Discovering Partial-Value Associations and Applications

by

Ting Yan Fok

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved October 2023 by the
Graduate Supervisory Committee:

Nong Ye, Chair
Ashif Iquebal
Feng Ju
James Collofello

ARIZONA STATE UNIVERSITY

December 2023

ABSTRACT

Existing machine learning and data mining techniques have difficulty in handling three characteristics of real-world data sets altogether in a computationally efficient way: (1) different data types with both categorical data and numeric data, (2) different variable relations in different value ranges of variables, and (3) unknown variable dependency.

This dissertation developed a Partial-Value Association Discovery (PVAD) algorithm to overcome the above drawbacks in existing techniques. It also enables the discovery of partial-value and full-value variable associations showing both effects of individual variables and interactive effects of multiple variables. The algorithm is compared with Association rule mining and Decision Tree for validation purposes. The results show that the PVAD algorithm can overcome the shortcomings of existing methods.

The second part of this dissertation focuses on knee point detection on noisy data. This extended research topic was inspired during the investigation into categorization for numeric data, which corresponds to Step 1 of the PVAD algorithm. A new mathematical definition of knee point on discrete data is introduced. Due to the unavailability of ground truth data or benchmark data sets, functions used to generate synthetic data are carefully selected and defined. These functions are subsequently employed to create the data sets for this experiment. These synthetic data sets are useful for systematically evaluating and comparing the performance of existing methods. Additionally, a deep-learning model is devised for this problem. Experiments show that the proposed model surpasses existing methods in all synthetic data sets, regardless of whether the samples have single or multiple knee points.

The third section presents the application results of the PVAD algorithm to real-world data sets in various domains. These include energy consumption data of an

Arizona State University (ASU) building, Computer Network, and ASU Engineering Freshmen Retention. The PVAD algorithm is utilized to create an associative network for energy consumption modeling, analyze univariate and multivariate measures of network flow variables, and identify common and uncommon characteristics related to engineering student retention after their first year at the university. The findings indicate that the PVAD algorithm offers the advantage and capability to uncover variable relationships.

DEDICATION

I dedicate this dissertation to my family. To my loving and supportive husband Kevin, your constant support and encouragement have been instrumental in my journey towards completing this dissertation. To my amazing parents and beloved brother Hubert, thank you for your love and patience throughout my journey of pursuing and successfully completing this degree, especially the times you needed me the most. Finally, I dedicate this work to the Lord, God, as I am sincerely thankful for everything I have. You are truly faithful and loving.

ACKNOWLEDGEMENTS

First, I would like to give thanks to my advisor, Dr. Nong Ye for her invaluable support, guidance, and mentorship throughout my Ph.D. studies. Dr. Ye is an exceptional research leader and a caring individual who genuinely fosters her students' growth and success. Dr. Ye's insights and feedback have enhanced the quality of my work and broadened my perspective in the field. Other than academia, she has taught me the importance of prioritizing self-care. I am confident that the skills and values she has instilled in me will continue to benefit me in the future.

I would also like to extend my sincere gratitude to my committee members: Dr. Ashif Iquebal, Dr. Feng Ju, and Dr. James Collofello. Their insightful suggestions and gentle guidance have played an indispensable role in shaping the outcome of my research. I am truly grateful to have the privilege of having such exceptional and eminent researchers as my dissertation committee members.

I am incredibly grateful to all my amazing professors and colleagues who have played a significant role in my academic journey. I would like to extend my heartfelt appreciation to Dr. Jing Li, Dr. Teresa Wu, Dr. Daniel McCarville, Dr. Adolfo Escobedo, Dr. Pan Rong, Dr. Ya Yan Lu, Yanzhe (Josh) Xu, Jiajing Huang, Ali Sarabi for their warm and caring support, as well as encouragement. You all are incredibly remarkable individuals that I have had the pleasure of meeting, particularly during my difficult time in the U.S.

Lastly, I would like to thank my husband Kevin Ding, family and friends for their company and assistance: Haiqin Lin, Si Wang, Chris Luo, John Qiang, Helen Liang, Yan Yu, Michaela Wu, Jiner, Xin Li, Enhui Shao, Daniel Yang, Zheng Zhu, Hazel Yang, Dunchuan Wu, Yiqun Dai, Bo Peng, Xiaorong Zhang.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Background

Real-world data sets often have the following three properties: (1) the existence of variable relations in different ranges of variable values, (2) unknown variable dependency, and (3) the inclusion of both numeric and categorical data.

Regarding (1), existing data mining or machine learning methods fit a model by considering the variables' full range of values. The fitted model is used to find or explain the relationship among variables. By fitting with only one model, these methods assume that one single model is sufficient to explain the data. However, there are several drawbacks when making such an assumption. The first problem is that we may have different relationships over different ranges of data values. The second problem is that a relationship may exist for a particular range and no relationships exist in other data value ranges. What is more, fitting one single model over the entire data value range results in a poor fit to the data if different relationships exist in different ranges of data. This can be seen from the Fisher's Iris data set (Frank and Asuncion, 2008) in which the classification of the target variable (Plant Type) using all independent variables only works on the target value *Iris Sentosa* but not on the remaining two (*Iris Versicolor* and *Iris Virginica*). For such data where variable relations only hold for partial ranges of variable values or different variable relations hold for different ranges of variable values, fitting a model using the same variable relations for all variable values does not fit all data values well, that is, the model explains or represents the whole data set poorly. Hence, it is desirable to have a

method that discovers both full-value and partial-value relations among variables.

What is more, variable dependency is often not known a priori in real-world data. In existing techniques, it is necessary to select a variable as either an independent or target variable before training the model. Once a variable is designated as an independent variable, it remains fixed and cannot be changed to be a dependent variable. This limitation adds an additional constraint when using a single model to analyze data relationships. In fact, a variable can be an independent variable in one relation while it can be a dependent variable in other relations.

The third characteristic is the inclusion of both numeric and categorical data. For example, the engineering student data at Arizona State University (ASU) that we analyze to understand engineering retention (Ye *et al.*, 2018c; Ye, 2017b) has both numeric data fields such as age and GPA and categorical data fields such as gender and race. While some of the current techniques can only handle data of the same type, it is desirable to have a method that can overcome this shortcoming.

However, the above three properties cannot be handled by existing data mining or machine learning techniques all at once. The Partial-Value Association Discovery (PVAD) algorithm is therefore developed to overcome those drawbacks of existing techniques.

1.2 State of the Art

There are several methods that can deal with one or some of the real-world data set properties. Taking the examples of Decision and Regression Tree (Ye, 2013a, 2003) as well as Random Forest (Breiman, 2001; Ho, 1995, 1998) these are supervised learning methods that can build tree-like models. Although the computation required to split an internal node for continuous variables is not efficient, they can handle mixed-type data (3). Nonetheless, decision tree methods require prior knowledge of variable

dependency. To find out all the associations among variables, these methods have to be applied multiple times, and each time one variable is selected as a target/dependent variable and the rest as attribute/independent variables. Therefore, the Decision and Regression Tree can handle (1) and (3) but not (2). In addition, techniques such as parametric and non-parametric regression (Friedman, 1991; Hastie *et al.*, 2009; Zhang and Singer, 2010), as well as Support Vector Machine (Gasse *et al.*, 2012; Breiman, 1998, 1996; Freund and Schapire, 1997; Mason *et al.*, 1999), are some commonly used supervised learning models. Depending on the data type of target variables, classification or prediction can be performed. With categorical data presented in the data set, the corresponding variables have to be transformed into dummy variables before fitting into a model (Draper and Smith, 1998). Nonetheless, these models assume that the role of a variable in a variable relation is known (i.e., which variable is an independent or dependent variable) and a variable can only play one role of being either an independent variable or a dependent variable in one layer of variable relations. Once a variable is considered as an independent variable, it can no longer be utilized as a dependent variable which is a main disadvantage, especially when the role of a variable is not known or when multiple layers of variable relations are required where a variable can play different roles of being an independent or dependent variable in different variable relations at different layers. Thus, these methods are also capable of handling (3) but not (1) and (2). The same drawbacks can be found in Artificial Neural Networks Tso and Yau (2007) which cannot work with categorical data directly and requires prior knowledge of independent and dependent variables.

Bayesian networks (Frank and Asuncion, 2008; Ye *et al.*, 2018c; Tsamardinos *et al.*, 2006; Ellis and Wong, 2008), structural equation models (Jones *et al.*, 2014a) and reverse engineering methods (AKUTSU *et al.*, 1999; Bazil *et al.*, 2011) are examples of the few options left that do not require prior knowledge of variables. However,

those techniques discover only variable relations for full ranges of all variable values instead of relations for specific values. Hence, these methods can work on (2) and (3) but not (1).

The last but most related approach is Association Rule Mining, which is one of the data mining techniques for discovering associative relationships among data. It was originally used for market basket analysis to find out items that are frequently purchased together. An example of a rule is "diaper $\rightarrow$ beer". The strength of a rule can be evaluated by its support and confidence (Agrawal *et al.*, 1993). An association rule with high confidence suggests a high probability that a customer buying a diaper will also grab a beer. Apriori (Agrawal *et al.*, 1996) is the first implemented algorithm for Association Rule Mining. In general, there are two main steps in the algorithm (Han *et al.*, 1999; Tseng and Chen, 2005). Step 1 finds frequent itemsets that meet the minimum support threshold and Step 2 discovers association rules that satisfy the minimum confidence threshold using itemsets found in Step 1. Though pruning is applied in each step of candidate rule generation, the exponential time and space complexity lead to the mining process being prohibitive. This expensive process also makes Association Rule Mining for high-dimensional data a challenging task

After all, Association Rule Mining cannot return all the associations among variables. This method generates rules based on the notion of frequent itemset. In every iteration of Step 1, any itemset that has less frequency count than the minimum support threshold is pruned and no longer be considered in the next iteration. However, these less supported itemsets may also have associations among the variables. What is more, it is natural that certain values of a variable have lower frequency counts. This can be seen in examples of minorities or people with higher incomes. Early pruning will lose all the associations involving those variable values.

1.3 Research Objective and Contributions

The inadequacies of existing techniques in dealing with real-world data are discussed in Section 1.2. These methods are shown to be only able to handle some of the characteristics of real-work data, but not all of them. Association Rule Mining is also considered but its outputs do not return a full set of associations.

Hence, the primary objective of this dissertation is to develop an algorithm that can overcome the deficiencies in the current techniques. This algorithm is capable of identifying both partial-value and full-value associations in the variables. We have proven the advantage and the capability of the algorithm in discovering variable relations by applying it to analyze data sets in different fields. In particular, we have deployed it to:

1. Identify student characteristics that affect the retention of first-year engineering students at ASU. (Ye and Fok, 2019; Ye *et al.*, 2019, 2021a),

2. Construct an associative network to model variable relations of energy consumption data for a building at ASU (Ye *et al.*, 2018b,a),

3. Establish both univariate and multivariate metrics to analyze computer network traffic data in order to identify network attacks and anomalies Ye *et al.* (2019, 2021b).

Another significant contribution we have made is the development of a computationally efficient method, called YFM3 (Ye and Fok, 2019), for finding the longest associations. Unlike existing methods that exhaustively examine all possible combinations of variable values, our method offers a shortcut approach. Bypassing any iterations, YFM3 directly provides the longest associations with exceptional computational efficiency. This breakthrough saves a significant amount of time and resources

in the process.

This research was further expanded during the investigation of Step 1 of the PVAD's algorithm, which is essentially categorizing numerical data into categorical data. This process involves identifying knee point(s) on the curve of sorted data. Therefore, our contributions to this problem are as follows:

1. Introducing a new mathematical definition for a knee point in discrete data sets. We have demonstrated and explained the necessity of rescaling the data,

2. Creating a benchmark data set that provides noisy data within the original data range, along with ground truth labels that are independent of any underlying algorithm/techniques,

3. Developing a new deep-learning model, UNetConv, for detecting knee points in discrete data sets. We have demonstrated that the proposed model outperforms other state-of-the-art methods in both seen and unseen data.

1.4 Dissertation Organization

The proposed dissertation research will be presented in the following chapters. Chapter 2 presents the PVAD Algorithm. Details are given in Section 2.1. The method was validated through a sensitivity analysis to assess the impact of parameter values on the results. Section 2.2 describes the results of the analysis. Furthermore, in Sections 2.3 and 2.4, the results of the PVAD algorithm are compared with those of Association Rule Mining and Decision Tree for verification purposes. The results show that the PVAD algorithm can overcome the shortcomings of existing methods.

Chapter 3 presents an extended research topic which is known as the Knee Point Detection Problem. This topic was inspired during the investigation into categorization for numeric data, which corresponds to Step 1 of the PVAD algorithm. Section

3.3 provides a novel mathematical definition of knee point on continuous functions, along with its extension to discrete data. The deep-learning model devised for this problem is described in Section 3.4, and its performance compared with existing methods is given in Section 3.5.

The PVAD algorithm is eligible to deal with real-world data. Chapter 4 describes the applications of the PVAD algorithm to real-world data sets in different domains. Sections 4.1 - 4.5 presents the PVAD method's application to the real-world data sets: energy consumption data of an Arizona State University (ASU) building, Computer Network, and ASU Engineering Freshmen. The findings show that the PVAD algorithm has the advantage and the capability of discovering variable relations.

Chapter 2

PARTIAL-VALUE ASSOCIATION DISCOVERY

2.1 Methodology

The PVAD algorithm consists of three main steps.

Step 1. Identify value intervals/categorical values of variables,

Step 2. Discover partial-value associations of variable values,

Step 3. Construct a multi-layer structural model using established partial-value associations.

In Step 1, we consider two kinds of variables presenting in a data set: categorical variables and numeric variables. A categorical variable already comes with its categorical values which can be used directly in Step 2 of the PVAD algorithm. This step is illustrated by the chemical data set (http://www.stat.columbia.edu/ gelman/book/data/) as shown in Table 2.1. This data set has four numeric variables: Temperature, Ratio, Contact, and Conversion. The variable Temperature has three numeric values: 1100, 1200, and 1300, which can be directly taken as three categorical values of Low, Medium and High, respectively. To transform the numeric data into categorical, we plot the sorted values of the variable, identify data clusters, and use the non-overlapping intervals of data clusters to define the categorical values of the numeric variable. Figure 2.1a shows an example of using this method to determine the data clusters and the categorical values of the numeric variable, Contact, from the chemical data set. This variable has the following 16 values in the data set:

0.0120	0.0120	0.0115	0.0130	0.0135	0.0120	0.0400	0.0380
0.0320	0.0260	0.0340	0.0410	0.0840	0.0980	0.0920	0.0860

The data plot in Figure 2.1a shows these 16 values in the sorted order of increasing values in three data clusters with the non-overlapping intervals to define three categorical values of c1, c2, and c3: $[0.0115, 0.0135]$ = c1, $[0.0260, 0.0410]$ = c2, and $[0.0840, 0.0980]$ = c3. These three data clusters are identified by inspecting visually the biggest jumps in the value differences or distances of consecutive data points in the data plot. As shown in Figure 2.1a, the value distances of consecutive data points within each of these three data clusters are smaller than the value distances between the data clusters. Based on this principle of having distances of data points within a data cluster smaller than the distances of data points in different data clusters, clustering techniques such as hierarchical clustering (Ye, 2013a, 2003) can also be used to produce the same data clusters. We can also use elbow points in the data plot that start line segments with different slopes to identify data clusters since changes in line segment slopes indicate big changes in the distances of consecutive data points. Similarly, to transform the variable Ratio, we plot the sorted values of Ratio in Figure 2.1b and identify three data clusters and their corresponding categorical intervals: $[5.3, 13.5]$ = Low, $[17, 17]$ = Medium, and $[23, 23]$ = High. Following the same procedure. While for the variable Conversion, the identified clusters are plotted in Figure 2.1c and the corresponding categorical intervals are: $[15.0, 20.5]$ = Low, $[28.0, 38.5]$ = Medium, and $[44.5, 50.5]$ = High. The above data clusters are all identified by visually inspecting the largest distances of adjacent data points. Using clustering techniques such as hierarchical clustering can produce the same data clusters.

If both Method 1 and Method 2 are used to transform a numeric variable into a categorical variable, the two sets of categorical values from Method 1 and Method 2 are compared to select the set of categorical values with the smallest number of categorical values based on the MDL principle (Ye, 2013a, 2003).

The data in Table 2.1 are transformed into categorical values as shown in Table 2.2.

Table 2.1: Original Chemical Data Set.

Instance	**Temperature**	**Ratio**	**Contact**	**Conversion**
1	1300	7.5	0.012	49
2	1300	9	0.012	50.2
3	1300	11	0.0115	50.5
4	1300	13.5	0.013	48.5
5	1300	17	0.0135	47.5
6	1300	23	0.012	44.5
7	1200	5.3	0.04	28
8	1200	7.5	0.038	31.5
9	1200	11	0.032	34.5
10	1200	13.5	0.026	35
11	1200	17	0.034	38
12	1200	23	0.041	38.5
13	1100	5.3	0.084	15
14	1100	7.5	0.098	17
15	1100	11	0.092	20.5
16	1100	17	0.086	19.5

When we collect new data, we may have data values not present in the original data set which are used to derive intervals of values and define categorical values. It is possible that new data may have values falling in the gaps of the intervals used to define categorical values. For any numeric value falling in a gap of intervals, the numeric value can take the categorical value for the interval which has a value closest to the numeric value. For example, the transformed variable Conversion has categorical values for the following intervals: $[15.0, 20.5]$ = Low, $[28.0, 38.5]$ = Medium, and $[44.5, 50.5]$ = High. A new data record with a Conversion value of 23.1 takes the categorical value of Low because it is closest to the upper interval of $[15.0, 20.5]$.

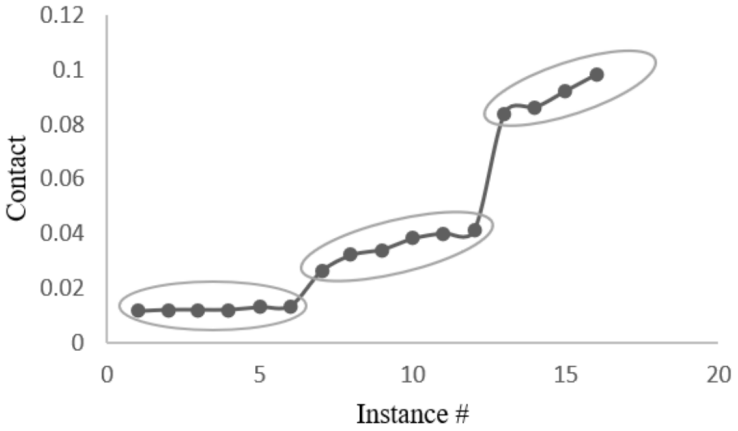

Figure 2.1(a): The plot of the sorted values of Contact.

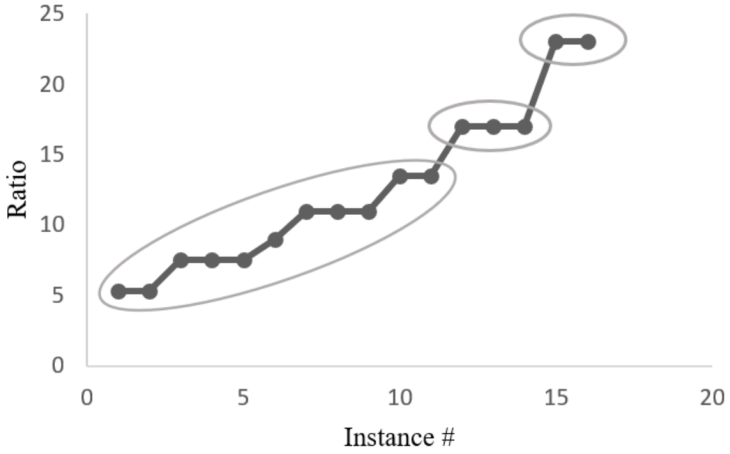

Figure 2.1(b): The plot of the sorted values of Ratio

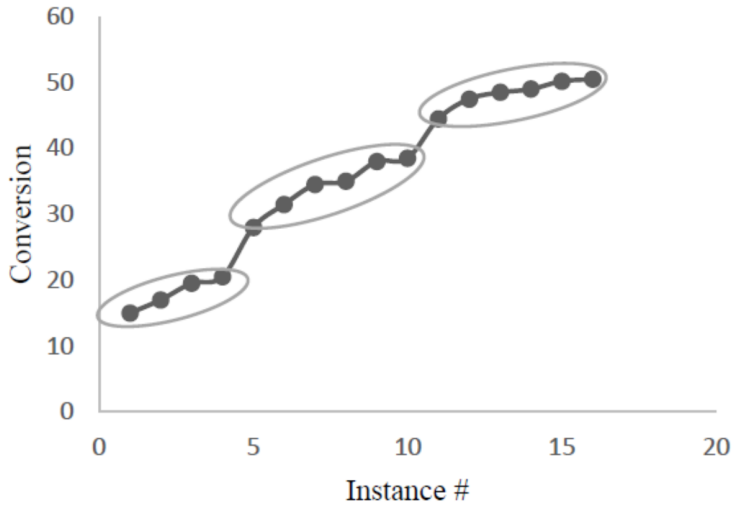

Figure 2.1(c): The plot of the sorted values of Conversion.

Figure 2.1: Determining data clusters and categorical values of variables in the chemical data set

Table 2.2: Transformed Chemical Data Set by Step 1 of PVAD

Instance	**Temperature**	**Ratio**	**Contact**	**Conversion**
1	High	Low	Low	High
2	High	Low	Low	High
3	High	Low	Low	High
4	High	Low	Low	High
5	High	Medium	Low	High
6	High	High	Low	High
7	Medium	Low	Medium	Medium
8	Medium	Low	Medium	Medium
9	Medium	Low	Medium	Medium
10	Medium	Low	Medium	Medium
11	Medium	Medium	Medium	Medium
12	Medium	High	Medium	Medium
13	Low	Low	High	Low
14	Low	Low	High	Low
15	Low	Low	High	Low
16	Low	Low	High	Low

Step 2 of the PVAD algorithm discovers partial-value associations of variable values, and each partial-value association is in the form of X = $A \rightarrow$ Y = B where X and Y are the vectors of one or more variables, A and B are the values of X and Y, respectively, X is called conditional variable(s), and Y is called associative variable(s), X = A are called conditional variables' values (CV), and Y = B are called associative variables' values (AV). Each data record in the data set is called an instance with instance #. Step 2 of the PVAD algorithm consists of the following steps.

Step 2.1. Discover 1-to-1 partial-value associations, x = $a \rightarrow$ y = b, where the association involves only one conditional variable and only one associative variable. For each value a of each variable x, each value b of each variable y, and the candidate association, x = $a \rightarrow$ y = b, we carry out the following steps:

Step 2.1.1. Compute the co-occurrence ratio (cr) of each candidate associ-

ation, x = a → y = b as follows:

$$cr(\mathrm{x} = a \rightarrow \mathrm{y} = b) = \frac{N_{\mathrm{x}=a,\ \mathrm{y}=b}}{N_{\mathrm{x}=a}} \tag{2.1}$$

, where $N_{\mathrm{x}=a,\ \mathrm{y}=b}$ is the number of instances containing both x = a and y = b, and $N_{\mathrm{x}=a}$ is the number of instances containing x = a.

Step 2.1.2. Store each candidate 1-to-1 association, including its cr value, N_{CV}, and the instance indices # which is the set of instances supporting this association (i.e. instances supporting x = a and y = b), where N_{CV} is the number of instances containing CV(x = a). A candidate 1-to-1 association is defined as an association with $cr \neq 0$ or ∞ in Equation 2.1, that is, $N_{\mathrm{x}=a \text{ and } \mathrm{y}=b} \neq 0$ and $N_{\mathrm{x}=a} \neq 0$

Step 2.1.3. Establish the 1-to-1 partial-value association, x = a → y = b, if $cr(\mathrm{x} = a \rightarrow \mathrm{y} = b) \geq \alpha$, where α is set to a value in the range of $(0, 1]$ and is close or equal to 1.

Note that an established association has $cr \geq \alpha$, and a candidate association may have any cr value in $(0, 1]$.

In Step 2.2, we discover p-to-q partial-value associations, X = A → Y = B, where the association involves either multiple conditional variables or multiple associative variables, using the methods of YFM1, and YFM2 and the procedure shown in Figure 2.2 to generate all the established p-to-q partial-value associations.

