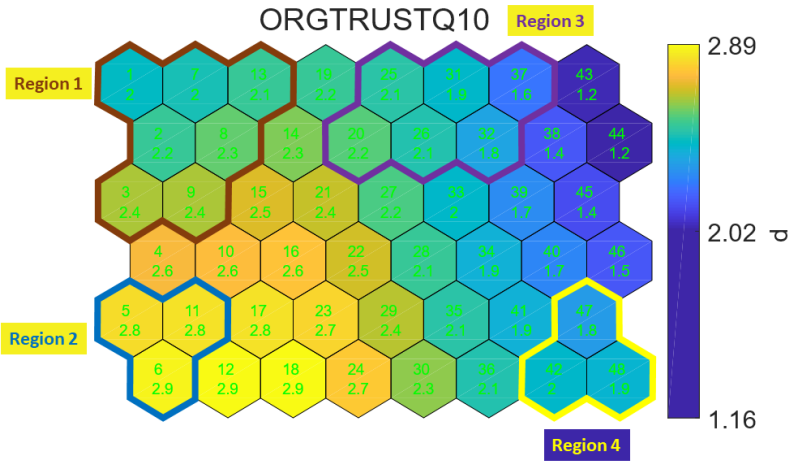


Figure 2.10: Most of my peers/colleagues efficiently do their work even if the unit manager is not around. 0: Definitely Disagree; 1: Inclined to Disagree; 2: Inclined to Agree; 3: Definitely Agree.

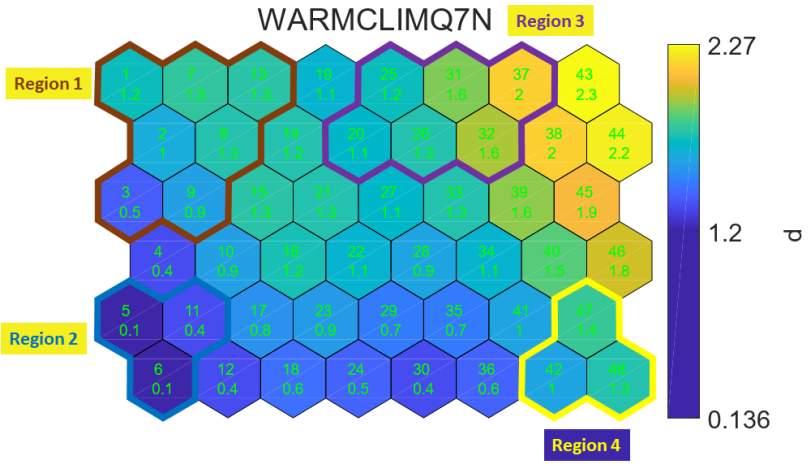


Figure 2.11: People in this unit really do not trust each other. 0: Definitely Disagree; 1: Inclined to Disagree; 2: Inclined to Agree; 3: Definitely Agree.

Colored Map Interpretation In the visualizations, the bright orange color is associated with the higher value of the feature, while the dark blue color means a lower value of the feature. The precised color-value relationship is represented on the reference bar on the right.

The numbers on every cell consist of two elements: the upper number is the cell number; the bottom number is the feature value of the cell (the codebook value of SOM).

The map is organized in such a way: each cell is a small cluster for several subjects, that are **overall** very similar in the aspects of the selected features. The subjects that are in the nearby cells are more similar than the subjects from cells that

are not adjacent.

The colored map is showing the individual feature values (including the target values) one feature at a time. Although different feature has a different colored map, the cluster of the subjects are fixed for every map.

The add-on boarders mark the regions of interests on the map. Further analysis is given on each of the region in the latter part of the Chapter.

Same interpretation for the colored maps is applied for all the colored maps.

Region one: cells 1, 2, 3, 7, 8, 9, and 13. Subjects in these cells have high values (above 2.3) for the output variable 1, $ERREPQ_1$, which indicates that they are more willing to report when a mistake is made, but caught and corrected before affecting the patient. The outstanding characteristic for them is that they have been worked on average a very long time in the current unit: between 14 years and 26 years (indicating by the $EXPCURRUNIT$ map). However, in general these subjects do not give very high score for the peer trust questions (indicating by the rest of the maps).

Conclusion: subjects have worked in the current unit for over 14 years are likely to report the $ERREPQ_1$ error.

Region two: cells 5, 6, and 11. Subjects in these cells also give above average scores for the variable $ERREPQ_1$. It can be noticed easily that they all worked in the current unit for 4 to 6 years, which is relatively short comparing to subjects in other cells. Moreover, they tend to trust their peers very much, giving very high

score (around 3) to $ORGTRUSTQ_5$ and $ORGTRUSTQ_{10}$, and very low score to $WARMCLIMQ_7N$, which is a reverse question (the lower the score, the higher they feel trust).

Conclusion: subjects have worked in the current unit for under 6 years, but have very high trust levels for their peers are likely to report $ERREPQ_1$ error.

Region three: cells 20, 25, 26, 31, 32 and 37. Subjects in theses cells are more willing to report as well. They are also relatively "young" to the current unit, between 4 to 5 years. However, their trust to the peers are not too strong, on the margin of the low trust level: around 2 for both $ORGTRUST$ questions and between 1 for to 2 for the $WARMCLIM$ question.

Conclusion: subjects have worked in the current unit for around 5 years, but somehow feel the lack of the peer trust, are likely to report $ERREPQ_1$ error.

Region four: cells 42 47 and 48. Subjects in these cells are very unwilling to report the error (average score is around or bellow 0.3). They worked in the current unit for 8 to 10 years. They feel somewhat trust among peers but far from strong trust.

Conclusion: subjects who have very high trust and who have very low trust are both likely to report the error. However, subjects who have medium or medium low level of peer trust are uncertain whether they will report the error or not. How long have they been working in the unit also has some effect on the subjects for reporting the error.

2.4.3.2 ERREPQ2

ERREPQ2: When a mistake is made, but has no potential harm to the patient, how likely are you to report this error?

Feature Selection Results

The best feature sets $\mathbf{S}_k^{**}$, where $k = 1, 2, ...20$, for $\mathbf{Y}_2$ are selected by ELM-WFS. The following table lists selected feature names for $k = 1$ to $k = 5$.

Table 2.3: Selected Features for ERREPQ2

	Feature Names				
$\mathbf{S}_1^{**}$	LSHPQ5				
$\mathbf{S}_2^{**}$	Age	WARMCLIMQ8N			
$\mathbf{S}_3^{**}$	SHIFTWRK	LSHPQ5	LSHPQ9		
$\mathbf{S}_4^{**}$	Age	WARMCLIMQ8N	ORGTRUSTQ5	ORGTRUSTQ8	
$\mathbf{S}_5^{**}$	LSHPQ1	LSHPQ10	LSHPQ11	LSHPQ17	ORGTRUSTQ5

The R^2 values for the best features are showing in Figure 2.12.

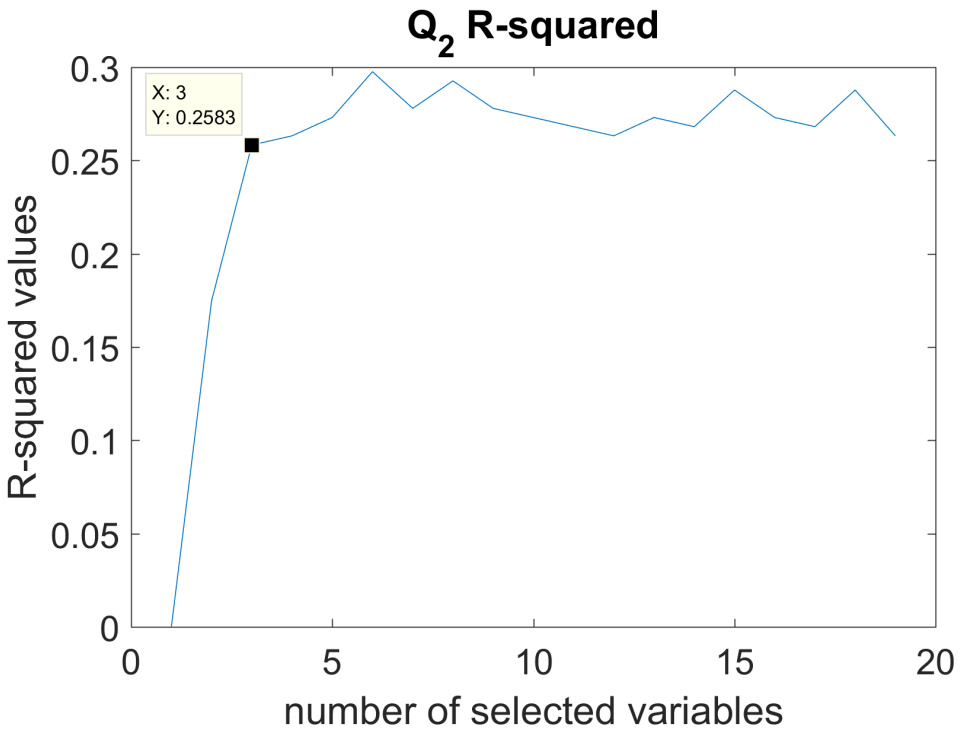


Figure 2.12: R Squared Values for ERREPQ2

Since the R^2 value for $ERREPQ_2$ does not increase a lot after $\mathbf{S}_3^{**}$, $\mathbf{S}_3^{**}$ is the **Optimal** feature set for visualization. The features are:

- SHIFTWRK: Typical working shift.
- LSHPQ5: Seeks differing perspectives when solving problems.
- LSHPQ9: Talks enthusiastically about what needs to be accomplished.

Visualization Results

SOM is built upon the optimal feature set and the outcome variable 2: $(\mathbf{S}_3^{**}, \mathbf{Y}_2)$.

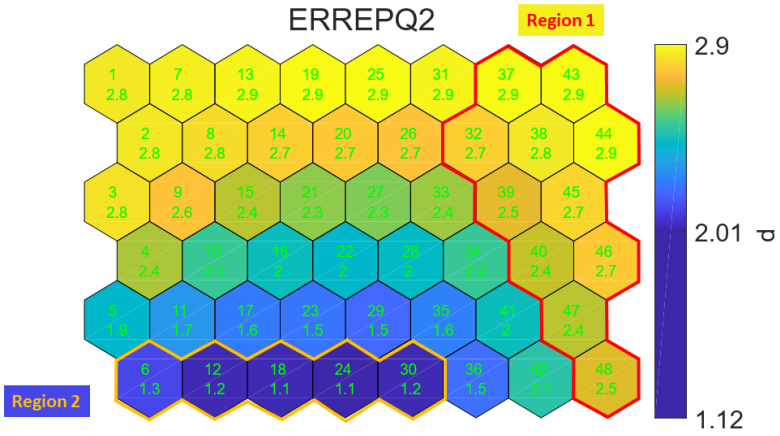


Figure 2.13: When a mistake is made, but has no potential harm to the patient, how likely are you to report this error? 0: Not Likely at All; 1: Somewhat Not Likely; 2: Somewhat Likely; 3: Very Likely.

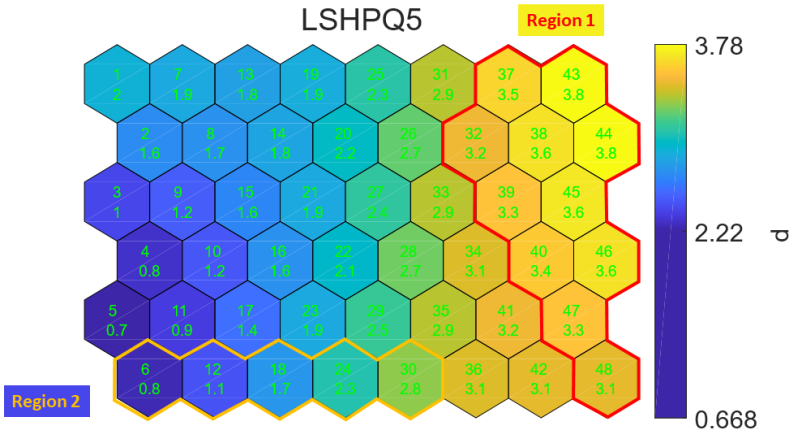


Figure 2.14: Seeks differing perspectives when solving problems. 0: Not at all; 1: Once in a while; 2: Sometimes; 3: Fairly often; 4: Frequently if not always.

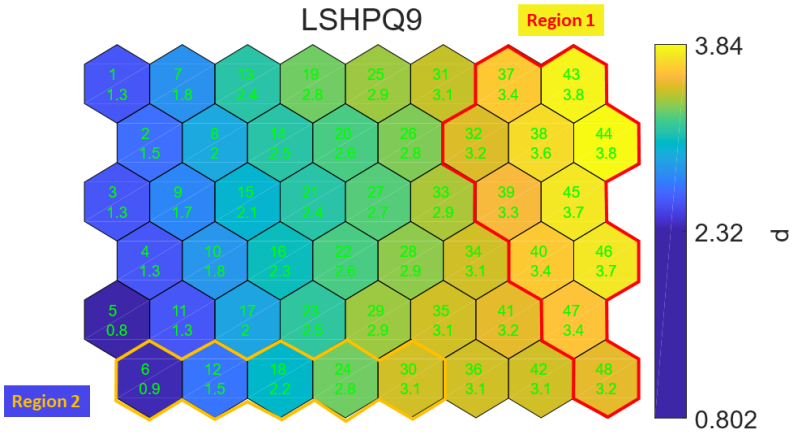


Figure 2.15: Talks enthusiastically about what needs to be accomplished. 0: Not at all; 1: Once in a while; 2: Sometimes; 3: Fairly often; 4: Frequently if not always.

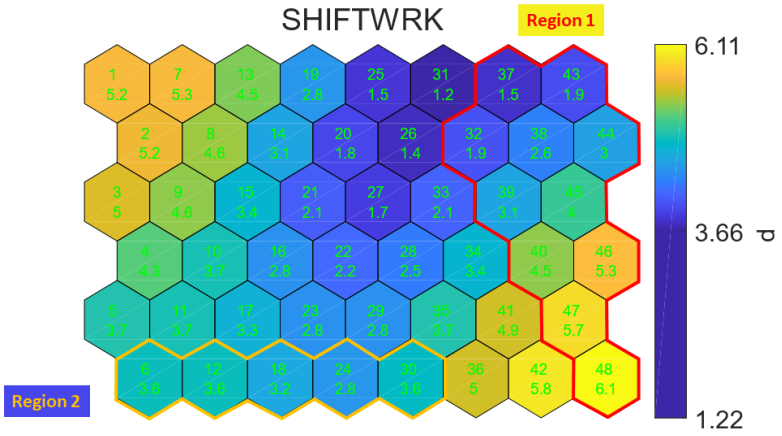


Figure 2.16: Please indicate your typical shift (the shift that your work most of your time). 0=7am-3pm; 1=3pm-11pm; 2=11pm-7am; 3=7am-7pm; 4=7pm-7am; 5=8am-5pm; 6=other; 7= no specific shift/rotating.

Region one: cells 32, 37-40, and 43-48. Subjects in these cells are somewhat likely or very likely to report the error. The outstanding character for these subjects is that they all give very high score for the two unit manager leadership measurement questions.

Conclusion: subjects who believe their unit manager is creative when solving the problems and has enthusiasm about the goal are likely to report the error.

Region two: cells 6, 12, 18, 24, and 30. Subjects in these cells are not likely at all or somewhat unlikely to report the error. However the reason why they are

not motivated to report is not obvious. For subjects in the cell 6 and 12, the low recognition level of the unit manager's leadership may cause the unwillingness to report. For the rest subjects the long work-shift (many of the subjects in these cell are working at a 12 hour work-shift) may be the reason of lack of motivation to report.

Conclusion: subjects who work at a long shift and think their manager are not seeking differing perspective when solving the problems or lack of enthusiasm when speaking of the goals are unlikely to report the error.

2.4.3.3 ERREPQ3

ERREPQ3: When a mistake is made that could harm the patient, but does not, how likely are you to report this error?

Feature Selection Results

The best feature sets $\mathbf{S}_k^{**}$, where $k = 1, 2, ...20$, for $\mathbf{Y}_2$ are selected by ELM-WFS. The following table lists selected feature names for $k = 1$ to $k = 5$.

Table 2.4: Selected Features for ERREPQ3

	Feature Names				
$\mathbf{S}_1^{**}$	LSHPQ18				
$\mathbf{S}_2^{**}$	Age	WORKHRS			
$\mathbf{S}_3^{**}$	EXPCURRUNIT	LSHPQ6	ORGTRUSTQ6		
$\mathbf{S}_4^{**}$	Age	LSHPQ15	LSHPQ17	LSHPQ20	
$\mathbf{S}_5^{**}$	LSHPQ10	LSHPQ18	ORGTRUSTQ6	COMPLERREPQ2	ERREPTIMEQ5

The R^2 values for the best features are showing in Figure 2.17.

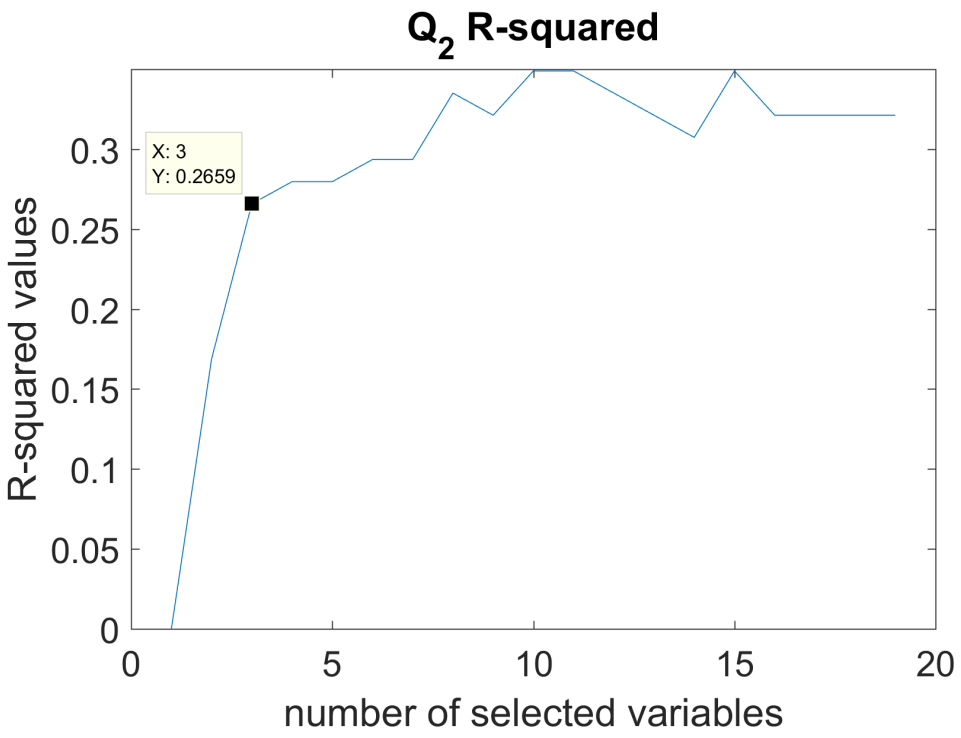


Figure 2.17: R Squared Values for ERREPQ3

Since the R^2 value for $ERREPQ_3$ does not increase a lot after $\mathbf{S}_3^{**}$, $\mathbf{S}_3^{**}$ is the **Optimal** feature set for visualization. The features are:

- EXPCURRUNIT: Years of experience in the current unit.
- LSHPQ6: Talks optimistically about the future.
- ORGTRUSTQ6: My unit manager seems to do an efficient job.

Visualization Results

SOM is built upon the optimal feature set and the outcome variable 2: $(\mathbf{S}_3^{**}, \mathbf{Y}_2)$.

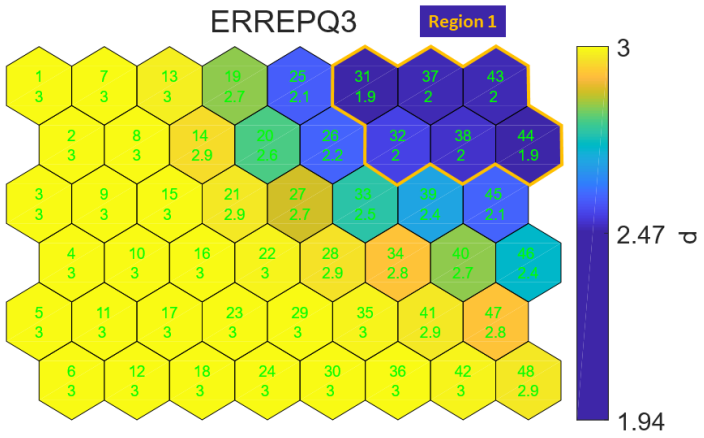


Figure 2.18: When a mistake is made, that could harm the patient, but does not, how likely are you to report this error? 0: Not Likely at All; 1: Somewhat Not Likely; 2: Somewhat Likely; 3: Very Likely.

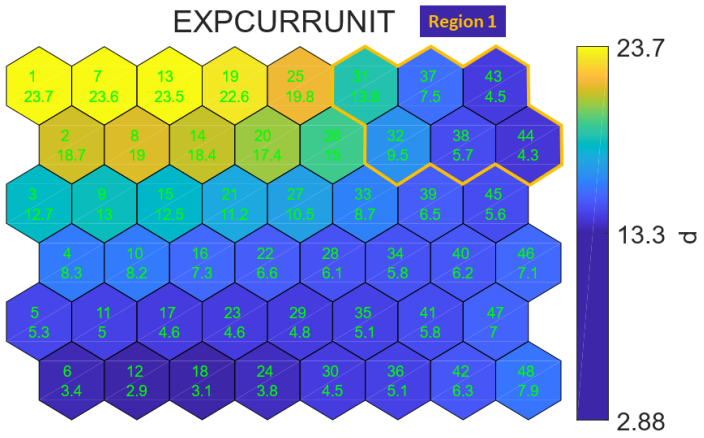


Figure 2.19: How long you have been working in your current unit?

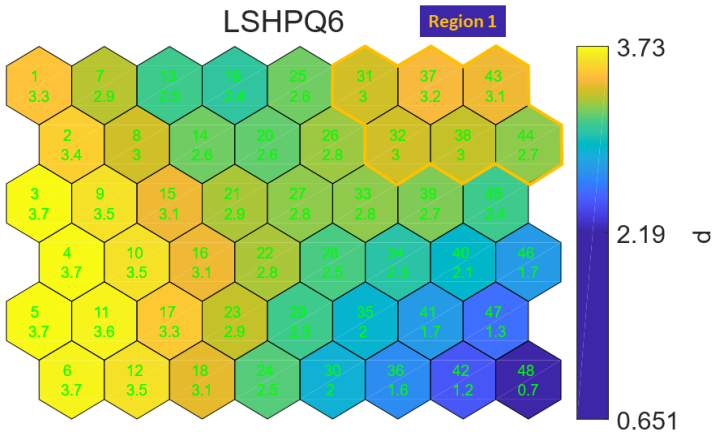


Figure 2.20: Talks optimistically about the future. 0: Not at all; 1: Once in a while; 2: Sometimes; 3: Fairly often; 4: Frequently if not always.