The method of YFM1 takes each group of the established k-to-l associations (e.g. the established 1-to-1 associations from Step 2.1) having the same or inclusive set of

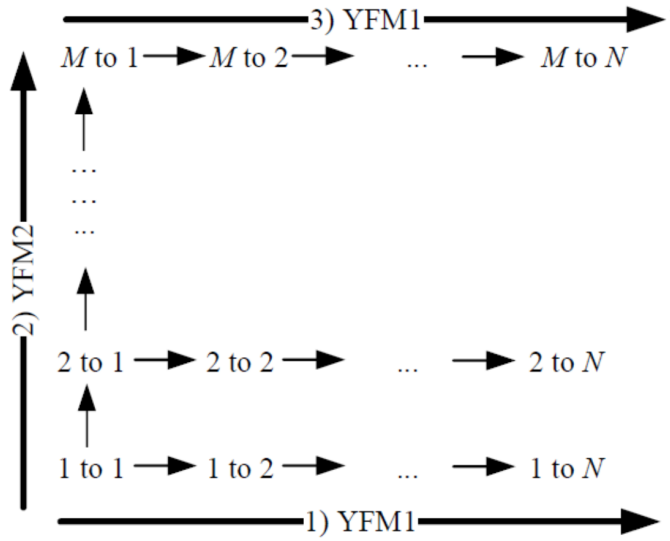

Figure 2.2: The procedure of using YFM1 and YFM2 to establish partial-value associations.

supporting instances and the same CV, and establishes k-to-q associations, $\{\text{CV}\} \rightarrow \{AV_q\}$, where $q > l$ and $\{AV_q\}$ is any combination of AVs from the associations in the group, because of the following:

$$\begin{aligned} cr(CV \rightarrow \{AV_q\}) &= \frac{N_{CV \text{ and } \{AV_q\}}}{N_{CV}} = \frac{\min\{N_{CV \text{ and } AV_i \mid i=1,\ldots,q}\}}{N_{CV}} \\ &= \min\{cr(CV \rightarrow AV_1), \ldots, cr(CV \rightarrow AV_q)\} \geq \alpha \end{aligned} \tag{2.2}$$

The YFM1 method also takes each group of the established k-to-l associations having the same or inclusive set of supporting instances and the same AV, and establishes p-to-l associations, $\{CV_p\} \rightarrow \{AV\}$, where $p > k$ and $\{CV_p\}$ is any combination of CVs from the associations in the group, because of the following:

$$
\begin{aligned}
cr(\{CV_p\} \rightarrow AV) &= \frac{N_{\{CV_p\} \text{ and } AV}}{N_{\{CV_p\}}} \\
&\geq \max\{cr(CV_1 \rightarrow AV), \ldots, cr(CV_p \rightarrow AV)\} \\
&\geq \alpha
\end{aligned} \tag{2.3}
$$

where $N_{\{CV_p\}} \leq N_{\{CV_i\}},\ i = 1, \ldots, p$ and $N_{\{CV_p\} \text{ and } AV} = min\{N_{CV_i \text{ and } AV} \mid i = 1, \ldots, p\}$. Moreover, the method of YFM1 takes the associations in the group having the same CV and their largest common subset satisfying the following condition:

$$
\frac{N_{\text{Common Subset}}}{N_{CV}} \geq \alpha \tag{2.4}
$$

where $N_{\text{Common Subset}}$ is the number of instances in the largest common subset, and establishes k-to-q associations, $\{CV\} \rightarrow \{AV_q\}$, where $q > l$ and $\{AV_q\}$ is any combination of AVs from the associations in the group, because of the following:

$$
cr(CV \rightarrow \{AV_q\}) = \frac{N_{\text{Common Subset}}}{N_{CV}} \geq \alpha \tag{2.5}
$$

For example, suppose that we have the group of the established 1-to-1 associations from Step 2.1, $\mathrm{x}_1 = a_1 \rightarrow \mathrm{y}_1 = b_1$ with the supporting set of instances {1, 2, 3, 4, 5} and $\mathrm{x}_1 = a_1 \rightarrow \mathrm{y}_2 = b_2$ with the same supporting set of instances 1, 2, 3, 4, 5 or a subset of instances {1, 2, 3, 4}, where 1, 2, 3, 4, and 5 are instance #. {1, 2, 3, 4} is an inclusive set to {1, 2, 3, 4, 5} as {1, 2, 3, 4, 5} includes {1, 2, 3, 4}, in other words, the inclusive set {1, 2, 3, 4} is a subset of {1, 2, 3, 4, 5}. We establish the 1-to-2 association, $\mathrm{x}_1 = a_1 \rightarrow \mathrm{y}_1 = b_1, \mathrm{y}_2 = b_2$, using YFM1.

The p-to-q associations established in YFM1 provide more specific CV or AV than the k-to-l associations. For example, the associations established for the energy con-

sumption data set, [ ] include the following 1-to-1 association and 2-to-1 associations:

$$\text{1-to-1: E = Medium} \rightarrow \text{ A = High}$$

$$\text{2-to-1: E = Medium, C = High} \rightarrow \text{ A = High}$$

where E stands for electricity consumption, A stands for outside air temperature, and C stands for cooling consumption. Both cooling and heating may use electricity. The 1-to-1 association indicates the association of the medium level of electricity consumption with the high level of air temperature in general. The two CVs in the 2-to-1 association provide a more specific association of the cooling consumption part of the electricity consumption with the high level of air temperature.

The method of YFM2 establishes p-to-1 associations from the candidate $(p-1)$-to-1 associations in the following steps:

1. Sort the $(p-1)$-to-1 associations by their CV
2. For each group of associations with the same CV

    a) For each association t_i in the group, determine the minimum number of instances required for the common subset, m_i, as follows:

$$m_i = \lceil n_i \times \alpha \rceil \tag{2.6}$$

    where n_i is the total number of instances in the set of instances supporting association t_i.

    b) For every other association t_j in the group, if the variable in the AV of association t_j is not the same as the variable in the AV of association t_i AND $n_{\text{intersection}} \geq m_i$, where $n_{\text{intersection}}$ is the number of instances in the intersection of the two instance sets supporting t_i and t_j, we establish a

new p-to-1 association with CV = {CV and AV in t_i} and AV = {AVin t_j}, because this new association has the following cr value $\geq \alpha$:

$$cr(\ \{CV \text{ and } AV \text{ in } t_i\} \rightarrow AV \text{ in } t_j) = \frac{n_{\text{intersection}}}{n_i} \geq \frac{m_i}{n_i} \geq \alpha \qquad (2.7)$$

For example, suppose that we have $\alpha = 0.5$ and a group of two 2-to-1 associations with the same CV as follows:

$\mathrm{x}_1 = a_1, \mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_3 = a_3$, with the supporting set of instances {1, 2}, thus $n_1 = 2$

$\mathrm{x}_1 = a_1, \mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_4 = a_4$, with the supporting set of instances {1, 3}, thus $n_2 = 2$.

For the first association, t_1, we have:

$$n_1 = 2, m_1 = \lceil 2 \times 0.5 \rceil = 1.$$

The second association, t_2, has AV which is not the same as AV in the first association. The intersection of the two instance sets supporting the first and second associations is {1}. The number of instances in the intersection is $1 \geq m_1$. Thus, we establish a new 3-to-1 association:

$$\mathrm{x}_1 = a_1, \mathrm{x}_2 = a_2, \mathrm{x}_3 = a_3 \rightarrow \mathrm{x}_4 = a_4$$

It is computationally fast to discover 1-to-1 associations in Step 2.1. It takes time for YFM2 to examine possible combinations of multiple variable values. However, there is no shortcut by only considering established associations. This is illustrated by the following example. The data set is shown below and YFM2 is used to establish

2-to-1 associations from the given 1-to-1 candidate associations:

Instance	**x1**	**x2**	**x3**
1	1	0	1
2	1	0	1
3	1	0	1
4	0	0	0
5	0	0	0
6	0	1	1
7	0	1	1
8	0	1	1

If we set $\alpha = 0.9$, we have for this data set:

$$cr(\mathrm{x}_1 = 0 \rightarrow \mathrm{x}_2 = 0) = \frac{2}{5} = 0.4, \text{with the supporting set of instances } \{4, 5\}$$
$$cr(\mathrm{x}_1 = 0 \rightarrow \mathrm{x}_3 = 0) = \frac{2}{5} = 0.4, \text{with the supporting set of instances } \{4, 5\}$$

For the first 1-to-1 association, we compute:

$$m_1 = \lceil 2 \times 0.9 \rceil = 2$$

Since the intersection of the above two 1-to-1 associations has 2 instances, 4, 5, we establish the new 2-to-1 association from the above two 1-to-1 associations:

$$\mathrm{x}_1 = 0, \mathrm{x}_2 = 0 \rightarrow \mathrm{x}_3 = 0$$

Note that in this example, both 1-to-1 associations have $cr < \alpha$ and so they are not established. From this result, we show that attribute values in a $(p-1)$-to-1 candidate association can become part of an established p-to-1 association in a later iteration of YFM2. Hence, it is necessary to keep these candidate 1-to-1 associations

in order to establish all 2-to-1 associations. In general, we need to store all candidate p-to-1 associations ($p = 1, 2, \ldots, M-1$) with their crs and supporting instance sets as inputs to the YFM2.

In the procedure of using YFM1 and YFM2 in Step 2.2 as shown in Figure 2.2, at first we use YFM1 to establish 1-to-2, ..., 1-to-M associations from the established 1-to-1 associations from Step 2.1, where M is the total number of variables in the data set. Secondly, we use YFM2 to establish 2-to-1 associations from the candidate 1-to-1 associations, 3-to-1 associations from the candidate 2-to-1 associations, ..., and M-to-1 associations from the candidate (M-1)-to-1 associations. At last, we use YFM1 to establish 2-to-2, ..., 2-to-M associations from the established 2-to-1 associations, ..., M-to-2, ..., M-to-M associations from the established M-to-1 associations. However, this procedure of Step 2.2 can be cut short (that is, stopped earlier) if an application needs p-to-q associations up to the given values of p and q, where $p < M$ and $q < M$. If we are interested in associations whose AV involves a certain value of a given variable only, the procedure of YFM2 can be stopped as soon as a k-to-1 association with such an AV is established, because any p-to-1 association with the same AV, where $p > k$, will be established using the same set of supporting instances for the k-to-1 association by taking in more CVs in these instances and the k-to-1 association is more generic and powerful association than the p-to-1 association.

Recently we developed YFM3, a computationally fast method of discovering the longest associations which include m-to-1 associations and 1-to-m associations, where $m + 1$ is the total number of variables in a data set. Suppose that we have an Excel file containing a data set with 5 columns for 5 variables, x_1, x_2, x_3, x_4, x_5, and 16 rows for 16 instances shown below.

x_1	x_2	x_3	x_4	x_5
S	Y	S	A	T
S	Y	S	C	T
D	Y	S	A	T
D	Y	S	C	T
S	Y	L	A	T
S	Y	L	C	F
D	Y	L	A	F
D	Y	L	C	F
S	P	S	A	T
S	P	S	C	F
D	P	S	A	F
D	P	S	C	F
S	P	L	A	T
S	P	L	C	F
D	P	L	A	F
D	P	L	C	F

Using this Excel file, we create five new Excel files with only two columns for two new variables in each file. In File 1, the first column is x_1, and the second column is named $x_2x_3x_4x_5$ which is the concatenation of the other four variables, x_2, x_3, x_4, x_5, as shown below.

x_1	$x_2x_3x_4x_5$
S	YSAT
S	YSCT
D	YSAT
D	YSCT
S	YLAT
S	YLCF
D	YLAF
D	YLCF
S	PSAT
S	PSCF
D	PSAF
D	PSCF
S	PLAT
S	PLCF
D	PLAF
D	PLCF

File 2 has x_2 as the first column and the concatenation of $x_1x_3x_4x_5$ as the second column. File 3 has x_3 as the first column and the concatenation of $x_1x_2x_4x_5$ as the second column. File 3 has x_3 as the first column and the concatenation of $x_1x_2x_4x_5$ as the second column. File 4 has x_4 as the first column and the concatenation of $x_1x_2x_3x_5$ as the second column. File 5 has x_5 as the first column and the concatenation of $x_1x_2x_3x_4$ as the second column. It is computationally fast to generate 1-to-1 associations using each of the five new data files. Putting together all 1-to-1 associations for all new data files produces all 1-to-4 associations and 4-to-1 associations. For example, a 1-to-1 association produced from File 1, $x_2x_3x_4x_5 = PLAF \rightarrow x_1 = D$, gives a 4-to-1 association of $x_2 = P, x_3 = L, x_4 = A, x_5 = F \rightarrow x_1 = D$.

Step 2.3 of the PVAD algorithm takes all the established associations from Step 2.1 and Step 2.2, uses β to remove the associations whose supporting set of instances has less than the β number of instances, uses γ to remove the 1-to-1 associations whose CV or AV exists in more than or equal to γ of all the instances in the data set, where β denotes the number of instances and can be set to a positive integer equal to or greater than 1, and γ is the percentage of instances in the data set and can be set to a value in (0%, 100%]. Hence, β is used to remove any association which is not supported by a sufficient number of instances in the data set. γ is used to remove any 1-to-1 association with its CV or AV being present in too many instances of the data set (too common in the data set). A 1-to-1 association with such a common CV or AV gives little meaningful, useful information on variable relations since the association is established solely due to the common presence of CV or AV in the data set.

In Step 3 of the PVAD algorithm, a multi-layer structural system model of partial/full-value associations is constructed. Step 3 of the PVAD algorithm consists of the

following steps.

Step 3.1 Consolidate and generalize the partial-value associations from Step 2 as follows:

If we have one association, x = a_1 → y = b, and another association, x = a_2 → y = b, also

- If a_1 and a_2 are two different but non-consecutive values of x, we replace x = a_1 → y = b and x = a_2 → y = b by a consolidated/generalized association, x = a_1/a_2 → y = b, where the operator / of two terms represents either of two terms but not both terms;
- If a_1 and a_2 are two consecutive values of x, we replace x = a_1 → y = b and x = a_2 → y = b by a consolidated/generalized association, x = a → y = b, where a is a new categorical value of x including a_1 and a_2.

Hence, if all values of variables have the same association, this step will consolidate and generalize the same associations for various values of variables in the associative network into one association for all values of variables. Hence, the PVAD algorithm can identify both partial-value and full-value associations of variables. We use x = $*$, to represent all values of x.

If one association has a more specific CV or AV than the CV or AV of another association, we can express the two associations using a consolidated representation. For example, x = a, y = b, z = c → s = d has a more specific CV than the CV of x = a, y = b → z = c. Meanwhile, the CV of x = a, y = b → z = c also has a more specific CV than the CV of $x = a \rightarrow z = c$.

We can express these three associations using the following representation:

$$\mathrm{x} = a\ (y = b\ (z = c)) \rightarrow s = d.$$

We can read each CV from the above representation by reading inside out of parentheses to get x = a and y = b and z = c at first, x = a and y = b secondly, and x = a at last.

We use the operator | of two terms to represent one or both terms to be present. Hence, the following expression:

$$\mathrm{x}_1 = a_1 \mid \mathrm{x}_2 = a_2 \rightarrow (\mathrm{x}_3 = a_3)\ \mathrm{x}_4 = a_4/a_5,$$

represents the following associations:

$$\begin{aligned}
&\mathrm{x}_1 = a_1 \rightarrow \mathrm{x}_4 = a_4\\
&\mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_4 = a_4\\
&\mathrm{x}_1 = a_1, \mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_4 = a_4\\
&\mathrm{x}_1 = a_1 \rightarrow \mathrm{x}_4 = a_5\\
&\mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_4 = a_5\\
&\mathrm{x}_1 = a_1, \mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_4 = a_5\\
&\mathrm{x}_1 = a_1 \rightarrow \mathrm{x}_3 = a_3, \mathrm{x}_4 = a_4\\
&\mathrm{x}_1 = a_1 \rightarrow \mathrm{x}_3 = a_3, \mathrm{x}_4 = a_5\\
&\mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_3 = a_3, \mathrm{x}_4 = a_4\\
&\mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_3 = a_3, \mathrm{x}_4 = a_5\\
&\mathrm{x}_1 = a_1, \mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_3 = a_3, \mathrm{x}_4 = a_4\\
&\mathrm{x}_1 = a_1, \mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_3 = a_3, \mathrm{x}_4 = a_5.
\end{aligned}$$

The pair of brackets [ ], is used to group some terms. For example, the

following expression:

$$[\mathrm{x}_1 = a_1, \mathrm{x}_2 = a_2]/\mathrm{x}_3 = a_3 \rightarrow \mathrm{x}_4 = a_4$$

represents the following associations:

$$\mathrm{x}_1 = a_1, \mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_4 = a_4$$

$$\mathrm{x}_3 = a_3 \rightarrow \mathrm{x}_4 = a_4.$$

Step 3.2 Use partial/full-value associations from Step 3.1 to construct an associative network which is a multi-layer structural model of partial/full-value associations from Step 3.1. A node is added to the associative network to represent each unique CV or AV of associations from Step 3.1. A directed link, called an associative link, is drawn in the associative network to represent each association. An associative network is drawn using a N-diagram. Figure 2.2 shows how $\mathrm{x} = a\ (\mathrm{y} = b\ (z = c)) \rightarrow s = d$ is represented in the N-diagram of an associative network. In an N-diagram of an associative network, a link starts from or points to the inside of an oval, and we read CV or AV inside out. For example, in Figure 2.2, the link starting from the inside of the oval which contains $\mathrm{x} = a$ has no other ovals outside this oval and thus represents $\mathrm{x} = a$ only. The same link pointing to the inside of the oval containing $\mathrm{s} = d$ represents $\mathrm{s} = d$ only since there are no other ovals outside this oval. The link starting from the inside of the oval containing $\mathrm{z} = c$ represents $\mathrm{x} = a$, $\mathrm{y} = b$, and $\mathrm{z} = c$ by reading inside out. Hence, the three links in Figure 2.2 represent:

$$\mathrm{x} = a\ (\mathrm{y} = b\ (\mathrm{z} = c)) \rightarrow \mathrm{s} = d,$$

that is, three associations:

$$\mathrm{x} = a\ (\mathrm{y} = b\ (\mathrm{z} = c)) \rightarrow \mathrm{s} = d,$$

Step 3.3 Remove a direct link between one CV node and one AV node if there are multiple paths from the CV node to the AV node since the direct link can be derived from a path of associative links from this CV node to this AV node. For example, if we have two paths going to node $\mathrm{x}_3 = a_3$: $\mathrm{x}_1 = a_1 \rightarrow \mathrm{x}_3 = a_3$ and $\mathrm{x}_1 = a_1 \rightarrow \mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_3 = a_3$, we remove the direct link $\mathrm{x}_1 = a_1 \rightarrow \mathrm{x}_3 = a_3$ because the direct link $\mathrm{x}_1 = a_1 \rightarrow \mathrm{x}_3 = a_3$ can be derived from $\mathrm{x}_1 = a_1 \rightarrow \mathrm{x}_2 = a_2 \rightarrow \mathrm{x}_3 = a_3$.

2.2 Sensitivity Analysis

Three parameters α, β, and γ in the PVAD algorithm are used to discover associations. To better understand the effect of these parameters on the results, the PVAD algorithm is run on ASU energy consumption system data with three different settings:

1. $\alpha = 0.8$, $\beta = 50$, $\gamma = 0.95$ vs. $\alpha = 1.$ $\beta = 50$, $\gamma = 0.95$;
2. $\alpha = 1$, $\beta = 10$, $\gamma = 0.95$ vs. $\alpha = 1.$ $\beta = 50$, $\gamma = 0.95$;
3. $\alpha = 1$, $\beta = 50$, $\gamma = 0.95$ vs. $\alpha = 1.$ $\beta = 50$, $\gamma = 1$.

In each setting, only one parameter value is varied and the remaining two are fixed. For simplicity, only 4-to-1 established associations are shown and compared. The metric used in this section is the percentage of instances covered by the established associations. Concretely, as instance # covered is stored along with the corresponding association, by taking the union of all those instance #s, we know that an instance # in the union set is covered by at least one of the established associations. Lastly,

by counting the instances in the union set and dividing it by the number of data records in the data set, the resulting fraction measures the percentage of instances "explained" by the established associations.

The first setting is to test how α affects the results. We run the PVAD two times with configurations $\alpha = 0.8$, $\beta = 50$, $\gamma = 0.95$ vs. $\alpha = 1$. $\beta = 50$, $\gamma = 0.95$ As α is the threshold for cr such that a candidate association is established if its $cr \geq \alpha$. It is expected that a higher value of α will lead to less established associations. The results are shown in Table 2.3. The first four rows show the associations commonly found in both configurations. In fact, if keeping β and γ constant, using a lower valuer of α will return a superset of results that uses a larger α. Regarding the percentage of covered instances, it is 62.37% for $\alpha = 0.8$ and 10.48% for $\alpha = 1$. There is a significant decrease because $\alpha = 1$ looks for the strongest association – wherever a row contains the CV, in the same row, it should also have the AV.

Table 2.3: Sensitivity Analysis of α.

$\alpha = 0.8, \beta = 50, \gamma = 0.95$		$\alpha = 1. \beta = 50, \gamma = 0.95$	
Established Associations	***cr***	**Established Associations**	***cr***
T=12:15 PM - 5:30 PM, E=High, C=Low, A=Medium → H=Medium	1	T=12:15 PM - 5:30 PM, E=High, C=Low, A=Medium → H=Medium	1
T=12:15 PM - 5:30 PM, E=Medium, C=High, H=Low → A=High	1	T=12:15 PM - 5:30 PM, E=Medium, C=High, H=Low → A=High	1
T=12:15 PM - 5:30 PM, E=Medium, C=Low, H=Low → A=High	1	T=12:15 PM - 5:30 PM, E=Medium, C=Low, H=Low → A=High	1
T=5:45 PM - 11 PM, E=High, C=Low, A=Medium → H=Medium	1	T=5:45 PM - 11 PM, E=High, C=Low, A=Medium → H=Medium	1
T=11:15 PM - 6 AM, E=Low, C=Low, A=Medium → H=Medium	0.81		
T=11:15 PM - 6 AM, C=Low, H=High, A=Low → E=Low	0.99		

Continued on next page

Table 2.3: Sensitivity Analysis of α. (Continued)

T=11:15 PM - 6 AM, C=Low, H=Low, A=High → E=Low	0.99		
T=11:15 PM - 6 AM, C=Low, H=Medium, A=Low → E=Low	0.95		
T=11:15 PM - 6 AM, C=Low, H=Medium, A=Medium → E=Low	0.95		
T=12:15 PM - 5:30 PM, E=High, C=Low, H=Medium → A=Medium	0.86		
T=12:15 PM - 5:30 PM, E=Low, C=Low, H=Medium → A=High	0.84		
T=12:15 PM - 5:30 PM, E=Medium, C=High, A=High → H=Low	0.95		
T=12:15 PM - 5:30 PM, E=Medium, C=Low, A=Medium → H=Medium	0.83		
T=12:15 PM - 5:30 PM, C=High, H=Low, A=High → E=Medium	0.95		
T=5:45 PM - 11 PM, E=Medium, C=Low, A=High → H=Medium	0.95		
T=5:45 PM - 11 PM, E=Medium, C=Low, A=Medium → H=Medium	0.81		
T=5:45 PM - 11 PM, C=Low, H=High, A=Medium → E=Medium	0.82		
T=6:15 AM - 8 AM, E=Medium, C=Low, H=High → A=Low	0.9		
T=6:15 AM - 8 AM, E=Medium, C=Low, A=Medium → H=Medium	0.88		
T=8:15 AM - 12 PM, E=Medium, C=Low, A=High → H=Medium	0.84		
T=8:15 AM - 12 PM, E=Medium, C=Low, A=Medium → H=Medium	0.84		
T=8:15 AM - 12 PM, C=Low, H=Medium, A=High → E=Medium	0.91		
T=8:15 AM - 12 PM, C=Low, H=Medium, A=Medium → E=Medium	0.82		
% of covered instances: 62.37%		**% of covered Instances: 10.48%**	

To test for β value's effect on the results, the second experiment has configurations $\alpha = 1$, $\beta = 10$, $\gamma = 0.95$, and $\alpha = 1$. $\beta = 50$, $\gamma = 0.95$. The role of β is to make sure that the CVs of an association appear "frequent" enough in the data set. As a higher value of β requires the CVs to have a larger frequency count in the data set, it

is expected that fewer associations can be established as β increases. Similar to α, a lower value of β generates more rules and the rules are a superset to those discovered by using a larger β value. The associations are presented in Table 2.4 which all have $cr = 1$. It can be observed from the table that there are only four associations found for β=50. The percentage of covered instances also drops from 78.73% to 10.48% when changing β from 10 to 50. It is not surprising as we do not have many rows having these four attribute values altogether.

Table 2.4: Sensitivity Analysis of β

Established Associations of Setting $\alpha = 1, \beta = 10, \gamma = 0.954$	**Established Associations of Setting** $\alpha = 1, \beta = 50, \gamma = 0.95$
T=12:15 PM - 5:30 PM, E=High, C=Low, A=Medium → H=Medium	T=12:15 PM - 5:30 PM, E=High, C=Low, A=Medium → H=Medium
T=12:15 PM - 5:30 PM, E=Medium, C=High, H=Low → A=High	T=12:15 PM - 5:30 PM, E=Medium, C=High, H=Low → A=High
T=12:15 PM - 5:30 PM, E=Medium, C=Low, H=Low → A=High	T=12:15 PM - 5:30 PM, E=Medium, C=Low, H=Low → A=High
T=5:45 PM - 11 PM, E=High, C=Low, A=Medium → H=Medium	T=5:45 PM - 11 PM, E=High, C=Low, A=Medium → H=Medium
T=11:15 PM - 6 AM, E=Low, H=High, A=Low → C=Low	
T=11:15 PM - 6 AM, E=Low, H=High, A=Medium → C=Low	
T=11:15 PM - 6 AM, E=Low, H=Low, A=High → C=Low	
T=11:15 PM - 6 AM, E=Low, H=Low, A=Medium → C=Low	
T=11:15 PM - 6 AM, E=Low, H=Medium, A=High → C=Low	
T=11:15 PM - 6 AM, E=Low, H=Medium, A=Lo → C=Low	
T=11:15 PM - 6 AM, E=Low, H=Medium, A=Medium → C=Low	
T=11:15 PM - 6 AM, E=Medium, H=Medium, A=Low → C=Low	
T=11:15 PM - 6 AM, E=Medium, H=Medium, A=Medium → C=Low	

Continued on next page

Table 2.4: Sensitivity Analysis of β (Continued)

T=11:15 PM - 6 AM, C=Low, H=High, A=Medium → E=Low	
T=12:15 PM - 5:30 PM, E=High, H=Medium, A=Medium → C=Low	
T=12:15 PM - 5:30 PM, E=Low, C=Low, H=Low → A=High	
T=12:15 PM - 5:30 PM, E=Low, H=Low, A=High → C=Low	
T=12:15 PM - 5:30 PM, E=Low, H=Medium, A=High → C=Low	
T=12:15 PM - 5:30 PM, E=Low, H=Medium, A=Medium → C=Low	
T=12:15 PM - 5:30 PM, E=Medium, H=High, A=Low → C=Low	
T=12:15 PM - 5:30 PM, E=Medium, H=High, A=Medium → C=Low	
T=12:15 PM - 5:30 PM, E=Medium, H=Medium, A=Low → C=Low	
T=12:15 PM - 5:30 PM, E=Medium, H=Medium, A=Medium → C=Low	
T=12:15 PM - 5:30 PM, C=Low, H=High, A=Low → E=Medium	
T=12:15 PM - 5:30 PM, C=Low, H=Medium, A=Low → E=Medium	
T=5:45 PM - 11 PM, E=High, C=Low, H=Low → A=High	
T=5:45 PM - 11 PM, E=High, C=Low, A=Low → H=Medium	
T=5:45 PM - 11 PM, E=High, H=Medium, A=Low → C=Low	
T=5:45 PM - 11 PM, E=Low, H=Low, A=High → C=Low	
T=5:45 PM - 11 PM, E=Low, H=Medium, A=High → C=Low	
T=5:45 PM - 11 PM, E=Low, H=Medium, A=Medium → C=Low	
T=5:45 PM - 11 PM, E=Medium, C=High, H=Low → A=High	
T=5:45 PM - 11 PM, E=Medium, H=High, A=Low → C=Low	
T=5:45 PM - 11 PM, E=Medium, H=High, A=Medium → C=Low	
T=5:45 PM - 11 PM, E=Medium, H=Medium, A=Low → C=Low	

Continued on next page

Table 2.4: Sensitivity Analysis of β (Continued)

T=5:45 PM - 11 PM, E=Medium, H=Medium, A=Medium → C=Low	
T=6:15 AM - 8 AM, E=Low, H=High, A=Low → C=Low	
T=6:15 AM - 8 AM, E=Low, H=Medium, A=Low → C=Low	
T=6:15 AM - 8 AM, E=Low, H=Medium, A=Medium → C=Low	
T=6:15 AM - 8 AM, E=Medium, C=Low, A=High → H=Medium	
T=6:15 AM - 8 AM, E=Medium, H=High, A=Low → C=Low	
T=6:15 AM - 8 AM, E=Medium, H=Medium, A=High → C=Low	
T=6:15 AM - 8 AM, E=Medium, H=Medium, A=Low → C=Low	
T=6:15 AM - 8 AM, E=Medium, H=Medium, A=Medium → C=Low	
T=6:15 AM - 8 AM, C=Low, H=Medium, A=High → E=Medium	
T=8:15 AM - 12 PM, E=Low, H=Medium, A=High → C=Low	
T=8:15 AM - 12 PM, E=Low, H=Medium, A=Low → C=Low	
T=8:15 AM - 12 PM, E=Low, H=Medium, A=Medium → C=Low	
T=8:15 AM - 12 PM, E=Medium, H=High, A=Low → C=Low	
T=8:15 AM - 12 PM, E=Medium, H=High, A=Medium → C=Low	
T=8:15 AM - 12 PM, E=Medium, H=Medium, A=Low → C=Low	
T=8:15 AM - 12 PM, E=Medium, H=Medium, A=Medium → C=Low	
E=High, C=High, H=Medium, A=High → T=5:45 PM - 11 PM	
% of covered instances: 2343/2976 = 78.73%	% of covered instances: 312/2976 = 10.48%

While β sets a lower bound of occurrence frequency in an association, γ sets an upper bound to the percentage of AV frequency count. Recall that the purpose of setting γ is to remove associations with AV being present in too many instances of the

data set (too common in the data set), a lower value of γ has a stronger restriction on the number of times that AV can be found in the data set. Consequently, less associations will be established for a lower value of γ. The last experiment in this section compares the results of settings $\alpha = 1$, $\beta = 50$, $\gamma = 0.95$ vs. $\alpha = 1$. $\beta = 50$, $\gamma = 1$.

As shown in Table 2.5, setting $\gamma = 1$ gives more established associations than $\gamma = 0.95$. The last eleven associations in the left column all have AV equal to C=Low which can be found in 2848 out of 2976 (= 96%) instances in the dataset.

Table 2.5: Sensitivity Analysis of γ

Established Associations	**% of AV in data**	**Established Associations**	**% of AV in data**
T=12:15 PM - 5:30 PM, E=High, C=Low, A=Medium → H=Medium	0.7	T=12:15 PM - 5:30 PM, E=High, C=Low, A=Medium → H=Medium	0.7
T=12:15 PM - 5:30 PM, E=Medium, C=High, H=Low → A=High	0.34	T=12:15 PM - 5:30 PM, E=Medium, C=High, H=Low → A=High	0.34
T=12:15 PM - 5:30 PM, E=Medium, C=Low, H=Low → A=High	0.34	T=12:15 PM - 5:30 PM, E=Medium, C=Low, H=Low → A=High	0.34
T=5:45 PM - 11 PM, E=High, C=Low, A=Medium → H=Medium	0.7	T=5:45 PM - 11 PM, E=High, C=Low, A=Medium → H=Medium	0.7
T=11:15 PM - 6 AM, E=Low, H=High, A=Low → C=Low	0.96		
T=11:15 PM - 6 AM, E=Low, H=Low, A=High → C=Low	0.96		
T=11:15 PM - 6 AM, E=Low, H=Medium, A=Low → C=Low	0.96		
T=11:15 PM - 6 AM, E=Low, H=Medium, A=Medium → C=Low	0.96		
T=12:15 PM - 5:30 PM, E=High, H=Medium, A=Medium → C=Low	0.96		

Continued on next page

Table 2.5: Sensitivity Analysis of γ (Continued)

T=12:15 PM - 5:30 PM, E=Low, H=Medium, A=High → C=Low	0.96		
T=12:15 PM - 5:30 PM, E=Medium, H=Medium, A=Medium → C=Low	0.96		
T=5:45 PM - 11 PM, E=Medium, H=Medium, A=Low → C=Low	0.96		
T=5:45 PM - 11 PM, E=Medium, H=Medium, A=Medium → C=Low	0.96		
T=6:15 AM - 8 AM, E=Low, H=Medium, A=Low → C=Low	0.96		
T=8:15 AM - 12 PM, E=Medium, H=Medium, A=Medium → C=Low	0.96		
% of covered instances: 2343/2976 = 78.73%		**% of covered instances: 312/2976 = 10.48%**	

2.3 Verification of PVAD: Comparison with Association Rule Results

As discussed in Section 1.2, Association Rule Mining is the most similar technique to the PVAD algorithm. The technique first uses the Aprori algorithm to determine frequent item sets that satisfy the minimum support (Ye, 2013a, 2003). Then each frequent item set is broken up into all possible combinations of association rules which are then evaluated to see if any of them satisfy the minimum confidence threshold.

The main shortcoming of Association Rule mining is its inability to discover all associations. While that deficiency is only briefly mentioned in Section 1.2, it is further investigated in this section. Both techniques are applied to the chemical data set in Table 2.2. The respective values of minimum support and confidence threshold are 50 and 0.8. For comparison purposes, the PVAD parameters α, β and γ are set to be 0.8, 30, and 1 respectively. The reason for setting β to be lower than the minimum support is that we only want to look at associations with co-occurrence frequency larger than or equal to 50 but not the occurrence frequency.

Restricting β=50 will lead to an early drop of candidate associations which will then return fewer associations. Nonetheless, we will only consider associations with co-occurrence frequency ≥ 50 when comparing the results. As all the association rules derived from Association Rule Mining are also discovered in the PVAD algorithm, the common results are displayed in Table 2.6. Table 2.7 shows associations that are only found by PVAD and this finding supports our claim at the start of this section that Association Rule Mining cannot discover all the associations. From the viewpoint of covered instances, the association rules altogether cover 2032/2976 = 68.28% of data set instances. Not only those 2032 instances, associations of PVAD also cover other instances and its % of covered instances is 2976/2976 = 100% meaning that every instance in the data set is covered by at least one PVAD association.

Table 2.6: Common Results Found by Association Rule Mining and PVAD.

Established Associations $\alpha = 0.8, \beta = 30, \gamma = 1$ and Co-Occ Freq ≥ 50)	***cr***
2-to-3	
T=12:15 PM - 5:30 PM, C=High → E=Medium, H=Low, A=High	0.9
4-to-1	
T=11:15 PM - 6 AM, E=Low, C=Low, A=Medium → H=Medium	0.81
T=11:15 PM - 6 AM, E=Low, H=High, A=Low → C=Low	1
T=11:15 PM - 6 AM, E=Low, H=Low, A=High → C=Low	1
T=11:15 PM - 6 AM, E=Low, H=Medium, A=Low → C=Low	1
T=11:15 PM - 6 AM, E=Low, H=Medium, A=Medium → C=Low	1
T=11:15 PM - 6 AM, C=Low, H=High, A=Low → E=Low	0.99
T=11:15 PM - 6 AM, C=Low, H=Low, A=High → E=Low	0.99
T=11:15 PM - 6 AM, C=Low, H=Medium, A=Low → E=Low	0.95
T=11:15 PM - 6 AM, C=Low, H=Medium, A=Medium → E=Low	0.95
T=12:15 PM - 5:30 PM, E=High, C=Low, H=Medium → A=Medium	0.86
T=12:15 PM - 5:30 PM, E=High, C=Low, A=Medium → H=Medium	1
T=12:15 PM - 5:30 PM, E=High, H=Medium, A=Medium → C=Low	1
T=12:15 PM - 5:30 PM, E=Low, C=Low, H=Medium → A=High	0.84
T=12:15 PM - 5:30 PM, E=Low, H=Medium, A=High → C=Low	1
T=12:15 PM - 5:30 PM, E=Medium, C=High, H=Low → A=High	1

Continued on next page

Table 2.6: Common Results Found by Association Rule Mining and PVAD. (Continued)

T=12:15 PM - 5:30 PM, E=Medium, C=High, A=High → H=Low	0.95
T=12:15 PM - 5:30 PM, E=Medium, C=Low, H=Low → A=High	1
T=12:15 PM - 5:30 PM, E=Medium, C=Low, A=Medium → H=Medium	0.83
T=12:15 PM - 5:30 PM, E=Medium, H=Medium, A=High → C=Low	0.98
T=12:15 PM - 5:30 PM, E=Medium, H=Medium, A=Medium → C=Low	1
T=12:15 PM - 5:30 PM, C=High, H=Low, A=High → E=Medium	0.95
T=5:45 PM - 11 PM, E=High, C=Low, A=Medium → H=Medium	1
T=5:45 PM - 11 PM, E=High, H=Medium, A=Medium → C=Low	0.96
T=5:45 PM - 11 PM, E=Medium, C=Low, A=High → H=Medium	0.95
T=5:45 PM - 11 PM, E=Medium, C=Low, A=Medium → H=Medium	0.81
T=5:45 PM - 11 PM, E=Medium, H=Medium, A=High → C=Low	0.92
T=5:45 PM - 11 PM, E=Medium, H=Medium, A=Low → C=Low	1
T=5:45 PM - 11 PM, E=Medium, H=Medium, A=Medium → C=Low	1
T=6:15 AM - 8 AM, E=Low, H=Medium, A=Low → C=Low	1
T=8:15 AM - 12 PM, E=Medium, C=Low, A=High → H=Medium	0.84
T=8:15 AM - 12 PM, E=Medium, C=Low, A=Medium → H=Medium	0.84
T=8:15 AM - 12 PM, E=Medium, H=Medium, A=High → C=Low	0.96
T=8:15 AM - 12 PM, E=Medium, H=Medium, A=Medium → C=Low	1
T=8:15 AM - 12 PM, C=Low, H=Medium, A=High → E=Medium	0.91

Table 2.7: Associations only found by PVAD algorithm

1-to-1	***cr***	**3-to-1**	***cr***
T=11:15 PM - 6 AM → E=Low	0.96	T=11:15 PM - 6 AM, E=Low, H=High → C=Low	1
T=11:15 PM - 6 AM → C=Low	1	T=11:15 PM - 6 AM, E=Low, H=Low → C=Low	1
T=12:15 PM - 5:30 PM → C=Low	0.91	T=11:15 PM - 6 AM, E=Low, H=Medium → C=Low	1
T=5:45 PM - 11 PM → C=Low	0.92	T=11:15 PM - 6 AM, E=Low, A=High → C=Low	1
T=6:15 AM - 8 AM → C=Low	1	T=11:15 PM - 6 AM, E=Low, A=Low → C=Low	1
T=8:15 AM - 12 PM → E=Medium	0.83	T=11:15 PM - 6 AM, E=Low, A=Medium → C=Low	1
T=8:15 AM - 12 PM → C=Low	0.98	T=11:15 PM - 6 AM, E=Low, A=Medium → H=Medium	0.81

Continued on next page

Table 2.7: Associations only found by PVAD algorithm (Continued)

E=High → C=Low	0.91	T=11:15 PM - 6 AM, C=Low, H=High → E=Low	0.99
E=High → H=Medium	0.89	T=11:15 PM - 6 AM, C=Low, H=Low → E=Low	0.98
E=Low → C=Low	1	T=11:15 PM - 6 AM, C=Low, H=Medium → E=Low	0.94
E=Medium → C=Low	0.93	T=11:15 PM - 6 AM, C=Low, A=High → E=Low	0.94
C=High → E=Medium	0.8	T=11:15 PM - 6 AM, C=Low, A=Low → E=Low	0.97
C=High → A=High	0.96	T=11:15 PM - 6 AM, C=Low, A=Medium → E=Low	0.96
H=High → C=Low	1	T=11:15 PM - 6 AM, C=Low, A=Medium → H=Medium	0.82
H=Low → A=High	0.88	T=11:15 PM - 6 AM, H=High, A=Low → E=Low	0.99
H=Medium → C=Low	0.98	T=11:15 PM - 6 AM, H=High, A=Low → C=Low	1
A=High → C=Low	0.88	T=11:15 PM - 6 AM, H=Low, A=High → E=Low	0.99
A=Low → C=Low	1	T=11:15 PM - 6 AM, H=Low, A=High → C=Low	1
A=Medium → C=Low	1	T=11:15 PM - 6 AM, H=Medium, A=Low → E=Low	0.95
A=Medium- → H=Medium	0.84	T=11:15 PM - 6 AM, H=Medium, A=Low → C=Low	1
1-to-2	***cr***	T=11:15 PM - 6 AM, H=Medium, A=Medium → E=Low	0.95
T=11:15 PM - 6 AM → E=Low, C=Low	0.96	T=11:15 PM - 6 AM, H=Medium, A=Medium → C=Low	1
T=8:15 AM - 12 PM → E=Medium, C=Low	0.81	T=12:15 PM - 5:30 PM, E=High, C=Low → H=Medium	0.88
E=High → C=Low, H=Medium	0.83	T=12:15 PM - 5:30 PM, E=High, H=Medium → C=Low	1
C=High-E=Medium, A=High	0.8	T=12:15 PM - 5:30 PM, E=High, H=Medium → A=Medium	0.86
A=Medium-C=Low, H=Medium	0.84	T=12:15 PM - 5:30 PM, E=High, A=Medium → C=Low	1
1-to-3, 1-to- 4	***cr***	T=12:15 PM - 5:30 PM, E=High, A=Medium → H=Medium	1
None		T=12:15 PM - 5:30 PM, E=Low, C=Low → A=High	0.87
2-to-1	***cr***	T=12:15 PM - 5:30 PM, E=Low, H=Medium → C=Low	1
T=11:15 PM - 6 AM, E=Low → C=Low	1	T=12:15 PM - 5:30 PM, E=Low, H=Medium → A=High	0.84

Continued on next page

Table 2.7: Associations only found by PVAD algorithm (Continued)

T=11:15 PM - 6 AM, C=Low → E=Low	0.96	T=12:15 PM - 5:30 PM, E=Low, A=High → C=Low	1
T=11:15 PM - 6 AM, H=High → E=Low	0.99	T=12:15 PM - 5:30 PM, E=Medium, C=High → H=Low	0.95
T=11:15 PM - 6 AM, H=High → C=Low	1	T=12:15 PM - 5:30 PM, E=Medium, C=High → A=High	1
T=11:15 PM - 6 AM, H=Low → E=Low	0.98	T=12:15 PM - 5:30 PM, E=Medium, H=Low → A=High	1
T=11:15 PM - 6 AM, H=Low → C=Low	1	T=12:15 PM - 5:30 PM, E=Medium, H=Medium → C=Low	0.99
T=11:15 PM - 6 AM, H=Medium → E=Low	0.94	T=12:15 PM - 5:30 PM, E=Medium, A=High → C=Low	0.84
T=11:15 PM - 6 AM, H=Medium → C=Low	1	T=12:15 PM - 5:30 PM, E=Medium, A=Medium- → C=Low	1
T=11:15 PM - 6 AM, A=High → E=Low	0.94	T=12:15 PM - 5:30 PM, E=Medium, A=Medium-H=Medium	0.83
T=11:15 PM - 6 AM, A=High → C=Low	1	T=12:15 PM - 5:30 PM, C=High, H=Low → E=Medium	0.95
T=11:15 PM - 6 AM, A=Low → E=Low	0.97	T=12:15 PM - 5:30 PM, C=High, H=Low → A=High	1
T=11:15 PM - 6 AM, A=Low → C=Low	1	T=12:15 PM - 5:30 PM, C=High, A=High → E=Medium	0.95
T=11:15 PM - 6 AM, A=Medium → E=Low	0.96	T=12:15 PM - 5:30 PM, C=High, A=High → H=Low	0.95
T=11:15 PM - 6 AM, A=Medium → C=Low	1	T=12:15 PM - 5:30 PM, C=Low, H=Low → A=High	1
T=11:15 PM - 6 AM, A=Medium → H=Medium	0.82	T=12:15 PM - 5:30 PM, C=Low, A=Medium → H=Medium	0.89
T=12:15 PM - 5:30 PM, E=High → C=Low	0.96	T=12:15 PM - 5:30 PM, H=Medium, A=High → C=Low	0.99
T=12:15 PM - 5:30 PM, E=High → H=Medium	0.84	T=12:15 PM - 5:30 PM, H=Medium, A=Medium → C=Low	1
T=12:15 PM - 5:30 PM, E=Low → C=Low	1	T=5:45 PM - 11 PM, E=High, C=Low-H=Medium	0.92
T=12:15 PM - 5:30 PM, E=Low → A=High	0.87	T=5:45 PM - 11 PM, E=High, H=Medium → C=Low	0.9
T=12:15 PM - 5:30 PM, E=Medium → C=Low	0.88	T=5:45 PM - 11 PM, E=High, A=Medium → C=Low	0.96
T=12:15 PM - 5:30 PM, C=High → E=Medium	0.95	T=5:45 PM - 11 PM, E=High, A=Medium → H=Medium	1
T=12:15 PM - 5:30 PM, C=High → H=Low	0.95	T=5:45 PM - 11 PM, E=Medium, C=Low → H=Medium	0.82
T=12:15 PM - 5:30 PM, C=High → A=High	1	T=5:45 PM - 11 PM, E=Medium, H=High → C=Low	1

Continued on next page

Table 2.7: Associations only found by PVAD algorithm (Continued)

T=12:15 PM - 5:30 PM, H=Low → A=High	1	T=5:45 PM - 11 PM, E=Medium, H=Medium → C=Low	0.98
T=12:15 PM - 5:30 PM, H=Medium → C=Low	0.99	T=5:45 PM - 11 PM, E=Medium, A=Low → C=Low	1
T=12:15 PM - 5:30 PM, A=High → C=Low	0.87	T=5:45 PM - 11 PM, E=Medium, A=Medium → C=Low	1
T=12:15 PM - 5:30 PM, A=Medium → C=Low	1	T=5:45 PM - 11 PM, E=Medium, A=Medium → H=Medium	0.81
T=12:15 PM - 5:30 PM, A=Medium → H=Medium	0.89	T=5:45 PM - 11 PM, C=Low, H=High → E=Medium	0.82
T=5:45 PM - 11 PM, E=High → C=Low	0.89	T=5:45 PM - 11 PM, C=Low, A=Medium → H=Medium	0.84
T=5:45 PM - 11 PM, E=High → H=Medium	0.91	T=5:45 PM - 11 PM, H=High, A=Medium → C=Low	1
T=5:45 PM - 11 PM, E=Low → C=Low	1	T=5:45 PM - 11 PM, H=Medium, A=High → C=Low	0.86
T=5:45 PM - 11 PM, E=Medium → C=Low	0.92	T=5:45 PM - 11 PM, H=Medium, A=Low → C=Low	1
T=5:45 PM - 11 PM, C=High → A=High	0.91	T=5:45 PM - 11 PM, H=Medium, A=Medium → C=Low	0.98
T=5:45 PM - 11 PM, C=Low → H=Medium	0.81	T=6:15 AM - 8 AM, E=Low, H=Medium → C=Low	1
T=5:45 PM - 11 PM, H=High → E=Medium	0.82	T=6:15 AM - 8 AM, E=Low, A=Low → C=Low	1
T=5:45 PM - 11 PM, H=High → C=Low	1	T=6:15 AM - 8 AM, E=Medium, H=High-C=Low	1
T=5:45 PM - 11 PM, H=Medium → C=Low	0.95	T=6:15 AM - 8 AM, E=Medium, H=Medium-C=Low	1
T=5:45 PM - 11 PM, A=Low → C=Low	1	T=6:15 AM - 8 AM, E=Medium, A=Low → C=Low	1
T=5:45 PM - 11 PM, A=Medium → C=Low	0.99	T=6:15 AM - 8 AM, E=Medium, A=Medium → C=Low	1
T=5:45 PM - 11 PM, A=Medium → H=Medium	0.84	T=6:15 AM - 8 AM, C=Low, H=High → A=Low	0.87
T=6:15 AM - 8 AM, E=Low-C=Low	1	T=6:15 AM - 8 AM, C=Low, A=Medium → H=Medium	0.87
T=6:15 AM - 8 AM, E=Medium → C=Low	1	T=6:15 AM - 8 AM, H=High, A=Low → C=Low	1
T=6:15 AM - 8 AM, H=High → C=Low	1	T=6:15 AM - 8 AM, H=Medium, A=Low → C=Low	1
T=6:15 AM - 8 AM, H=High → A=Low	0.87	T=6:15 AM - 8 AM, H=Medium, A=Medium → C=Low	1
T=6:15 AM - 8 AM, H=Medium → C=Low	1	T=8:15 AM - 12 PM, E=Low, H=Medium → C=Low	1
T=6:15 AM - 8 AM, A=Low → C=Low	1	T=8:15 AM - 12 PM, E=Medium, H=High → C=Low	1

Continued on next page

Table 2.7: Associations only found by PVAD algorithm (Continued)