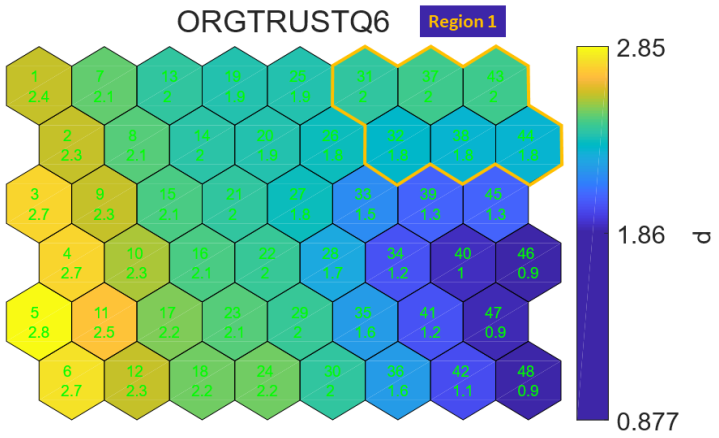


Figure 2.21: My unit manager seems to do an efficient job. 0: Definitely Disagree; 1: Inclined to Disagree; 2: Inclined to Agree; 3: Definitely Agree.

Region one: cells 31, 32, 37, 38, 43 and 44. Subjects in these cells are somewhat likely to report the error. For the out come variable $ERREPQ_3$ the majority people are choosing "very likely to report". However, for this region, subjects are hesitating. The subjects belive their manager are very optimistic about the future according to the $LSHPQ_6$ map, but they don't think their manager can do his/her job efficiently.

Conclusion: Subjects who have some doubts about their manager's efficiency and think the manager is optimistic about the future are somewhat likely to report the error.

2.5 Conclusions for Chapter 2

Results of this data analysis using SOM showed that nurses willingness to report medication error is contingent on three factors of experience in the unit, nursing experience, organizational trust particularly trust in peers, and nurse manager leadership behaviors. Furthermore, the results showed that outcome predictors varied based on level of error severity. Based on this result, hospital administrators should consider focusing on the previously outlined predictors if they want to improve nurses' willingness to report medication errors regardless its level of severity. Using SOM, accounted for the non-liner relationship that exist among the different study variables. Most importantly it showed the pattern of organizational trust development. This information was not evident when we used traditional liner modeling.

The new methodology that is combining ELMs and SOMs has provided an clear understanding of the studied dataset. Some of the analysis are obviously right and similar to the conclusions that can be obtained with traditional data analysis. Nevertheless, more understanding has been obtained. For example, the model is sparse (few variables). It is a well-known results in the field of perception that only 5 to 6 variables can be easily understood by humans [141, 145]. Furthermore unknowns nonlinear interactions between variables have been discovered using our approach. It has to be mentioned that our methodology is suitable for big data: it can handle the 3 attributes of big data: Volume, Velocity and Variety.

In the future, we are planing to use the same methodology to other medical and nursing problems. It is important to work together with practitioners to validate

the results but we are willing to make the methodology nearly automatic and usable by any person that does not have a strong background in machine learning.

CHAPTER 3
A MACHINE-LEARNING-ENHANCED HIERARCHICAL MULTISCALE METHOD FOR BRIDGING FROM MOLECULAR DYNAMICS TO CONTINUA

This Chapter is based on the collaboration with Professor Shaoping Xiao. My contribution is in the experimental part and the implementation of Machine Learning technics. My contribution is roughly 35% of this Chapter, therefore, I will not present this Chapter during the comprehensive exam.

3.1 Introduction for Chapter 3

To accelerate and foster the maturation of technology in designing novel engineering materials and devices, numerical methods [92] play an important role in exploiting new engineering design procedures. Recent developments in nanotechnology demand that molecular building blocks complement and enhance new engineering techniques at the macroscale [153–155]. Therefore, an aggressive development of new computational methods, including multiscale methods [91], is required to address complex physical phenomena at various length and time scales for the integrated design of multiscale, multifunctional materials and products [100].

Multiscale methods have been categorized into two classes: concurrent and hierarchical multiscale methods. Concurrent multiscale methods [108] employ an appropriate model to couple multiple length/time scales so that simulations at different scales are conducted simultaneously. Most of the developed concurrent multiscale methods are atomistic/continuum coupling methods [1, 11, 28, 108, 132, 144, 159], in

which the molecular model is overlapped with the continuum model. One of the main challenges in concurrent multiscale methods is how to couple the scales [34,134] without spurious nonphysical phenomena occurring at scale interfaces or overlapping domains. This challenge motivates recent state-of-the-art developments in concurrent multiscale modeling and simulation [41,119,135].

On the contrary, the scale-coupling or scale-overlapping challenge in concurrent multiscale methods doesn't exist in hierarchical approaches [133], in which the molecular and continuum models are simulated sequentially. Indeed, researchers pay more attention to how to pass information between scales, especially from the molecular model to the continuum model. Homogenization, including the Representative Volume Element (RVE) techniques, is commonly employed to obtain effective material properties from the molecular model for continuum simulations. The Cauchy-Born (CB) rule [32] is one of the most-used homogenization techniques. It assumes that the lattice vectors deform as line elements within a locally homogeneous deformation so that stress-deformation relationships can be derived. It has been extended to study curved membranes [8] and crystalline solids with temperature effects [157,158,160][21–23]. Recently, Ademiloye *et al.* [2] proposed a hierarchical multiscale model based on the CB rule to investigate the elastic properties and biomechanical responses of the erythrocyte membrane. Other than the CB rule, Bogdanor *et al.* [18] adopted a homogenization-based reduced-order multiscale computational model to predict the progressive damage accumulation and failure in composite materials. In addition, Meng *et al.* [102] developed a cohesive law to characterize the interfacial properties

between cellulose nanofibrils by considering the hydrogen bond breaking and reforming at the molecular scale. The developed cohesive model then rendered a superior toughness in continuum simulations.

On the other hand, the RVE techniques utilize a periodic subdomain in the molecular model to calculate effective material properties, which are then passed to the continuum model. Jiang *et al.* [72] used molecular dynamics (MD) to predict basic mechanical behaviors, including elastic and damage responses to external loading conditions. Then, the MD results were used to generate a preliminary elastodamage model for continuum simulations. Grabowski *et al.* [40] developed a multiscale electromechanical model to study carbon nanotube (CNT) composites. They used MD simulations to provide information about the elastic properties and density of polymeric material and CNTs for simulations at the meso- and macroscales. Subramanian *et al.* [131] presented a framework of point-information-to-continuum-level analysis to characterize the behavior of CNT composites. In their method, the stochastic distributions obtained from MD simulations provide a basis to simulate local variations of matrix properties in the continuum model. In addition, Ghaffari *et al.* [37] studied the lubricant between sliding solids via MD simulations and passed the friction coefficient to the continuum model to predict the rolling contact fatigue life.

Recently, due to the development of computer technology and explosive data generation and consumption, data science and analysis via machine learning (ML) has become an efficient tool in science and engineering [121]. Machine learning has been widely applied in the biomedical engineering domain for real-time simulations.

Jahya *et al.* [69] used a validated finite element (FE) model of the prostate and its surrounding structures to generate training data for deep learning (DL), i.e. ML with artificial neural networks (ANNs). Then, the trained ANN could predict a three-dimensional phantom deformation based on given input variables, which include boundary conditions. In another application, Lorente *et al.* [95] used ML regression models, including three tree-based methods and two simpler ones, to simulate the biomechanical behaviors of the human liver during breathing in real time.

In the community of computational mechanics and materials science, data sciences and informatics [70] have been used to accelerate materials development and deployment. Kalidindi *et al.* [74] described a few computational protocols to accelerate significantly the process of building microstructure informatics in the integrated computational materials engineering infrastructure. Gupta *et al.* [44] used a data science approach to establish reduced-order linkages between the material microscale internal structure and its associated macroscale properties. Their training dataset was generated from the mechanical responses of an ensemble of representative microstructures based on FE simulations. In molecular simulations, ML has been used to predict molecular properties [47] for accurate atomistic simulations. Chen *et al.* [21] presented a highly accurate force field for molybdenum by ML on a large material dataset. In a similar work, Glielmo *et al.* [39] proposed a novel scheme to predict atomic forces as vector quantities by ML regression. Artrith and Urban [9] implemented ANN potentials in atomistic material simulations to study titanium dioxide (TiO2). Bélisle *et al.* [13] evaluated a few ML techniques to predict material proper-

ties from training data obtained via MD simulations. In addition, Ibanez *et al.* [64,65] discussed the difficulties in the data-driven or data-intensive approaches that link experimental data to numerical simulations. They proposed the solution by using a data-driven inverse approach to generate the whole constitutive manifold from few complex experimental tests.

A few works have been done by using ML in multiscale modeling and simulation. Matouš *et al.* [97] reviewed predictive nonlinear theories for multiscale modeling of heterogeneous materials. They discussed a predictive image-based multiscale material model, in which statistically representative unit cells were generated via ML to optimally preserve the statistical description of the microstructure. Le *et al.* [83] employed ANNs and proposed a decoupled computational homogenization method for nonlinear elastic materials. In their method, the training samples of the effective potential were computed through random sampling in the parameter space, and then ANNs were used to approximate the surface response and to derive the macroscopic stress and tangent tensor components. Liu *et al.* [93] developed a data-driven approach to predict the behavior of general heterogeneous materials under inelastic deformation. One of their innovations was using an unsupervised clustering algorithm to homogenize the local features of the material microstructure into a group of clusters. This research group [94] recently developed a microstructural database based on the self-consistent clustering analysis to accurately predict a nonlinear material response. In another pioneering work, Fritzen and Kunc [35] used a data-driven approach to investigate the nonlinear behavior of materials with a three-dimensional

microstructure. They performed finite element method (FEM) simulations on the microstructural level first, and the generated simulation data then served as input for a reduced-order model at the macroscale level.

In this Chapter, we propose an alternative data-driven approach by using ML to pass information from the molecular model to the continuum model in a hierarchical multiscale framework. First, MD simulations are conducted to generate the dataset, including training and testing sets, in which the input variables contain deformation and temperature while the output variables are stress components and material failure mode. Then, the generated data is used to train several ML classification and regression models. Finally, the well-trained learning machines are directly implemented in continuum simulations to predict material failure mode and stress components. In this approach, neither constitutive relations nor effective material properties are explicitly derived as achieved in existing hierarchical multiscale methods. The learning machines serve as "black boxes" to replace constitutive relations and failure mode decisions in the continuum model. Such "black boxes" are trained based on the dataset from molecular simulations; therefore, the propose scheme is physical-based and data-driven.

The outline of this Chapter is described as follows. After the introduction, MD simulations and data collection are described in Section 2. The examples include a one-dimensional molecule chain and an aluminum crystalline solid. Sections 3 describes the proposed ML-enhanced hierarchical multiscale modeling as well as the ML regression and classification algorithms. Details about the training processes

are also explained. Continuum modeling and simulation with the implementation of ML-trained predictive models are discussed in Section 4 followed by conclusions and future outlook.

3.2 Molecular Dynamics Simulations

Molecular dynamics has been a powerful tool to elucidate physical phenomena at the nanoscale [107, 152, 154, 156]. In MD simulations, the atoms or molecules in the simulated system follow the laws of classical mechanics. The motion of an atom, e.g. atom i, with mass m_i, is due to its interaction with other atoms in the system according to Newton's second law:

$$m_i \overrightarrow{a}_i = \overrightarrow{f}_i = -\nabla U\left(\overrightarrow{r}_i\right) \tag{3.1}$$

where $\overrightarrow{a}_i$ is the acceleration of atom i , and the interatomic force, $\overrightarrow{f}_i$, applied on atom i is derived from the total potential energy U, which is a function of the position vector, $\overrightarrow{r}_i$.

In MD simulations, the accelerations are calculated via Eq. 3.1. The velocity Verlet method is commonly used to conduct time integrations within the time step of Δt to update velocities and displacements:

$$u\left(t+\Delta t\right) = u\left(t\right) + v\left(t\right)\Delta t + \frac{1}{2}a\left(t\right)\Delta t^2 \tag{3.2}$$

$$a\left(t+\Delta t\right) = \frac{f\left(\overrightarrow{r}\left(t+\Delta t\right)\right)}{m} \tag{3.3}$$

$$v\left(t+\Delta t\right)=v\left(t\right)+\frac{\Delta t}{2}\left[a\left(t\right)+a\left(t+\Delta t\right)\right] \tag{3.4}$$

In this Chapter, MD simulations are conducted to generate a dataset. Two examples are considered here: a one-dimensional Lennard-Jones molecule chain and an aluminum (AL) crystalline solid. The collected dataset includes training and testing samples, which are used to train predictive models, i.e. learning machines, for predicting stresses and material failure/defect modes. Then, the well-trained predictive models are implemented in the continuum model of the proposed hierarchical multiscale method.

3.2.1 One-dimensional Lennard-Jones Molecule Chain

We first consider a one-dimensional molecule chain, which contains 1000 atoms, with periodic boundary conditions. Each atom has a mass of $1.993\times10^{-26}\ kg$. The following Lennard-Jones (LJ) 6-12 potential function is employed to describe the interatomic interactions between the nearest neighboring atoms,

$$U\left(l\right)=4\varepsilon\left[\frac{1}{4}\left(\frac{l_0}{l}\right)^{12}-\frac{1}{2}\left(\frac{l_0}{l}\right)^{6}\right] \tag{3.5}$$

where $l_0=1nm$ is the initial bond length, l is the deformed bond length, and $\varepsilon=1.65\times10^{-18}\ J$ is the depth of the energy well.

At given deformation gradient (F) and temperature (T), a canonical (NVT) MD simulation is conducted until the molecule chain reaches a thermodynamic equilibrium state. Then, the atomic-level Cauchy stress tensor [163] can be calculated

via

$$\boldsymbol{\sigma} = \frac{1}{V} \sum \left(\frac{1}{2} \sum_{i \neq j} \boldsymbol{r}_{ij} \otimes \boldsymbol{f}_{ij} \right) \tag{3.6}$$

where $\boldsymbol{r}_{ij} (= \boldsymbol{r}_i - \boldsymbol{r}_j)$ represents the interatomic distance between atoms i and j, and $\otimes$ denotes the tensor product of two vectors. The cross-sectional area is assumed as $1nm^2$. The sign convention adopted for interatomic forces, $\boldsymbol{f}_{ij}$, is positive for attraction and negative for repulsion. Accordingly, a positive stress indicates tension and a negative stress indicates compression in the one-dimensional case here. It shall be noted that a temperature-related homogenization technique can theoretically derive the stress-deformation gradient relation [157, 158, 160]. However, it was developed for crystalline solids only and has difficulties for other materials without regular lattice structures.

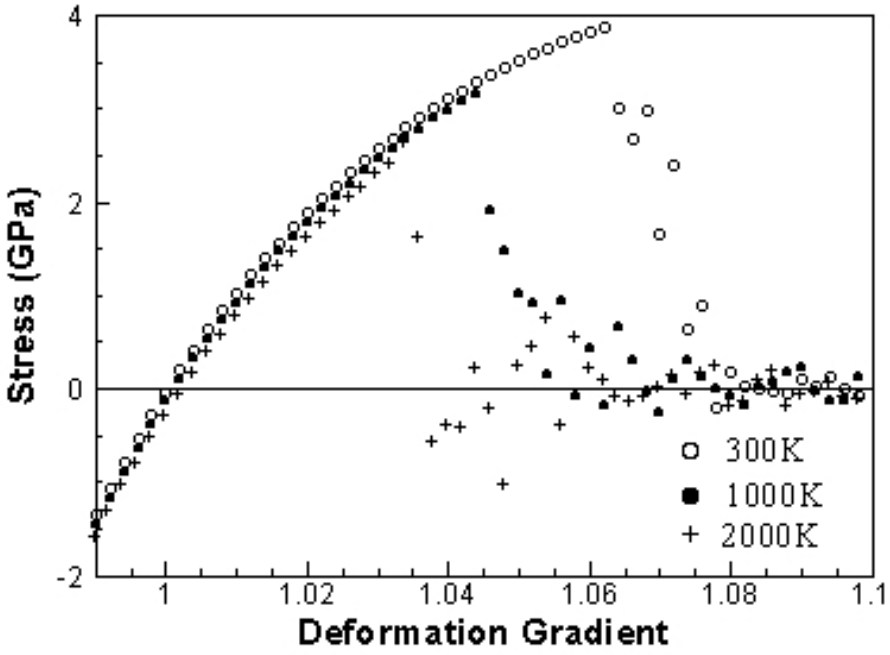


Figure 3.1: Stress-Deformation Gradient Data at Various Temperatures

Figure 3.1 shows the collected stress-deformation gradient data at various temperatures: 300 K, 1000 K and 2000 K. Due to the nature of LJ potential, the stress-deformation gradient relations exhibit hyperelasticity in compression. Although the relations, shown in Figure 3.1, have no big differences between each other when tensile deformation is small, the failure stresses vary significantly at different temperatures. Figure 3.2 demonstrates the failure modes in terms of deformation gradient and temperature. At a higher temperature, the LJ molecule chain would be broken at a smaller deformation gradient.

A number of MD simulations are conducted at various deformation gradients

and temperatures to generate the dataset. To pass the information from the molecular model to the continuum model in this example, two predictive models need to be trained. One is a material failure predictive model, in which the input variables are deformation gradient and temperature, while the output is a Boolean to represent two different material failure modes: failure or non-failure. The other is a stress predictive model, in which the input variables are deformation gradient and temperature while the output target is stress. The above two predictive models will be trained by using ML classification and regression methods, respectively, based on the corresponding dataset collected from MD simulations.

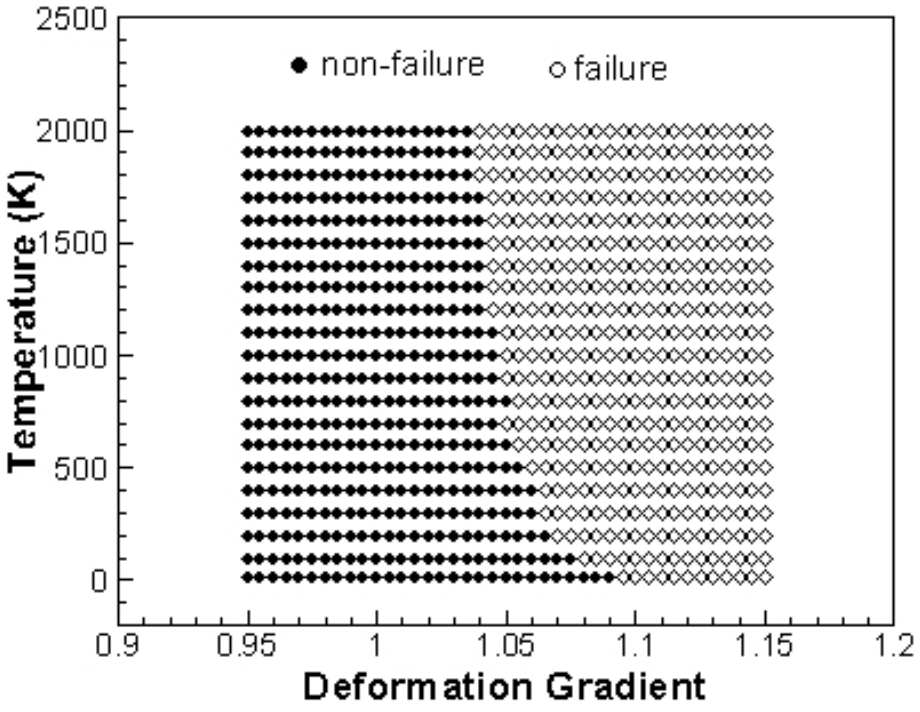


Figure 3.2: Failure and Non-Failure Domains

3.2.2 Aluminum Crystalline Solid

In another example, an FCC Al crystalline solid with <100> orientations in the X, Y and Z directions is studied at a temperature of 300 K via MD simulations. The simulated atomistic model contains twelve lattice units in each direction so that the total number of atoms is 6912. The potential function is a many-body interatomic potential developed by Mishin *et al.* [109]. Molecular dynamics simulations in this study are carried out with the Larger-scale Atomic Molecular Massively Parallel Simulator (LAMMPS) [117]. The time step is 1 fs. Periodic boundary conditions are employed in each direction, and the deformation, represented by the engineering strains, is applied in the $X-Y$ plane only to approximate a two-dimensional simulation model with the plane strain condition.

We first elucidate the mechanical behavior of this FCC Al crystal under uniaxial tension at 0% shear strain. In MD simulations, the first step is equilibration, in which the simulated model is equilibrated in the isothermal-isobaric (NPT) ensemble at a pressure of 0 bar for 20ps. Then, the model is deformed at a constant strain rate of $5 \times 10^{-7}\ fs^{-1}$ in the X direction only, while no deformations are applied in the Y and Z directions. This is a strain-free condition, which is different from the stress-free conditions used in Tschopp and McDowell's studies [139]. Various strain rates are tested, and the above strain rate is chosen due to its minimal effect on stress calculation.

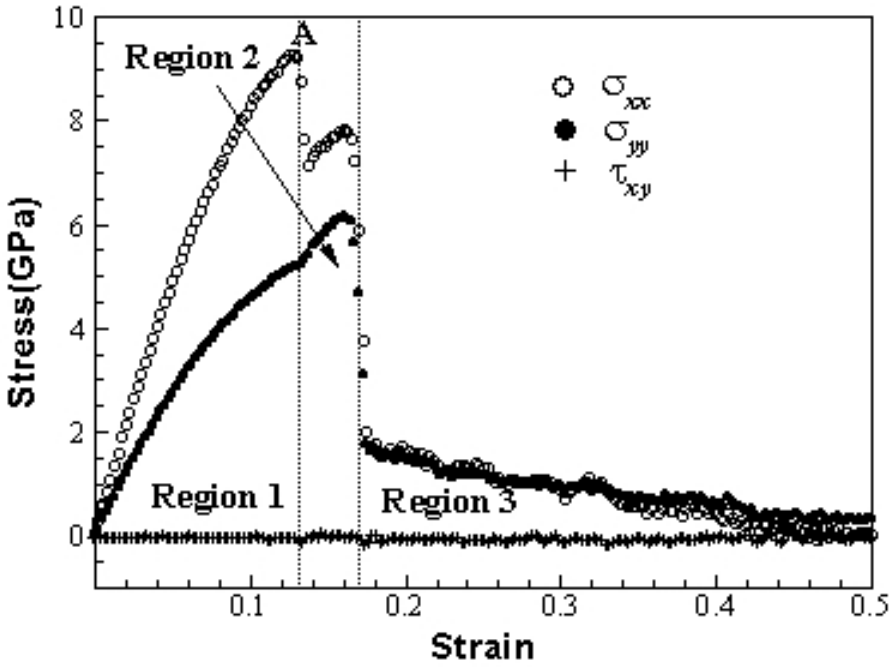
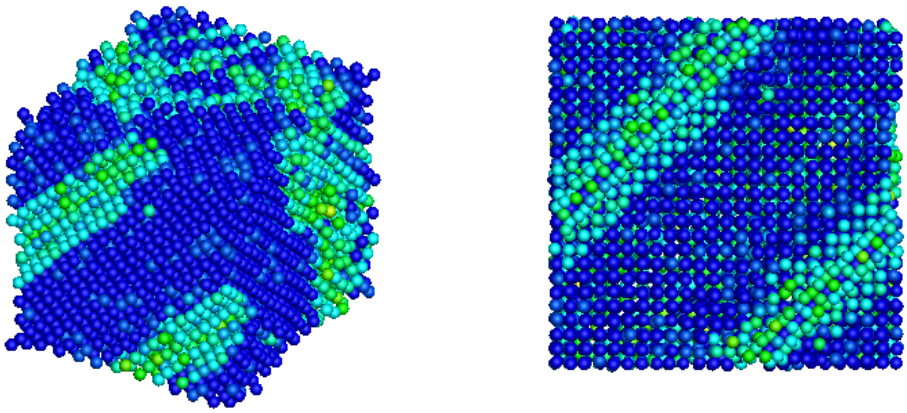


Figure 3.3: Stress-Strain Relation of an FCC Al Crystal in Uniaxial Tension at 300 K

Figure 3.3 shows the stress-strain relation of the FCC Al crystal under uniaxial tension up to 50% strain. Due to the Poisson effect, σ_y is non-zero although $\varepsilon_y = 0$. It can be seen that there are three regions separated by the discontinuities. In Region 1, the Al crystal maintains an almost perfect crystalline structure so that the stress-strain relation represents nearly elasticity until point A, at which a discontinuity occurs. Then, dislocation nucleation and growth are observed in Region 2.