T=6:15 AM - 8 AM, A=Medium → C=Low	1	T=8:15 AM - 12 PM, E=Medium, H=Medium → C=Low	0.98
T=6:15 AM - 8 AM, A=Medium → H=Medium	0.87	T=8:15 AM - 12 PM, E=Medium, A=High → C=Low	0.94
T=8:15 AM - 12 PM, E=Low → C=Low	1	T=8:15 AM - 12 PM, E=Medium, A=High → H=Medium	0.82
T=8:15 AM - 12 PM, E=Medium → C=Low	0.97	T=8:15 AM - 12 PM, E=Medium, A=Low → C=Low	1
T=8:15 AM - 12 PM, C=Low → E=Medium	0.83	T=8:15 AM - 12 PM, E=Medium, A=Medium → C=Low	1
T=8:15 AM - 12 PM, H=High → C=Low	1	T=8:15 AM - 12 PM, E=Medium, A=Medium → 4=Medium	0.84
T=8:15 AM - 12 PM, H=Medium → E=Medium	0.84	T=8:15 AM - 12 PM, C=Low, H=Medium → E=Medium	0.84
T=8:15 AM - 12 PM, H=Medium → C=Low	0.98	T=8:15 AM - 12 PM, C=Low, A=High → E=Medium	0.89
T=8:15 AM - 12 PM, A=High → E=Medium	0.89	T=8:15 AM - 12 PM, C=Low, A=High-H=Medium	0.82
T=8:15 AM - 12 PM, A=High → C=Low	0.94	T=8:15 AM - 12 PM, C=Low, A=Medium → E=Medium	0.83
T=8:15 AM - 12 PM, A=High → H=Medium	0.8	T=8:15 AM - 12 PM, C=Low, A=Medium → H=Medium	0.84
T=8:15 AM - 12 PM, A=Low → C=Low	1	T=8:15 AM - 12 PM, H=Medium, A=High → E=Medium	0.92
T=8:15 AM - 12 PM, A=Medium → E=Medium	0.83	T=8:15 AM - 12 PM, H=Medium, A=High → C=Low	0.96
T=8:15 AM - 12 PM, A=Medium → C=Low	1	T=8:15 AM - 12 PM, H=Medium, A=Low → C=Low	1
T=8:15 AM - 12 PM, A=Medium → H=Medium	0.84	T=8:15 AM - 12 PM, H=Medium, A=Medium → E=Medium	0.82
E=High, C=Low → H=Medium	0.91	T=8:15 AM - 12 PM, H=Medium, A=Medium → C=Low	1
E=High, H=Medium → C=Low	0.93	E=High, C=Low, A=Medium → H=Medium	1
E=High, A=Medium → C=Low	0.97	E=High, H=Medium, A=Medium → C=Low	0.97
E=High, A=Medium → H=Medium	1	E=Low, H=High, A=Low → 3=Low	1
E=Low, H=High- → C=Low	1	E=Low, H=High, A=Medium → C=Low	1
E=Low, H=Low → C=Low	1	E=Low, H=Low, A=High → C=Low	1
E=Low, H=Medium → C=Low	1	E=Low, H=Medium, A=High → C=Low	1
E=Low, A=High → C=Low	1	E=Low, H=Medium, A=Low → C=Low	1

Continued on next page

Table 2.7: Associations only found by PVAD algorithm (Continued)

E=Low, A=Low → C=Low	1	E=Low, H=Medium, A=Medium → C=Low	1
E=Low, A=Medium → C=Low	1	E=Medium, C=High, H=Low → A=High	1
E=Medium, C=High → H=Low	0.84	E=Medium, C=High, A=High → H=Low	0.84
E=Medium, C=High → A=High	1	E=Medium, C=Low, H=Low → A=High	0.89
E=Medium, H=High → C=Low	1	E=Medium, C=Low, A=Medium → H=Medium	0.84
E=Medium, H=Low → A=High	0.93	E=Medium, H=High, A=Low → C=Low	1
E=Medium, H=Medium → C=Low	0.99	E=Medium, H=High, A=Medium → C=Low	1
E=Medium, A=High → C=Low	0.85	E=Medium, H=Medium, A=High → C=Low	0.97
E=Medium, A=Low → C=Low	1	E=Medium, H=Medium, A=Low → C=Low	1
E=Medium, A=Medium → C=Low	1	E=Medium, H=Medium, A=Medium → C=Low	1
E=Medium, A=Medium → H=Medium	0.84	C=High, H=Low, A=High → E=Medium	0.93
C=High, H=Low → E=Medium	0.93	**3-to-2**	***cr***
C=High, H=Low → A=High	1	T=11:15 PM - 6 AM, E=Low, A=Medium → C=Low, H=Medium	0.81
C=High, A=High → E=Medium	0.84	T=11:15 PM - 6 AM, H=High, A=Low → E=Low, C=Low	0.99
C=Low, H=Low → A=High	0.84	T=11:15 PM - 6 AM, H=Low, A=High → E=Low, C=Low	0.99
C=Low, A=Medium → H=Medium	0.84	T=11:15 PM - 6 AM, H=Medium, A=Low → E=Low, C=Low	0.95
H=High, A=Low → C=Low	1	T=11:15 PM - 6 AM, H=Medium, A=Medium → E=Low, C=Low	0.95
H=High, A=Medium → C=Low	1	T=12:15 PM - 5:30 PM, E=High, H=Medium → C=Low, A=Medium	0.86
H=Medium, A=High → C=Low	0.95	T=12:15 PM - 5:30 PM, E=High, A=Medium → C=Low, H=Medium	1
H=Medium, A=Low → C=Low	1	T=12:15 PM - 5:30 PM, E=Low, H=Medium → C=Low, A=High	0.84
H=Medium, A=Medium-C=Low	0.99	T=12:15 PM - 5:30 PM, E=Medium, C=High → H=Low, A=High	0.95
2-to-2	***cr***	T=12:15 PM - 5:30 PM, E=Medium, A=Medium → C=Low, H=Medium	0.83

Continued on next page

Table 2.7: Associations only found by PVAD algorithm (Continued)

T=11:15 PM - 6 AM, H=High → E=Low, C=Low	0.99	T=12:15 PM - 5:30 PM, C=High, H=Low → E=Medium, A=High	0.95
T=11:15 PM - 6 AM, H=Low → E=Low, C=Low	0.98	T=12:15 PM - 5:30 PM, C=High, A=High → E=Medium, H=Low	0.9
T=11:15 PM - 6 AM, H=Medium → E=Low, C=Low	0.94	T=5:45 PM - 11 PM, E=High, A=Medium → C=Low, H=Medium	0.96
T=11:15 PM - 6 AM, A=High → E=Low, C=Low	0.94	T=5:45 PM - 11 PM, E=Medium, A=Medium → C=Low, H=Medium	0.81
T=11:15 PM - 6 AM, A=Low → E=Low, C=Low	0.97	T=8:15 AM - 12 PM, E=Medium, A=Medium → C=Low, H=Medium	0.84
T=11:15 PM - 6 AM, A=Medium → E=Low, C=Low	0.96	T=8:15 AM - 12 PM, H=Medium, A=High → E=Medium, C=Low	0.88
T=11:15 PM - 6 AM, A=Medium → C=Low, H=Medium	0.82	T=8:15 AM - 12 PM, H=Medium, A=Medium → E=Medium, C=Low	0.82
T=12:15 PM - 5:30 PM, E=High → C=Low, H=Medium	0.84		
T=12:15 PM - 5:30 PM, E=Low → C=Low, A=High	0.87		
T=12:15 PM - 5:30 PM, C=High → E=Medium, H=Low	0.9		
T=12:15 PM - 5:30 PM, C=High → E=Medium, A=High	0.95		
T=12:15 PM - 5:30 PM, C=High → H=Low, A=High	0.95		
T=12:15 PM - 5:30 PM, A=Medium → C=Low, H=Medium	0.89		
T=5:45 PM - 11 PM, E=High → C=Low, H=Medium	0.82		
T=5:45 PM - 11 PM, H=High → E=Medium, C=Low	0.82		
T=5:45 PM - 11 PM, A=Medium → C=Low, H=Medium	0.83		
T=6:15 AM - 8 AM, H=High → C=Low, A=Low	0.87		
T=6:15 AM - 8 AM, A=Medium → C=Low, H=Medium	0.87		

Continued on next page

Table 2.7: Associations only found by PVAD algorithm (Continued)

T=8:15 AM - 12 PM, H=Medium → E=Medium, C=Low	0.83		
T=8:15 AM - 12 PM, A=High → E=Medium, C=Low	0.84		
T=8:15 AM - 12 PM, A=Medium → E=Medium, C=Low	0.83		
T=8:15 AM - 12 PM, A=Medium → C=Low, H=Medium	0.84		
E=High, A=Medium → C=Low, H=Medium	0.97		
E=Medium, C=High → H=Low, A=High	0.84		
E=Medium, A=Medium → C=Low, H=Medium	0.84		
C=High, H=Low → E=Medium, A=High	0.93		

2.4 Verification of PVAD: Comparison with Decision Tree Results

Decision tree is a data mining technique to learn decision rules that express relations of the dependent variable y with independent variables x in a directed and acyclic graph (Ye, 2013a, 2003). The software, Weka, was used to construct decision trees of the energy consumption system data, To construct a decision tree in Weka, there are different algorithms such as ID3 (Quinlan, 1986) and J48 (Quinlan, 2014). The latter is an extended version of ID3 with additional features such as dealing with missing values and continuous attribute value ranges. It also addresses the over-fitting problem that decision trees are prone to by pruning. The pruning process requires the computation of the expected error rate. If the error rate of a subtree is greater than that of a leaf node, a subtree is pruned and replaced by the leaf node.

In our research, ID3 was used for the comparison with the PVAD algorithm because ID3 produces comparable results with associations produced by the PVAD

Table 2.8: Decision rules from the ID3 Tree with Air Temperature as the target variable same as PVAD association rules.

#	Decision rules that appear the same as PVAD association rules
1	H=Medium, E=High, T=12:15 PM to 5:30 PM → A=Medium
2	H=Medium, E=Low, T=12:15 PM to 5:30 PM → A=High
3	H=Low, T=12:15 PM to 5:30 PM → A=High
4	H=High, E=Medium, T=6:15 AM to 8 AM → A=Low

algorithm. Leaf nodes produced by ID3 are pure in that the class labels of instances are the same in each leaf node. The purity of the leaf node corresponds to AV in associations from the PVAD algorithm having the same variable value. The PVAD algorithm produces all associations up to N-to-1 associations, where N+1 is the number of variables. In other words, the PVAD algorithm can generate the longest CVs and find the AV that they are associated with. The combination of CVs corresponds to the path from the root of a decision tree down to a leaf node.

Because the decision tree technique requires the identification of one dependent variable (the target variable) and independent variables (attribute variables) for each decision tree, five decision trees need to be constructed for each of the five variables as the dependent variable. Tables 2.8-2.11 list decision rules produced by one of the five ID3 trees.

Table 2.9: Decision rules from the ID3 Tree with Air Temperature = Low as the target variable same as PVAD association rules.

5	H=Medium, E=High, T=11:15 PM to 6 AM → A=Low
6	H=High, E=Low, T=11:15 PM to 6 AM → A=Low
7	H=High, E=Medium, T=11:15 PM to 6 AM →A=Low
8	H=High, E=Medium, T=12:15 PM to 5:30 PM → A=Low
9	H=High, E=Low, T=6:15 AM to 8 AM → A=Low
10	H=Medium, E=Low, T=6:15 AM to 8 AM → A=Low
11	H=High, E=Low, T=8:15 AM to 12 PM → A=Low
12	H=High, E=Medium, T=8:15 AM to 12 PM → A=Low
13	H=Low, C=Low, E=Medium, T=5:45 PM to 11 PM → A=Low

Table 2.10: Decision rules from the ID3 Tree with Air Temperature = Medium as the target variable.

14	H=Low, T=6:15 AM to 8 AM → A=Medium
15	H=Medium, E=Low, T=11:15 PM to 6 AM → A=Medium
16	H=Low, E=Medium, T=11:15 PM to 6 AM → A=Medium
17	H=Medium, E=Medium, T=11:15 PM to 6 AM → A=Medium
18	H=High, E=Low, T=12:15 PM to 5:30 PM → A=Medium
19	H=High, E=Low, T=5:45 PM to 11 PM → A=Medium
20	H=High, E=Medium, T=5:45 PM to 11 PM → A=Medium
21	H=Medium, E=Medium, T=6:15 AM to 8 AM → A=Medium
22	H=Medium, E=High, T=8:15 AM to 12 PM → =Medium
23	H=Medium, E=Low, T=8:15 AM to 12 PM → A=Medium
24	H=Medium, C=Low, E=High, T=5:45 PM to 11 PM → A=Medium
25	H=Medium, C=Low, E=Medium, T=5:45 PM to 11 PM → A=Medium
26	H=Medium, C=Low, E=Medium, T=8:15 AM to 12 PM → A=Medium

Table 2.11: Decision rules from the ID3 Tree with Air Temperature = High as the target variable.

27	H=Low, E=Low, T=11:15 PM to 6 AM → A=High
28	H=Low, E=High, T=5:45 PM to 11 PM → A=High
29	H=Low, E=Low, T=5:45 PM to 11 PM → A=High
30	H=Low, E=Low, T=8:15 AM to 12 PM → A=High
31	H=Medium, C=High, E=High, T=5:45 PM to 11 PM → A=High
32	H=Medium, C=Low, E=Low, T=5:45 PM to 11 PM → A=High
33	H=Low, C=High, E=Medium, T=5:45 PM to 11 PM → A=High
34	H=Medium, C=High, E=Medium, T=5:45 PM to 11 PM → A=High

Continued on next page

Table 2.11: Decision rules from the ID3 Tree with Air Temperature = High as the target variable. (Continued)

35	C=High, H=Low, E=Medium, T=8:15 AM to 12 PM → A=High
36	H=Medium, C=High, E=Medium, T=8:15 AM to 12 PM → A=High
37	H=Low, C=Low, E=Medium, T=8:15 AM to 12 PM → A=High
38	H=Medium, C=High, E=Medium, T=12:15 PM to 5:30 PM → A=High
39	H=Medium, C=Low, E=Medium, T=12:15 PM to 5:30 PM → A=High

Although the decision rules from the decision trees appear to have the same form as associations from the PVAD algorithm, a decision rule has a different meaning from an association from the PVAD algorithm. A decision rule derived from the root of a decision tree to a leaf node of the decision tree represents a frequent item set with instances in the leaf node having the values of the target variable and the attribute variables in the decision rule. This is why we see a path in a decision tree is also present in another tree even though different decision trees have different target variables. For example, the attribute values in E=Medium, A=High, H=Medium, C=Low, T=12:15 PM to 5:30 PM, are found in all four decision trees. Note that the energy consumption data set has only five variables. Redundant paths of different decision trees can be found more often for larger data sets with more variables. This means the waste of computation time and space and the difficulty of sorting out results from a number of decision trees. Hence, a decision rule corresponds to a frequent item set in the association rule technique, whereas an association from the PVAD algorithm corresponds to an association rule in the association rule technique. This is why there are decision rules in Tables 2.8-2.11 that are not found in associations of the PVAD algorithm because frequent item sets for those decision rules were eliminated in the process of forming associations. Hence, the PVAD algorithm has the advantage over the decision tree technique because the PVAD algorithm discovers associations rather than frequent item sets.

There is another difference between the decision tree technique and the PVAD algorithm. Each step of constructing a decision tree performs the splitting of a data subset for data homogeneity based on the comparison of splits using only one variable and its values rather than combinations of multiple variables due to the large number of combinations and the enormous computation costs. Hence, the resulting decision tree contains decision rules with the consideration of only one variable at a time and may miss decision rules that can be generated if multiple variables and their values are considered and compared at a time. However, the PVAD algorithm examines one to multiple variables at a time and does not miss any associations that exist. This can be seen from the results in Table 2.8-2.11. All the associations found have only one AV: x_2. While there are associations identified by PVAD can have two or more AVs, such as "x_1=11:15 PM - 6 AM, x_5=Medium $\rightarrow$ x_2=Low, x_3=Low" and "x_1=12:15 PM - 5:30 PM, x_3=High $\rightarrow$ x_2=Medium, x_4=Low, x_5=High", the PVAD algorithm thus has the advantage to the decision tree technique that it does not miss any established associations and using YFM1 and YFM2.

Moreover, the decision tree algorithm requires the identification of the dependent variable (the target variable) and the independent variables (the attribute variables) although there may be no prior knowledge for the identification of which variable is a dependent or independent variable. This is why five decision trees, with one decision tree taking each of the five variables as the target variable, had to be constructed for the energy consumption data. The PVAD algorithm does not require the distinction of dependent and independent variables but discovers variable value relations and the role of each variable in each variable value relation.

Furthermore, the PVAD algorithm can generate p-to-q associations with $q > 1$ that the decision tree technique cannot generate because a decision tree is constructed for only one target variable and produces only p-to-1 decision rules. Therefore, associ-

ations found by the PVAD algorithm are more comprehensive. As discussed in the previous section, the PVAD algorithm overcomes shortcomings of existing statistical analysis and data mining techniques and produces partial/full-value associations that cannot be produced from other existing techniques.

Chapter 3

DEEP LEARNING MODEL FOR KNEE POINT DETECTION ON NOISY DATA

3.1 Introduction

Researchers in various fields frequently encounter the task of identifying knees/elbows. In this context, "knees" are points where the concavity of a curve is negative (concave downward), while "elbows" are points where the concavity is positive (concave upward). Generally speaking, a knee point represents an advantageous operation point that optimizes the balance between system performance and operational costs. Therefore, a reliable and precise knee/elbow point detection method is desired as selecting the "right" operating point can lead to efficient utilization of system resources, which in turn results in cost savings and performance benefits. In the field of system behavior, a knee point is a point at which the cost of altering system parameters is no longer justified by the expected performance benefit. This concept can be observed in the Network Congestion Control problem, where an ideal sending rate is desired to ensure fair traffic share and prevent congestion. If the curve of packet delay increases significantly and then levels off, indicating there is network congestion, the protocol should halt increasing the sending rate. In the application of lithium-ion batteries, the knee point on a Capacity Fade Curve hints the beginning of lithium-ion cell degradation and the battery is approaching its End-of-Life (Neubauer and Pesaran, 2011; Williard, 2011; Yang *et al.*, 2017; Schuster *et al.*, 2015). In the application of BotNet Detection, a knee point can help identify potential controllers used by a master host to relay to bots. When it comes to clustering applications, the Elbow method is one of the most popular approaches for determining the ideal number of clusters. The

elbow point on the plot of an evaluation criteria curve, such as the within-cluster sum of squares as a function of the number of clusters, represents the ideal number of clusters. Choosing the appropriate number of clusters can help in preventing over-fitting and ensuring precise outcomes.

In the most common practice, researchers typically use a rule-of-thumbs approach. This intuitive and heuristic method involves plotting the graph and identifying the knee point(s) by visual inspection. An example of this practice can be seen in (Ye *et al.*, 2019), where a Partial-Value Association Discovery Algorithm (PVAD) was developed to discover relations in mixed-type real-world data. The first step of the PVAD algorithm requires converting the numeric values of each continuous attribute into categorical values. The technique used in the dissertation is a heuristic process that involves pinpointing the most substantial jumps in value differences between consecutive data points (known as elbow points) to form data clusters (intervals). However, this ad hoc approach has two main drawbacks: it is highly subjective and the determination of knee points is non-repeatable. Another commonly used method is to define a metric based on system-specific or operational characteristics, which requires prior knowledge. It must be pointed out that system-specific approaches are not practical in the scenario that the data set being analyzed contains attributes from various domains.

Our hypothesis regarding this knee point detection problem aligns with the authors in (Satopaa *et al.*, 2011). They state that a knee point estimation method should be: "[quote] not require tuning for a specific system or operational characteristics is applicable in a wide range of settings". It is worth noting that the first formal definition of a knee point was documented in that same paper. On top of that, we would like to make an additional assumption that the identification of a knee point should be independent of the data unit, and thus we propose a new definition of knee point

in this dissertation. For the above reasons, we are interested in developing a concise and reproducible knee point determination process. Our objective is to develop a technique that requires minimal human intervention, which can help to identify the data point(s) for grouping data values into intervals that also capture the distribution of data values.

Our main contributions are: i) we provide a novel mathematical definition of knee/elbow, ii) develop a benchmark data set that includes ground truth of knee point and synthetic samples, iii) propose a new deep learning approach that supports multiple knee point detection, and iv) compare our method with existing methods.

The rest of the dissertation is structured in the following way. We begin by reviewing related work in Section 3.2. We then provide a formal definition of knee point in discrete data in Section 3.3. Section 3.4 presents our proposed method and the architecture of our network. The details of the experiment implementation and results are described in Section 3.5.

3.2 Related Work

Various approaches (Antunes *et al.*, 2018, 2019; Salvador and Chan, 2004; Zhao *et al.*, 2008; Tolsa, 2000) have been proposed to identify knees/elbows in discrete data. In this section, we present the most commonly used approaches and compare them in Section 3.5.

L-Method

For every point on the curve except the endpoints, the *L*-Method (Salvador and Chan, 2004) selects a candidate point and fits a line from the first data point to the candidate point, and fits another line from the candidate point to the endpoint. Root Mean Squared Error (RMSE) is then computed to measure how close the fitted lines

are to data points. The candidate point with the lowest score is selected as an elbow point. However, this method performs best when the size of data points on each side of the elbow is reasonably balanced. It has the tendency to predict a larger elbow index for curves with long tails (more data points on the right side). To overcome this issue, the authors also proposed an iterative refinement method to cut the curve tail and reduce the focus region in each iteration. In each iteration, one candidate elbow point is selected each time. This process stops until the selected elbow value converges. Since this method is designed primarily for determining the ideal number of clusters in cluster analysis, it is only effective when dealing with simple concave curves or curves with limited data size nonetheless. One needs to check the curve shape beforehand and decide the method to be deployed.

Dynamic First Derivative Threshold

The Dynamic First Derivative Threshold (*DFDT*) method (Antunes *et al.*, 2018) is designed to determine the ideal number of clusters in an evaluation criteria plot. It first approximates the first derivative of a curve, followed by using a threshold algorithm (*IsoData*) (Ridler *et al.*, 1978) that computes the threshold value for separating the first derivative approximation values into higher-value and lower-value groups. The elbow point is selected as the data point whose derivative value is closest to the threshold value. One major drawback of *DFDT* is that it is prone to curve with a nearly vertically straight line at the beginning (long curve head), causing the threshold algorithm to always return a larger threshold value. This in turn misleads the method into predicting the elbow point close to the curve head. As such, the authors have incorporated an iterative refinement to the method, similar to the one in the conventional *L*-Method. Instead of removing the curve tail, *DFDT* removes a small segment from the head of the curve in each iteration, specifically the por-

tion from the origin to half of the distance from the previously selected elbow. This process is repeated until the selected elbows converge at the same point.

AL-Method

The *AL*-Method (Antunes *et al.*, 2019) is an extension of the traditional *L*-method. The method attempts to determine the point with a sharper angle as an elbow point, so it considers an additional angle score while selecting an elbow point. The angle score is computed as the square of the deviation of the angle between the fitted lines (θ_i) from 90 degrees: $|90 - \theta_i|^2$. Same as the *L*-Method, it requires using linear regression to fit two straight lines for every point on the curve, except the endpoints. The RMSEs and the angle scores are then respectively rescaled to a range of $[0, 1]$, and combined to calculate the overall score. The point with the lowest score is selected as an elbow point. The authors also deployed the same tail-cutting iterative refinement method to address curves with long heads or long tails, as in *DFDT*.

S-Method

The authors of the *AL*-Method also proposed the *S*-Method (Antunes *et al.*, 2019) as a further development in the same paper. This method fits three straight lines to a curve to handle curves with long heads or tails. The first and third lines fit for the curve head and tail respectively, while the middle fitted line captures the nature of the curve shape and thus is able to detect the elbow point. The criterion for selecting an elbow point is the weighted RMSE scores, which are weighted by the number of data points in the corresponding line segment used in curve fitting. The authors found that using linear regression to fit the points in the selected range introduces bias to the slope of the fitted line. They suggested fitting the lines by using the first and last point in the range but left this as an area for future exploration. This observation is

applicable to the AL-Method as well.

The above-mentioned methods are primarily designed to meet the needs of detecting elbows in clustering applications. The common drawback of these methods is that they are only effective for a narrow range of the x-interval (expected number of clusters). Experiments show that these methods have low accuracy when the number of expected clusters is large. Another limitation is that these methods have only been tested on curves with a single elbow point. If the data points are divided into smaller regions and these methods are called recursively and applied to those regions, non-elbow points may also be incorrectly identified as elbow points. Consequently, these methods do not perform well when used recursively. Though AL and S-Methods have demonstrated excellent performance on error curves from specific clustering algorithms. This indicates that the accuracy of the models can be affected by the underlying clustering algorithm. Therefore, we believe it is crucial to develop a method that is independent of any underlying algorithms and conduct experiments that can test the method's true ability to detect knee points. Nonetheless, both AL and S methods have a high computational cost, which can be prohibitively expensive for curves with a large number of data points.