Dislocation is one type of defect in crystals, and it occurs when the atoms are out of position in the crystal structure. Dislocations can be identified by the



(a) Three-Dimensional View

(b) Side View

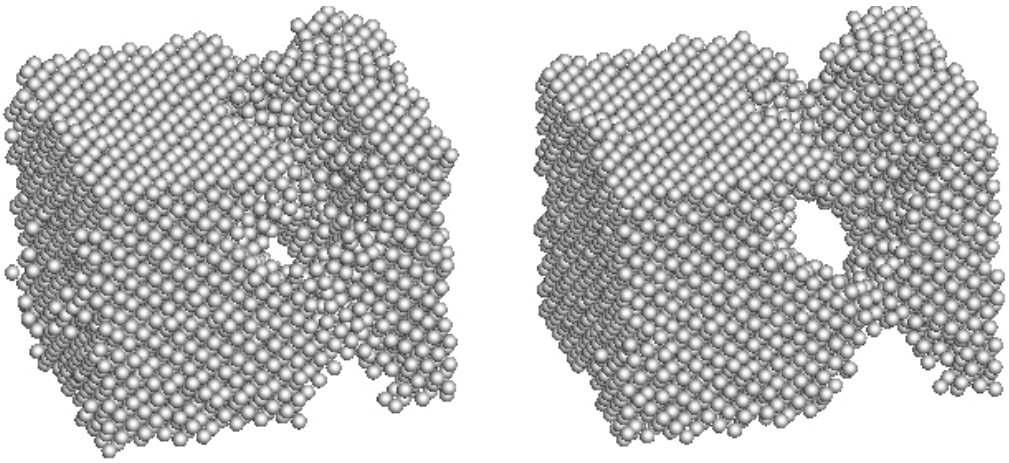
Figure 3.4: Nucleation of Dislocation in Al Crystal at 15% Strain

centro-symmetry parameter (CSP) [76], which is a metric to quantify the local loss of centro-symmetry at an atomic position. The non-centro-symmetry environment is characteristic for most crystal defects, including dislocation. The CSP is calculated as

$$CSP = \sum_{i=1}^{N/2} \left| \boldsymbol{r}_i + \boldsymbol{r}_{i+N/2} \right|^2 \tag{3.7}$$

where $\boldsymbol{r}_i$ and $\boldsymbol{r}_{i+N/2}$ are position vectors from the central atom to a pair of opposite neighbors, and $N = 12$ is the number of nearest neighbors taken into account for FCC crystals. When an FCC crystal is pulled along a <100> direction, dislocation always occurs on a <111> plane and in a <110> direction as shown in Figure 3.4.

When the simulated Al crystal is continuously elongated, voids nucleate and grow. This phenomenon occurs when the stress-strain relationship reaches to Region 3, as shown in Figure 3.3. At the microscale, the nucleation and growth of voids can



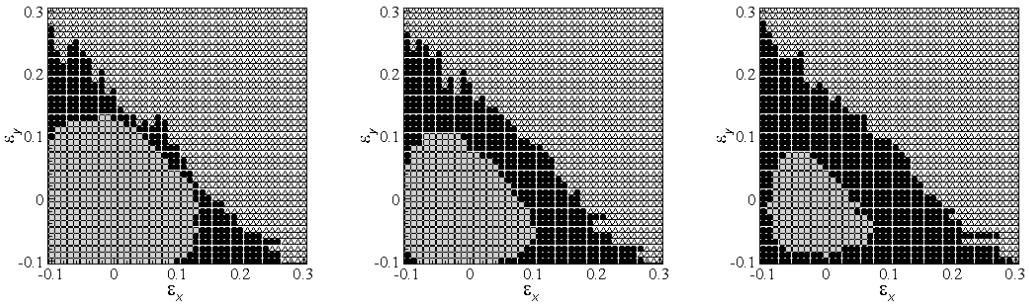
(a) $\varepsilon_x = 25\%$ (b) $\varepsilon_x = 35\%$

Figure 3.5: Voids Nucleation and Growth in the Al Crystal

be viewed as the initiation and growth of damage. Once the damage reaches the pre-defined threshold, a microscale crack initiates. Figure 3.5 illustrates configurations of Al crystal with voids at various strains.

The uniaxial tension simulation described above is conducted at $\varepsilon_{xy} = 0$. We conduct additional biaxial tension, compression and tension-compression simulations with the normal strains ranging from -20% to 50%. The similar physical phenomena of dislocation and void nucleation and growth are observed. The same simulations are repeated at various shear strains ε_{xy} in a range of -20% to 20% to generate the dataset for ML methods to train the predictive models for continuum modeling and simulation.

We only consider the plane strain condition at a room temperature of 300 K in this example. Therefore, in the collected dataset, the input variable of training



(a) $\varepsilon_{xy} = 0\%$ (b) $\varepsilon_{xy} = 4\%$ (c) $\varepsilon_{xy} = 6\%$

Figure 3.6: Domain of Material Defect Modes (∘ Defect-Free; • Dislocation; Δ Void)

and testing samples are strains, including ε_x, ε_y, and ε_{xy}. The output variables are either stresses or material defect modes according to the targets of predictive models. Three predictive models are trained in this example. The first model trained by ML classification is to predict material defect modes, including defect-free, dislocation and void modes. The output target is -1, 0, or 1 to represent three different material defect modes respectively. It shall be noted that Figure 3.3 only shows the case of uniaxial tension with $\varepsilon_{xy} = 0$ and $\varepsilon_y = 0$. Indeed, the domains of the material defect modes are three-dimensional in terms of strain components, as shown in Figure 3.6. The other two predictive models, trained by ML regression, are for stress prediction. One is to predict the stresses when the material is at the defect-free mode, and the other is to predict stresses at the dislocation mode. The output targets of both models are stress components: σ_x, σ_y, and σ_{xy}. We assume that material failure occurs at the location where the void mode is predicted.

3.3 Hierarchical Multiscale Modeling with Machine Learning

Figure 3.7 illustrates the proposed hierarchical multiscale modeling enhanced by ML for solving dynamic problems of continuum and structural mechanics. As discussed in the previous section, two types of data are collected via molecular dynamics simulations. In both types of data, the input variables include strain (or deformation gradient) and temperature if the temperature effect is considered. However, the output variables are different. One type of data is for ML to train a material failure classification model, and the output variable is an integer to represent material failure or defect mode. Another type of data is for ML to train stress regression models so that the output variables are stress components. Both failure classification and stress regression models are implemented in the continuum model to substitute the explicit constitutive relations, which are commonly used in conventional continuum simulations. Consequently, information is passed from the molecular model to the continuum model in our hierarchical multiscale method as shown in Figure 3.7. The details about training predictive models by ML are described below.

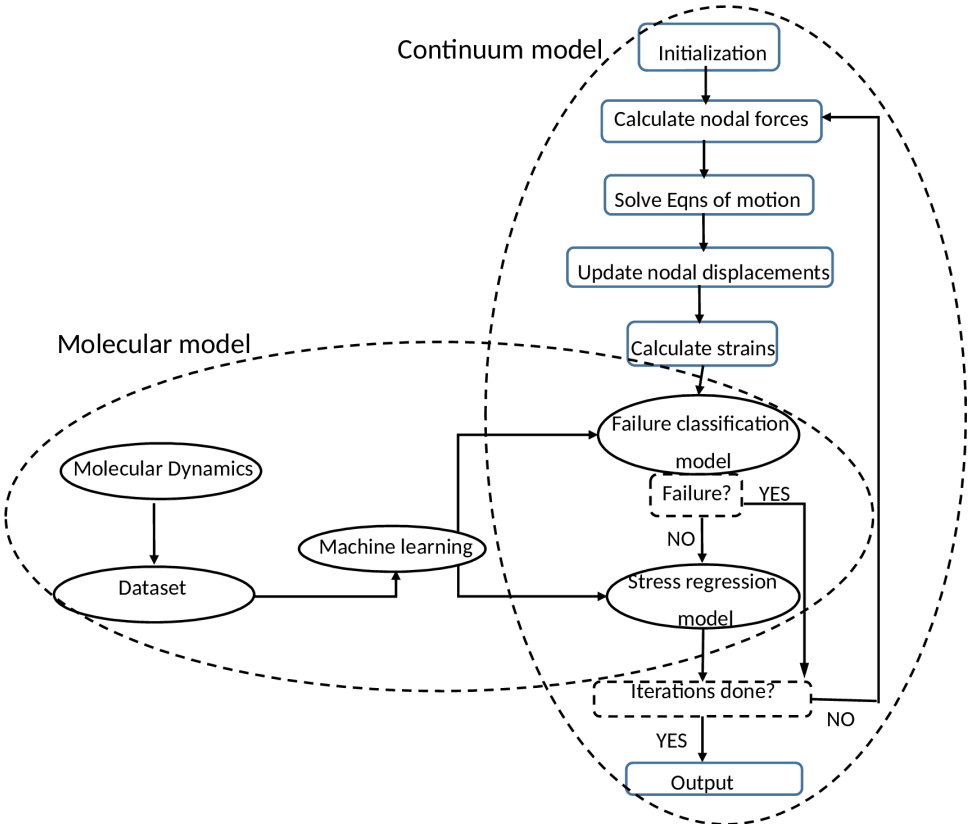


Figure 3.7: Hierarchical Multiscale Modeling Enhanced by Machine Learning

Support Vector Machines (SVMs) are supervised ML algorithms commonly used for regression [126], classification [23], and outliers detection. The estimated output in an SVM nonlinear regression algorithm can be written as

$$\tilde{y}(\boldsymbol{x}) = \sum_{J=1}^{N} (\alpha_J - \alpha_J^*) K(\boldsymbol{x}_J, \boldsymbol{x}) + b \tag{3.8}$$

where N is the number of training samples, α_J and α_J^* are Lagrange multipliers, and b is the bias. K is the kernel function, which transforms the training data from the input to the feature space. There are a few optimization algorithms [126], which

can be used to minimize the error function and to generate predictive models in SVM regression. Similarly, SVMs can conduct classification tasks by constructing hyperplanes in a multidimensional space that separates various labeled cases.

SVMs are adopted in the example of the one-dimensional LJ molecule chain. The input and output variables of data samples are:

$$\boldsymbol{x} = \left\{ \begin{array}{c} F \\ \\ T \end{array} \right\} \quad y_R = \{\sigma\} \quad or \quad y_C = \{-1 \ or \ 1\} \tag{3.9}$$

where F is the deformation gradient, T is the temperature, and σ is the Cauchy stress. In addition, 1 represents material non-failure mode while -1 represents material failure mode. It shall be noted that 90% of the collected dataset is used as the training set, with the remaining 10% comprising the testing set.

There are a total of 436 data samples of $\{\boldsymbol{x}_I, \ (y_R)_I\}$ used to train and test the stress regression model. The radial basis function (RBF) [22], as the kernel function, in our SVM training is expressed as

$$K\left(\boldsymbol{x}_J, \boldsymbol{x}\right) = e^{-\gamma \|\boldsymbol{x}_J - \boldsymbol{x}\|^2} \tag{3.10}$$

where $\gamma = 0.1$. The Normalized Mean Square Error (NMSE) of testing data is 0.38%.

To train the material failure classification model, a total of 861 data samples of $\{\boldsymbol{x}_I, \ (y_C)_I\}$ are used, and the model accuracy for the testing set is 99%. Figure 3.8 demonstrates the material non-failure/failure interface predicted via the learning machine and compared with the collected data samples.

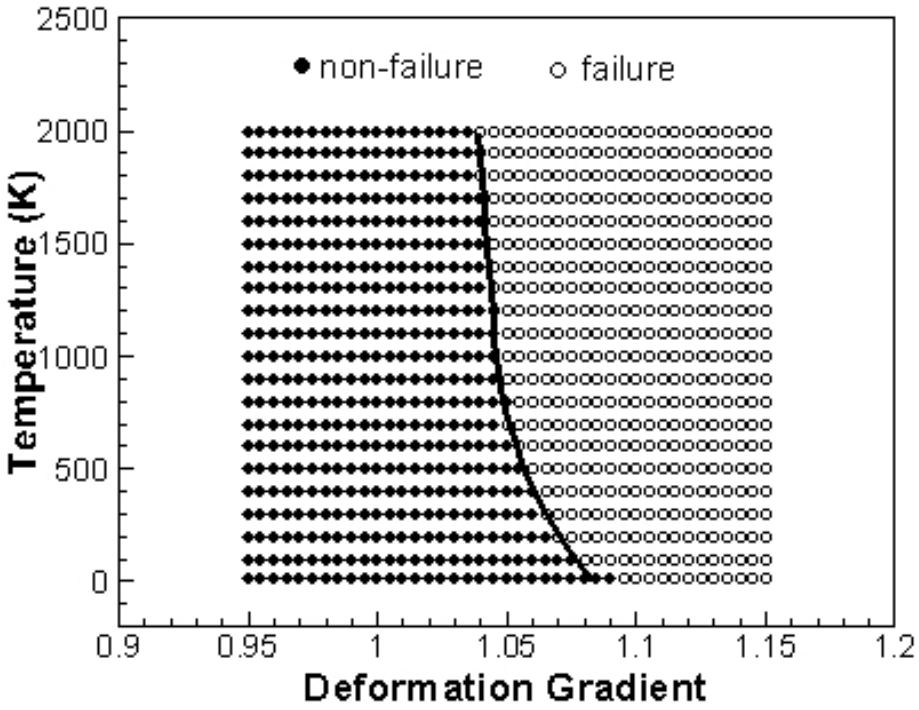


Figure 3.8: The Material Non-Failure/Failure Interface Predicted by Machine Learning

In the framework of hierarchical multiscale modeling described in Figure 3.7, if the FEM is employed in continuum simulations, the well-trained predictive models will be used to predict material failure mode and stresses at each quadrature point at every iteration. It has been shown that SVMs are computationally intensive [5]. Therefore, in the example of the Al crystalline solid, Extreme Learning Machines (ELMs) [43, 53], one of the important emergent ML techniques, are adopted. It has been shown that an ELM is a fast training method for Single-Layer Feed-forward

Networks (SLFNs) [63]. Therefore, an SLFN is used in our ELM model, as shown in Figure 3.9. Although a SLFN has three layers of neurons, the term of "Single" stands for the only layer of linear/nonlinear neurons in the model, and it is the hidden layer. In addition, the input layer provides input variables, while the output layer targets output variables.

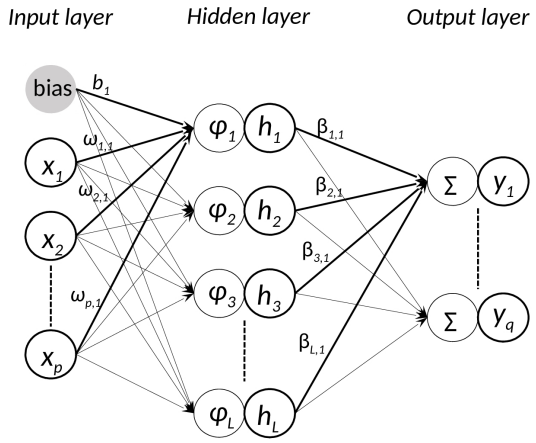


Figure 3.9: An ELM Model with a Single-Layer Neural Network

The ELM model in Figure 3.9 is described below. Considering a set of N distinct training samples $(\boldsymbol{x}_I,\ \boldsymbol{y}_I)$ where $I \in [1,\ N]$, input data $\boldsymbol{x}_I \in R^p$ and corresponding output data $\boldsymbol{y}_I \in R^q$. There are L hidden neurons transforming the input data into a different representation $\boldsymbol{h}$, which is used as output layer weights to esti-

mate outputs. There are two steps in the transformation. First, the data is projected into the hidden layer using the input layer weights, $\boldsymbol{\omega}$, and biases, $\boldsymbol{b}$. Then, the projected data is transformed via the transformation functions, φ. As a result, the transformation of the input data can be mathematically expressed as

$$h_j = \varphi_j \left(\boldsymbol{\omega}_j^T \boldsymbol{x} + b_j \right) = \varphi_j \left(\sum_{i=1}^{p} \omega_{ij} x_i + b_j \right) j = 1 \dots L \tag{3.11}$$

It is known that the hidden layer is not constrained to have only one type of transformation function, i.e., activation function, in neurons. Different functions can be used: linear, sigmoid, hyperbolic tangent, and some radial basis functions (RBFs) [22, 52]. Particularly, linear neurons learn linear relationships between input and output data. In addition, the RBF neurons use distances between samples and centroids as inputs, and any norm including L^1, L^2, and L^∞ norms of distances can be used. Consequently, the estimated outputs of the kth training sample are then calculated as

$$\tilde{y}_k = \boldsymbol{\beta}_k^T \boldsymbol{h} = \sum_{j=1}^{L} \beta_{jk} \varphi_j \left(\sum_{i=1}^{p} \omega_{ij} x_i + b_j \right) = y_k + \varepsilon_k k = 1 \dots q \tag{3.12}$$

where ε_k is the noise, i.e. the estimate residual. The 10-fold cross-validation technique [90] is used in ELMs training.

It shall be noted that an ELM model can be used for multi-layered feedforward neural networks [56] as well. As a difference from traditional ML theories, the hidden neurons don't need to be tuned in ELM models, and all the parameters of hidden neurons can be randomly generated and independent of the training data. Indeed, an ELM [5, 55] can universally approximate any continuous function with almost

any nonlinear and piecewise hidden neurons. Therefore, it can solve any regression problem with a desired accuracy when enough hidden neurons and training data are given. In addition, multi-label classification problems [140] can be handled similarly. On the other hand, unlike the back-propagation [66] training procedure, there is no dependence between the input and output weights so a non-iterative linear solution for the output weights becomes possible. Therefore, it provides a speedup of 5 orders of magnitude in ELMs compared to Multilayer Perceptron (MLP) [51,122], or a speedup of 6 orders of magnitude compared to SVMs [25] based on the studies of [5] .

In the Al crystal example, the ELM classification model, used to classify material defect modes, is trained with a total of 229,881 data samples, in which the input and output variables are

$$\boldsymbol{x} = \left\{ \begin{array}{c} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{array} \right\}, y_C = \{1,\ 0\ or - 1\} \tag{3.13}$$

where ε_{xx}, ε_{yy} and ε_{xy} are engineering strains. Among the three output classes, 1 represents the material defect-free mode, 0 represents the material dislocation mode, and -1 represents the material void mode. In the ELM classification without overfitting, various numbers of neurons with hyperbolic tangent functions are used, and the accuracies are listed in Table 3.2. It shall be noted that the ELM neural network with 2,000 neurons is implemented in the continuum model because it utilizes fewer neurons but achieves a sufficient accuracy.

Table 3.1: Accuracies of ELM Classifications

Number of neurons	Classification accuracy
100	93.7%
500	97.4%
1,000	97.9%
2,000	98.4%
3,000	98.6%
20,000	99.1%

It has been shown in Figure 3.3 that the stress is dramatically reduced once voids are nucleated. Therefore, we consider material failure occurring at the location where material void mode is predicted. Consequently, two ELM regression models are trained to predict stresses. One is for the material defect-free mode, and the other is for the material dislocation mode. The input and output variables of data samples are

$$\boldsymbol{x} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{Bmatrix}, \boldsymbol{y}_R = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} \tag{3.14}$$

where σ_{xx}, σ_{yy}, and σ_{xy} are Cauchy stresses. There are 25,278 and 78,084 data samples collected to train those two predictive models respectively. In the ELM regression model for defect-free stress prediction, there are 500 nonlinear neurons with

hyperbolic tangent functions. The NMSEs of the three stress components are 0.02%, 0.03%, and 0.14%, respectively. In the ELM regression neural network for dislocation stress prediction, there are a total of 3000 nonlinear neurons with hyperbolic tangent functions in the hidden layer. The NMSEs of the three output stress components are 0.16%, 0.16%, and 4.1%, respectively. It can be seen that the learning machine to predict stresses in the material dislocation mode has larger NMSEs than the learning machine for the material defect-free mode. The reason is that stresses vary due to dislocation nucleation, growth, and movement in the material dislocation mode. More features, including dislocation density and orientation, need to be considered in ELM training. In addition, temperature could be an additional feature if the temperature effect is considered.

3.4 Continuum Modeling and Simulation

3.4.1 One-Dimensional Lennard-Jones Molecule Chain

After the predictive models are trained via SVM methods, shock wave propagation in a 200 μm-long L-J molecule chain is modeled as a continua and simulated by using FEM. The chain is discretized with 200 two-node elements. Each element contains 1000 atoms, and the nodal mass is $1.993 \times 10^{-23} kg$. The cross-sectional area is still $1nm^2$, and the time step is set as 5.0 fs.

We first study the wave propagation along the molecule chain at 300 K when a square pulse load is applied at the left end of the chain while the right end is free. This compressive pulse has an amplitude of $1nN$ and a period of $0.5ps$. Figure

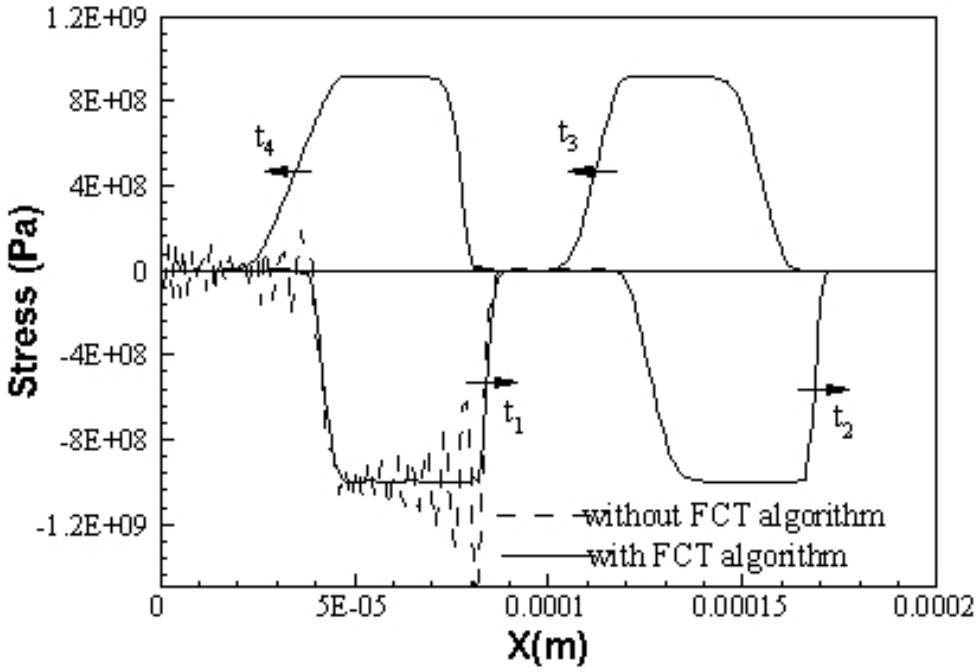


Figure 3.10: Stress Shock Wave Propagation in LJ Molecule Chain Subjected to a Square Pulse Load at 300K

3.10 shows the configurations of stress shock wave propagation at four various times: $t_1 = 1ps$, $t_2 = 2ps$, $t_3 = 3.5ps$, and $t_4 = 4.5ps$. After the compressive pulse is applied, there is a compressive stress shock wave propagating along the chain as shown in Figure 3.10. It can be seen that the oscillations are generated behind the shock wave fronts: the loading and unloading wave fronts. Generally, the oscillation occurs because numerical methods have difficulty reproducing strong discontinuities. A common solution is using artificial viscosity to smooth the shock wave fronts. In this Chapter, the flux-corrected transport (FCT) algorithm [151] is applied to eliminate the oscillations.