Kneedle

Kneedle (Satopaa *et al.*, 2011) is the only algorithm that is capable of detecting multiple knees without the need for recursive calls. The algorithm works by first fitting the data to a smoothing pine, which reduces noise and in an attempt to preserve the original curve shape. The (x, y) values are then normalized into a unit square. After projecting the smoothed points to $y = x$, the method defines a unique threshold value for each local maximum point and determines those local maxima meeting certain conditions as knee points. The rationale behind this is that knees are points

further from a straight line. The threshold value is based on the distance between consecutive x-values and a user-specified sensitivity parameter ζ. A lower ζ value tends to declare a knee point more aggressively, which can increase the risk of false positives. The major weakness of the method is that the fitted smoothing spline may return data points that fall outside the original data range and return irrelevant results.

U-Net

U-Net (Ronneberger *et al.*, 2015) is a popular convolutional neural network for biomedical image segmentation. Its architecture consists of successive downsampling and upsampling layers that enable it to learn global features. The network also includes skip connections that can pass local features learned in the same level of the downsampling layer to the upsampling layer at the same level. These local features are then combined with spatial information learned through a sequence of upsampling layers to yield more precise segmentation. Because of its impressive performance in capturing both local and global contextual information of the input image, U-Net has been modified and successfully applied to other visual computing domains such as medical image reconstruction (Zhou *et al.*, 2022; Lee *et al.*, 2018; Andersson *et al.*, 2019; Ding *et al.*, 2019) and pansharpening (Yao *et al.*, 2018; Cao *et al.*, 2021).

3.3 Knee Point Definition

As in previous works (Satopaa *et al.*, 2011; Salvador and Chan, 2004; Antunes *et al.*, 2018, 2019), a mathematical definition of curvature for continuous function has been used as a foundation for knee/elbow definition. For a twice-differentiable function $f(x)$, the signed curvature of f at point $(x, f(x))$ is given by:

$$K_y(x) = \frac{y''}{\left(1 + (\,y'\,)^2\right)^{3/2}} \tag{3.1}$$

Curvature measures the amount by which the tangent vector of the curve changes as the point moves along the curve. The notion of selecting the point of minimum curvature as the knee point is well-suited to heuristics, as minimum curvature captures the exact point at which the curve reaches a peak and then stabilizes instead of continuing to increase or decrease, and as a result, can be used to identify knees. It is noteworthy to mention that in the case of single knee/elbow detection, the problem of finding knee point or elbow point is interchangeable. If a curve presents positive concavity, it can be inverted to a negative concavity curve by replacing the $\mathbf{x}$ and $\mathbf{y}$ data points with the difference of the corresponding maximum value to the original data values (i.e. replace x_i by $x_{\max} - x_i$ and y_i by $y_{\max} - y_i$), where $x_{\max}$ and $y_{\max}$ are the maximum values of x and y respectively.

However, the above curvature definition is limited to continuous functions, it is not well-defined for discrete data sets. Fitting a continuous function to a set of noisy data is one possible way to extend the definition of curvature on discrete data. Despite the difficulty of fitting, the point identified in the fitting curve may fall outside the valid data range or shift the true knee point position, leading to irrelevant or inaccurate results.

Nevertheless, our goal is to develop an algorithm that performs effectively for data sets having different ranges of values. This is important because real-world data can have a wide range of possible values, and it is crucial for our algorithm to be reliable irrespective of data magnitude. To achieve this, it is necessary to normalize the data into a unit square beforehand.

Let $D^N = \left\{(\mathbf{x}^1, \mathbf{y}^1), \cdots, (\mathbf{x}^N, \mathbf{y}^N)\right\}$ be a set of N samples, where the i-th sample

$(\mathbf{x}^i, \mathbf{y}^i)$ consists of L data points such that $\mathbf{x}^i = (x_1^i, \cdots, x_L^i)$ and $\mathbf{y}^i = (y_1^i, \cdots, y_L^i)$. The rescaling operation that normalizes $(\mathbf{x}^i, \mathbf{y}^i)$ to $(\tilde{\mathbf{x}}^i, \tilde{\mathbf{y}}^i)$ is:

For $j = 1, ..., L$,

$$\tilde{x}_j^i = \frac{x_j^i - x_{min}^i}{x_{max}^i - x_{min}^i} \tag{3.2a}$$

$$\tilde{y}_j^i = \frac{y_j^i - y_{min}^i}{y_{max}^i - y_{min}^i} \tag{3.2b}$$

, where $x_{min}^i = min(x_1^i, \cdots, x_L^i)$ and $y_{min}^i = min(y_1^i, \cdots, y_L^i)$. The values of $\tilde{\mathbf{x}}^i$ and $\tilde{\mathbf{y}}^i$ both fall in the range of $[0, 1]$. If we re-arrange Equation 3.2a and 3.2b, then we have

$$\begin{aligned} x_j^i &= \tilde{x}_j^i(x_{max}^i - x_{min}^i) + x_{min}^i \\ &:= a_x^i \, x_j^i + b_x^i \end{aligned} \tag{3.3a}$$

$$\begin{aligned} y_j^i &= \tilde{y}_j^i(y_{max}^i - y_{min}^i) + y_{min}^i \\ &:= a_y^i \, y_j^i + b_y^i \end{aligned} \tag{3.3b}$$

Applying the above results to Equation 3.1, the resulting curvature equation of normalized data becomes:

$$K_{\tilde{y}}(\tilde{x}) = \frac{\frac{a_x^2}{a_y} f''(a_x \, \tilde{x} + b_x)}{\left[1 + \left(\frac{a_x}{a_y} f'(a_x \, \tilde{x} + b_x)\right)^2\right]^{3/2}} \tag{3.4}$$

It is important to note that normalizing data does change the curve shape and thus alter the knee/elbow point position. Figure 3.1 (a) and (b) demonstrate how normalizing data changes the curvature shape and knee position. Figure 3.1 (a) shows the curve of $y = 5 \times \frac{1}{1+e^{-10x+5}} = 5 \times \tilde{f}(x)$ generated by 1000 evenly-spaced

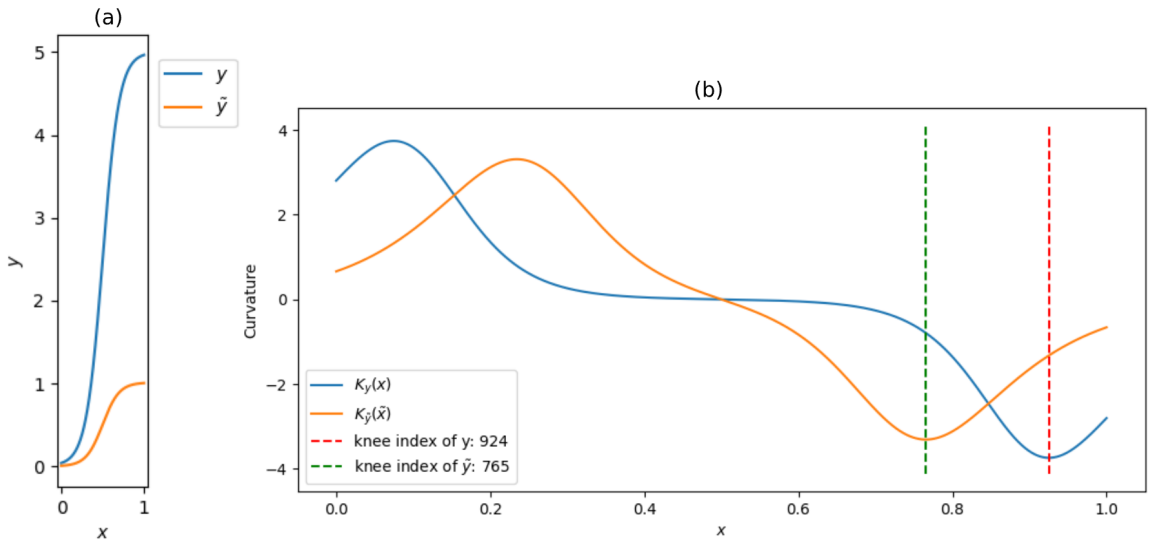

Figure 3.1: An example showing data normalization changes the curvature shape and knee position. (a) The curve of $y = 5 \times \frac{1}{1+e^{-10x+5}}$ generated by 1000 evenly-spaced x values in $[0, 1]$. The normalized values are plotted as $\tilde{y}$ in the figure; (b) Curvatures and the corresponding knee point indices of the curves. The normalization operation applies a squeezing effect to the curve of y, resulting in a smaller rate of change as observed in $\tilde{y}$. This reduces the range of values of $K_{\tilde{y}}(\tilde{x})$ and causes a shift in the position of the knee point.

x values in $[0, 1]$. Its corresponding curvature is $K_y(x) = \frac{5 \cdot 10 \cdot \tilde{f}(1-\tilde{f})(1-5\tilde{f})}{\left[1+\left(5 \cdot 10 \cdot \tilde{f}(1-\tilde{f})\right)^2\right]^{3/2}}$. By inputting the corresponding values of $x_{min}, x_{max}, y_{min}, y_{max}$ which are 0, 1, 0.033, and 4.967 into Equation 3.4 and making the necessary substitutions, we have $K_{\tilde{y}}(\tilde{x}) = \frac{\frac{1}{4.934} f''(\tilde{x})}{\left[1+\left(\frac{1}{4.934} f'(\tilde{x})\right)^2\right]^{3/2}}$. One can observe that the curvature of y ranges from -3.740 and 3.740, while that for $\tilde{y}$ is -3.308 to 3.308. The change in curvature value is a result of the rescaling operation applied to the x and y values. This operation compressed the 1000 x and y values into shorter intervals of $[0, 1]$ respectively. As a result, the curve of $\tilde{y}$ is flatter than y, resulting in a slower deviation of the tangent to the curve of $\tilde{y}$ in the interval. This in turn leads to a decrease in curvature values and a shift of the knee point to a forwarder (leftward) position.

3.4 Proposed Approach

This section describes the architecture of our proposed convolutional neural network (*UNetConv*) and introduces the loss function and inference method used for knee point detection.

3.4.1 Model Architecture

The proposed network architecture is displayed in Figure 3.2. It should be pointed out that the height and width of each layer output are not entirely drawn to scale. The model is comprised of two main components: a U-Net model and a sequence of convolutional layers. The first part receives the input and processes it through the encoding path, then through a bottleneck, and finally through the decoding path. Both paths consist of four levels of blocks. In the encoding path, each block has a convolutional layer with 11×11 kernel with same padding, followed by a batch normalization (BN) layer and a ReLU activation function. The last layer of each block is a 2×2 max pooling layer with a stride of 2, which reduces the feature map width by half for the purpose of downsampling. The bottleneck layer consists of a single convolutional layer, which also has an 11×11 kernel, same padding, and 256 channels.

In the decoding path, additional layers are added to every level. First, an up-convolutional layer with 2×2 kernel is applied to upsample a feature map, followed by a BN layer and a ReLU activation function. A skip connection then takes place, which concatenates the feature map with the output from the encoding part at the same level. Lastly, the feature map is fed to a convolutional layer of 11×11 kernel with same padding. In both encoding and decoding paths, the number of channels in each block is 32, 64, 128, and 256.

The second part of the model is a sequence of convolutional layers. Each layer has a 2×2 convolutions and same padding. The number of channels in the layers is 16, 8, 4, and 1 respectively. The final step is normalization, which maps the output to a probability, indicating the likelihood of the current data point being a knee point. The network contains about 3.3M parameters in total.

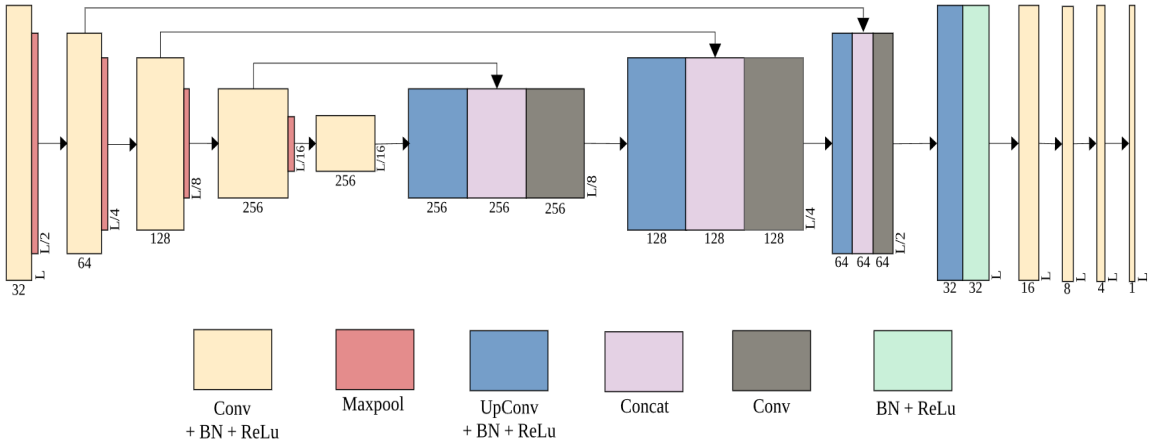

Figure 3.2: An illustration of the architecture of our proposed method, *UNetConv*. The model is comprised of two main components: a U-Net model and a sequence of convolutional layers. The U-Net model component part passes the input through the encoding path, followed by a bottleneck layer and then to the decoding path. Both the encoding path and decoding path contain four levels of blocks. The numbers beneath and in the bottom right corner of each block respectively indicate the number of channels and size of the resulting feature map passed through that specific layer.

3.4.2 Soft F_1 score

The F_1 score is a statistical measure to evaluate the accuracy of a classification model. It is particularly useful when the classes/labels in the data set are imbalanced, which is the case of our scenario - there are at most 5 knee points in each sample. The traditional F_1 score is a harmonic mean of two other metrics - precision and accuracy. Details are discussed in Section 3.5.3.

The issue with the traditional F_1 score is that it is not differentiable. It accepts

binary (0 or 1) inputs of prediction and ground truth, then it computes integer values of True Positives, False Positives, and False Negatives. Thus, it can not be used as a loss function to compute the gradient and update the model's weights during the training phase. This limitation can be overcome by modifying the F_1 metric to accept probabilities as inputs and calculate the required counting numbers as a continuous sum of likelihood. The equation is given in the following.

Let $(\hat{p}_1^i, \ldots, \hat{p}_n^i)$ be a set of predicted probabilities by the model and $(p_1^i, \ldots, p_n^i)$ be the binary ground truth of sample i respectively, where $p_j^i = 1$ if a knee is attained at index j, 0 otherwise. The soft F1-score $\widetilde{F}_1$ is :

$$\widetilde{F}_1(\hat{\mathbf{p}}^i, \mathbf{p}^i) = \frac{\sum_{j=1}^{L} \hat{p}_j^i \, p_j^i}{\sum_{j=1}^{L} \hat{p}_j^i + \sum_{j=1}^{L} p_j^i} \tag{3.5}$$

$\widetilde{F}_1$ naturally approximates the traditional F_1 classification metric and shares the same property of indicating better precision and accuracy with a higher score (with 1 being the best value).

3.4.3 Non-Maximal Suppression

Non-Maximal Suppression (*NMS*) (Redmon *et al.*, 2016) is a common technique in computer vision to eliminate multiple detections of the same object. With pre-specified threshold value δ and the area of interest (suppression area), the process of *NMS* involves dropping all detections whose prediction values are below δ. The algorithm selects the highest-scoring candidate and suppresses all other overlapping candidates within the area of interest. This process repeats until no candidates remain.

In this research, the *UNetConv* model is designed to predict the probability of a data point being a knee point. Often it is clear which data point is a knee and the

probability curve predicted by the model presents a tall narrow spike shape. However, there are cases where multiple spikes may occur near a knee point, which could be due to noise or the model naturally predicting higher probability near a knee point, making it difficult to determine. For this reason, we implement *NMS* to fix this issue.

3.5 Experiments

This section explains how we create noisy data for the training and test sets. We then provide details on how we implement the experiment. Finally, we evaluate the proposed network and compare it to other existing methods.

3.5.1 Synthetic Data

To evaluate the model performance, we select twelve functions to generate samples and create data sets. These functions (*FT 1-12*) are listed in Table 3.1. One important note here is that we make certain assumptions about the curve being used. Specifically, we assumed that the curve has two main characteristics: (1) monotone increasing, and (2) having at least one knee point in the interval. *FT 1-9* are functions that have only one knee point in the interval. *FT6* is the translated Scaled Exponential Linear Unit (SELU) function and *FT9* is the cumulative distribution function (CDF) of normal distribution. On the other hand, the number of knees $K(\geq 2)$ for *FT 10-12* samples can be specified, which allows for multiple knee points in these functions. In these multi-knee functions, *FT10* is the sum of multiple logistic functions and its curve is a smooth step function. *FT11* is a combination of sine functions, the resulting function can be described as a translated tilted sine function. *FT12* (see Figure 3.3) is a synthetic function formed by the summation of functions from the single-knee family. The number of functions chosen to generate a sample of *FT12* corresponds to the number of knees in the sample. Each time, a function

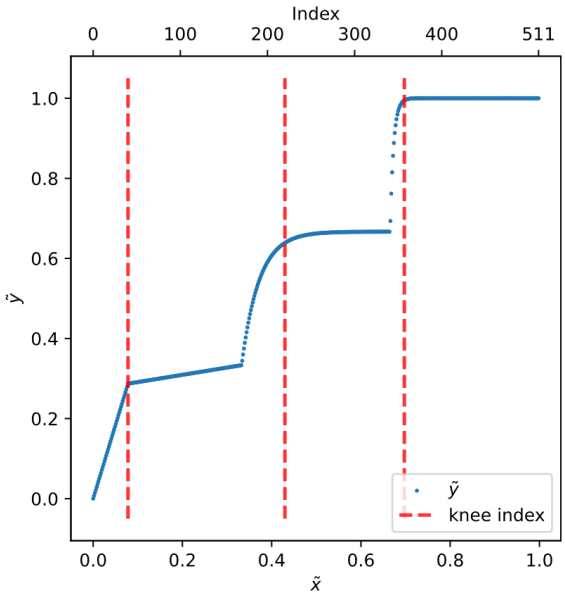

Figure 3.3: Graphical representation of a *FT12* multi-knee sample. This sample is formed by summing graphs from three single-knee functions, which is a combination of *FT8, 1* and *6*.

from the single-knee family is randomly selected to concatenate with the currently connected curve. There is one restriction on the slope when joining the curves: the slope formed by the last two points in the existing curve should not be greater than the slope of the first two points of the next curve being connected. This prevents the creation of additional knee points upon connection.

To introduce some noise to the generated sample, while maintaining the function data range, we consider each function in Table 3.1 as a cumulative distribution function (CDF). We then generate noisy data points $\hat{\mathbf{y}}$ by making use of the empirical distribution function and Inverse Transform Sampling method. Mathematically, given a CDF $f_{\tilde{X}}$ and a uniform variable $U \sim Uniform[0, 1]$, the random variable $R = f_{\tilde{X}}^{-1}(U)$ can be described by $f_{\tilde{X}}$. Therefore, the empirical distribution of R can be written as the following:

Table 3.1: Selected functions to generate samples and create data sets.

Code	Function	Description	Flipped?
$FT1$	$\ln(x)$	Logarithm	Y
$FT2$	$(-1)^{m+1}x^m$, for $m = 3, 5, 9$ or 11	Polynomial	Y
$FT3$	$x^{\frac{1}{m}}$, for $m = 3, 5, 9, \ldots, 17$	Rational	Y
$FT4$	$\frac{1}{1+e^{-x}}$	Logistic	Y
$FT5$	$-\ln\left(1+e^{-x}\right)$	Translated Softplus	Y
$FT6$	$1-e^{-x}$	Translated SELU	Y
$FT7$**	$(\frac{mx}{s})^p - (\frac{mx}{s})^q e^{-(\frac{x}{s})^r}$	Product of exponential and rational function	N
$FT8$	$y(x) = \begin{cases} m_1 x, & \text{if } x \in [0,\ x[\eta-1]] \\ m_2 x + c_2, & \text{if } x \in [x[\eta],\ 1] \end{cases}$	Piecewise Linear	N
$FT9$	$\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{(t-\mu)^2}{2\sigma^2}}$, where $\begin{cases} \mu = 13, \\ \sigma = 5 \end{cases}$	Normal Distribution CDF	Y
$FT10$	$\sum_{i=1}^{K} \frac{c_{1,i}}{1+e^{-c_{2,i}(x-c_{3,i})}}$	Sum of K Logistic functions	N
$FT11$	$\frac{1}{m}\sum_{i=1}^{K} \frac{\binom{2K}{K-i}}{i \cdot 2^{2K-1}} \sin(ix) + (x+t)\cdot q \cdot \ln(x)$	Translated tilted sine	N
$FT12$	Sum of FT 1-8	Sum of single-knee functions	N

Notation: 1) η: knee index of a sample, where $0 \leq \eta \leq L-1$; 2) K: number of knee(s) in a sample; 3) $**$: Values of p, q, r are determined by a brute force search in the set of values $[1, 2, 3, 4, 5]$, the possible values of s are $[10, 20]$ and that for m is $[0.1, 0.2, \ldots, 5.0]$.

For $r \in \tilde{\mathbf{X}}$,

$$\begin{aligned} \widehat{f}_R(r) &= \frac{\sum_{j=1}^{L'} \mathbf{1}_{R_j \leq r}}{L} \\ &= \frac{\sum_{j=1}^{L'} \mathbf{1}_{f^{-1}(U_j) \leq r}}{L'} \qquad \left(\because R_j = f_{\tilde{X}}^{-1}(U_j)\right) \\ &= \frac{\sum_{j=1}^{L'} \mathbf{1}_{U_j \leq f(r)}}{L'} \end{aligned} \tag{3.6}$$

Based on the above results, we can obtain noisy data points $\hat{\mathbf{y}}^i$ by using the cumu-

lative count of randomly generated points that follow a standard uniform distribution with a value less than $\tilde{y}_j$. The complete procedure to generate a sample is summarized in the following:

i. Generate L pairs of noise-free data points (x_j^i, y_j^i) where

$$x_j^i = x_{lb}^i + \frac{j}{L-1}(x_{ub}^i - x_{lb}^i)$$
$$y_j^i = f(x_j^i)$$

for $j = 0, 1, \cdots, L-1$

ii. Compute the normalized values $(\tilde{\mathbf{x}}^i, \tilde{\mathbf{y}}^i)$ of sample i by inputting the results from the previous step into Equations 3.2a and 3.2b

iii. Generate $\{u_m\}_{m=1}^{L'}$ from the standard uniform distribution $Uniform[0, 1]$, where L' is not necessarily same as L

iv. Compute the noisy data points $(\hat{x}_j^i, \hat{y}_j^i)$ for sample i, where

$$\hat{x}_j^i = \tilde{x}_j^i \tag{3.7a}$$
$$\hat{y}_j^i = \frac{\sum_{m=1}^{L'} \mathbf{1}_{u_m \leq \tilde{y}_j^i}}{L'} \tag{3.7b}$$

It is noteworthy to mention that the x interval varies between samples, even if they are generated from the same function. Generally speaking, a wider x interval leads to a curve with sharper curvature after normalization. Varying x interval allows us to create samples with different ranges of curvature values. Another benefit is that the curve may have different shapes for different intervals of x. Taking $FT4$ as an example, the curve exhibits an elbow point at approximately $x = -1.36$ (see curve

y_1 in Figure 3.4 (a). We pick the regions $[-30, 10]$ (blue) and $[0, 25]$ (orange), and generate 512 (x, y) data points in these intervals. The respective normalized data points are displayed as $\tilde{y}_1$ and $\tilde{y}_2$ in 3.4 (b). The blue region has a wider x interval. Its curve segment displays an elbow point, as its x interval includes $x = -1.36$. The curve shape of $\tilde{y}_1$ is noticeably different from $\tilde{y}_2$, even though both are produced by the same function. Therefore, their respective knee point position is also notably distinct from each other.

To further improve the diversity of data, we also randomly flip a normalized sample along the $y = 1 - x$ axis if there exists an analytical expression for the inverse of the chosen function. Again taking $FT4$ as an example, its inverse function is the logit function $y = ln(x/(1-x))$ which has a significantly different curve shape and different knee point position. Figure 3.4 (b) shows the flipped data of $\tilde{y}_1$. The knee point of *flipped* $\tilde{y}_1$ occurs at the very beginning of the curve.

3.5.2 Implementation Details

Synthetic Training Set and Test Sets

We create three distinct data sets using the functions outlined in Table 3.1 in order to test the model performance. The network is trained on 7000 samples from *FT1-8, 10-12*, reserving *FT9* for building a separate test set since we want to investigate the model's performance on samples from unseen functions. In the 7000-sample training set, there is an equal portion of single-knee and multi-knee samples, with 3500 of each category. Each function is proportionally represented within its respective portion. Specifically, functions *FT10-12* each account for 33.33% in all multiple knee samples, while the remaining distributions each account for 12.5% of all the single knee samples. The model performance is tested by identifying knee point index/indices in three test

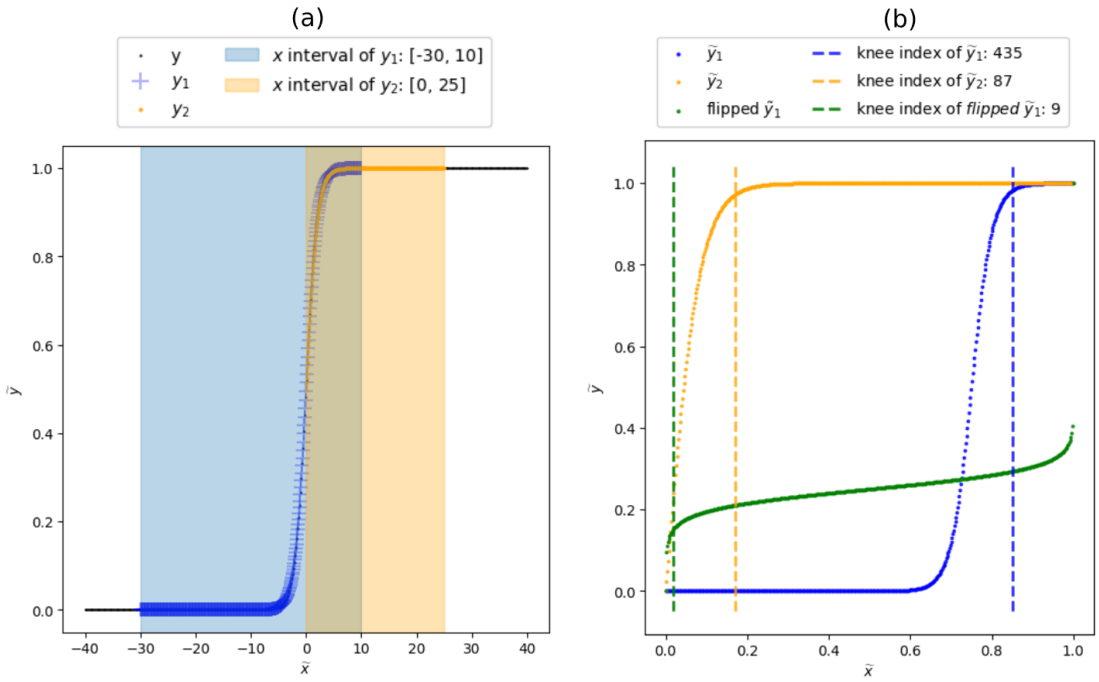

Figure 3.4: An example showing varying the x interval can generate samples with a variety of curve shapes, different ranges of curvature values, and thus different positions of knee point(s). (a) A graph showing the logistic function, $y = \frac{1}{1+e^{-x}}$, for $x \in [-40, 40]$. (b) The curve shape of $\tilde{y}_1$ is noticeably different from $\tilde{y}_2$, even though both are produced by the same function. The figure also shows the flipped curve of y_1. Unlike the logit function, the knee point of *flipped* $\tilde{y}_1$ occurs at the very beginning of the curve.

sets containing 300, 800, and 100 samples, respectively. The 300-sample test set (denoted as *mknee*) includes 100 samples from each multi-knee distribution (*FT10-12*), and the 800 test set (denoted as *sknee*) also comprises 100 samples from each single-knee distribution (*FT1-8*). The last test set contains 100 samples from $FT9$ distribution (denoted as *ng*). Since the model has not been trained with samples from this distribution, its outcome will demonstrate the model's capability in capturing knee point properties. Every sample in the data sets has 512 (x, y) data points. Upon analyzing the data, the range of knee curvature values per data set is as follows: [-337.32, -3.00] for the training set, [-339.39, -3.00] for the 800-sample *sknee* test set, [-326.32, -5.78] for the 300-sample *mknee* test set, and [-40.85 to -6.82] for the 100-

Table 3.2: Values of configuration that were attempted when applying the Exponentially Weighted Moving Average (EWM) to smooth data. The configuration that achieves the lowest MSE between the smoothed data and noise-free data is chosen. The optimal configuration varies for each sample.

Configuration	Values
Center of Mass	0.2, 0.4, ..., 10.0
Span	1.2, 1.4, ..., 10.0
Half-life	0.2, 0.4, ..., 10.0
Alpha	0.1, 0.3, ..., 0.9

sample ng test set. To sum up, the range of knee curvature in this experiment is between -339.39 and -3.00. In all of the synthetic data sets, the knees are not located within 10 indices from the boundary.

Real-World Data Set

To further assess the model's performance on real-world data, three variables (Birth Date (Year), Math ALEKS, and Fall Hours) are selected from the ASU 2017 Engineering Freshmen Data (Engr2017) which is analyzed in Section 4.5.1. The corresponding data points for these variables are 2717, 2579, and 2668. The target knee points in these variables were determined by using the YFM1 method of the PVAD algorithm. It is worth noting that all of these variables have multiple knee points. However, the only variable that has a knee point within the first 10 indices is Birth Date (Year), which occurs at index 7.

Data Preprocessing for Other Model/Methods

Since the *Kneedle* method requires curve smoothing in the data preprocessing stage, each sample is first smoothed by Exponentially Weighted Moving Average (EWM)

before being projected/rotated. We test various configurations (as shown in Table 3.2) and select the one that achieves the lowest MSE between the true noise-free y values and the fitted curve for each sample. In addition, for methods primarily designed for detecting elbows in clustering applications such as *DFDT* and *AL*-Method, we translate the data points of each single-knee sample by $(\tilde{x}_j^i,\ 1 - \tilde{y}_j^i)$ such that the translated curve has consistent positive concavity, which is similar to the loss function shape in clustering applications.

Loss Function

As discussed in Section 3.4.2, the traditional F_1 is an intractable step function for gradient descent. To overcome this limitation, we implement the soft $\widetilde{F}_1$ score as a surrogate function of F_1. Strictly speaking, our loss function is defined as:

$$\min \frac{\alpha}{\widetilde{F}_1} + 1 - \widetilde{F}_1 \tag{3.8}$$

, where α is a constant. To determine the value of α, we run our model with α set to 0.01, 0.1, and 1, each with a single trial, using the loss function described in Equation 3.8. We select the α value that results in the lowest loss value for the subsequent experiment. The selected α value is 0.1.

Optimization

We train the network for 200 epochs with batch size = 64. AdaDelta is employed with an initial learning rate = 0.5 and momentum = 0.5. For every 10 epochs, the learning rate is decreased by half. The loss function used is given in Equation 3.5.

Post-Processing on Model Output

In the testing stage, the model output of the i-th sample, $\hat{\mathbf{p}}^i$, undergoes further processing using *NMS*. This method predicts the knee index/indices and returns a binary prediction $\hat{\mathbf{p}}^i_{NMS}$ with a value of 1 at index j, indicating the detection of a knee at the j-th data point. The probability threshold selected for *NMS* is 0.5. The suppression area is set to be $\pm$ 10 indices, dropping any candidates located 10 indices from the left and 10 indices from the right of the selected knee point in each iteration. The resulting binary output $\hat{\mathbf{p}}^i_{NMS}$ is compared with the binary ground truth $\mathbf{p}^i$ to compute the traditional F_1 score for model performance evaluation.

3.5.3 Metric

The F_1 score is one of the most widely used metrics in classification analysis. It is the harmonic mean of precision and recall, providing a single score that balances these two metrics. This is useful because a model with high recall but low precision may correctly identify a lot of true positives, but may also identify many false positives.

Since some algorithms cannot detect at the exact same knee point, we incorporate allowable index error in the calculation of F_1 score to accommodate for this issue as in (Satopaa *et al.*, 2011). As an illustration, suppose we have a data set with points at x = 1, 2, 3, ..., 7, and the knee occurs at x = 5 and 7. We consider the algorithm to have "correctly" identified the knees if it identifies any points at x = 3, 4, 5, 6, or 7 as knees, with a margin of error of 2.

3.5.4 Evaluation

Results on Synthetic Data

Table 3.3: Quantitative results of UNetConv and other methods, with an allowable index error of 2.

Test Set	DFDT	DFDT Ref	AL	AL Ref	S	S Ref	Kneedle Proj	Kneedle Rot	UNetConv
sknee	0.021	0.024	0.274	⋈	0.044	⋈	0.09	0.09	**0.740**
mknee	–	–	–	–	–	–	0.063	0.063	**0.720**
ng	0.000	0.000	0.040	⋈	0.000	⋈	0.11	0.11	**0.810**

⋈: Fail to Converge

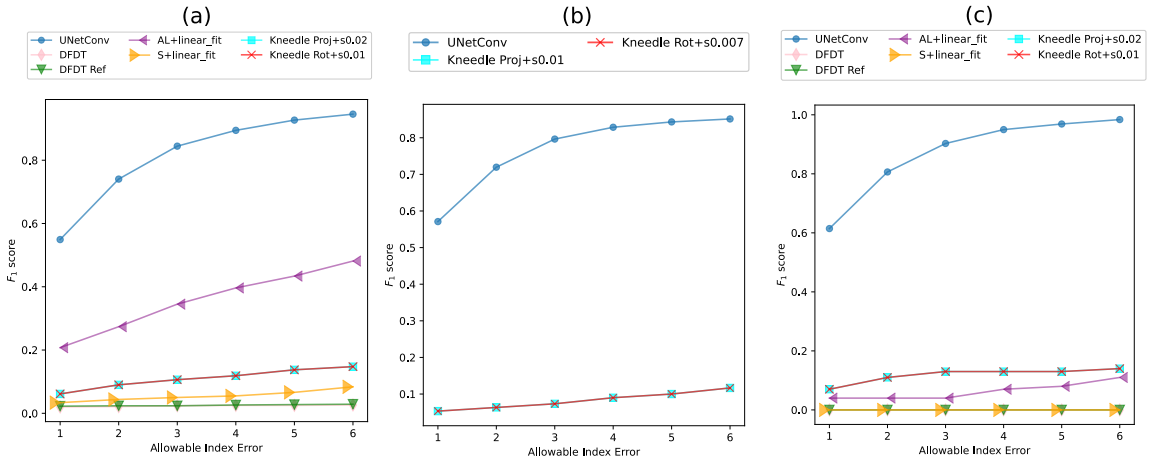

Figure 3.5: F_1 scores of (a) *UNetConv*, *DFDT*,*AL*, *S* and *Kneedle* methods for varying allowable index error on the *sknee* data set; (b) *UNetConv* and *Kneedle* methods for varying allowable index error on the *mknee* data set; (c) *UNetConv*, *DFDT*, *AL*, *S* and *Kneedle* methods for varying allowable index error on the *ng* data set.

We evaluate our network by running 20 trials on all three synthetic test sets. The average of the F_1 scores per test set is then compared with other methods mentioned in Section 3.2. Since the *AL*-Method is an extension of the traditional *L*-Method, we only consider the *AL*-Method in this evaluation. In Figure 3.5, the F_1 scores for each method per test set are plotted against allowable errors ranging from 1 to 6. Our method is denoted by *UNetConv*. We use two curve-fitting methods for both *AL* and *S* methods: fitting a straight line that best matches all the data within the specified range (*best fit*), and fitting the first and last data point within the specified range (*linear fit*). Since the results using *linear fit* are better than *best fit* irrespective of *AL*

or S methods in all scenarios, we only list the results using *linear fit* in Figure 3.5 and Table 3.3. The iterative refinement methods of the AL-Method and S-Method are denoted as AL Ref and S Ref, respectively. However, we are unable to obtain results as these methods fail to converge to the same knee point for some samples. In the original work of *Kneedle*, projection is implemented to transform data points instead of rotation. We perform both projection (*Kneedle Proj*) and rotation (*Kneedle Rot*) when comparing the results. For each test set and each data transformation method, we select the ζ value that achieves the highest mean F_1 score among the allowable index errors and only consider these results when making comparisons. For *Kneedle Rot*, the best ζ values for $sknee, mknee$ and ng are 0.01, [0.007, 0.008] and [0.006, 0.02] respectively. As for *Kneedle Proj*, the corresponding ideal ζ values for $sknee, mknee$ and ng are 0.02, 0.01, and [.008, 0.03]. For simplicity, we take the lowest ζ value since there are multiple ζ values achieving the same mean F_1 score. To demonstrate the overall performance of the knee detectors, we plot the knees detected by *UNetConv* and the techniques that achieve the top three F_1 scores in Figures 3.6 and 3.7. Table 3.3 shows the quantitative results of the proposed model and other methods, with an allowable error of 2.

Results on Real-World Data

Regarding the variables in the real-world dataset Engr2017, we use the same configurations except the allowable index errors are set to 50 and 60. This adjustment is made to account for the fact that these variables contain over 2000 data points. Table 3.4 shows the resulting F_1 scores for each index error allowed for each variable. Figure 3.8 displays the data points (blue), the probability output by *UNetConv* (aqua), the predicted knee indices (green), and the knees identified (red) using YFM1 of the PVAD algorithm for each of the variables.

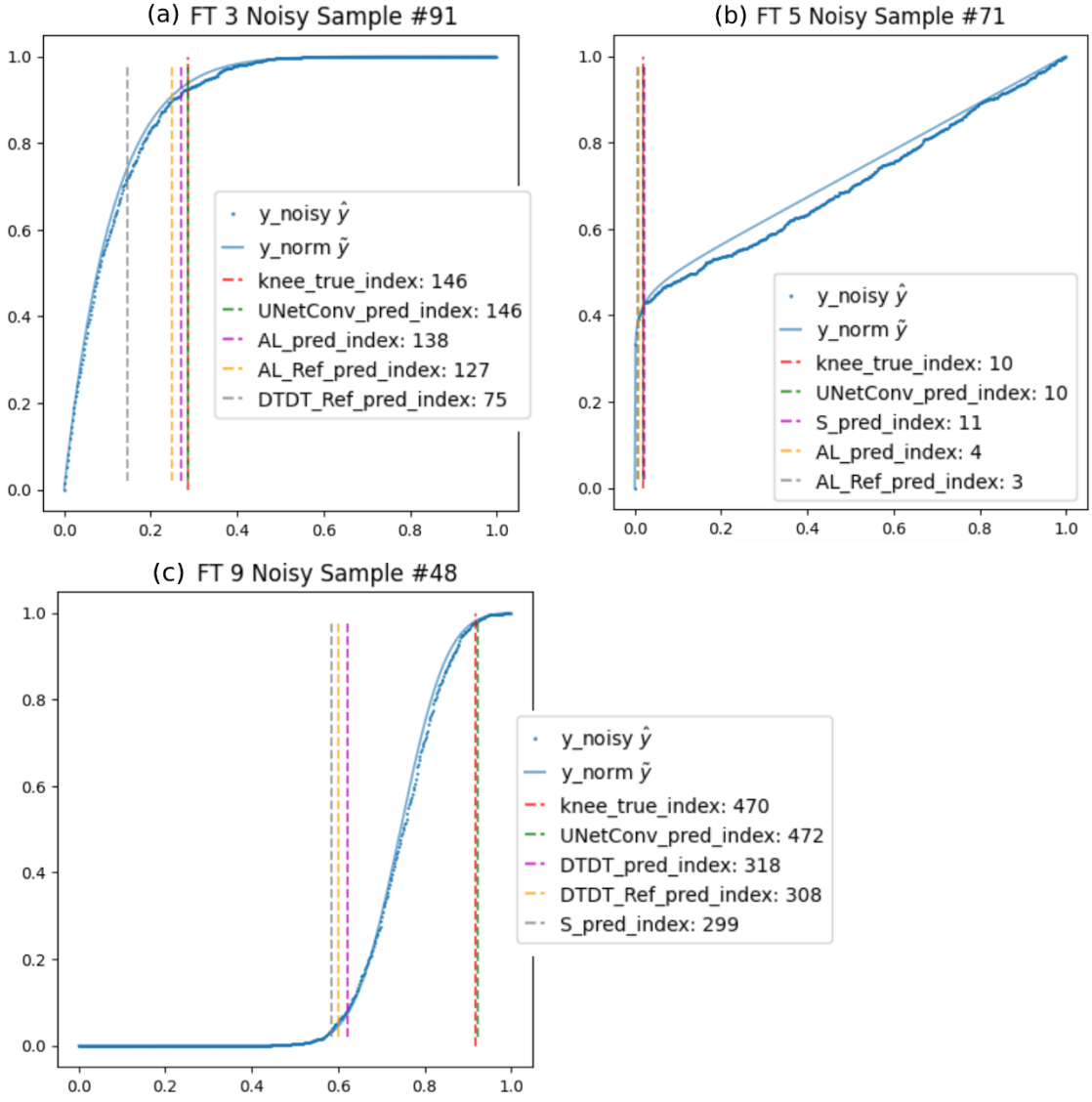

Figure 3.6: A demonstration of the overall performance of the knee detectors on single-knee noisy data $\hat{\mathbf{y}}$. (a) *UNetConv*, *AL*-Method, *AL*-Method with Refinement and *DFDT* with Refinement for *FT8* sample; (b) *UNetConv*, *S*-Method, *AL*-Method and *AL*-Method with Refinement for *FT5* sample; (c) *UNetConv*, *DTDT*, *DFDT* with Refinement and S-Method for *FT9* sample.

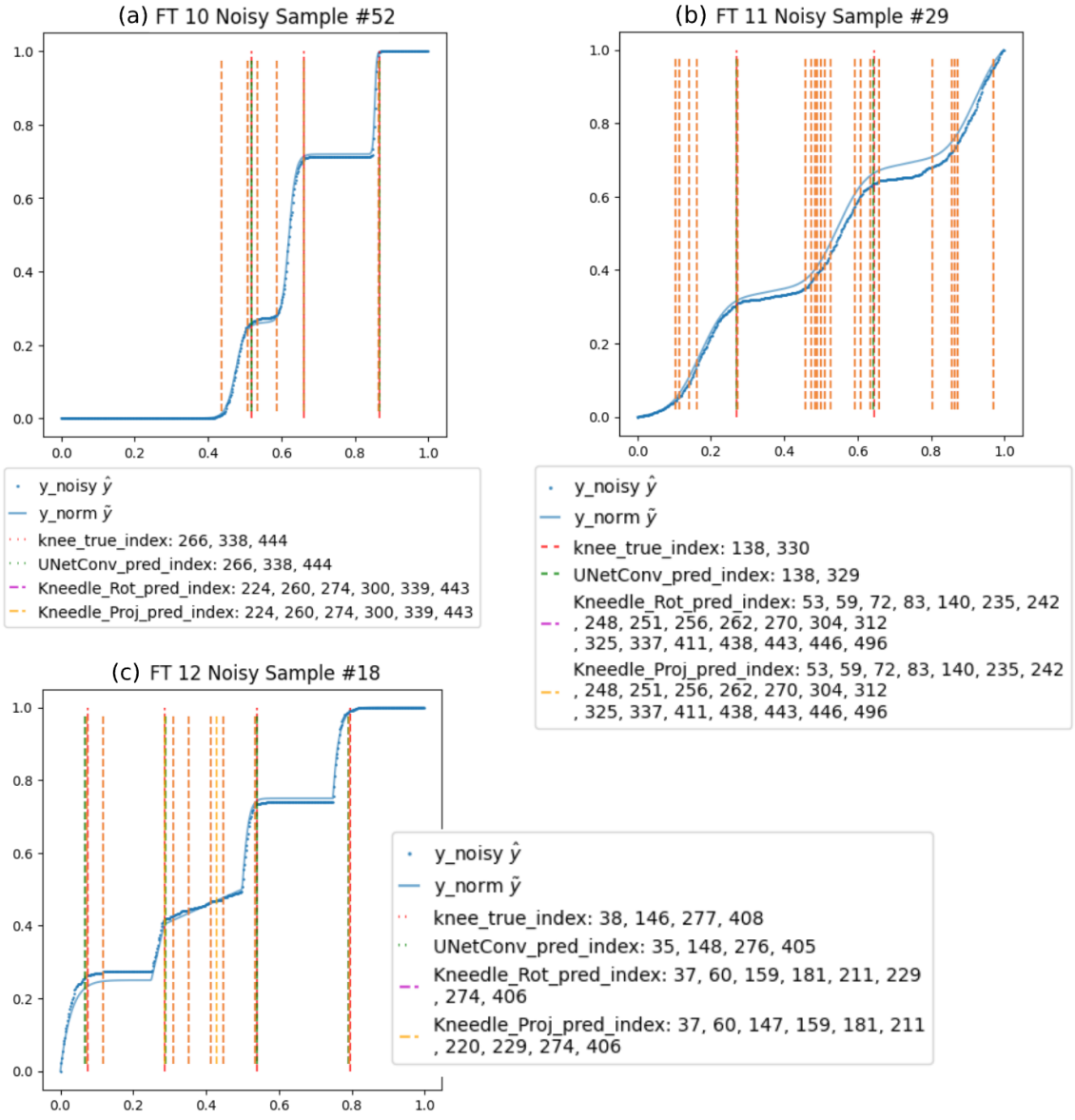

Figure 3.7: A demonstration of the overall performance of the knee detectors on multiple-knee noisy data $\hat{\mathbf{y}}$. The figures show *UNetConv*, *Kneedle* with Rotation and *Kneedle* with Projection for (a) *FT10* sample; (b) *FT11* sample; (c) *FT12* sample.

Table 3.4: Quantitative results of *UNetConv* on Engr2017 Variables

Allowable Index Error	Birth Date (Year)	Math ALEKS	Fall Hours
50	0.333	0.571	0.286
60	0.667	0.571	0.286

3.5.5 Discussions on the results

In all the synthetic test sets and all settings of allowable index error, *UNetConv* outperforms the existing methods. From Table 3.3, our model attains the highest F_1 score of 0.74 in the *sknee* test set, with the second best result of 0.274 attained by the *AL*-Method. The results of other methods are all below 0.1. For the multiple-knee *mknee* test set, *UNetConv* obtains the highest score of 0.72, followed by *Kneedle* (0.063), which results in the same score regardless of using rotation or projection to transform the curve. For the unseen 100-sample noisy Gaussian *ng* test set, *UNetConv* again surpasses other methods. Even in the most extreme scenario where the allowable index error is set to 1, our model reaches 0.55, 0.57, and 0.61 F_1 scores on the test sets, which are double the best results achieved by other methods.

The *DFDT* techniques are not capable of locating knee points as most of the samples have both an elbow point and a knee point on the curves as their first derivative values are both close to zero. The use of only first derivative values makes it challenging to differentiate between an elbow and a knee. It is thus understandable that the method struggles to have good performance. Though there is an iterative refinement method to help overcome this shortcoming, there is only a minor improvement in the model performance. Based on our analysis of the test sets, we have found that these methods perform better on the *FT8* data set, which consists of noisy piecewise straight-line functions. This is due to the fact that the first-derivative values of these

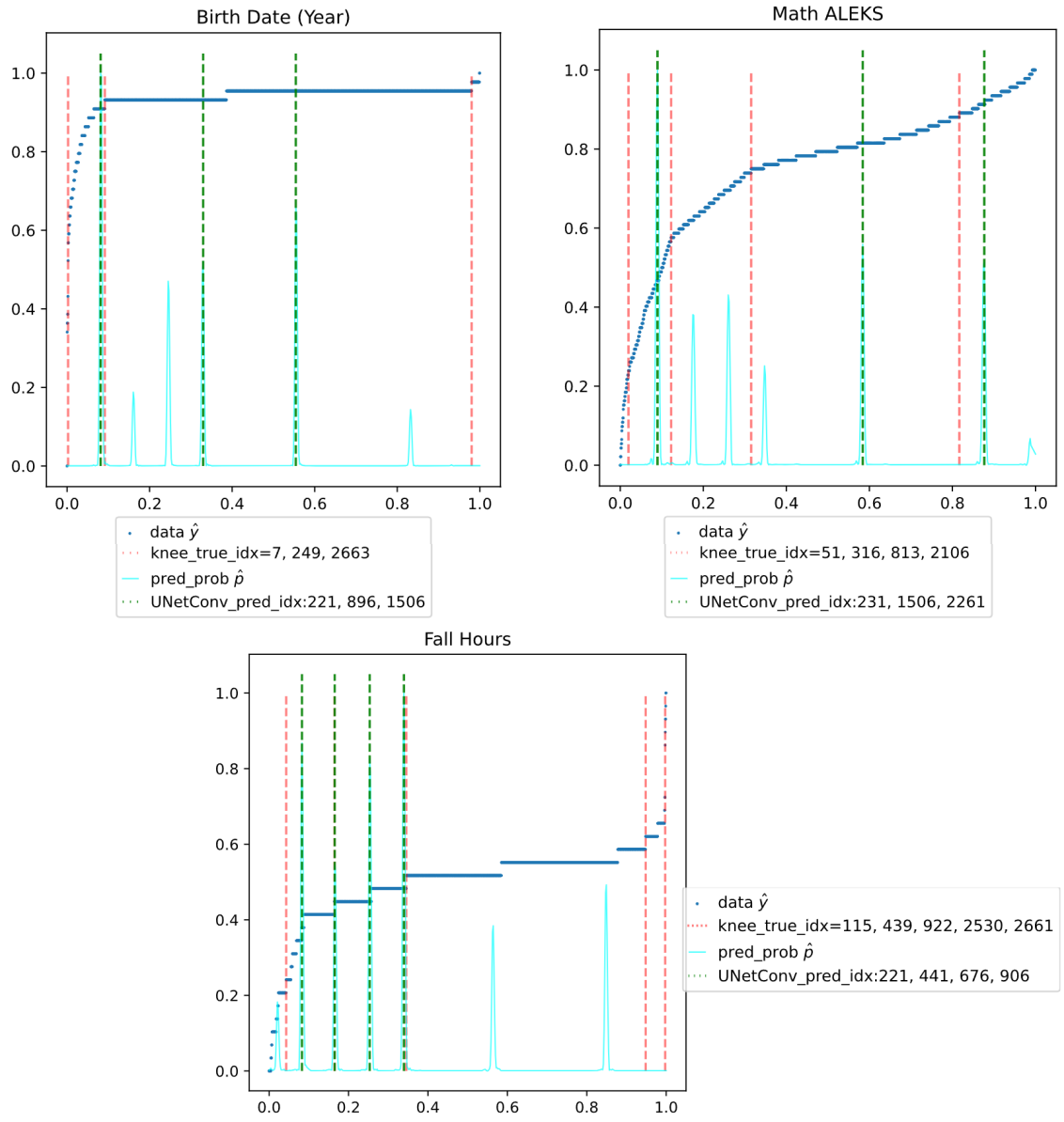

Figure 3.8: A demonstration of *UNetConv* performance of variables (a) x16: Birth Date (Year), (b) x29: Math ALEKS and (c) x35: Fall Hours in Engr2017 data set.

samples can be distinctly separated into higher and lower value groups, thus making it easier for the method's threshold algorithm *IsoData* to accurately estimate the optimal knee point.