Due to the hyperelastic nature of LJ potential when LJ bonds are compressed, the secant modulus is larger at a higher compressive stress, and the wave speed is faster. Consequently, the unloading wave front becomes gentler while the loading wave front remains steep. The phenomena can be observed in the wave profile at time t_2 in Figure 3.10. After the stress wave is reflected by the right end, which is free, the compressive stress wave becomes a tensile stress wave, and different phenomena are then observed. Since the stress-deformation gradient relations in Figure 1 indicate that the secant modulus is lower at a higher tensile stress, the wave speed is slower. Therefore, in the wave profiles at time t_3 and t_4 in Figure 3.10, the loading wave becomes gentler while the unloading becomes steeper.

When a sinusoidal pulse load with an amplitude of 10 nN and a period of 0.5 ps is applied, the phenomena similar to those shown in Figure 3.11 are observed. After wave reflection at the right end, the tensile wave propagates. Theoretically, when

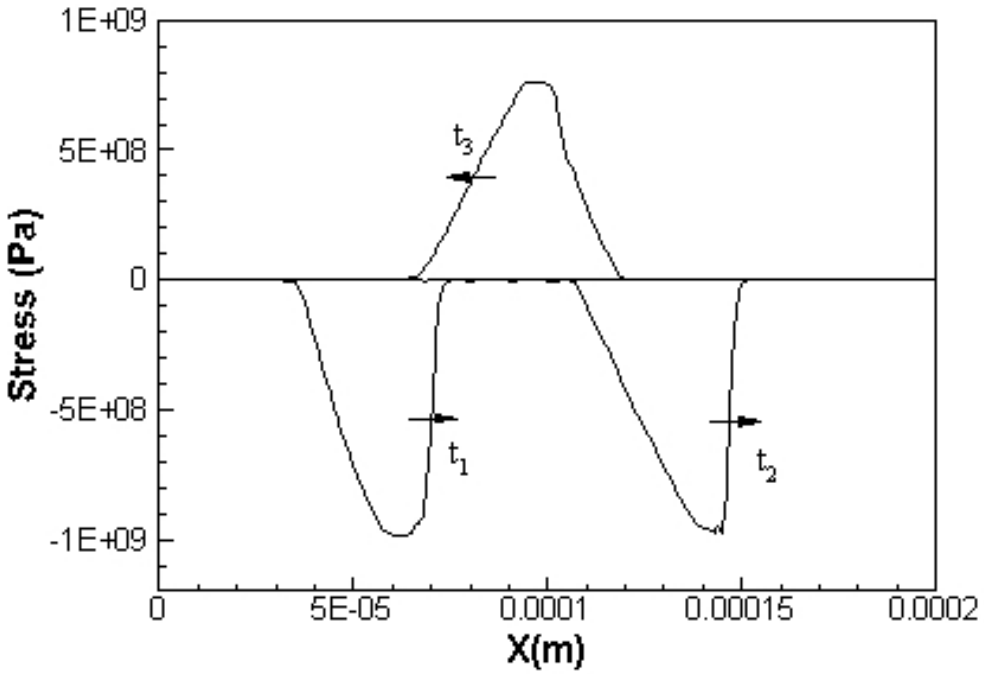


Figure 3.11: Stress Shock Wave Propagation in LJ Molecule Chain Subjected to a Sine Pulse Load at 2000K ($t_1 = 1ps$, $t_2 = 2ps$, $t_3 = 4.5ps$)

the tensile tress reaches to a critical level, i.e., the failure stress at the deformation gradient threshold, the material failure occurs as well as spallation. In our study, the SVM classification model is used to check each element for failure occurrence. Once the failure is predicted, the spall thickness and speed are then calculated. Table 2 lists the spall thicknesses and speeds under various temperatures between 300 K and 1800 K. Obviously, at a higher temperature, material failure occurs earlier, so that the spall has smaller thickness and higher speed.

Table 3.2: Spall Thicknesses and Speeds at Various Temperatures

Temperature (K)	Spall thickness (μm)	Spall speed (m/s)
300	32	4095.2
600	27	4339.1
900	25	4442.8
1200	22	4687.4
1500	21	4887.8
1800	20	5076.3

3.4.2 Al Crystalline Solid

Here an Al crystalline solid subjected to uniaxial tension is considered. Plane strain is assumed, and the simulated object has a length of 2 mm and a height of 2

mm. There is a hole with a radius of 0.15 mm located at the center of the solid. 0.01 μs is chosen as the time step, and a prescribed extension of 0.01 μm is applied on the top and bottom surfaces of the solid at every time step. Such a small extension is chosen in order to approximate quasi-static simulations. There are a total of 1766 nodes and 3379 triangular elements in the FEM model as shown in Figure 3.12. Since three-node linear triangular elements are used in the continuum model, the calculated strain tensor in each element is a constant at every time step. The failure classification model is used first to identify material failure (defect) mode for each element. In our simulation, if the material void mode is detected in an element, the material failure occurs in this element, and the element becomes a void. Otherwise, for an element in either material defect-free or dislocation mode, the stress regression model is used to predict the stress tensor. The above procedure is repeated on each non-failure element at every time step.

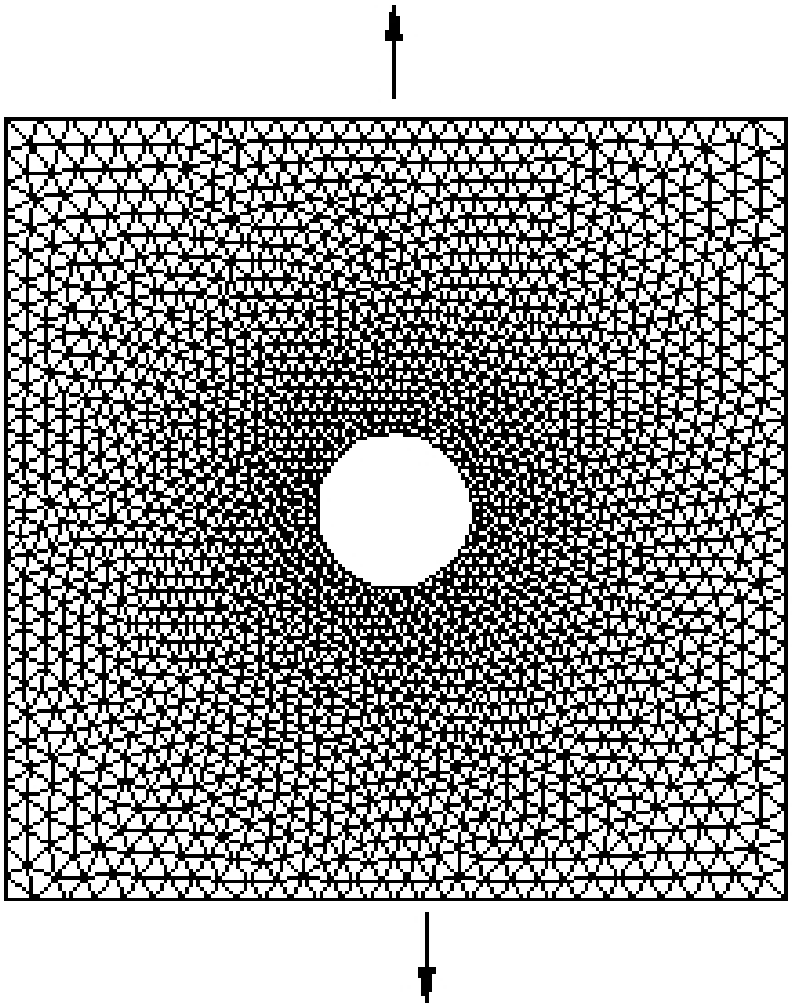
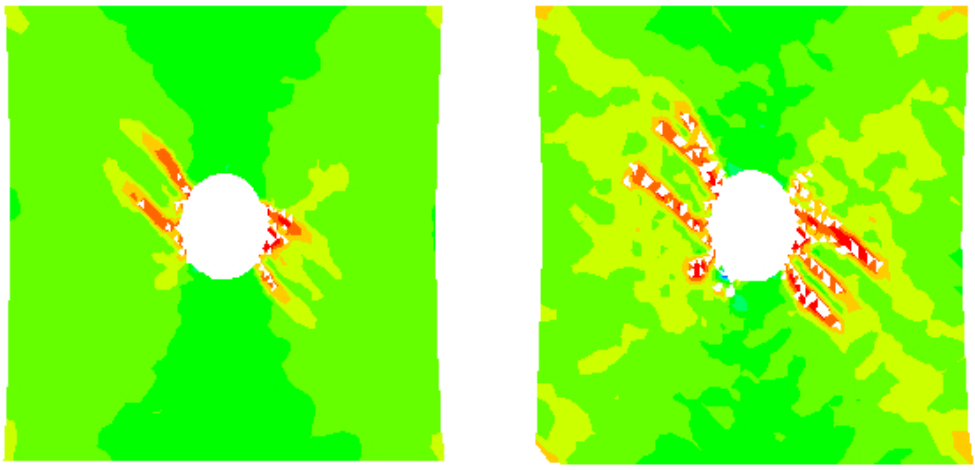


Figure 3.12: An Al Crystalline Solid Is Subject to Uniaxial Tension

Figure 3.13 illustrates the evolution of strain localization in the simulated Al crystalline solid under uniaxial tension. It can be clearly noted that the elements with material dislocation or void modes are detected and shear band paths [127] are observed. Particularly, micro-cracks are initiated in the element with material void modes and then form macro-cracks along the shear band paths.



(a) 6.4% Strain Applied (b) 8.1% Strain Applied

Figure 3.13: Evolution of Strain Localization in a Central-Holed Al Crystalline Solid

3.5 Conclusions for Chapter 3

In the proposed hierarchical multiscale method, ML played an important role in bridging different scales. The dataset, collected via MD simulations in the molecular model, was used to train a few predictive models, i.e., learning machines, and then the well-trained learning machines were implemented in continuum modeling and simulation. The collected dataset represents material physical phenomena at the nanoscale, including stress-strain relations, dislocation phenomenon, and failure occurrence. Based on the collected molecular database, either SVMs or ELMs were trained for evaluation of stress and determination of the material failure/defect mode, which were related to the macroscale mechanical behaviors: stress response, shear band phenomenon, and micro/macro-crack initiation and propagation. The ex-

amples of molecule chain and aluminum crystalline solid demonstrated the proposed hierarchical modeling and simulation enhanced by ML. Our simulation experience confirmed that ELMs were faster and more accurate than SVMs. It is expected that learning machines can catch detailed physical phenomena in the molecular model and pass the information to the continuum model as long as sufficient neurons without overfitting and proper physical features (i.e., input variables) in the dataset are provided. Although only simple examples were demonstrated in this Chapter, more potential in-depth research can be done within the same framework. Of course, the larger number of the features involved, the more challenges will be raised in big data collection and mining.

In general, the data science process includes framing the problem, collecting/processing data, exploring data, performing in-depth analysis, and communicating the results of the analysis. In the engineering domain, the first step of framing a problem is transforming a physical problem to a mathematical model, which can be solved by numerical methods, including multiscale methods. In addition, the features of the dataset, which need to be collected in the next step, must be identified based on the original physical problem. In the examples discussed in this Chapter, only deformation and temperature were considered as the features. Indeed, more features can be considered according to the nanostructured materials to be studied and the physical phenomena to be investigated. For example, to study the mechanical behaviors of nanocomposites, additional features could include inclusion density, orientation, and distribution. In another example of studying the role of defects in mechanics of

materials, defect density, defect size, and dislocation orientation could be considered as features as well.

The proposed machine-learning-enhanced multiscale method is closely accompanied by the rest of the data science process. First, the data collection is conducted via MD simulations in the proposed framework. Generally, a set of randomly generated input variables initiates one MD simulation and then generates a single data. Therefore, a large dataset requires intensive MD simulations, especially for high-dimensional feature spaces. Fortunately, those MD simulations can be run independently, and researchers can take advantage of current parallel, grid, or cloud computing techniques. The effort of processing data, i.e., cleaning data, is minimal because the dataset is collected from physical-based molecular simulations. Next, based on the physical phenomena that we intend to investigate at the nanoscale and the messages that we want to pass to the macroscale, data exploration is the step during which the output targets, including stress, damage initiation and growth, and failure occurrence, are identified, as shown in this Chapter. Then, the key to perform in-depth analysis is employing appropriate ML algorithms. In the proposed multiscale method, the learning machines play an important role in predicting outputs in continuum simulations at each material point and at every iteration after being trained. Consequently, a fast-speed learning machine with high accuracy is required in data analysis to avoid computational intensity in the next step. It is obvious that ELMs, reinforced by parallelization, offers a much better solution than the others, including SVMs. In the last step, communicating results is passing the information from the

molecular model to the continuum model via the well-trained learning machines.

Ideally, MD is one way to sample the ensemble by generating configurations deterministically at the nanoscale. The domain average is then used to evaluate mechanical and thermodynamic quantities, including stresses. However, randomly generated initial configurations and temperature regulations can introduce statistical noises. For example, the same strain state may result in different stress states at various MD simulations. To reduce the noises, time averages shall be used as the ensemble averages to generate output data based on the ergodic hypothesis. On the other hand, even for big data with such statistical noises, ELMs can provide confidence intervals around the best predictions.

In this Chapter, only the nano- and macroscales are considered, and learning machines are employed to pass the information from the molecular model to the continuum model. Indeed, the hierarchical multiscale model can be extended to include various scales, and massages can be passed in the same manner. The quantum scale can be added as the smallest scale. Not only the interatomic forces but also bond breaking and reforming in the molecular model can be determined by learning machines, which are trained based on the dataset collected from quantum calculations. In addition, the microscale or the mesoscale can be added between the nano- and the macroscales to link nanomechanics and structure mechanics by micromechanics. Such a bottom-up multiscale strategy will enhance novel material design coupled with engineering product design, which is usually a top-down process.

CHAPTER 4
ELM-SOM+: A CONTINUOUS MAPPING FOR VISUALIZATION

4.1 Introduction for Chapter 4

In this Chapter, we propose a new topology-preserving nonlinear dimensionality reduction tool: ELM-SOM+. The method incorporates the idea of SOM, using a 2-D manifold to capture the data topology. However, it creates a continuous projection instead, meanwhile giving small reconstruction error. We first describe the basic components of ELM-SOM+, and its applicability in Section II. In Section III, we successfully present and analyze the results of ELM-SOM+ for nine diverse datasets. Conclusion and future work are shown in Section IV.

4.2 Methodology

This thesis presents a nonlinear dimensionality reduction method for visualization: ELM-SOM+. This method is based on both Extreme Learning Machine (ELM) [20, 106] and Self-Organizing Map (SOM). The outline of ELM-SOM+ algorithm is demonstrated in Figure 4.1. At the beginning, the data topology is captured by SOM, creating a discrete projection X_p of the original data X, in 2-D space. Then, the initial projection X_p is imitated by an encoder: (ELM_{ENC}), which creates a continuous projection: $\hat{X}_p$. Next, $\hat{X}$ in the original dimension is reconstructed from $\hat{X}_p$ by a decoder: ELM_{DEC}, generating an approximate version of the original data. This allows the calculation of the reconstruction error between $\hat{X}$ and X. Lastly, the reconstruction error is minimized by optimizing the ELM weights in the ELM_{ENC}, which

improves the quality of the projection: $\hat{X}_p$, therefor, the quality of the reconstruction: $\hat{X}$ as well.

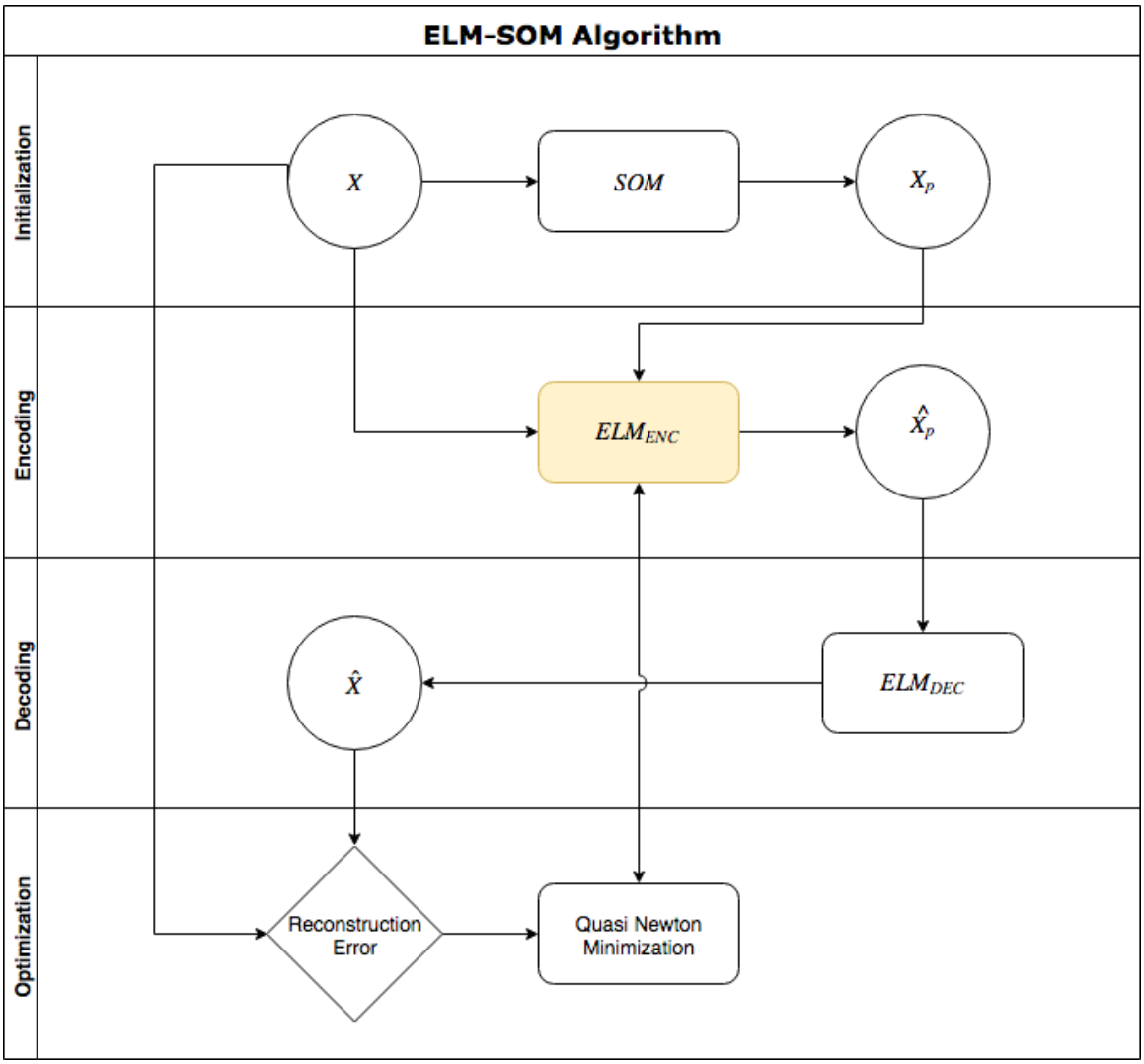


Figure 4.1: ELM-SOM Algorithm

4.2.1 Self-Organizing Maps

SOM is introduced by the Finnish professor Teuvo Kohonen in the 1980s [78]. It is an unsupervised learning tool [29,89,103], and a popular nonlinear dimensionality reduction tool that uses a predefined 2-D lattice (see Figure 4.2) to capture the topology of the data in the high dimension [3].

Each node in the lattice attains a weight vector $\mathbf{w}_i$ in the original d-dimensional data space as the input vectors $\mathbf{x}$.

At the beginning all the weight vectors are randomly initialized. For each input vector $\mathbf{x}_k, k \in [1, N]$, the pairwise distances between $\mathbf{x}_k$ and every weight vector $\mathbf{w}_i$ is calculated. The Best Matching Unit (**BMU**) for $\mathbf{x}_k$ is the note whose weight vector $\mathbf{w}_u$ has the smallest distance with $\mathbf{x}_k$.

When the **BMU** is found, the lattice weights are updated as:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) - \epsilon(t)\lambda(\mathbf{m}_{BMU}, \mathbf{m}_i, t)(\mathbf{m}_i(t) - \mathbf{x}_k), \tag{4.1}$$

where $\epsilon(t)$ is the adaption rate, and the $\lambda(\mathbf{m}_{BMU}, \mathbf{m}_i, t)$ is neighborhood function that decide the influence range of the updating. Finally, after a considerable number of iterations, these weight vectors will converge, hence the SOM is trained.

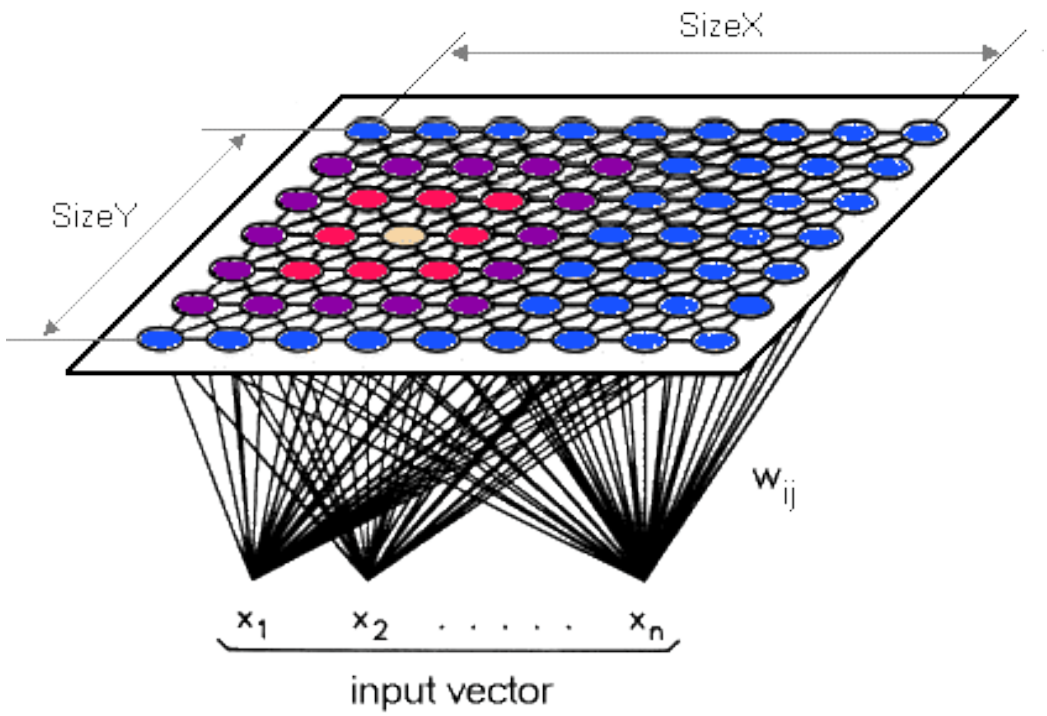


Figure 4.2: Self-Organizing Maps

After the training, according to SOM algorithm, each input vector $\mathbf{x}_k$, is projected to the corresponding **BMU** on the 2-D lattice. Therefore, Self-Organizing Maps are performing a discrete nonlinear dimensionality reductions.