For the *AL*-method, it achieves the second highest F_1 score on the *sknee* data set. This method works best on simple curve shapes, like *FT8* and *FT9*, where each curve segment is close to a straight line and the angle score can accurately capture a knee point. However, when it comes to the curve segment with a quadratic or higher degree of curve shape (as in *FT7*), the fitted straight lines and the angle between no longer contribute positively towards locating a knee. We also observe that this method is not as effective when being used on curves with a gradual rate of change or less sharp curvature.

The *S*-method works poorly on all the single-knee samples. One primary reason is the imprecise fitting of straight lines onto curved segments. The fitted curve with the lowest RMSE is not a reliable indicator of the presence of a knee point. The *S*-method not only shares the same limitations as the *AL*-method, but it is also limited to working with simple concave curves. As a result, it consistently forecasts the initial point as a knee point. This explains its poor performance on the *ng* test set (*FT9*), which always has an elbow point before a knee point.

Regardless of whether rotation or projection is used, *Kneedle* produces consistent results for all test sets. One must define a single sensitivity value to compute a unique threshold at every local maximum point of the transformed data curve. However, determining a universal sensitivity value that effectively applies to all local maximum points of a sample is challenging due to the variation in the transformed data magnitude. Furthermore, *Kneedle* is highly susceptible to noise, which can erroneously classify a spike as local maximum and thus consider a noisy data point as a candidate. It is worth noting that the knee point may not always be reached at a local maximum.

The *Kneedle* algorithm has a tendency to detect multiple false negatives.

Lastly, it is observed that the performance of *UNetConv* drops significantly when being applied to the Engr2017 data set. The F_1 scores fall below 0.6 for all the variables. There is no notable improvement despite increasing the allowable indices from 50 to 60, which means a total allowance of 20 indices. This observation suggests that some of the knee points predicted by *UNetConv* are far from the target knee point(s). From Figure 3.8, it can be seen that the model blindly identifies a knee point on a relatively flat curve segment. For example, the model predicts knee points at indices 896 and 1506 for the variable Birth Date (Year), even though the curve shows a generally horizontal shape in the interval [0.3, 0.6]. Furthermore, as expected, the model could not detect the first knee point of variable Birth Date (Year), which is located at index 7, since the training samples all have knee points $\pm$ 10 indices away from the boundaries.

Another possible reason for a decrease in the model performance is that there are predictions being indiscriminately dropped by *NMS* because of failing to meet the threshold value requirement. An example can be seen in Figure 3.8 (b). The probability output curve reaches its peaks relatively close to the target knee points at 316 and 813, they are aborted by *NMS* due to the probability values at these points below 0.5. On the other hand, it is worth noting that the presence of peaks near a knee point implies that *UNetConv* effectively learns the features for knee point identification, but the selection of *NMS* threshold value does exert influence on the results. After all, one can still utilize the model's probability output and leverage the peaks on a curve to aid in detecting knee point(s).

3.6 Conclusion, Limitations and Future Work

In this dissertation, we introduce a novel mathematical definition for a knee point in discrete data sets. We show and explain the necessity of rescaling the data. We develop a benchmark data set that provides noisy data within the original data range, along with ground truth labels that are independent of any underlying algorithm/techniques. We believe that this benchmark data set can serve as a common ground for evaluating future knee detection designs. We propose a new model, *UNetConv*, for detecting knee points in discrete data sets, and compare its performance with existing approaches using the developed benchmark data set and real-world data. Our results indicate that *UNetConv* outperforms other existing methods and exhibits exceptional performance on unseen data. The evaluation results on real-world data show that *UNetConv* demonstrates potential by giving reasonable results for keen point prediction and justifies additional investigation.

The limitations of this study include: (1) The noise introduced to the samples is Gaussian noise, which can make it relatively easier for the model to detect knee points due to its distinct characteristics. However, in real-world scenarios, the noise may not always follow a Gaussian distribution, which can affect the accuracy and generalizability of the model's performance. (2) The target knee points in the real-world data set are detected initially through visual inspection without normalization. This process imports a certain level of subjectivity when determining the knee points. (3) There is a lack of versatility in the functions chosen to generate samples. As a result, there are numerous curve shapes that have yet to be explored by the model. (4) In all the synthetic data sets, the highest number of knee points observed in a curve is five. It remains uncertain whether the proposed model is capable of dealing with scenarios involving more than five knee points.

Considering the aforementioned limitations, it is essential to conduct an extensive and in-depth investigation. Future work includes but is not limited to incorporating a wider range of samples with varying levels of noise. This will provide valuable insights into the model's robustness and its ability to handle noisy data, thereby revealing the model's sensitivity to noise.

Chapter 4

APPLICATIONS OF PVAD RESULTS ON REAL-WORLD DATA

The PVAD algorithm was validated through a sensitivity analysis to evaluate the impact of parameter values on the results. In Chapter 2, we also demonstrate how the PVAD algorithm can address the limitations of existing methods. The objective of this chapter is to illustrate that the PVAD algorithm is eligible to analyze real-world data in various domains, including energy consumption (Section 4.1), engineering student retention (Sections 4.4 and 4.5), and network traffic (Sections 4.2 and 4.3). The applications show that the PVAD algorithm is capable of learning variable relations for both full and partial value ranges. The findings show that the PVAD algorithm has the advantage and capability of discovering variable relations.

4.1 ASU Energy Consumption System Dataset

The PVAD algorithm has been applied to energy consumption system data collected at ASU in an attempt to build a PVAD-based system modeling. The energy consumption data was collected from an ASU building in the entire month of January 2013. The data was sampled every 15 minutes on each day of January 2013 and contained 2976 data records. The data has four numeric variables: E for electricity consumption in kilo watts per hour (kWh), C for energy consumption for cooling in tons per hour (TonHr), H for energy consumption for heating in mmBTU, and A for outside air temperature in Fahrenheit (F), as well as the variable, T for TimeStamp, taking such values as "Jan 1 2013 12:00AM", "Jan 1 2013 12:15AM", etc. Although TimeStamp appears like a categorical variable with 2976 different values in 2976 data records, TimeStamp is a numeric variable with values being sampled every 15 min-

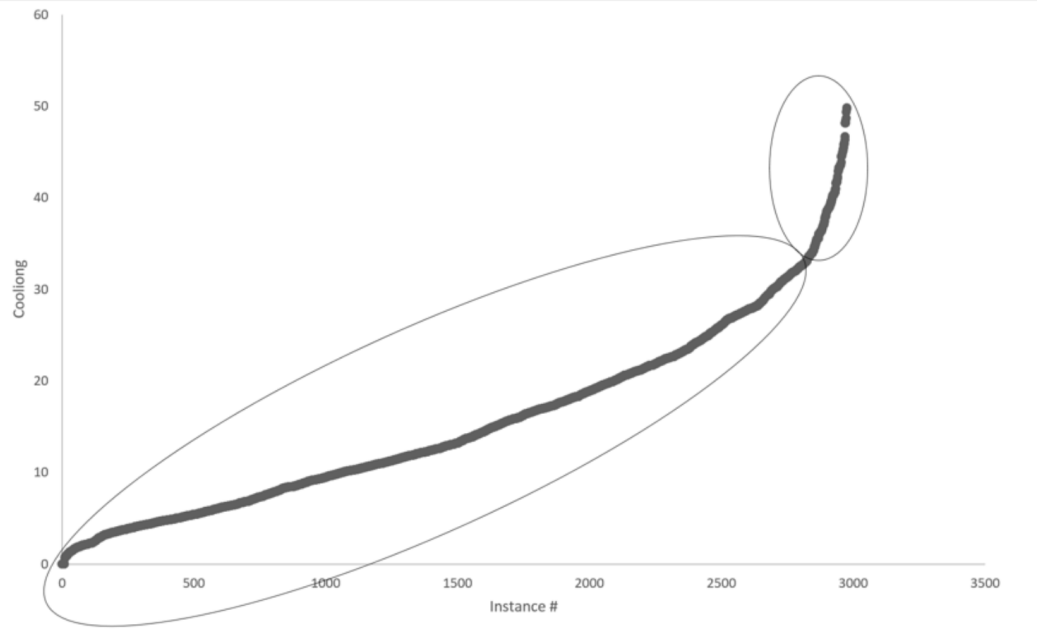

Figure 4.1: The plot of variable C (Energy Consumption for Cooling) in the increasing order of values with data clusters.

utes. TimeStamp is an important variable since it is related to changes in outside temperature and changes of occupants and their activities in the building at various times of each day. Hence, it is expected that TimeStamp has close relations with other numeric variables in the data set. The PVAD algorithm is used to analyze the energy consumption data using $\alpha = 0.8$, $\beta = 10$ (out of 2976 instances in the data set), and $\gamma = 95\%$. The set of the results from each combination of α, β and γ can be examined for meanings and information which the results reveal in the context of the application. The most meaningful, useful set of the results from a given combination of α, β and γ can be used to establish the final results and system model.

4.1.1 Analysis of PVAD results on ASU Energy Consumption System Dataset

In Step 1 of the PVAD algorithm, because we do not have any categorical variable in the data set to use Method 2, we first use Method 1 to transform each of the four

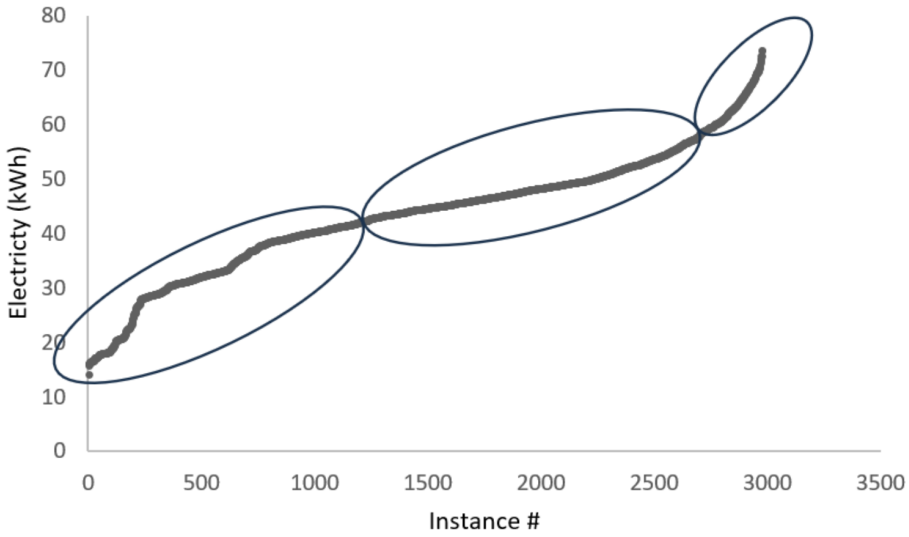

Figure 4.2: The plot of variable E (Electricity Consumption) in the increasing order of values with data clusters.

Table 4.1: Categorical values of variables and their corresponding numeric values for the energy consumption data.

Variable	**Categorical Value**	**Corresponding Numeric Values**
T (TimePeriod)	11:15 PM – 6 AM	Sampling every 15 minutes
	6:15 AM – 8 AM	Sampling every 15 minutes
	8:15 AM – 12 PM	Sampling every 15 minutes
	12:15 PM – 5:30 PM	Sampling every 15 minutes
	5:45 PM – 11 PM	Sampling every 15 minutes
E (Electricity)	Low	< 42
	Medium	[42, 58)
	High	≥ 58
C (Cooling)	Low	< 34
	High	≥ 34
H (Heating)	Low	< 0.18
	Medium	[0.18, 0.28)
	High	≥ 0.28
A (Air Temperature)	Low	< 40
	Low	[40, 55)
	Medium	≥ 55

numeric variables, E, C, H, and A, into a categorical variable by plotting values of each variable in an increasing order and identifying the initial data clusters. The initial data clusters and their intervals of values may be adjusted so that values of each variable at similar times are included in the same intervals. Figures 4.1 and 4.2 show examples of plotting the data points of E and C, along with the corresponding final data clusters. Table 4.1 shows categorical values of all the variables E, C, H and A which are determined using the intervals of values for the data clusters. With the newly transformed categorical variables of E, C, H and A, we then use Method 2 to transform TimeStamp into a categorical variable, TimePeriod, by using each of the four categorical variables (E, C, H and A) and their categorical values as a guide to determine intervals of TimeStamp values corresponding to different categorical values of each categorical variable. Table 4.1 shows the categorical values of TimePeriod (T).

Tables 4.2-4.3 list the most specific association(s) in each group of the associations with the same AV. Table 4.4 lists the most generic association(s) in each group of the associations with the same AV. Variable relations for energy consumption are revealed by each association in Tables 4.2-4.4. In Tables 4.2-4.4, there are groups that give similar associations. For example, the associations in Group 1 and Group 2 in Table 4.4 are similar. For the groups with similar associations, we marked only one group using the symbol $\wedge$ in the column of group #. Most of the associations in Tables 4.2-4.4 involve C=Low for cooling being low in CV or AV because most of the instances in the data set (2848 out of a total of 2976 instances) contain C=Low due to the month of January when the data was collected. Since C=Low is so common in the data set, C=Low can be dropped from the associations when interpreting associations.

Table 4.2: The most specific associations in each group of associations with the same AV: Set 1.

Group #	**The most specific association(s) in group**
1	A=Medium, [T=12:15 PM to 11 PM, E=High]/ [T=6:15 AM to 11 PM, E=Medium]/[T=11:15 PM to 6 AM, E=Low] → H=Medium, C=Low
2^	A=Medium, C=Low, [T=11:15 PM to 6 AM, E=Low]/[T=6:15 AM to 12 PM, E=Medium]/[T=12:15 PM to 11 PM, E=High/Medium] → H=Medium A=High, C=Low, E=Medium, T=8:15 AM to 12 PM → H=Medium
3^	A=High, E=Medium, C=High, T=12:15 PM to 5:30 PM → H=Low
4	C=High, E=Medium, T=12:15 PM to 5:30 PM → A=High | H=Low
5^	H=High, E=Medium, T=6:15 AM to 8 AM → A=Low, C=Low
6	H=Medium, E=Low, T=12:15 PM to 5:30 PM → A=High, C=Low
7^	[E=Medium, C=*, H=Low]/[E=Low, C=Low, H=Medium], T=12:15 PM to 5:30 PM → A=High C=High, T=5:45 PM - 11 PM → A=High
8	H=Medium, E=High, T=12:15 PM to 5:30 PM → A=Medium, C=Low
9^	H=Medium, C=Low, E=High, T=12:15 PM to 5:30 PM, → A=Medium
10	H=High, C=Low, E=Medium, T=6:15 AM to 8 AM → A=Low
11^	[A=Low, H=High]/[A=High, H=Low]/[A=Low/Medium, H=Medium], C=Low, T=11:15 PM to 6 AM → E=Low
12	[A=Low, H=High]/[A=High, H=Low]/[A=Medium/Low, H=Medium], T=11:15 PM to 6 AM → C=Low, E=Low

Table 4.3: Specific associations in each group of associations with the same AV: Set 2

13	A=Medium, H=High, T=5:45 PM to 11 PM → C=Low, E=Medium T=8:15 AM to 12 PM, H=Medium, A=High/Medium → E=Medium, C=Low

Continued on next page

Table 4.3: Specific associations in each group of associations with the same AV: Set 2 (Continued)

14^	A=High, H=Low, C=High, T=12:15 PM to 5:30 PM → E=Medium A=Medium, H=High, C=Low, T=5:45 PM to 11 PM → E=Medium A=Medium/High, H=Medium, C=Low, T=8:15 AM to 12 PM → E=Medium
15	H=Low, C=High, T=12:15 PM to 5:30 PM → A=High | E=Medium
16^	A=High, C=High, T=12:15 PM to 5:30 PM, → H=Low | E=Medium A=Medium/High, C=Low, T=8:15 AM to 12 PM → H=Medium | E=Medium

Table 4.4: Generic associations in each group of associations with the same AV

Group #	**The most generic association(s) in each group**
1^	A=Medium/E=High → H=Medium
2	E=High/A=Medium, C=Low → H=Medium
3	E=Medium, C=High, A=High → H=Low
4	E=Medium, C=High → H=Low | A=High
5^	H=High, T=6:15 AM to 8 AM → A=Low.
6^	E=Low, T=12:15 PM to 5:30 PM → A=High
7^	H=Low → A=High T=5:45 PM - 11 PM, C=High → A=High C=Low, E=Low, T=12:15 PM to 5:30 PM → A=High
8^	H=Medium, E=High, T=12:15 PM to 5:30 PM → A=Medium
9	H=Medium, C=Low, E=High, T=12:15 PM to 5:30 PM → A=Medium
10	H=High, C=Low, T=6:15 AM to 8 AM → A=Low
11	C=Low, T=11:15 PM to 6 AM → E=Low
12^	T=11:15 PM to 6 AM → E=Low
13^	T=8:15 AM to 12 PM → E=Medium H=High, T=5:45 PM to 11 PM → E=Medium
14	H=High, C=Low, T=5:45 PM to 11 PM → E=Medium C=High → E=Medium
15	C=High → A=High | E=Medium
16^	A=High, C=High, T=12:15 PM to 5:30 PM → H=Low | E=Medium A=Medium/High, C=Low, T=8:15 AM to 12 PM → H=Medium | E=Medium

The associative network of the energy consumption system model shown in Figure 4.3 was constructed using the associations in the groups marked with $\wedge$ in Table 4.4. Figure 4.3 shows the factors associated with the high, medium and low air temperatures (from the associations with A as the AV), the factors associated with the Medium and Low heating consumption (from the associations with H as the AV), and the factors associated with the medium and low electricity consumption (from the associations with E as the AV).

Figure 4.3 shows that E, C, H and A are related differently in different time periods. For example, in the afternoon, T = 12:15 PM to 5:30 PM, the medium heating consumption (H = Medium) along with the high electricity consumption (E = High) is associated with the medium air temperature (A = Medium), whereas in the early morning, T = 6:15 AM to 8 AM, the high heating consumption (H = High) is associated with the low air temperature (A = Low). Similarly, the most specific associations in Tables 4.2-4.3, even the most generic associations in Table 4.4 and in Figure 4.3 show that associations of T, E, C, H and A differ in different value ranges of these variables. This illustrates that the PVAD algorithm can discover full/partial-value variable relations that exist in many real-world systems.

4.2 Analysis of PVAD results on 2016 Computer Network Dataset

To protect the security of computer networks and detect anomalous network behaviors including cyber attacks, computer networks need to be monitored by collecting and analyzing network traffic data (Ye, 2008). Network anomalies are detected by first establishing profiles of normal network behaviors and detecting large deviations from profiles of normal network behaviors. A set of TCP flow data between the Internet and all hosts from a medium size enterprise (with approximately 5,000 users and 30,000 devices on computer networks) was collected on July 12, 2016 without re-

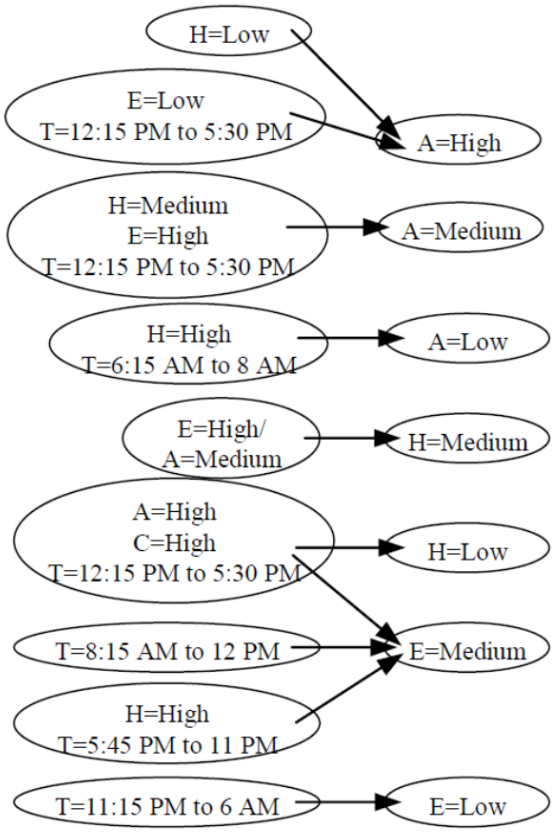

Figure 4.3: The most generic associations in the groups marked by $\wedge$ in Table 4.4 represented in an associative network.

ports of any cyber attacks or network anomalies. This data set was used to establish characteristics of normal network behaviors.

A network session between a source host and a destination host for a certain network application is defined by the source IP address, the destination IP address, the source port, the destination port, and the protocol (e.g., TCP or UDP), and contains a sequence of packets called a flow. Flow data for a flow captures features of packets in each flow (Ye, 2008). TCP flow data in this data set are bi-flow data that contain features of packets flowing in both directions from the source to the destination and from the destination to the host in each flow.

This set of bi-flow data contains 168,655 data records and it has the following

twenty data fields in each data record.

- sIP (source IP address)
- dIP (destination IP address) dPort (destination port)
- sTime (start time) eTime (end time)
- Dir (direction of flow)
- Dur (ms) (duration in ms) Dur (s) (duration in s)
- Pro (protocol = 6 for TCP)
- cRTT (time difference in ms between the first query packet and the first response packet)
- QPkts (query/source packets) QBytes (query/source bytes)
- RPkts (response/destination packets) RBytes (response/destination bytes)
- QiFlags (TCP flags in the first query packet)
- QrFlags (union of TCP flags in remaining query packets) RiFlags (TCP flags in the first response packet)
- RrFlags (union of TCP flags in remaining response packets)

Traditionally normal network behaviors are established using univariate analytical techniques (Ye, 2008). However, multivariate characteristics of normal network behaviors looking into relations of multiple network flow attributes should also be added to obtain complete profiles of normal network behaviors. In this dissertation, we illustrate how the PVAD algorithm was used to establish multivariate characteristics of normal network behaviors.

This data set was analyzed using the PVAD algorithm with $\alpha = 0.95$, $\beta = 2$, and $\gamma = 100\%$ to uncover multivariate data associations. In Step 1 of the PVAD algorithm, the following sixteen data fields were extracted from the original twenty data fields, and were transformed into categorical data.

- x1: frequency of source IP address, from sIP
- x2: frequency of destination IP address, from dIP
- x3: frequency of source port, from sPort
- x4: frequency of destination port, from dPort
- x5: start time in minute, from sTime-min
- x6: direction of flow, from Dir
- x7: duration in ms, from Dur
- x8: time difference in ms between the first query packet and the first response packet, from cRTT
- x9: query/source packets, from QPkts
- x10: query/source bytes, from QBytes
- x11: response/destination packets, from RPkts
- x12: response/destination bytes, from RBytes
- x13: TCP flags in the first query packet, from QiFlags
- x14: union of TCP flags in remaining query packets, from QrFlags
- x15: TCP flags in the first response packet, from RiFlags
- x16: union of TCP flags in remaining query packets, from RrFlags

The frequency of each field value was computed to give univariate characteristics of TCP flow data in this data set. Table 4.5 gives the dominant value of each data field. The dominant value of a data field has the highest percentage of data in the data set having this value.

The PVAD algorithm produced 1-to-1, ..., 15-to-1 associations. Table 4.6 gives the 1-to-1 associations with 100,000 or more supporting instances and the 15-to-1 association with the three largest numbers of supporting instances.