4.2.2 ELM-SOM+

Self-Organizing Map is a powerful visualization tool to create 2-D projections, nonetheless, the projection is discrete. The projection is on the pre-defined grid, which has at most s^2 possible values, where s^2 is the total number of the nodes in SOM. As a result, the reconstruction of the data can only be discrete. This Chapter

extends the original idea of SOM, creating a topology-preserving projection: ELM-SOM, which is similar to SOM, yet without the limitation of the discrete projection. In ELM-SOM projection, points that are close in the original space, are projected to close yet distinguishable places, instead of projecting to the same BMU as SOM does. Therefore, the ELM-SOM projection is continues. This continuity allows a better reconstruction of the data. The reconstruction error is used to measure the quality of the projection [114]. Comparing with SOM, ELM-SOM+ can largely decrease the reconstruction error. The next paragraphs (**Phase-I, II, III and IV**) are presenting the ELM-SOM+ algorithm in detail. They are presented in Figure 4.1.

Phase-I: Learning the Data Topology with SOM: A SOM is built at the first step of ELM-SOM+ to preserve the topology of the data. For each data point $\mathbf{x}_i \in \mathbb{R}^d,\ i \in [1, N]$, the BMU of $\mathbf{x}_i$ is $\mathbf{c}_k \in \mathbb{R}^d,\ k \in [1, s^2]$. Each vector $\mathbf{c}_k$ is a node of the SOM, and can be seen as a cluster center for the data points. The total number of the centers is s^2, since for simplicity, we define the map to be a square of the size of $s \times s$. We define the set C_k as the index set for the points whose BMUs are center $\mathbf{c}_k$. For example, $\forall j \in C_k,\ \mathbf{x}_j$ have the BMU of $\mathbf{c}_k$.

For each center $\mathbf{c}_k$, the corresponding projection is $\mathbf{x}_{pk} \in \mathbb{R}^2,\ k \in [1, s^2]$. After the SOM training, the data topology is learned by the following transformation:

$$P(\mathbf{x}_j) = \mathbf{x}_{pk},\ \forall j \in C_k, \tag{4.2}$$

where $j = [1, \ldots, N], k = [1, \ldots, s^2]$.

All the data points in the set C_k have the same projection. The projection

$\mathbf{x}_{pk}$ are discrete in space. In ELM-SOM+, we do not use $\mathbf{x}_{pk}$ as the final projection. However, both C_k and $\mathbf{x}_{pk}$ are used as the "input-output" pairs for the next phase of ELM-SOM+, because C_k and $\mathbf{x}_{pk}$ together preserve the topology information of the data.

Phase-II: Initial reconstruction of the Data with the second ELM: ELM_{DEC} reconstructs the data from the projection space provided by the SOM onto the original space. The input is the SOM projection, and the target is the corresponding data point $\mathbf{x}_i$. The following transformation is learned by ELM_{DEC}:

$$R(\hat{\mathbf{x}}_{pk}) = \hat{\mathbf{x}}_k, \quad k = [1, \ldots, s^2]. \tag{4.3}$$

This allows the reconstruction error (Mean Square Error) to be calculated:

$$E = \frac{\sum_{i=1}^{N} \|R(\hat{\mathbf{x}}_{pk}) - \mathbf{x}_k\|^2}{N}. \tag{4.4}$$

We want to minimize the error E by tuning the model ELM_{DEC}, selecting the optimal number of neurons n_B^* for ELM_{DEC}. For that purpose, we minimize the Leave-One-Out error for ELM_{DEC}.

By tuning n_B, we find an E', which similar to the one provided by the original SOM. This means that we are selecting a continuous projection that is as good as the SOM projection.

Phase-III: Creating Continuous Projection with ELM_{ENC}: Based on the topology information learned from the SOM, the first ELM: ELM_{ENC} is built to

create the continuous encoding projection. The inputs of the ELM_{ENC} are the data points $\mathbf{X}$. The targets are the corresponding projection $\mathbf{X}_p$ learned from the SOM in phase-I. Thus, ELM_{ENC} is trained to link $\mathbf{x}_i \in \mathbb{R}^d$ to $\hat{\mathbf{x}}_{pi} \in \mathbb{R}^2$:

$$\tilde{P}(\mathbf{x}_i) = \hat{\mathbf{x}}_{pi}, \quad i = [1, \ldots, N]. \tag{4.5}$$

For data points that have the same BMU: $\mathbf{x}_j,\ j \in C_k$, the projection are different, yet similar since although they have the same target values in ELM, their input values from the original space are different.

The optimal number of hidden neurons (n_A) for ELM_{ENC} is also selected based to minimize the LOO error. If there exists too many neurons in ELM_{ENC}, it might lead into an over-fitting problem. In case of the over-fitting, ELM_{ENC} learns a perfect mapping relationship $P(\mathbf{x}_j) = \mathbf{x}_{pk},\ \forall j \in C_k$, and projects every data point perfectly on the target $\mathbf{x}_{pk}$. Thus, the results are exactly the same as SOM for the phase-I, which also leads to a larger reconstruction error. If we have too few neurons in ELM_{ENC}, the model is not sophisticated enough and is not able to approximate the transformation $P(\mathbf{x}_j) = \mathbf{x}_{pk},\ \forall j \in C_k$. We, therefore, end up with a poor projection result, as well as a large reconstruction error, and lose the topology information of the data learned from the phase-I, .

Phase-IV: The linear weights of ELM_{ENC} are now tuned using the fminunc from Matlab. It is based on a Large-Scale Optimization. This algorithm is a subspace trust region method and is based on the interior-reflective Newton method. Each iteration involves the approximate solution of a large linear system using the method

of preconditioned conjugate gradients (PCG). All random weights of the ELM_{ENC} and the ELM_{DEC} are kept constant. The linear weights of the ELM_{DEC} are tuned as in any traditional ELM model.

The ELM-SOM+ algorithm is summarized as follows: (there is a bug, I will correct it later.)

Algorithm 4.1 ELM-SOM+ Algorithm

1: Train a SOM, with a size of $s \times s$ on the dataset

2: Build ELM_{DEC}, with n_B minimizing the LOO error (SOM projection is the input)

3: Build ELM_{ENC}, with n_A minimizing the LOO error (SOM projection is the output)

4: Project the data with ELM_{ENC}

5: Reconstruct the data with ELM_{DEC}

6: Calculate the Reconstruction Error E'

7: The linear weights of ELM_{ENC} are tuned to minimize further E'

4.3 Experiments

Several experiments are performed to examine the performance of the proposed method on diverse datasets. Our approach is used together with PCA, and the performances are compared. Two main criteria, reconstruction error, and visual projection performance are used.

4.3.1 Data

Nine different and diverse datasets are selected to perform experiments to evaluate our methodology for different circumstances. These datasets, which are listed in Table 4.1.

4.3.1.1 Abalone Data

Abalone dataset which has been measured to predict the age of abalone according to various physical measurements [10]. This data consists in 4177 instances with nine different features including gender (Male, Female, and Infant), length, diameter, height, whole weight, shucked weight, viscera weight, shell weight, and rings.

4.3.1.2 Countries Data

This dataset contains a part of a larger dataset of total wealth estimates and per capita wealth estimates dataset for 209 countries in different years, in which regional and income group aggregates are computed for each year [150]. We merely utilize total wealth estimates in 2005 as our countries dataset which consists of 209 instances and 19 different features including for example: Population, Net foreign assets, Produced Capital, Crop, Pasture Land, Oil, Natural Gas, Hard coal, Soft coal, Minerals, and Subsoil Assets.

4.3.1.3 Sculpture Data

This dataset which is widely utilized as a benchmark to recover the neighborhood structure for instance in [142], includes a set of 698 sculpture face images.

These sculpture images are computer renderings of a 3-D sculpture head under different poses and lighting directions [142]. Each image consists in a 4096-dimensional vector built from an array of 64 by 64 brightness values of pixels.

4.3.1.4 Glass Identification Data

The Glass Identification data [36], which is utilized in criminal investigation, consists in 241 instances and nine features. This data is created to classify different types of glasses.

4.3.1.5 MNIST Handwritten Digits Data

The Modified National Institute of Standards and Technology (MNIST) dataset consists of 60,000 images of handwritten digits in which the black and white digits are normalized in size, and centered in a fixed size image. In each image, the center of gravity lies at the center of the image with 28 by 28 pixels. For simplicity, we use 1000 image samples, and the dimensionality of each sample vector is $28 \times 28 = 784$ [84].

4.3.1.6 Wisconsin Breast Cancer Data

The Wisconsin Breast Cancer Data (WBCD) [149] consists in 699 breast mass pattern instances and nine different measurement features computed from a digitized image of a fine needle aspirate of a breast mass. Among these patterns 458 of them are benign samples and 241 are malignant samples.

4.3.1.7 SantaFeA

The main benchmark of the Santa Fe Time Series Competition, time series A, is composed of a clean low-dimensional nonlinear and stationary time series with 1,000 observations [146]. Competitors were asked to correctly predict the next 100 observations (SantaFe.A.cont). The performance evaluation done by the Santa Fe Competition was based on the NMSE errors of prediction found by the competitors.

4.3.1.8 Blood Transfusion

To demonstrate the RFMTC marketing model (a modified version of RFM), this study adopted the donor database of Blood Transfusion Service Center in Hsin-Chu City in Taiwan [161]. The center passes their blood transfusion service bus to one university in Hsin-Chu City to gather blood donated about every three months. To build a FRMTC model, they selected 748 donors at random from the donor database. These 748 donor data, each one included R (Recency - months since last donation), F (Frequency - total number of donation), M (Monetary - total blood donated in c.c.), T (Time - months since first donation), and a binary variable representing whether he/she donated blood in March 2007 (1 stand for donating blood; 0 stands for not donating blood).

4.3.1.9 Wine Quality

The two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine [26]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about

grape types, wine brand, wine selling price, etc.). These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are munch more normal wines than excellent or poor ones). It is not sure if all input variables are relevant.

4.3.2 Performance Criteria

We consider two different criteria, reconstruction error and visual projection performance to evaluate each of utilized methods for these nine different datasets. The reconstruction error (defined in Eq. 4.4) is calculated to validate how far are the reconstructed data points from the original data points on average. In other words, the smaller the value of reconstruction error, the higher the quality of the dimensionality reduction. Reconstruction quality is also validated using visual projection performance.

4.3.3 Procedure

To compare our novel ELM-SOM+ method with another dimensionality reduction methods (PCA), we perform experiments on nine diverse datasets. For each experiment, we compare the reconstruction errors. Because each data has different feature variables regarding concept and scaling units, we perform normalization on each dataset to make each variable have the same influence on pair-wise Euclidean distances among data points. Besides, for each dataset, we remove response variable for regression problems and class label variable for classification problems. This final pre-processing is done to avoid any impact of response variable or class label on the

visualization. Therefore, we can investigate the strength of dimensionality reduction and visualization for regression/classification without having any predefined response variable or class label.

4.3.4 Results

The results of the experiments reveal that our ELM-SOM+ method outperforms PCA according to the reconstruction error. Furthermore, ELM-SOM+ is providing a continuous projection rather than a discrete one in SOM. The results are listed in Table 4.1.

Table 4.1: Comparison of the Reconstruction Errors

Dataset Name	***PCA***	***ELM-SOM***
Abalone	0.088	**0.023**
Countries	0.351	**0.077**
Sculpture	0.564	**0.327**
Glass	0.493	**0.073**
Handwritten Digits	0.883	**0.677**
Breast Cancer	0.368	**0.288**
SantaFeA	0.222	**0.017**
Blood Transfusion	0.089	**0.020**
Wine Quality	0.564	**0.418**

4.3.5 Visualizations

For each dataset, we present two graphs: the visualization using PCA and the visualization using ELM-SOM+.

Figure 4.3 shows ELM-SOM+ for the Abalone dataset. We can see that infant Abalones are nicely visualized between females and males using ELM-SOM+. Figure 4.4 shows ELM-SOM+ for Countries dataset. The visualization using ELM-SOM+ is consistent with geopolitical facts. For example, oil producers are projected together. Figure 4.5 shows ELM-SOM+ for Sculpture dataset. Figure 4.6 shows ELM-SOM+ for Glass dataset. Figure 4.7 shows ELM-SOM+ for MNIST dataset. Figure 4.8 shows ELM-SOM+ for Wisconsin dataset. Figure 4.9 shows ELM-SOM+ for SantaFe A dataset. Figure 4.10 shows ELM-SOM+ for Blood Transfusion dataset. Figure 4.11 shows ELM-SOM+ for Wine Quality dataset.

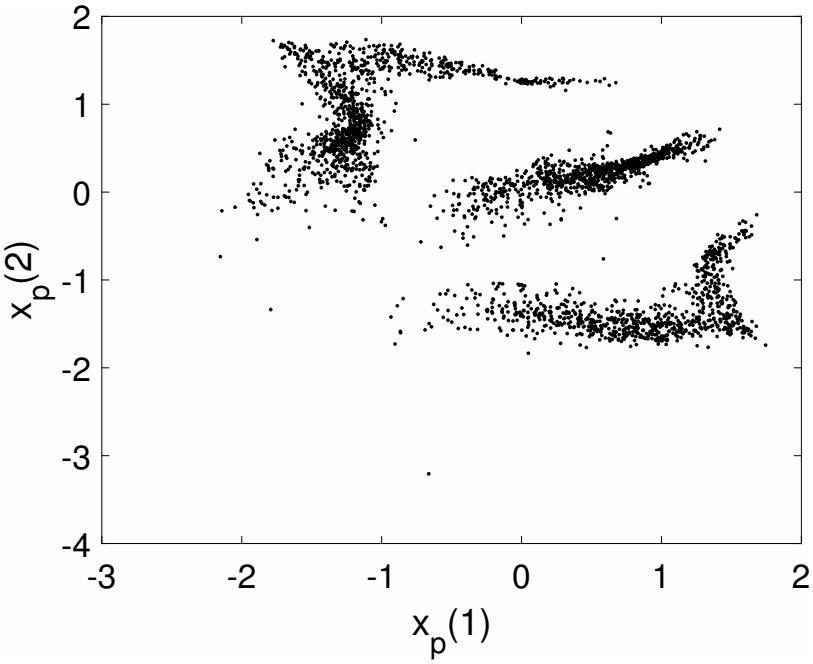


Figure 4.3: Abalone ELM-SOM+ Visualization

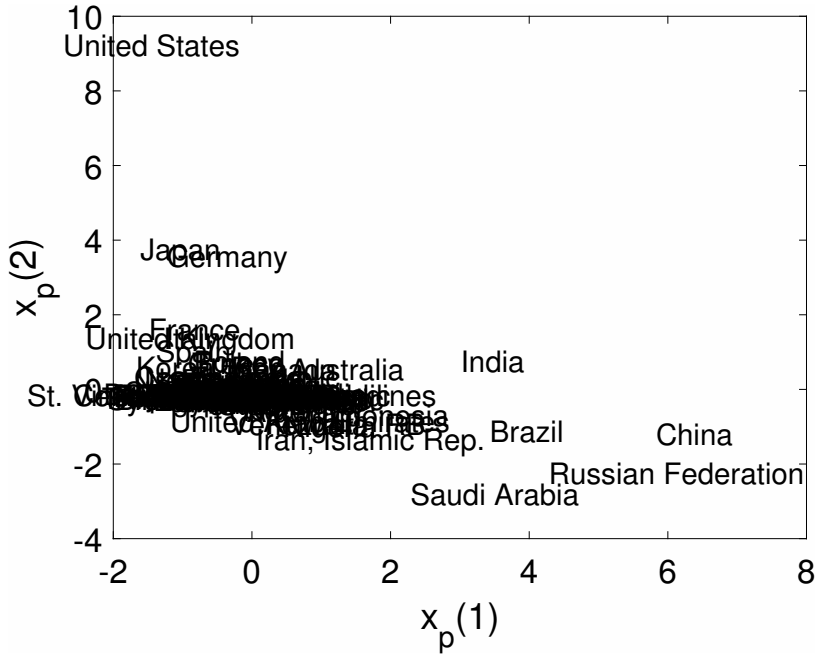


Figure 4.4: Countries ELM-SOM+ Visualization

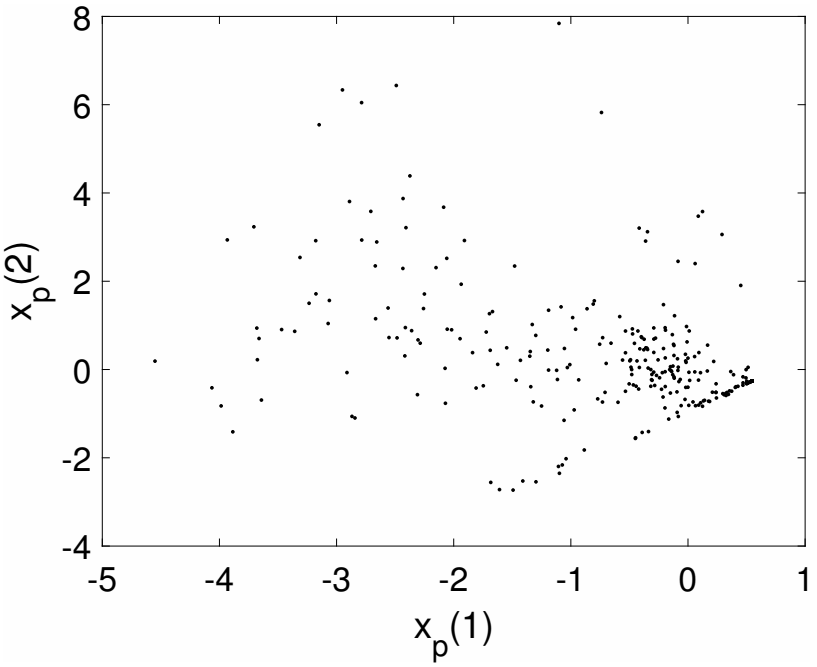


Figure 4.5: Sculpture ELM-SOM+ Visualization

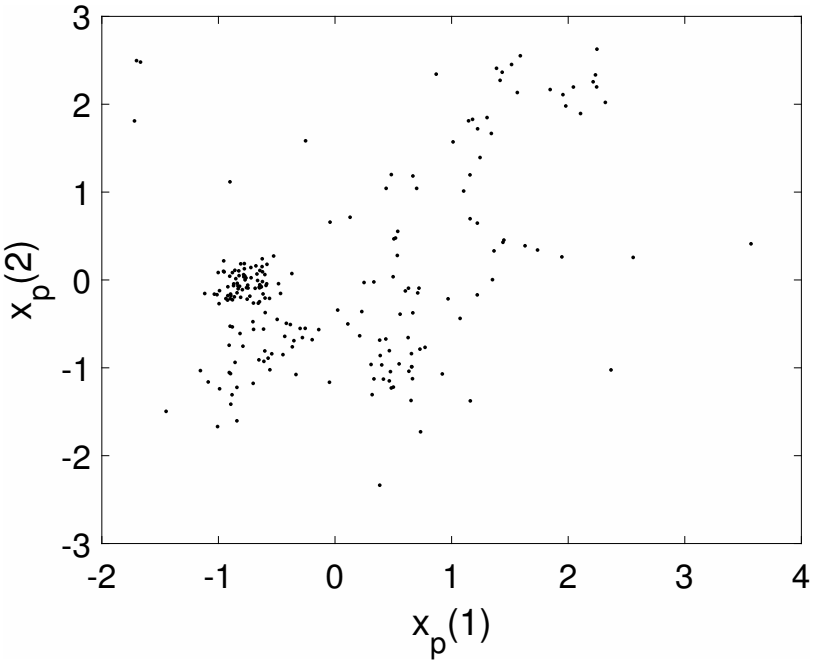


Figure 4.6: Glass ELM-SOM+ Visualization

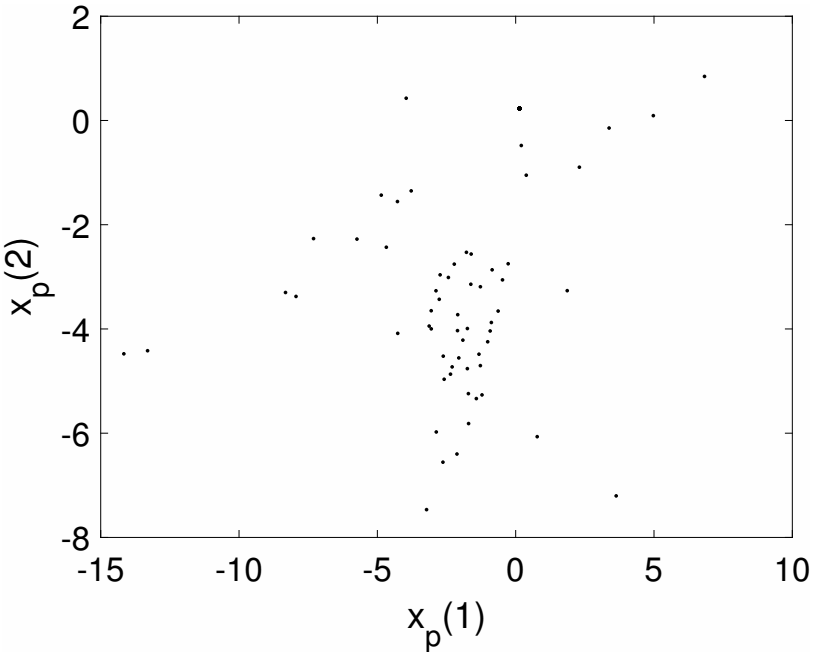


Figure 4.7: MNIST ELM-SOM+ Visualization

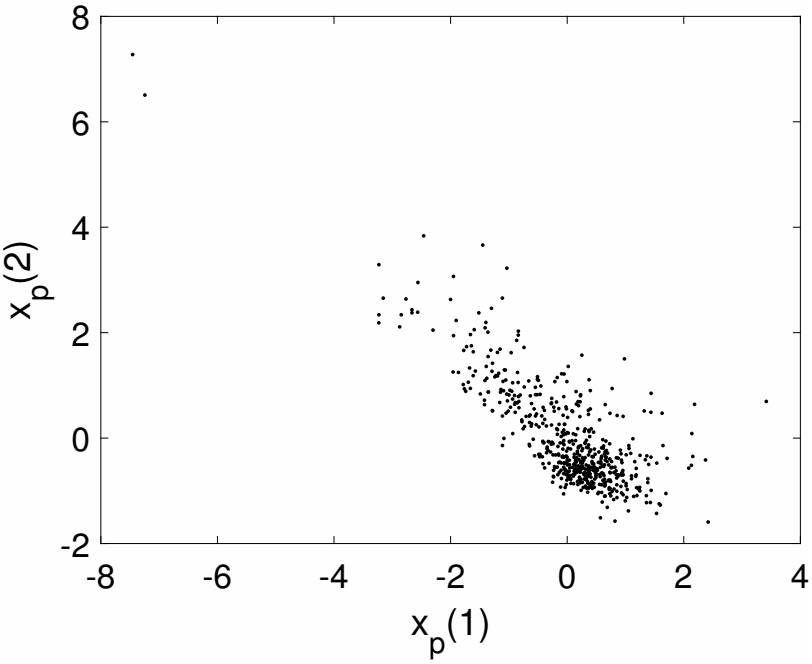


Figure 4.8: Wisconsin ELM-SOM+ Visualization

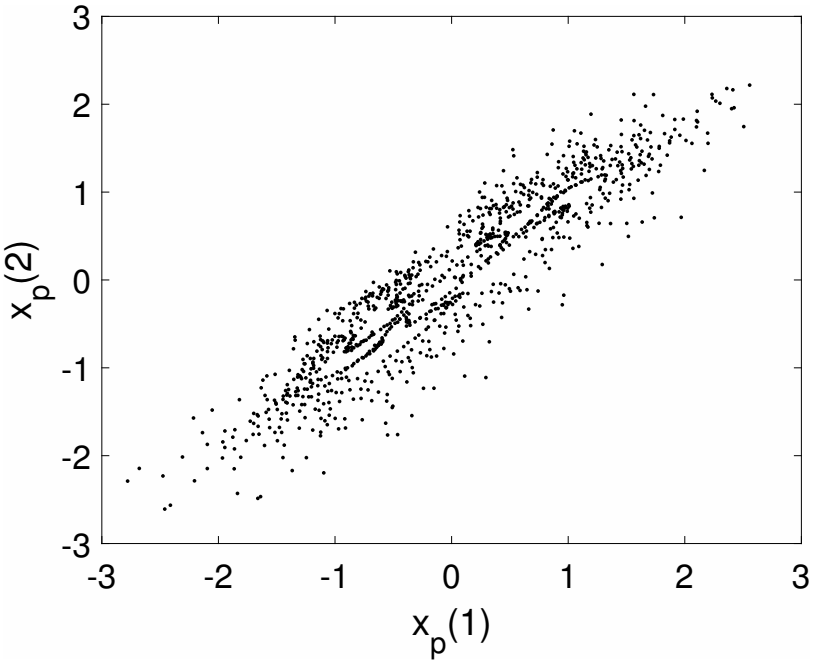


Figure 4.9: SantaFE A ELM-SOM+ Visualization

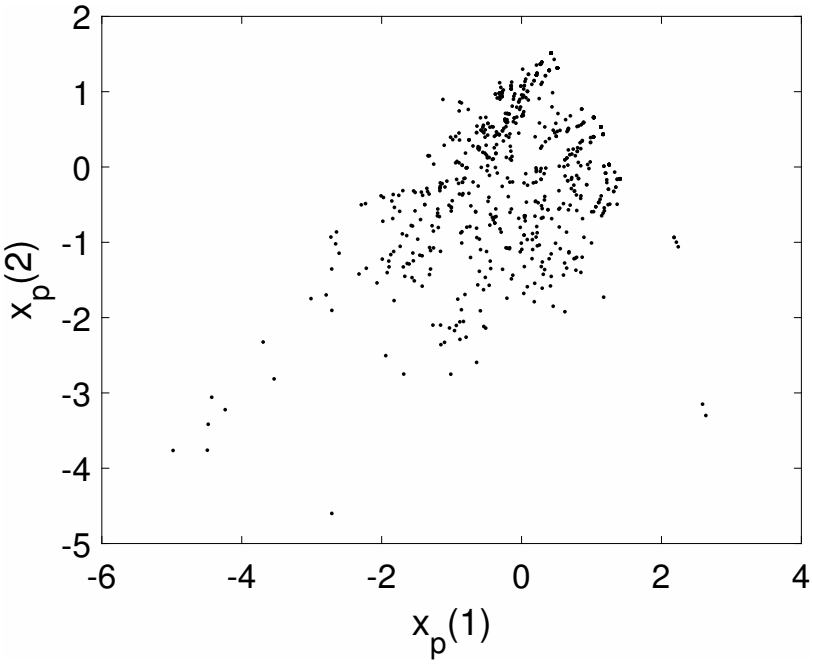


Figure 4.10: Blood Transfusion ELM-SOM+ Visualization

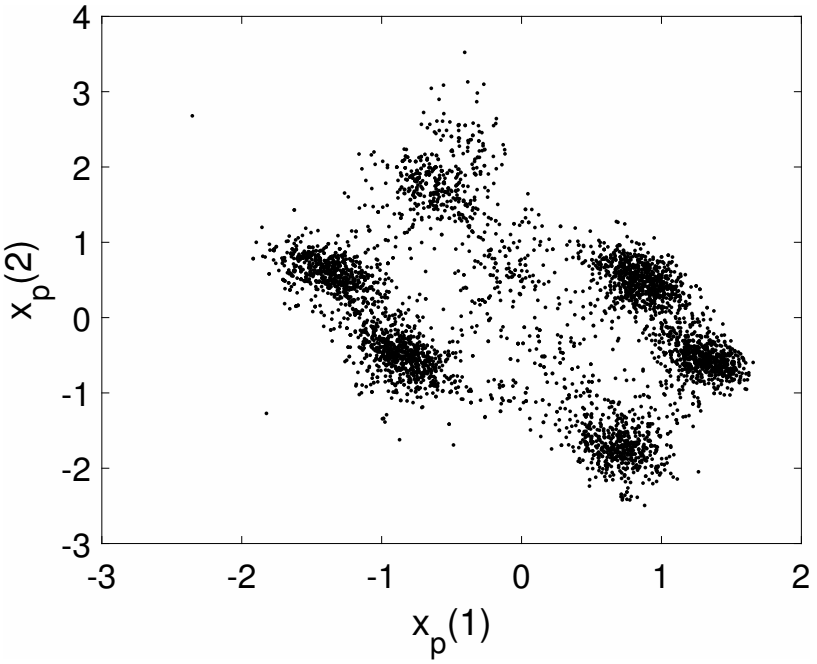


Figure 4.11: Wine ELM-SOM+ Visualization

4.4 Conclusions for Chapter 4

According to the performed experiments on diverse datasets, it is shown that the ELM-SOM+ technique can contribute to an efficient dimensionality reduction. In different data-based concepts, the ELM-SOM+ decreases the reconstruction error considerably compared to what PCA does. Furthermore, it can be concluded that ELM-SOM+ is capable of improving SOM algorithm. It not only has the nonlinearity feature and suitability for big data but also compensates for the discontinuity of SOM algorithm. It creates a continuous projection by using two Extreme Learning Machine models, the first one to perform the dimensionality reduction and the second one to perform the reconstruction. In the future, we will extend and test ELM-SOM+ for

big data applications.

CHAPTER 5
A MODIFIED LANCZOS ALGORITHM FOR FAST REGULARIZATION OF EXTREME LEARNING MACHINES

5.1 Introduction for Chapter 5

In this Chapter we are interested in the discussion of two common problems of Artificial Neural Networks (ANNs) [99]: 1) how to process large amount of data with reasonable computational time? 2) how to select the structure the complexity and the parameters of ANNs?

Although the significant improvement of the required computational power has been made for many complex algorithms, enabling the solution of large problems, such as SVM, and Deep Learning, the volume of data is growing even faster [6]. Therefore, reducing the computational time for machine learning algorithms is evermore desirable. Extreme Learning Machines (ELMs) [4, 42, 61, 82, 104] is a type of Randomized Neural Networks (RNNs) that is known for its fast training speed and good accuracy. Despite its merits, the performance of ELM is sensitive to the number of neurons. Underfitting can happen when there are not enough neurons, which leads to a poor approximation; while too many neurons often leads to overfitting problems, resulting in poor generalization performance. It is not easy to find the "correct" number of neurons that keeps the balance between a better network performance and simple network topology. Regularization is introduced to deal with this particular dilemma. Many algorithms have been applied to regularize the complexity of ELM, such as L_1 regularization like LASSO [125, 138] or L_2 regularization as known as Ridge Regres-

sion or Tikonov regression [105, 115]. Although, these regularization algorithms can significantly reduce the complexity of ELMs, they can't give a direct answer to the "correct" number of neurons, and the performance of ELMs is still largely influenced by the number of neurons it has. Lanczos Algorithm [80,81] originally was introduced to approximate the extreme eigenvalues of symmetric matrices. It is a fast iterative process that is proven to converge quickly [111]. Due to the distinct training process of ELMs, the last step of training ELMs is an Ordinary Least Square problem, which can be solved by the Lanczos Algorithm. This Chapter presents a modified Lanczos Algorithm for ELMs that can speed up the training process, but more importantly does regularization of ELMs and allows ELMs to have a very large number of neurons, while not encountering overfitting problems. In other words, Lanczos ELM can reduce the computational time for the model selection, and just use a large number of neurons to generate the robust outcome without overfitting.

In the Section 5.2 the original Lanczos Algorithm is presented. We then describe how to use the Lanczos Algorithm to solve a symmetric linear system. This is presented in Section 5.3. The proposed Lanczos ELM is explained in detail in Section 5.4, followed by the experiments to show the performance of Lanczos ELM on four diverse datasets in Section 5.5. Finally the conclusion is drawn in Section 5.6.

5.2 The Lanczos Algorithm

This Section presents the original Lanczos Algorithm as in [80, 81, 111, 113]. The basic Lanczos Algorithm for tridiagonalization of a symmetric $N \times N$ matrix $\mathbf{A}$

computes a sequence of Lanczos vectors $\boldsymbol{q_j}$ and scalars α_j, β_j at the jth step, following the well-know iteration rules:

Algorithm 5.1 Lanczos Algorithm

1: Initialization: $\mathbf{r_0}$, $\mathbf{r_0} \neq \mathbf{0}$; $\boldsymbol{q_0} = \mathbf{0}$; $\beta_1 = \|\mathbf{r_0}\|$

2: **for** $j = 1, 2, 3, ...$ **do**

3: $\quad \boldsymbol{q_j} = \boldsymbol{r_{j-1}}/\beta_j$

4: $\quad \boldsymbol{u_j} = \mathbf{A}\boldsymbol{q_j} - \beta_j \boldsymbol{q_{j-1}}$

5: $\quad \alpha_j = \boldsymbol{u_j}^* \boldsymbol{q_j}$

6: $\quad \boldsymbol{r_j} = \boldsymbol{u_j} - \alpha_j \boldsymbol{q_j}$

7: $\quad \beta_{j+1} = \|\boldsymbol{r_j}\|$

8: **end for**

Note: The $*$ sign is the (conjugate) transpose of a matrix.

To summarize, the Iteration process can be written as:

$$\boldsymbol{r_j} = \beta_{j+1}\boldsymbol{q_{j+1}} = \mathbf{A}\boldsymbol{q_j} - \alpha_j \boldsymbol{q_j} - \beta_j \boldsymbol{q_{j-1}}. \tag{5.1}$$

The matrix form of The fist j step is:

$$\mathbf{AQ_j} - \mathbf{Q_j Z_j} = \beta_{j+1}\boldsymbol{q_{j+1}}\boldsymbol{e_j^*}. \tag{5.2}$$

In this equation, $\mathbf{Q_j} \in \mathbb{R}^{N \times j}$ is composed by the orthonormal Lanczos vectors $\boldsymbol{q_j}$,

$\mathbf{Q_j} = (\boldsymbol{q_1}, ..., \boldsymbol{q_j})$ and

$$\mathbf{Q_j}^* \mathbf{Q_j} = \boldsymbol{I}_j. \tag{5.3}$$

$\boldsymbol{e_j}$ is the j's column of the $j \times j$ identity matrix $\boldsymbol{I_j}$. $\mathbf{Z_j}$ is the tridiagonalization of matrix $\mathbf{A}$:

$$\mathbf{Z_j} = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \dots & 0 \\ \beta_2 & \alpha_2 & \beta_3 & & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & & \beta_{j-1} & \alpha_{j-1} & \beta_j \\ 0 & \dots & 0 & \beta_j & \alpha_j \end{bmatrix} \tag{5.4}$$

.

In this Section, the termination criteria for Lanczos Algorithm is skipped, since it is with little relevance to the problems in this Chapter. Usually the iteration will stop with $j << N$. If $j = N$, the eigenvalues and eigenvectors of $\mathbf{Z_j}$ is the also the eigenvalues and eigenvectors of $\mathbf{A}$. The detail can be fond in [112].

5.3 Lanczos Algorithm for Solving Symmetric Linear Systems

Although Lanczos Algorithm was introduced for the eigenvalue and eigenvector problems initially, it can be applied to solve the symmetric linear systems. For a linear system:

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{Y} \tag{5.5}$$

where, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a symmetric coefficient matrix, the true solution is $\boldsymbol{\theta} = \mathbf{A}^{-1}\mathbf{Y}$. Usually, the on-hand initial approximation is $\boldsymbol{\theta_a}$, which leads to an initial residual term $\mathbf{r_0} = \mathbf{Y} - \mathbf{A}\boldsymbol{\theta_a}$. If no such initial approximation is available, then take $\boldsymbol{\theta_a} = \mathbf{0}$. After the rearrangement of the problem, the target of solving the linear system becomes finding the correction term $\boldsymbol{\theta_c}$ that is the solution for the non-singular N-rowed equation:

$$\mathbf{A}\boldsymbol{\theta_c} = \mathbf{r_0} \tag{5.6}$$

Lanczos Algorithm solves the symmetric linear system in 5.6 by finding improving approximations $\boldsymbol{\theta_j}$, whose residual gradually approaches $\mathbf{r_0}$, throughout the iteration process. The iteration is accomplished by computing a sequence of Lanczos vectors $\boldsymbol{q_j}$ and scalars α_j, β_j at the jth step, following the Lanczos iteration rules as in Algorithm 5.1. To reach the True solution, $j = N$ iterations must be done, and $\mathbf{x_N}$ is solved by:

$$\boldsymbol{\theta_N} = \mathbf{Q_N}\mathbf{Z_N}^{-1}\beta_1\boldsymbol{e_1}. \tag{5.7}$$

The proof for equation 5.7 can be fond in [112]. To summarize:

Algorithm 5.2 Lanczos Algorithm for solving symmetric linear systems

1: Initialization: $\mathbf{r_0} = \mathbf{Y}$ (For simplicity only. Other initialization can also be applied); $\boldsymbol{q_0} = \mathbf{0}$; $\beta_1 = \|\mathbf{r_0}\|$

2: **for** $j = 1, 2, 3, ...N$ **do**

3: $\quad \boldsymbol{q_j} = \boldsymbol{r_{j-1}}/\beta_j$

4: $\quad \alpha_j = \boldsymbol{q_j^T}\mathbf{A}\boldsymbol{q_j}$

5: $\quad \boldsymbol{r_j} = \mathbf{A}\boldsymbol{q_j} - \boldsymbol{q_j}\alpha_j - \boldsymbol{q_{j-1}}\beta_j$

6: $\quad \beta_{j+1} = \|\mathbf{r_j}\|$

7: **end for**

8: $\boldsymbol{\theta_N} = \mathbf{Q_N}\mathbf{Z_N}^{-1}\beta_1\boldsymbol{e_1}$

Lanczos Algorithm usually is used for approximating the solution of a symmetric linear system. Thus a more common situation is that the algorithm terminates before N iterations, once the residual norm decreases below the desired threshold. The details can be found in [112]. Since in this Chapter we are focusing on the regularization effect of Lanczos Algorithm and always converges to the true solution with N iterations, the discussion about the earlier termination is not included, but keep in mind that the earlier termination can also be applied to further speed-up the ELMs.

5.4 Iterative Lanczos ELM

From the Section 2.3.3.2 we know that the last step of ELM is solving a Ordinary Least Square Regression problem: $\min_{\boldsymbol{\theta}} \|\mathbf{H}\boldsymbol{\theta} = \boldsymbol{T}\|^2$. The direct solution to

this problem is:

$$\boldsymbol{\theta} = \mathbf{H}^{\dagger}\boldsymbol{T}, \tag{5.8}$$

$$\mathbf{H}^{\dagger} = (\mathbf{H}^{T}\mathbf{H})^{-1}\mathbf{H}^{T}. \tag{5.9}$$

Plug equation 5.8 in 5.9. The solution of $\boldsymbol{\theta}$ becomes:

$$\boldsymbol{\theta} = (\mathbf{H}^{T}\mathbf{H})^{-1}\mathbf{H}^{T}\boldsymbol{T}. \tag{5.10}$$

Note that $\mathbf{H} \in \mathbb{R}^{m \times N}$. N is the number of neurons in the ELM, and m is the number of the data samples. Calculating $\mathbf{H}^{T}\mathbf{H}$ and solving the linear system have a time complexity of $\mathcal{O}(N^2 m)$.

It will be convenient to rearrange the solution in equation 5.10, to see that it is equivalent to solve the following leaner system:

$$(\mathbf{H}^{T}\mathbf{H})\boldsymbol{\theta} = \mathbf{H}^{T}\boldsymbol{T}. \tag{5.11}$$

$\mathbf{H}^{T}\mathbf{H}$ is a symmetric matrix and is positive semi-definite (if $\mathbf{H}$ has independent columns then it is positive definite). Reformulating the problem like this allows Lanczos Algorithm to be applied to solve this linear system, with $\mathbf{A} = \mathbf{H}^{T}\mathbf{H}$ and $\mathbf{Y} = \mathbf{H}^{T}\boldsymbol{T}$. Moreover, $\mathbf{H}^{T}\mathbf{H}$ is supposed to have high collinearity, because each column of $\mathbf{H}$ is a nonlinear mixture of the same columns of $\boldsymbol{X}$. Thus only a few eigenvalues and eigenvectors of $\mathbf{H}^{T}\mathbf{H}$ are enough to approximate it. Therefore, the Lanczos Algorithm is efficient for training an ELM. This will be confirmed in the Section 5.5.

In addition, we have a strong incentive to avoid computing $\mathbf{H}^{T}\mathbf{H}$ directly

because of the high time complexity of doing so. Thus, a matrix $\mathbf{M}$ is crafted based on $\mathbf{H}$ as showing in equation 5.12 below:

$$\mathbf{M} = \mathbf{H}\boldsymbol{q_j}. \tag{5.12}$$

When applying Lanczos Algorithm, instead of computing $\mathbf{H}^T\mathbf{H}$, in each Lanczos iteration j only a relatively small matrix $\mathbf{H}\boldsymbol{q_j}$ needs to be multiplied. The complete process of the Iterative Lanczos ELM is as follows:

Algorithm 5.3 Iterative Lanczos ELM

1: Create ELM: generate the hidden layer weights $\boldsymbol{w}$, generate the hidden layer output $\mathbf{H}$.

2: Initialize the Lanczos Algorithm $\mathbf{r_0} = \boldsymbol{T}$; $\boldsymbol{q_0} = \mathbf{0}$; $\beta_1 = \|\mathbf{r_0}\|$

3: **for** $j = 1, 2, 3, ...N$ **do**

4: $\quad \boldsymbol{q_j} = \boldsymbol{r_{j-1}}/\beta_j$

5: $\quad \alpha_j = \mathbf{M}^{\boldsymbol{T}}\mathbf{M}$

6: $\quad \boldsymbol{r_j} = \mathbf{A}\boldsymbol{q_j} - \boldsymbol{q_j}\alpha_j - \boldsymbol{q_{j-1}}\beta_j$

7: $\quad \beta_{j+1} = \|\mathbf{r_j}\|$

8: **end for**

9: $\boldsymbol{\theta}_\mathbf{N} = \mathbf{Q_N}\mathbf{Z_N}^{-1}\beta_1\boldsymbol{e_1}$

Note that to compute $\boldsymbol{\alpha_j} = \mathbf{M^T M}(= \boldsymbol{q_j^T}\mathbf{H^T}\mathbf{H}\boldsymbol{q_j})$ only has a time complexity

of $\mathcal{O}(Nm)$ for each Lanczos Iteration. Since the number of necessary iterations k is greatly smaller than N, the overall complexity is $\mathcal{O}(Nkm)$ which is smaller than $\mathcal{O}(N^2m)$ (the original complexity for training an ELM).

5.5 Experiments

To examine the performance of Lanczos ELM, several machine learning datasets are used to test our algorithm. As stated in the introduction, the Lanczos Algorithm actually performs as a regularization of the ELM, which helps with the ELM model structure selection — to auto-select the effective number of neurons. It is a regularization since the training error is increased and the validation error is decreased. This is illustrated in the following experiment.

5.5.1 Datasets

Four different and diverse datasets are selected to perform experiments to evaluate our methodology for different circumstances.

5.5.1.1 Abalone

Abalone dataset which has been measured to predict the age of abalone according to various physical measurements [10]. This data consists in 4177 samples with nine different features including gender (Male, Female, and Infant), length, diameter, height, whole weight, shucked weight, viscera weight, shell weight, and rings.

5.5.1.2 The Boston Housing

The data was originally published by Harrison, D. and Rubinfeld, D.L. in [49]. The Boston Housing Dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass. It was obtained from the StatLib archive [48], and has been used extensively throughout the literature to benchmark algorithms. The dataset is small in size with only 506 samples and 14 attributes in each sample of the dataset.

5.5.1.3 Checkerboard

We created this dataset to test the Lanczos ELM for classification problems. In this dataset, there are two classes: "red" and "blue" that alternate in each direction. Each class is surrounded by four blocks of the different class. The data points are generated randomly with a small noise term. Figure 5.1 is the graph for the dataset. The dataset only has 2 variables, which are the coordinates of the points, and one class label.

5.5.1.4 SantaFeA

The main benchmark of the Santa Fe Time Series Competition [129], time series A, is composed of a clean low-dimensional nonlinear and stationary time series with 1,000 observations [146]. Competitors were asked to correctly predict the next 100 observations (SantaFe.A.cont). The performance evaluation done by the Santa Fe Competition was based on the NMSE errors of prediction found by the competitors.

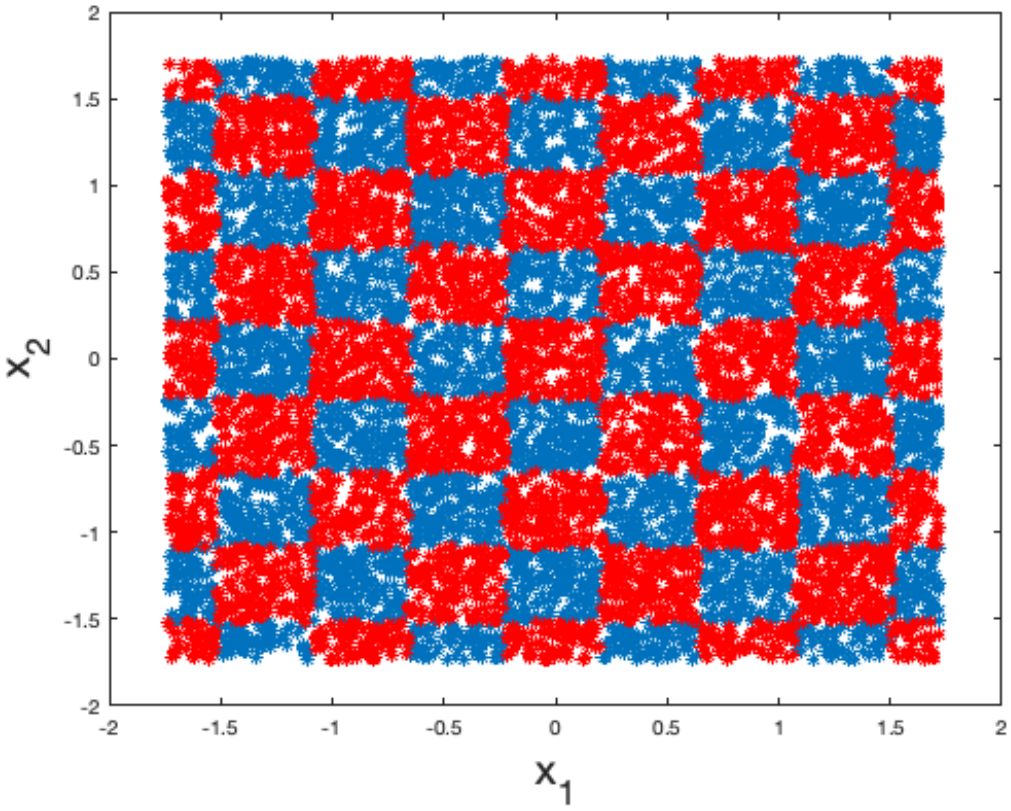


Figure 5.1: Checkerboard

5.5.2 Methodology

In this experiment, Lanczos ELM is compared against a regular ELM with growing number of neurons. The purpose of the experiment is twofold: 1) to test the regularization effect of Lanczos ELM. 2) to compare the performance of Lanczos ELM after j iterations with a j-neurons regular ELM.

In order to examine the regularization effect, a small validation set is created. Both the Lanczos ELM and the regular ELM were trained on the same training set first and tested on the same validation set. Training errors and validation errors are calculated for two ELMs, and compared to examine the overfitting problems of the ELMs.

The Number of neurons: N for both ELMs is pre-determined in the experiment. Typically, N is a very large number comparing with the number of attributes of the dataset, but is less than the number of samples of the dataset. The results of the experiment show how to determine the proper number of neurons for the real applications, which is much less than the N here used in the experiment. Further discussions are in the Section 5.5.3.

For the Lanczos ELM, N is the number of neurons built in ELM, and the training and validation errors are calculated with respect to the iterations of the Lanczos process. For the regular ELM, the number of neurons is growing from 1 to N, and the training and validation errors are calculated with respect to the number of neurons of the ELM.

5.5.3 Results

The experiment results are collected to create the training and validation error graph for each dataset.

5.5.3.1 Abalone

In Figure 5.2, the blue line is the training error of Lanczos ELM; the red line is the validation error of Lanczos ELM; the black line is the training error of the regular ELM; the green line is the validation error of the regular ELM. A few important yet trivial information can be found from the graph: 1) From two training error line, it proves that when Lanczos ELM finishes $N = 600$ iterations it leads to the same result as the regular ELM with $N = 600$ has. 2) From the validation and training errors of Lanczos ELM, it is noticeable that merely 9 iterations of Lanczos ELM can reach the lowest validation error. Hence the Lanczos ELM should terminate at iteration 9. Moreover, since Lanczos ELM is utilizing all N neurons, the validation error is even lower than the best validation error of the regular ELM.

5.5.3.2 The Boston Housing

In Figure 5.3, similar pattern of the four errors can be found: 1) Lanczos ELM gives the complete solve of the ELM when went through all the iterations. 2) Both the training and the validation error are lower than the regular ELM. 3) With only a dozen iterations Lanczos ELM gives the best validation error. Again, this is because Lanczos ELM is the ELM with N neurons and regularization.

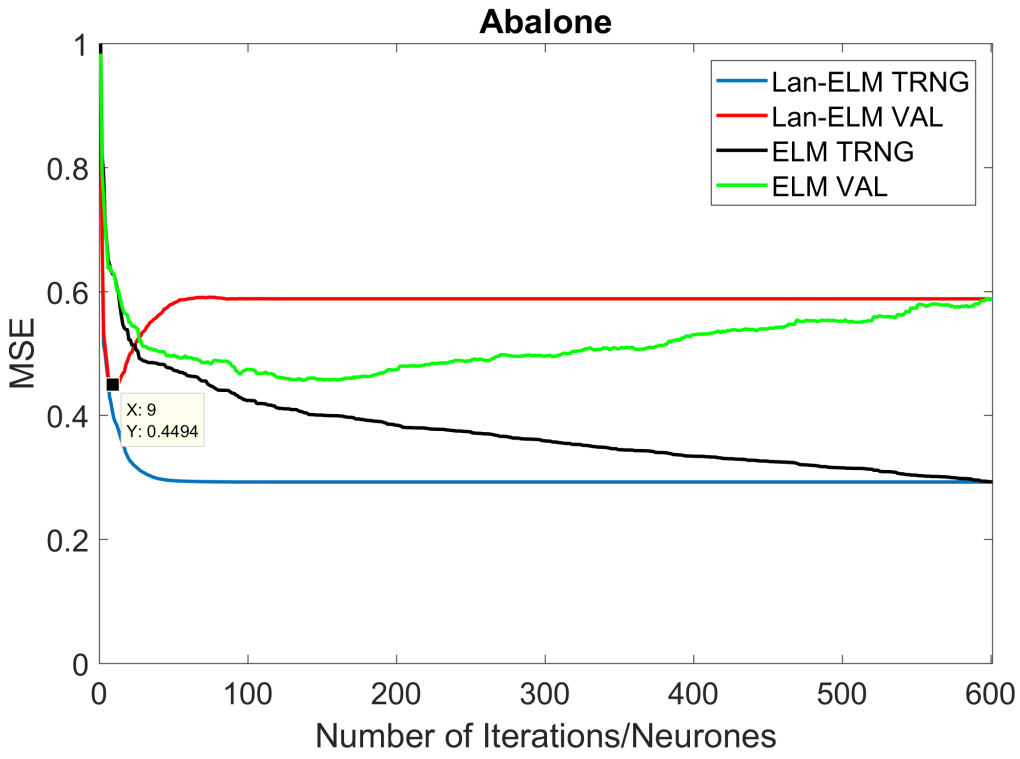


Figure 5.2: Abalone

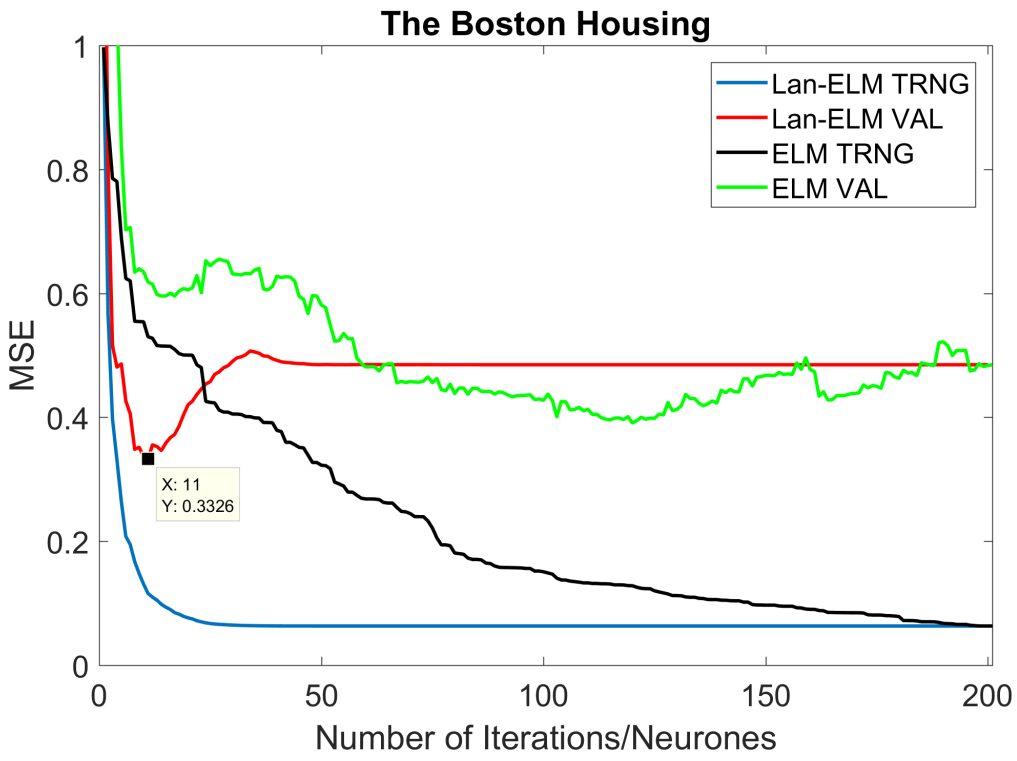


Figure 5.3: The Boston Housing

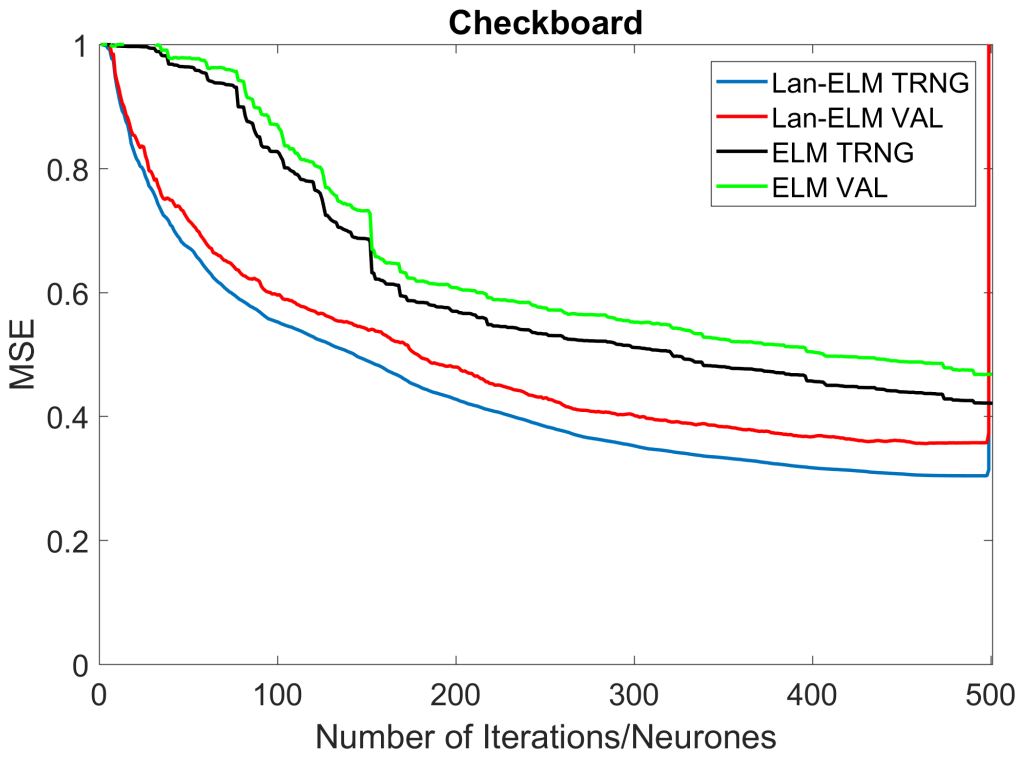


Figure 5.4: Checkerboard

5.5.3.3 "Checkerboard"

The "Checkerboard" (Figure 5.4) problem is one harder problem because of the degeneration of the linear system. Since the number of variables in "Checkerboard" is only 2, the determinant of the matrix $\mathbf{H}^T\mathbf{H}$ goes to zero very quickly as the dimension of $\mathbf{H}$ grows. This explains the behavior on the right end of the graph. Even though this is a harder problem, Lanczos ELM still outperforms the regular ELM.

5.5.3.4 SantaFeA

SantaFeA is a Time Series dataset. Figure 5.5 shows that Lanczos ELM has consistent performance on the time series data as well. Only 7 iterations of Lanczos

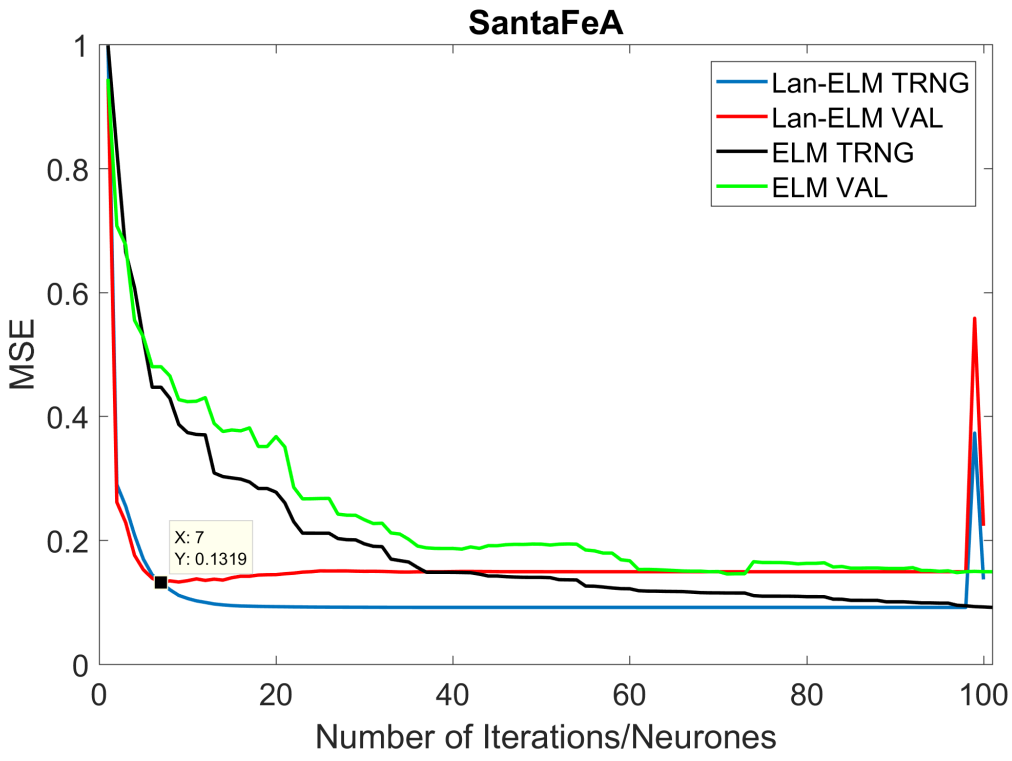


Figure 5.5: SantaFeA

ELM can reach the minimum validation error and outperform the regular ELM. Hence, the Lanczos ELM should terminate at the 7th iteration, which leads to a better validation error with less computational time than the regular ELM.

As summarized in Table 5.1, the computational time is on average divided by 20 and the Normalized MSE is reduced by 14%.

Table 5.1: Comparisons of Errors and Computational Time

	LanELM ValE	ELM ValE	LanELM Time	ELM Time
Abalone	0.45	0.46	0.02	0.55
Housing	0.33	0.39	0.003	0.09
SantaFeA	0.13	0.15	0.002	0.04
Checkboard	0.36	0.47	5.0	21.8

5.6 Conclusion for Chapter 5

The experiment above reveals many important merits of the Lanczos ELM. Although the motivation of applying Lanczos Algorithm to solve ELM is trying to speed up the ELM algorithm, the results clearly showed that other than the speed up, Lanczos ELM also performs as a regularized ELM. In terms of speed up, Lanczos ELM avoids calculating $\mathbf{H}^T\mathbf{H}$ directly, which leads to few times faster of calculation; Plus, Lanczos ELM only needs a few iterations to reach the minimum validation error, which leads to a very early termination of the algorithm, hence less calculation is needed. In terms of the regularization, Lanczos ELM nearly automatically solves the overfitting problem, hence, avoids the selection the optimal number of neurons. In general, a large number of neurons can be applied at the initialization, and Lanczos ELM will determine the optimal number of the iterations by searching the minimum of the validation errors. The algorithm is very robust that even the initial number of

neurons varies a lot, the validation error will still converge to the minimum after a similar number of iterations. This also means that the number of neurons can exceed the number of samples, yet still not overfit the problem. In addition, since Lanczos ELM is mainly matrix multiplications, it is naturally parallelized and can benefit from multi-core clusters for further speed up.

CHAPTER 6
FUTURE WORK

6.1 ELM-NG-LE

Although ELM-SOM+ can conduct dimensionality reduction very well, it is limited only to 2-D projection. If the intrinsic dimension of data is not 2-D, ELM-SOM+ projection can result in large information loss.

There exists better methods to learn the manifold of the data. Growing Neural Gas Algorithm (GNG) applies a neural network structure and is inspired by SOM. This method aims at finding a clustering and the structure of the clustering. Laplacian Eigenmap allows a projection of the GNG structure.

GNG Algorithm combined with LE can further improve ELM-SOM+ Algorithm, because it is not limited to 2-D projection. Instead of initializing with a SOM, the GNG+LE can be used as the initialization. Then the two ELMs continue update the manifold and further minimize the reconstruction errors.

6.2 Using ELM-NG-LE for Missing Data Imputation

Since ELM-NG-LE possess the ability to learn the topology of the data, it has the potential for data imputation as well.

Once the ELM-NG-LE is trained, the encoder, the learned manifold (the Laplacian Eigenmap), and the decoder are together preserving the topology information of the data. The model structure is an autoencoder and is able to reproduce the data. Once the model is trained, it is able to reconstruct the data with fairly

small reconstruction error.

First use a subset of the data, which is complete to train the model first. Then, for the incomplete part of the data, impute a large amount of values that span the entire possible values (estimated by the observed data). Since the trained model preserves the data topology, any imputed value that is close to the data manifold will result in small reconstruction error; on the other hand any imputed value that is not on this manifold will result in large reconstruction error.

This allows us to draw an reconstruction error curve for this imputation. The imputation with the lowest error will be applied as the final imputation.

Sometimes, multiple possible imputations may even been found, due to the manifold of the data is nonlinear.

This method also provides us with a useful side product: the "probability" of the imputation. If the reconstruction errors for the imputation is normalized, then the likelyhood for each imputation can be computed.

6.3 ELM-NG-LE for Video Compression

ELM-NG-LE is a dimensionality reduction tool. After captured the topology of the data, the manifold it learned is in lower dimensional space. It preserves the data information with minimum information loss.

If the dimension of the manifold is low enough, and ELM-NG-LE can reconstruct the data from the manifold with a small reconstruction error, therefore, this algorithm compress the data. Hence, ELM-NG-LE has the potential for video com-

pression.

Further experiments should be done on this part and test it on video data.

REFERENCES

[1] Farid F. Abraham, Jeremy Q. Broughton, Noam Bernstein, and Efthimios Kaxiras. Spanning the length scales in dynamic simulation. *Computers in Physics*, 12(6):538–546, 1998.

[2] A.S. Ademiloye, L.W. Zhang, and K.M. Liew. AtomisticâĂŞcontinuum model for probing the biomechanical properties of human erythrocyte membrane under extreme conditions. *Computer Methods in Applied Mechanics and Engineering*, 325:22–36, oct 2017.

[3] Anton Akusok, Stephen Baek, Yoan Miche, Kaj-Mikael Bjork, Rui Nian, Paula Lauren, and Amaury Lendasse. ELMVIS+: Fast nonlinear visualization technique based on cosine distance and extreme learning machines. *Neurocomputing*, 205:247 – 263, 2016.

[4] Anton Akusok, Kaj-Mikael Bjork, Yoan Miche, and Amaury Lendasse. High-performance extreme learning machines: A complete toolbox for big data applications. *IEEE Access*, 3:1011–1025, 2015.

[5] Anton Akusok, David Veganzones, Yoan Miche, Kaj-Mikael Bjork, Philippe Du Jardin, Eric Severin, and Amaury Lendasse. MD-ELM: Originally Mislabeled Samples Detection using OP-ELM Model. *Neurocomputing*, 2015.

[6] Omar Y. Al-Jarrah, Paul D. Yoo, Sami Muhaidat, George K. Karagiannidis, and Kamal Taha. Efficient machine learning for big data: A review. *Big Data Research*, 2(3):87 – 93, 2015.

[7] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2014.

[8] Marino Arroyo and Ted Belytschko. A finite deformation membrane based on inter-atomic potentials for the transverse mechanics of nanotubes. *Mechanics of Materials*, 35(3-6):193–215, mar 2003.

[9] Nongnuch Artrith and Alexander Urban. An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO 2. *Computational Materials Science*, 114:135–150, mar 2016.

[10] Arthur Asuncion and David Newman. Abalone. `https://archive.ics.uci.edu/ml/datasets/abalone`.

[11] S. Badia, P. Bochev, R. Lehoucq, M. L. Parks, Jacob Fish, Mohan A. Nuggehally, and M. Gunzburger. A Force-Based Blending Model forAtomistic-to-Continuum Coupling. *International Journal for Multiscale Computational Engineering*, 5(5):387–406, 2007.

[12] Michele C. Balas, Linda D. Scott, and Ann E. Rogers. Frequency and type of errors and near errors reported by critical care nurses. *Canadian Journal of Nursing Research*, 38(2):24–41, 6 2006.

[13] Eve Bélisle, Zi Huang, Sébastien Le Digabel, and Aïmen E. Gheribi. Evaluation of machine learning interpolation techniques for prediction of physical properties. *Computational Materials Science*, 98:170–177, feb 2015.

[14] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 585–591, Cambridge, MA, USA, 2001. MIT Press.

[15] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[16] J Benn, M Koutantji, L Wallace, P Spurgeon, M Rejman, A Healey, and C Vincent. Feedback from incident reporting: information and action to improve patient safety. *BMJ Quality & Safety*, 18(1):11–21, 2009.

[17] Christopher M. Bishop, Markus Svensen, and Christopher K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[18] Michael J. Bogdanor, Caglar Oskay, and Stephen B. Clay. Multiscale modeling of failure in composites under model parameter uncertainty. *Computational Mechanics*, 56(3):389–404, sep 2015.

[19] A. Brady, A. Malone, and S. Fleming. A literature review of the individual and systems factors that contribute to medication errors in nursing practice. *J Nurs Manag*, 17:679–697, 2009.

[20] E. Cambria et al. Extreme learning machines [trends controversies]. 28(6):30–59, Nov 2013.

[21] Chi Chen, Zhi Deng, Richard Tran, Hanmei Tang, Iek-Heng Chu, and Shyue Ping Ong. Accurate force field for molybdenum by machine learning large materials data. *Physical Review Materials*, 1(4):043603, sep 2017.

[22] S. Chen, C.F.N. Cowan, and P.M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2(2):302–309, mar 1991.

[23] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, mar 2002.

[24] Hedy Cohen, Eileen S. Robinson, and Michelle. Mandrack. Getting to the root of medication errors: survey results. *Nursing*, 33(9):36–45, 2003.

[25] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[26] YP. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine quality. `https://archive.ics.uci.edu/ml/datasets/wine+quality`.

[27] Christine L. Covell and Judith A. Ritchie. Nurses' Responses to Medication Errors: Suggestions for the Development of Organizational Strategies to Improve Reporting. *Journal of Nursing Care Quality*, 24(4), 2009.

[28] W A Curtin and Ronald E Miller. Atomistic/continuum coupling in computational materials science. *Modelling and Simulation in Materials Science and Engineering*, 11(3):R33–R68, may 2003.

[29] Simon Dablemont, Geoffroy Simon, Amaury Lendasse, Alain Ruttiens, François Blayo, and Michel Verleysen. Time series forecasting with SOM and local nonlinear models-application to the DAX30 index prediction. In *Proceedings of the workshop on self-organizing maps, Kitakyushu, Japan*. Citeseer, 2003.

[30] Pierre Demartines and Jeanny Herault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1):148–154, Jan 1997.

[31] Béatrice Duval, Jin-Kao Hao, and Jose Crispin Hernandez Hernandez. A memetic algorithm for gene selection and molecular classification of cancer. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 201–208. ACM, 2009.

[32] J.L. Ericksen. The Cauchy and Born hypotheses for crystals. In *Phase Transformations and Material Instabilities in Solids*, pages 61–77. Elsevier, 1984.

[33] Amany Ahmed Farag. *Multigenerational Nursing Workforce Value Differences and Work Environment: Impact on RNs' Turnover Intentions*. PhD thesis, Case Western Reserve University, 2008.

[34] Jacob Fish. Bridging the scales in nano engineering and science. *Journal of Nanoparticle Research*, 8(5):577–594, nov 2006.

[35] Felix Fritzen and Oliver Kunc. Two-stage data-driven homogenization for nonlinear solids using a reduced order model. *European Journal of Mechanics - A/Solids*, 69:201–220, may 2018.

[36] B. German and Vina Spiehler. Glass identification. `https://archive.ics.uci.edu/ml/datasets/glass+identification`.

[37] Mir Ali Ghaffari, Yan Zhang, and Shaoping Xiao. Multiscale modeling and simulation of rolling contact fatigue. *International Journal of Fatigue*, 108:9–17, mar 2018.

[38] Stanton A Glantz, Bryan K Slinker, and Torsten B Neilands. *Primer of applied regression and analysis of variance*, volume 309. McGraw-Hill New York, 1990.

[39] Aldo Glielmo, Peter Sollich, and Alessandro De Vita. Accurate interatomic force fields via machine learning with covariant kernels. *Physical Review B*, 95(21):214302, jun 2017.

[40] Krzysztof Grabowski, Paulina Zbyrad, Tadeusz Uhl, Wieslaw J. Staszewski, and Pawel Packo. Multiscale electro-mechanical modeling of carbon nanotube composites. *Computational Materials Science*, 135:169–180, jul 2017.

[41] Robert Gracie and Ted Belytschko. An adaptive concurrent multiscale method for the dynamic simulation of dislocations. *International Journal for Numerical Methods in Engineering*, 86(4-5):575–597, apr 2011.

[42] Andrey Gritsenko, Zhiyu Sun, Stephen Baek, Yoan Miche, Renjie Hu, and Amaury Lendasse. Deformable surface registration with extreme learning machines. In *International Conference on Extreme Learning Machine*, pages 304–316. Springer, 2017.

[43] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 2, pages 985–990. IEEE, 2004.

[44] Akash Gupta, Ahmet Cecen, Sharad Goyal, Amarendra K. Singh, and Surya R. Kalidindi. StructureâĂŞproperty linkages using a data science approach: Application to a non-metallic inclusion/steel composite system. *Acta Materialia*, 91:239–254, jun 2015.

[45] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[46] Jackie H Jones and Linda Treiber. When the 5 rights go wrong. 25:240–7, 02 2010.

[47] Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O. Anatole von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters*, 6(12):2326–2331, jun 2015.

[48] D. Harrison and D.L. Rubinfeld. The boston housing dataset. `http://lib.stat.cmu.edu/datasets/boston`.

[49] David Harrison and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81 – 102, 1978.

[50] Simon Haykin and Neural Network. *Neural Networks: A comprehensive foundation*. 2004.

[51] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, jan 1989.

[52] Renjie Hu, Venous Roshdibenam, Hans J Johnson, Emil Eirola, Anton Akusok, Yoan Miche, Kaj-Mikael Björk, and Amaury Lendasse. Elm-som: A continuous self-organizing map for visualization. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[53] Gao Huang, Guang-Bin Huang, Shiji Song, and Keyou You. Trends in extreme learning machines: A review. *Neural Networks*, 61:32–48, jan 2015.

[54] Guang-Bin Huang. An insight into extreme learning machines: random neurons, random features and kernels. *Cognitive Computation*, 6(3):376–390, 2014.

[55] Guang-Bin Huang. An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels. *Cognitive Computation*, 6(3):376–390, sep 2014.

[56] Guang-Bin Huang. What are Extreme Learning Machines? Filling the Gap Between Frank Rosenblatt's Dream and John von Neumann's Puzzle. *Cognitive Computation*, 7:263–278, 2015.

[57] Guang-Bin Huang. What are extreme learning machines? filling the gap between frank rosenblattâĂŹs dream and john von neumannâĂŹs puzzle. *Cognitive Computation*, 7:263–278, 2015.

[58] Guang-Bin Huang, Lei Chen, and Chee-Kheong Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17(4):879–892, 2006.

[59] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(2):513–529, April 2012.

[60] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 2, pages 985–990 vol.2, July 2004.

[61] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1):489 – 501, 2006. Neural Networks.

[62] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1âĂŞ3):489 – 501, 2006. Neural Networks Selected Papers from the 7th Brazilian Symposium on Neural Networks (SBRN '04) 7th Brazilian Symposium on Neural Networks.

[63] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, dec 2006.

[64] Rubén Ibañez, Emmanuelle Abisset-Chavanne, Jose Vicente Aguado, David Gonzalez, Elias Cueto, and Francisco Chinesta. A Manifold Learning Approach to Data-Driven Computational Elasticity and Inelasticity. *Archives of Computational Methods in Engineering*, 25(1):47–57, jan 2018.

[65] Ruben Ibañez, Domenico Borzacchiello, Jose Vicente Aguado, Emmanuelle Abisset-Chavanne, Elias Cueto, Pierre Ladeveze, and Francisco Chinesta. Data-driven non-linear elasticity: constitutive manifold construction and problem discretization. *Computational Mechanics*, 60(5):813–826, nov 2017.

[66] B. Igelnik and Yoh-Han Pao. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Transactions on Neural Networks*, 6(6):1320–1329, 1995.

[67] Institute of Medicine. *To Err Is Human: Building a Safer Health System*. The National Academies Press, Washington, DC, 2000.

[68] B J Wakefield, Douglas Wakefield, T Uden-Holman, and Mary Blegen. Nurses' perceptions of why medication errors occur. 7:39–44, 02 1998.

[69] Alex Jahya, Mark Herink, and Sarthak Misra. A framework for predicting three-dimensional prostate deformation in real time. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 9(4):e52–e60, dec 2013.

[70] Anubhav Jain, Geoffroy Hautier, Shyue Ping Ong, and Kristin Persson. New opportunities for materials informatics: Resources and data mining techniques for uncovering hidden relationships. *Journal of Materials Research*, 31(08):977–994, apr 2016.

[71] Choo Janet, Hutchinson Alison, and Bucknall Tracey. Nurses' role in medication safety. *Journal of Nursing Management*, 18(7):853–861, 2010.

[72] Shan Jiang, Jun Tao, Thomas D Sewell, and Zhen Chen. Hierarchical multiscale simulations of crystalline β-octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine (β-HMX): Generalized interpolation material point method simulations of brittle fracture using an elastodamage model derived from molecular dynamics. *International Journal of Damage Mechanics*, 26(2):293–313, mar 2017.

[73] Cook John and Wall Toby. New work attitude measures of trust, organizational commitment and personal need non-fulfilment. *Journal of Occupational Psychology*, 53(1):39–52, 1980.

[74] Surya R. Kalidindi, Stephen R. Niezgoda, and Ayman A. Salem. Microstructure informatics using higher-order statistics and efficient data-mining protocols. *JOM*, 63(4):34–41, apr 2011.

[75] S. Kaski and J. Peltonen. Dimensionality reduction for data visualization. *IEEE Signal Processing Magazine*, 28(2):100–104, March 2011.

[76] Cynthia L. Kelchner, S. J. Plimpton, and J. C. Hamilton. Dislocation nucleation and defect structure during surface indentation. *Physical Review B*, 58(17):11085–11088, nov 1998.

[77] Natalia V. Kireeva, Svetlana I. Ovchinnikova, Igor V. Tetko, Abdullah M. Asiri, Konstantin V. BalakinâĂŁ, and Aslan Yu. Tsivadze. Nonlinear dimensionality reduction for visualizing toxicity data: Distance-based versus topology-based approaches. *ChemMedChem*, 9(5):1047–1059, 2014.

[78] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, Jan 1982.

[79] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, Mar 1964.

[80] Cornelius Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office Los Angeles, CA, 1950.

[81] Cornelius Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bur. Standards*, 49(1):33–53, 1952.

[82] Paula Lauren, Guangzhi Qu, Feng Zhang, and Amaury Lendasse. Discriminant document embeddings with an extreme learning machine for classifying clinical narratives. *Neurocomputing*, 277:129 – 138, 2018. Hierarchical Extreme Learning Machines.

[83] B. A. Le, J. Yvonnet, and Q.-C. He. Computational homogenization of nonlinear elastic materials using neural networks. *International Journal for Numerical Methods in Engineering*, 104(12):1061–1084, dec 2015.

[84] Yann LeCun, Courant Institute, Corinna Cortes, and Christopher J.C. Burges. MNIST. `http://yann.lecun.com/exdb/mnist/`.

[85] John Lee, Amaury Lendasse, and Michel Verleysen. Curvilinear distance analysis versus isomap. In *European Symposium on Artificial Neural NetworksBruges (Belgium)*, pages 185–192, 04 2002.

[86] John A. Lee and Michel Verleysen. *Nonlinear Dimensionality Reduction*. Springer-Verlag New York, first edition, 2007.

[87] John Aldo Lee, Amaury Lendasse, and Michel Verleysen. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, 57:49 – 76, 2004. New Aspects in Neurocomputing: 10th European Symposium on Artificial Neural Networks 2002.

[88] Linda Lefrak. Moving Toward Safer Practice: Reducing Medication Errors in Neonatal Care. *The Journal of Perinatal & Neonatal Nursing*, 16(2), 2002.

[89] A. Lendasse, M. Cottrell, V. Wertz, and M. Verleysen. Prediction of electric load using kohonen maps - application to the polish electricity consumption. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, volume 5, pages 3684–3689 vol.5, May 2002.

[90] Amaury Lendasse, Vincent Wertz, and Michel Verleysen. Model Selection with Cross-Validations and Bootstraps âĂŤ Application to Time Series Prediction with RBFN Models. In Xu L. Kaynak O., Alpaydin E., Oja E., editor, *Artificial Neural Networks and Neural Information Processing âĂŤ ICANN/ICONIP 2003. Lecture Notes in Computer Science, vol 2714.*, pages 573–580. Springer, Berlin, Heidelberg, 2003.

[91] Wing Kam Liu, Dong Qian, Stefano Gonella, Shaofan Li, Wei Chen, and Shardool Chirputkar. Multiscale methods for mechanical science of complex materials: Bridging from quantum to stochastic multiresolution continuum. *International Journal for Numerical Methods in Engineering*, 83(8-9):1039–1080, aug 2010.

[92] W.K. Liu, E.G. Karpov, S. Zhang, and H.S. Park. An introduction to computational nanomechanics and materials. *Computer Methods in Applied Mechanics and Engineering*, 193(17-20):1529–1578, may 2004.

[93] Zeliang Liu, M.A. Bessa, and Wing Kam Liu. Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, 306:319–341, jul 2016.

[94] Zeliang Liu, Mark Fleming, and Wing Kam Liu. Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials. *Computer Methods in Applied Mechanics and Engineering*, 330:547–577, mar 2018.

[95] D. Lorente, F. Martínez-Martínez, M.J. Rupérez, M.A. Lago, M. Martínez-Sober, P. Escandell-Montero, J.M. Martínez-Martínez, S. Martínez-Sanchis, A.J. Serrano-López, C. Monserrat, and J.D. Martín-Guerrero. A framework for modelling the biomechanical behaviour of the human liver during breathing in real time using machine learning. *Expert Systems with Applications*, 71:342–357, apr 2017.

[96] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. 'neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, July 1993.

[97] Karel Matouš, Marc G.D. Geers, Varvara G. Kouznetsova, and Andrew Gillman. A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. *Journal of Computational Physics*, 330:192–220, feb 2017.

[98] Ann Mayo and Denise Duncan. Nurse perceptions of medication errors: What we need to know for patient safety. 19:209–17, 09 2004.

[99] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec 1943.

[100] David L. McDowell, Jitesh Panchal, Hae-jin Choi, Carolyn Seepersad, Janet Allen, and Farrokh Mistree. *Integrated design of multiscale, multifunctional materials and products*. Butterworth-Heinemann, 1st edition, 2010.

[101] Ronen Meiri and Jacob Zahavi. Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research*, 171(3):842–858, 2006.

[102] Qinghua Meng, Bo Li, Teng Li, and Xi-Qiao Feng. A multiscale crack-bridging model of cellulose nanopaper. *Journal of the Mechanics and Physics of Solids*, 103:22–39, 2017.

[103] Paul Merlin, Antti Sorjamaa, Bertrand Maillet, and Amaury Lendasse. X-som and l-som: A double classification approach for missing value imputation. *Neurocomputing*, 73(7):1103 – 1108, 2010. Advances in Computational Intelligence and Learning.

[104] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse. Op-elm: Optimally pruned extreme learning machine. *IEEE Transactions on Neural Networks*, 21(1):158–162, Jan 2010.

[105] Yoan Miche, Mark van Heeswijk, Patrick Bas, Olli Simula, and Amaury Lendasse. Trop-elm: A double-regularized elm using lars and tikhonov regularization. *Neurocomputing*, 74(16):2413 – 2421, 2011. Advances in Extreme Learning Machine: Theory and Applications Biological Inspired Systems. Computational and Ambient Intelligence.

[106] Yoan Miche, Mark van Heeswijk, Patrick Bas, Olli Simula, and Amaury Lendasse. TROP-ELM: A double-regularized elm using lars and tikhonov regularization. *Neurocomputing*, 74(16):2413 – 2421, 2011.

[107] Steven L. Mielke et al. The role of vacancy defects and holes in the fracture of carbon nanotubes. *Chemical Physics Letters*, 390(4-6):413–420, jun 2004.

[108] Ronald E Miller and E B Tadmor. A unified framework and performance benchmark of fourteen multiscale atomistic/continuum coupling methods. *Modelling and Simulation in Materials Science and Engineering*, 17(5):053001, jul 2009.

[109] Y. Mishin, D. Farkas, M. J. Mehl, and D. A. Papaconstantopoulos. Interatomic potentials for monoatomic metals from experimental data and ab initio calculations. *Physical Review B*, 59(5):3393–3407, feb 1999.

[110] V F Nieva and J Sorra. Safety culture assessment: a tool for improving patient safety in healthcare organizations. *BMJ Quality & Safety*, 12(suppl 2):ii17–ii23, 2003.

[111] Christopher C Paige and Michael A Saunders. Solution of sparse indefinite systems of linear equations. *SIAM journal on numerical analysis*, 12(4):617–629, 1975.

[112] Beresford N Parlett. A new look at the lanczos algorithm for solving symmetric systems of linear equations. *Linear algebra and its applications*, 29:323–346, 1980.

[113] Beresford N Parlett and David S Scott. The lanczos algorithm with selective orthogonalization. *Mathematics of computation*, 33(145):217–238, 1979.

[114] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[115] David L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *J. ACM*, 9(1):84–97, January 1962.

[116] Tu Minh Phuong, Zhen Lin, and Russ B Altman. Choosing snps using feature selection. In *Computational Systems Bioinformatics Conference, 2005. Proceedings. 2005 IEEE*, pages 301–309. IEEE, 2005.

[117] Steve Plimpton. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *Journal of Computational Physics*, 117(1):1–19, mar 1995.

[118] A.K. Qin and P.N. Suganthan. Robust growing neural gas algorithm with application in cluster analysis. *Neural Networks*, 17(8):1135 – 1148, 2004.

[119] Mohammad Mamunur Rahman, Yusheng Feng, Thomas E. Yankeelov, and J. Tinsley Oden. A fully coupled spaceâĂŞtime multiscale modeling framework for predicting tumor growth. *Computer Methods in Applied Mechanics and Engineering*, 320:261–286, jun 2017.

[120] James Reason. Human error: models and management. *British Medical Journal*, 320(7237):768–770, 2000.

[121] Yoram Reich and S.V. Barai. Evaluating machine learning models for engineering problems. *Artificial Intelligence in Engineering*, 13(3):257–272, jul 1999.

[122] Frank Rosenblatt and Contract Nonr. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, November 1958.

[123] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, May 1969.

[124] John P. Santell, Rodney W. Hicks, Judy McMeekin, and Diane D. Cousins. Medication errors: Experience of the united states pharmacopeia (usp) medmarx reporting system. *The Journal of Clinical Pharmacology*, 43(7):760–767, 2003.

[125] F. Santosa and W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.

[126] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, aug 2004.

[127] Jeong-Hoon Song, Pedro M. A. Areias, and Ted Belytschko. A method for dynamic crack and shear band propagation with phantom nodes. *International Journal for Numerical Methods in Engineering*, 67(6):868–893, aug 2006.

[128] Kim Keum Soon, Kwon So-Hi, Kim Jin-A, and Cho Sunhee. NursesâĂŹ perceptions of medication errors and their contributing factors in south korea. *Journal of Nursing Management*, 19(3):346–353, 2011.

[129] Antti Sorjamaa, Jin Hao, Nima Reyhani, Yongnan Ji, and Amaury Lendasse. Methodology for long-term prediction of time series. *Neurocomputing*, 70(16):2861 – 2869, 2007. Neural Network Applications in Electrical Engineering Selected papers from the 3rd International Work-Conference on Artificial Neural Networks (IWANN 2005).

[130] DuÅąan Sovilj, Emil Eirola, Yoan Miche, Kaj-Mikael BjÃűrk, Rui Nian, Anton Akusok, and Amaury Lendasse. Extreme learning machine for missing data using multiple imputations. *Neurocomputing*, 174:220 – 231, 2016.

[131] Nithya Subramanian, Ashwin Rai, and Aditi Chattopadhyay. Atomistically informed stochastic multiscale model to predict the behavior of carbon nanotube-enhanced nanocomposites. *Carbon*, 94:661–672, nov 2015.

[132] E. B. Tadmor, M. Ortiz, and R. Phillips. Quasicontinuum analysis of defects in solids. *Philosophical Magazine A*, 73(6):1529–1563, jun 1996.

[133] E.B Tadmor, R Phillips, and M Ortiz. Hierarchical modeling in the mechanics of materials. *International Journal of Solids and Structures*, 37(1-2):379–389, jan 2000.

[134] Ellad B Tadmor and Ronald E Miller. Benchmarking, validation and reproducibility of concurrent multiscale methods are still needed. *Modelling and Simulation in Materials Science and Engineering*, 25(7):071001, oct 2017.

[135] Hossein Talebi, Mohammad Silani, and Timon Rabczuk. Concurrent multiscale modeling of three dimensional crack and dislocation propagation. *Advances in Engineering Software*, 80:82–92, feb 2015.

[136] Joshua B. Tenenbaum. Mapping a manifold of perceptual observations. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, NIPS '97, pages 682–688, Cambridge, MA, USA, 1998. MIT Press.

[137] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[138] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[139] M.A. Tschopp and D.L. McDowell. Influence of single crystal orientation on homogeneous dislocation nucleation under uniaxial loading. *Journal of the Mechanics and Physics of Solids*, 56(5):1806–1830, may 2008.

[140] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *INT J DATA WAREHOUSING AND MINING*, 3(3):1–13, 2007.

[141] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986.

[142] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11(Feb):451–490, 2010.

[143] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In Joan Cabestany, Alberto Prieto, and Francisco Sandoval, editors, *Computational Intelligence and Bioinspired Systems*, pages 758–770, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[144] Gregory J. Wagner and Wing Kam Liu. Coupling of atomistic and continuum simulations using a bridging scale decomposition. *Journal of Computational Physics*, 190(1):249–274, sep 2003.

[145] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3 edition, 2012.

[146] Andreas S Weigend and Neil A Gershenfeld. Results of the time series prediction competition at the Santa Fe Institute. In *Neural Networks, 1993., IEEE International Conference on*, pages 1786–1793. IEEE, 1993.

[147] Wikipedia. Self-organizing map. Available at "https://en.wikipedia.org/wiki/Self-organizing_map".

[148] Wikipedia contributors. Feature selection — Wikipedia, the free encyclopedia. Available at "https://en.wikipedia.org/w/index.php?title=Feature_selection\&oldid=849580613".

[149] WIlliam H. Wolberg and Olvi Mangasarian. Breast cancer. https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original).

[150] Worldbank. Countries. http://databank.worldbank.org/data/.

[151] Shaoping Xiao. A non-oscillatory method for spallation studies. *International Journal for Numerical Methods in Engineering*, 66:364–380, 2006.

[152] Shaoping Xiao, David R Andersen, Ray Han, and Wenyi Hou. Studies of Carbon Nanotube-Based Oscillators Using Molecular Dynamics. *Journal of Computational and Theoretical Nanoscience*, 3(1):142–147, 2006.

[153] Shaoping Xiao, David R Andersen, and Weixuan Yang. Design and Analysis of Nanotube-Based Memory Cells. *Nanoscale Research Letters*, 3:416–420, 2008.

[154] Shaoping Xiao and Wenyi Hou. Fracture of vacancy-defected carbon nanotubes and their embedded nanocomposites. *Physical Review B*, 73(11):115406, mar 2006.

[155] Shaoping Xiao and Wenyi Hou. Studies of nanotube-based resonant oscillators through multiscale modeling and simulation. *Physical Review B*, 75(12):125414, mar 2007.

[156] Shaoping Xiao, Shaowen Wang, Jun Ni, Ransom Briggs, and Maciej Rysz. Reliability Analysis of Carbon Nanotubes Using Molecular Dynamics with the Aid of Grid Computing. *Journal of Computational and Theoretical Nanoscience*, 5(4):528–534, apr 2008.

[157] Shaoping Xiao and Weixuan Yang. Temperature-related CauchyâĂŞBorn rule for multiscale modeling of crystalline solids. *Computational Materials Science*, 37(3):374–379, sep 2006.

[158] Shaoping Xiao and Weixuan Yang. A temperature-related homogenization technique and its implementation in the meshfree particle method for nanoscale simulations. *International Journal for Numerical Methods in Engineering*, 69(10):2099–2125, mar 2007.

[159] S.P. Xiao and T. Belytschko. A bridging domain method for coupling continua with molecular dynamics. *Computer Methods in Applied Mechanics and Engineering*, 193(17-20):1645–1669, may 2004.

[160] Weixuan Yang and Shaoping Xiao. Extension of the temperature-related CauchyâĂŞBorn rule: Material stability analysis and thermo-mechanical coupling. *Computational Materials Science*, 41(4):431–439, feb 2008.

[161] I-Cheng Yeh, King-Jang Yang, Ting, and Tao-Ming. Blood transfusion. `https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center`.

[162] Hualong Yu, Yulong Yuan, Xibei Yang, and Yuanyuan Dan. A dynamic generation approach for ensemble of extreme learning machines. In Zhigang Zeng, Yangmin Li, and Irwin King, editors, *Advances in Neural Networks – ISNN 2014*, pages 294–302, Cham, 2014. Springer International Publishing.

[163] M. Zhou. A new look at the atomic level virial stress: on continuum-molecular system equivalence. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 459(2037):2347–2392, sep 2003.