Table 4.5: The Dominant Value of Each Data Field in TCP Flow Data

Dominant Data Field Value	Percentage of Data Instances with This Value
x1=[2]	0.34562865
x2=[151, 5748]	0.860342119
x3=[1, 6]	0.567727017
x4=[212, 78165]	0.962449972
x5=29	0.144407222
x6=in	0.762266165
x7=[0.507, 1.104]	0.287255047
x8=[0.001, 0.008]	0.30560019
x9=[1, 6]	0.866182443
x10=[168, 172]	0.294346447
x11=[1, 5]	0.854068957
x12=[40, 223]	0.519041831
x13=S	0.91663455
x14=FA	0.351872165
x15=SA	0.930669117
x16=FPA	0.624517506

Table 4.6: Examples of 1-to-1 Associations and 15-to-1 Associations for Computer Network Data

Association	Number of Supporting Instances
x2=[151, 5748]→x4=[212, 78165]	143429
x2=[151, 5748]→x15=SA	141269
x2=[151, 5748]→x13=S	138002
x6=in→x2=[151, 5748]	126971
x6=in→x4=[212, 78165]	125852
x6=in→x15=SA	124913
x8=[0]→x2=[151, 5748]	101980
x8=[0]→x4=[212, 78165]	101355
x8=[0]→x6=in	102245
x9=[1, 6]→x4=[212, 78165]	142464
x9=[1, 6]→x11=[1, 5]	142645
x11=[1, 5]→x4=[212, 78165]	140638
x11=[1, 5]→x9=[1, 6]	142645
x13=S→x4=[212, 78165]	150458

Continued on next page

Table 4.6: Examples of 1-to-1 Associations and 15-to-1 Associations for Computer Network Data (Continued)

Association	**Number of Supporting Instances**
x13=S→x15=SA	153412
x15=SA→x4=[212, 78165]	153144
x15=SA→x13=S	153412
x16=FPA→x13=S	102804
x16=FPA→x15=SA	105008
x16=FPA→x4=[212, 78165]	102572
x1=[2], x3=[1,6], x4=[212, 78165], x5=29, x6=in, x7=[0.049, 0.506], x8=[0], x9=[1, 6], x10=[168, 172], x11=[1, 5], x12=[40, 223],x13=S, x14=FA, x15=SA, x16=FA → x2=[151, 5748]	1329
x2=[151, 5748], x3=[1,6], x4=[212, 78165],x5=39, x6=out, x7=[0, 0.048], x8=[0.001,0.008], x9=[1, 6], x10=[173, 624], x11=[1, 5], x12=[224, 1561], x13=S, x14=FPA, x15=SA, x16=FPA → x1=[147]	303
x1=[178, 8872], x2=[151, 5748], x3=[1,6], x5=39, x6=out, x7=[0, 0.048], x8=[0.001, 0.008], x9=[1, 6], x10=[173, 624], x11=[1, 5], x12=[224, 1561], x13=S, x14=FPA, x15=SA, x16=FPA → x4=[212, 78165]	303

4.3 Analysis of PVAD results on Intrusion Detection Evaluation Dataset

Network anomaly detection aims at detecting anomalies which manifest deviations from normal network flows and behaviors, including an ever-evolving variety of new network intrusions/attacks whose signatures have not been captured from their past occurrences (Ye, 2008; Chandola *et al.*, 2009). Network anomaly detection is a fundamental part of day-to-day operations for Internet Service Providers (ISPs) and enterprises to maintain the efficiency and reliability of computer networks. Building Network Anomaly Detectors (NADs) requires our knowledge about robust measures of network flows which bring out differences in network flows of benign network activities and network attacks.

This dissertation presents an analytical study of network flow data in benign

network activities and network attacks provided by the Canadian Institute of Cybersecurity. Section 4.3.1 presents network flow data of benign network activities and network attacks analyzed in this study. Section 4.3.2 describes univariate and multivariate data analyses of network flow data. Section 4.3.3 gives analytical results and measures of network flows derived from analytical results to detect differences between benign network activities and network attacks. Section 4.3.4 provides an overview of the research findings.

4.3.1 Network Flow Data Of Benign Network Activities And Network Intrusions

The Intrusion Detection Evaluation Dataset (CICIDS2017) from Canadian Institute of Cybersecurity (http://www.unb.ca/cic/) is used in this study to investigate univariate and multivariate measures of network flow data that can be used to detect network flows of network attacks. Network flow data in this data set is collected on a testbed system with two separate networks including a victim network and an attack network (Sharafaldin *et al.*, 2018). The attack network has one router, one switch and four PCs. The victim network has three servers, one firewall, two switches and ten PCs. All incoming and outgoing network traffic to the victim network is captured through one port in the main switch of the victim network. This dataset contains data of benign and attack network traffic which is captured over five days, Monday to Friday (Sharafaldin *et al.*, 2018). Benign and intrusion network traffic on Monday, Tuesday and Wednesday is analyzed in this study and is described in this section.

Monday traffic includes only benign traffic. All benign network traffic is generated using the B-Profile which is based on profiling the abstract behavior of 25 users involving five protocols of HTTP, HTTPS, FTP, SSH, and email protocols (Sharafaldin *et al.*, 2017). For each protocol, the B-Profile uses the packet size distribution of the protocol, the number of packets per flow, patterns in the payload, the size of the

payload, and the request time distributions of the protocol to generate benign events on the testbed infrastructure (Sharafaldin *et al.*, 2017). Benign network traffic is also generated on each day of Tuesday and Wednesday as the background traffic on the testbed infrastructure.

Attacks are implemented on the testbed infrastructure on Tuesday and Wednesday as follows:

- Tuesday: brute force attacks of password cracking (FTP-Patator in the morning, and SSH-Patator in the afternoon)
- Wednesday: DoS attacks (Slowhttptest, Slowloris, GoldenEye and Hulk in the morning, and Heartbleed in the afternoon)

83 attributes of network flows are extracted from pcap files of captured packet data. The first packet of a network flow is considered from the source IP to the destination IP and determines the direction of the network flow. A TCP flow is usually terminated upon the connection teardown by a FIN packet. A UDP flow is terminated by a flow timeout. This study uses 72 attributes out of these 83 attributes since some attributes are not useful in determining differences in benign network flows and attack network flows. For example, the timestamp attribute is not useful as times of benign network traffic and attack network traffic in this dataset are not like event times in the real world. Table 4.7 lists these 72 attributes and their corresponding variables for each network flow. In Table 4.7, the following acronyms are used:

IAT: inter-arrival time representing the time between two packets of a flow,
Fwd: forward,
Bwd: backward,
Init: initial,

Win: window,

Pkt: packet,

Seg: segment,

Std: standard deviation,

Max: maximum,

Min: minimum.

Table 4.7: 72 attributes of network flows used in this study

Variable	Attribute	Variable	Attribute	Variable	Attribute
x1	Source IP Address	x25	Fwd IAT Total	x49	ACCount ACK Flag Count
x2	Source Port	x26	Fwd IAT Mean	x50	URG Flag Count
x3	Destination IP Address	x27	Fwd IAT Std	x51	ECE Flag Count
x4	Destination Port	x28	Fwd IAT Max	x52	Down/Up Ratio
x5	Protocol	x29	Fwd IAT Min	x53	Average Packet Size
x6	Flow Duration	x30	Bwd IAT Total	x54	Avg Fwd Segment Size
x7	Total Fwd Packets	x31	Bwd IAT Mean	x55	Avg Bwd Segment Size
x8	Total Bwd Packets	x32	Bwd IAT Std	x56	Fwd Header length
x10	Total Length of Bwd Packets	x34	Bwd IAT Min	x58	Subflow Fwd Bytes
x11	Fwd Packet Length Max	x35	Fwd PSH Flags	x59	Subflow Bwd Packets
x12	Fwd Packet Length Min	x36	Fwd Header Length	x60	Subflow Bwd Bytes
x13	Fwd Packet Length Mean	x37	Bwd Header Length	x61	Init Win bytes forward
x14	Fwd Packet Length Std	x38	Fwd Packets/s	x62	Init Win bytes backward
x15	Bwd Packet Length Max	x39	Bwd Packets/s	x63	Act data pkt fwd
x16	Bwd Packet Length Min	x40	Min Packet Length	x64	Min seg size forward

Continued on next page

Table 4.7: 72 attributes of network flows used in this study (Continued)

Variable	Attribute	Variable	Attribute	Variable	Attribute
x17	Bwd Packet Length Mean	x41	Max Packet Length	x65	Active Time Mean
x18	Bwd Packet Length Std	x42	Packet Length Mean	x66	Active Time Std
x19	Flow Bytes/s	x43	Packet Length Std	x67	Active Time Max
x20	Flow Packets/s	x44	Packet Length Variance	x68	Active Time Min
x21	Flow IAT Mean	x45	FIN Flag Count	x69	Idle Time Mean
x22	Flow IAT Std	x46	SYN Flag Count	x70	Idle Time Std
x23	Flow IAT Max	x47	RST Flag Count	x71	Idle Time Max
x24	Flow IAT Min	x48	PSH Flag Count	x72	Idle Time Min

4.3.2 Methods Of Univariate And Multivariate Data Analyses

Data of one activity (benign or attack) on each day is taken as one individual data set. For example, benign data on Monday is one data set, benign data on Tuesday is another data set, and FTP-Patator attack data on Tuesday is a different set. In each data set, network flow variables x1 to x5 have categorical values, and all other network flow variables of 72 variables in Table 4.7 are numeric variables with numeric values. For each numeric data variable, we transform the numeric data of the variable into categorical data. Then we perform both the univariate analysis and the multivariate analysis of categorical data by analyzing the frequency distribution of categorical values of each network flow variable and using the PVAD (Partial-Value Association Discovery) algorithm (Ye, 2018, 2016) to obtain 1-to-1 data associations of all network flow variables. The details of transforming numeric data to categorical data, univariate data analysis, and multivariate data analysis are provided below.

Table 4.8: Examples of *dd* values showing distribution differences

Wednesday Benign			Wednesday Slowhttp Attack			
Variable Value	Freq-uency	%	Variable Value	Freq-uency	%	Absolute Difference in Percentage
x1=g1_[1, 758]	60606	0.1377			0.00	0.14
x1=g2_[829, 24634]	105846	0.2405	x1= g2_[2213, 7950]	5499	1.00	0.76
x1= g3_[25018, 60303]	273579	0.6217			0.00	0.62
						Sum: 1.52 = *dd*
x20=Infinity	348	0.0008			0.00	0.00
x20= g1_[-2000000, 11904.7619]	290334	0.6598	x20= g1_[-2000000, 11904.7619]	5164	0.94	0.28
x20= g2_[11940.2985, 3000000]	149349	0.3394	x20= g2_[11940.2985, 3000000]	335	0.06	0.28
						Sum: 0.56 = *dd*
x12=g1_[0, 258]	439669	0.9992	x12=g1_[0, 258]	4898	0.89	0.11
x12=g2_[261, 680]	80	0.0002	x12=g2_[261, 680]	326	0.06	0.06
x12= g3_[681, 1460]	167	0.0004	x12=g3_[681, 1460]	82	0.01	0.01
x12= g4_[1472, 2065]	115	0.0003	x12= g4_[1472, 2065]	193	0.04	0.03
						Sum: 0.22 = *dd*
x18=g1_[0, 193.2924]	357135	0.8116	x18= g1_[0, 193.2924]	5119	0.93	0.12
x18= g2_[193.3699, 1102.3637]	78416	0.1782	x18= g2_[193.3699, 1102.3637]	138	0.03	0.15
x18= g3_[1102.4398, 3160.9941]	4480	0.0102	x18= g3_[1102.4398, 3160.9941]	242	0.04	0.03
						Sum: 0.31 = *dd*

Transformation of numeric data to categorical data

For each data set, we transform each numeric variable to a categorical variable. x5 is a categorical variable indicating the protocol of a network flow and has only a few categorical values. No data transformation is needed for x5. Although network flow variables x1 to x4 are categorical variables with categorical values, each of these variables has a large number of categorical values. For example, each value of x1 represents a source IP address, and there are a large number of source IP addresses in each data set. Because the frequency of each source IP address is more relevant than the source IP address itself for network anomaly detection, we compute the frequency of each x1 value, that is, each source IP address in each data set. The source IP address of x1 in each data record is then replaced by the frequency value of this source IP address for the data set. For example, if a source IP address occurs 300 times in the data set and a data record has this source IP address as the x1 value, the x1 value is now assigned to 300 – the frequency of this IP address in this data set. Similarly, x3 represents the destination IP address, and the x3 value in each data set is replaced by the frequency value of the destination IP address in the data record. The variable, x2, represents the source port. If x2 has a system port (in the range of 0-1023) in a data record, the x2 value is kept since the use of a specific system port is important for network anomaly detection. If x2 has a non-system port in a data record, the frequency of this non-system port in the data set is calculated, and the x2 value is replaced by the frequency of this non-system port. The variable, x4, representing the destination port, is transformed in the same way of transforming x2. x1 to x4 with frequency values are now numeric variables.

Hence, except x5, all network flow variables are numeric variables. For each numeric variable and its numeric values, the PVAD algorithm (Ye, 2018, 2016) is used

to transform numeric values into categorical values. First of all, the set of Monday benign data is transformed to obtain categorical variables for this data set. We consider benign network traffic on Monday as the normal profile to detect network anomalies. Hence, categorical values defined using the Monday benign data set are used to transform each of data sets for Tuesday benign, Wednesday benign, Tuesday FTP-Patator attack, Tuesday SSH-Patator attack, Wednesday Slowhttp attack, Wednesday Slowloris attack, Wednesday GoldenEye attack, Wednesday Hulk attack, and Wednesday Heartbleed attack. When a numeric variable in a data set of the benign or an attack on Tuesday and Wednesday has a numeric value falling outside value ranges of categorical values defined from the benign data set of Monday, the PVAD algorithm puts the numeric value into a category value whose value range is closest to the numeric value. For example, if we have two categorical values with value ranges of [1, 2.1] and [2.3, 4], a numeric value of 2.15 is closer to 2.1 than 2.3 and thus 2.15 is transformed to the categorical value with the range of [1, 2.1].

Univariate data analysis

For each data set with all categorical variables, frequencies of categorical values for each individual variable are computed, which gives us the frequency distribution of the variable which is a univariate data characteristic as it represents a data characteristic of one individual variable. A measure of distribution difference called *dd* is developed to measure the difference in frequency distributions of the same variable among data sets of Monday benign, Tuesday benign, Wednesday benign, Tuesday FTP-Patator attack, Tuesday SSH-Patator attack, Wednesday Slowhttp attack, Wednesday Slowloris attack, Wednesday GoldenEye attack, Wednesday Hulk attack, and Wednesday Heartbleed attack. *dd* between one activity y (benign or attack) and another activity z is computed as follows:

$$dd = \sum_{i} |v_{iy} - v_{iz}| \tag{4.1}$$

where v_{iy} is the percentage of the ith categorical value for activity y, v_{iz} is the percentage of the ith categorical value for activity z, and the percentage of the ith categorical value is computed by having the frequency of this categorical value divided by the total number of data records in the data set. Table 4.8 gives four examples of computing dd between Wednesday benign and Wednesday Slowhttp attack for x1, x20, x12 and x18. In the example of x1, $g2\_[829, 24634]$ is not the dominant value of x1 under Wednesday benign, whereas the same value is the dominant and only value of x1 under Wednesday Slowhttp attack. The *dd* value for this difference between two frequency distributions of x1 is 1.52. In the example of x20, $g1\_[-2000000, 11904.7619]$ is the dominant value under both Wednesday benign and Wednesday Slowhttp attack. However, this value is taken by 66% of network flows in Wednesday benign but 94% of network flows in Wednesday Slowhttp attack. The *dd* value for this difference between two frequency distributions is 0.56. The other two examples of x12 and x18 show similar frequency distributions between Wednesday benign and Wednesday Slowhttp Attack with the dd values of 0.22 and 0.31. In this study, we consider that two frequency distributions are different if the dd value is ≥ 0.5, and two frequency distributions are similar if the *dd* value is < 0.5.

Multivariate data analysis

The PVAD algorithm with the parameter setting of $\alpha = 0.95$, $\beta = 10$, and $\gamma = 0.95$ is applied to each data set to obtain 1-to-1 associations of two variables and their values, in the form of CV $\rightarrow$ AV, where CV denotes a Conditional variable's Value and AV denotes an Associative variable's value. For example, if the PVAD algorithm

discovers the following association:

$$x17 = g1\_[0, 75.3333] \rightarrow x18 = g1\_[0, 192.333]$$

it means that among all network flows whose backward packet length mean (x17) falls in the range of [0, 75.3333] (the first categorical value of x17), 95% or more of them ($\alpha = 0.95$) have their backward packet length standard deviation (x18) fall in the range of [0, 192.333] (the first categorical value of x18). That is, the first categorical value of x17 is closely associated with the first categorical value of x18. Then 1-to-1 associations are compared among benign and attacks to look into:

- associations present in benign but are absent in attacks,
- associations present in attacks but are absent in benign.

4.3.3 Univariate And Multivariate Measures Of Network Flows Derived From Analytical Results

This section presents results of univariate data analysis and multivariate data analysis. Univariate and multivariate measures of network flows are derived from analytical results to detect differences of attack network traffic from benign network traffic, and are reported in this section.

Univariate Measures of Attacks/Anomalies

For each variable, Monday benign, Tuesday benign and Wednesday benign have similar distributions with dd values < 0.5. This demonstrates the reliability and effectiveness of dd in measuring frequency distribution differences. Table 4.9 gives variables whose frequency distribution under each attack is different from that under the Monday benign.

To build a network anomaly detector, we can use frequency distributions of variables from Monday as the norm. We can use a sliding time window from the present time to a time in the past (e.g., a 10-minute time window or a one-hour time window) to get a sample of network flows at every given time (e.g., every minute or every hour), analyze frequency distributions of variables for network flows in the sample, and monitor the number and pattern of variables whose frequency distribution is different from the norm. An increase in the number of variables with distribution differences may need a close attention of security analysts or system administrators to look into the pattern of variables with frequency distribution differences (i.e., what specific variables have the difference) and investigate whether or not an anomaly should be detected. Especially, if the pattern of variables with frequency distribution differences matches closely with the specific pattern of a known attack or anomaly (e.g., the pattern of variables shown in seven attack columns in Table 4.9), the security analyst or system administrator can be alarmed with this particular attack or anomaly.

Network flow variables in each attack column in Table 4.9 are compared with network flow variables extracted for each attack using the feature selection of the random forest technique in (Sharafaldin *et al.*, 2018). Network flow variables, which are selected by both our method based on frequency distribution differences and the feature extraction method of the random forest technique, are underlined and highlighted in bold in Table 4.9. Network flow variables which are selected by the feature selection method of the random forest technique but not by our frequency distribution difference method are listed in the last row of Table 4.9. Hence, our frequency distribution difference method discovers some of the network flow variables which are selected by the feature selection method of the random forest technique. Moreover, our frequency distribution difference method uncovers many more network

flow variables which are useful to detect anomalies or attacks. For example, there are huge differences between the frequency distributions of the Monday benign and each attack in the frequency distributions of x1, x3, x4, and x5 representing the source IP address, destination IP address, protocol, and flow duration. However, x1, x3, x4, and x5 are not selected by the feature selection of the random forest technique because these variables have only one original numeric value in each attack whereas the benign data has this same value and other values, making it impossible to discriminate each attack from the benign using the classification of the random forest technique. Statistically, the frequency distribution gives the complete picture of data for a variable in comparison with any other univariate feature for data discrimination including that used by the random forest technique.

Table 4.9: Variables with different frequency distribution under each attack from benign

Variable	Tue. FTP-Patator	Tue. SSH-Patator	Wed. Slowhttp	Wed. Slowloris	Wed. Goldeye	Wed. Hulk	Wed. Heart-bleed
x1	x	x	x	x	x	x	x
x3	x	x	x	x	x	x	x
x4	x	x	x	x	x	x	x
x5	x	x	x	x	x	x	x
x6	x	x	**x**	**x**	x	**x**	x
x15					x	x	
x17	x		x	x	x	x	x
x18		x			**x**	**x**	
x20			x	x	x		x
x21			**x**	**x**		x	
x22	x	x	x	x	x	**x**	
x23	x	x	x	x	x	x	
x25	x	x	x	x	x	x	
x26	x	x	x	x	x	x	
x27	x	x	x	x	x	x	
x28	x	x	x	x	x	x	
x30	x	x		x	x		
x31				**x**	x		
x32	x	x			x		
x33	x			x			

Continued on next page

Table 4.9: Variables with different frequency distribution under each attack from benign (Continued)

Variable	Tue. FTP-Patator	Tue. SSH-Patator	Wed. Slowhttp	Wed. Slowloris	Wed. Goldeye	Wed. Hulk	Wed. Heart-bleed
x34				x			
x35	**x**						
x36							x
x37							x
x38			x	x	x		
x41				x		x	
x42				x		x	
x43		x		x		x	
x44				x		x	
x46	**x**			x			
x48	x	x	x	x	x		
x49	x	**x**				x	x
x50		x					
x53					x	x	
x55					x	x	
x56							x
x60							x
x61	**x**	**x**	x	x	x		
x65			**x**	x			
x66			x	x			
x68			**x**				
x69			x	x	x	x	
x70			x	x			
x71			x	x	x	x	
x72			x	x	x	x	
Total # of Variables	20	2	23	32	27	25	12
Missing from variables in (Sharafaldin *et al.*, 2017)	x38	x9, x58		x24	x21, x24, x29		x6, x9, x18, x58

Multivariate Measures of Attacks/Anomalies

Table 4.10 gives variables in CVs of 1-to-1 associations in each benign activity and each attack. In Table 4.10, M stands for Monday, T stands for Tuesday, and W

stands for Wednesday. Unlike benign network flows involving a large range of diverse activities by a large number of users, each attack usually involves a much narrower range of activities with specific goal(s). Hence, 1-to-1 associations of benign network flows have almost all network flow variables in CVs (i.e., 71 variables for Monday benign, 71 variables for Tuesday benign, and 70 for Wednesday benign, out of totally 72 variables) due to a variety of data associations in a variety of network flows, whereas fewer variables (i.e., 9 to 45 variables for Tuesday attacks and Wednesday attacks) are involved in CVs of 1-to-1 associations due to more consistent network flows during attacks.

Hence, the number of variables in CVs of 1-to-1 associations can be used as a multivariate measure to detect attacks/anomalies. This multivariate measure can be used together with the univariate measure in Section 4.3.3 in a network anomaly detector. Specifically, we can use a sliding time window from the present time to a time in the past (e.g., a 10-minute time window or a one-hour time window) to get a sample of network flows at every given time (e.g., every minute or every hour), extract 1-to-1 associations from network flow data in the sample from the sliding window using the PVAD algorithm, and monitor the number of variables in CVs of those 1-to-1 associations. When this number of variables in CVs drops to smaller than the usual number in the benign condition, security analysts or system administrators will be alerted for an anomaly/attack. They may look into variables in CVs to see whether or not they are similar to the group of variables for any known attack (e.g., the group of variables in each of the last seven columns for seven attacks in Table 4.10). If the group of variables matches closely with the group of variables associated with a known attack or anomaly, security analysts or system administrators can be alarmed with this particular attack or anomaly.

In addition to the number of variables in CVs of 1-to-1 associations, specific

variables present in CVs of 1-to-1 associations of benign network flows but absent in CVs of 1-to-1 associations of attack or anomaly network flows can be another multivariate measure of attacks or anomalies. For example, the following variables are in CVs of 1-to-1 associations of benign network flows on Monday, Tuesday and Wednesday but absent in CVs of 1-to-1 associations of all seven attacks: x1 – x5, x7 – x8, x11, x14, x16, x47, x51, x57 – x59, and x63. If a majority of these variables do not show up in the CVs of 1-to-1 associations from network flows in a sliding window, security analysts and system administrators can be alerted for an anomaly/attack.

It is also found that 1-to-1 associations of Tuesday FTP-Patator and Tuesday SSH-Patator attacks involve mostly associations of the following variables:

- x6: Flow Duration
- x22: Flow IAT Std
- x23: Flow IAT Min
- x25: Fwd IAT Total
- x26: Fwd IATMean
- x27: Fwd IAT Std
- x28: Fwd IAT Max
- x30: Bwd IAT Total
- x33: Bwd IAT Max
- x35: Fwd PSH Flags
- x38: Fwd Packets/s
- x48: PSH Flag Count
- x49: ACK Flag Count
- x61: Init_Win_Bytes_forward.

Table 4.10: Variables in CVs of 1-to-1 associations in benign activities and attacks

Variable	Mon. Benign	Tue. Benign	Wed. Benign	Tue. FTP-Patator	Tue. SSH-Patator	Wed. Slowhttp	Wed. Slowloris	Wed. Goldeye	Wed. Hulk	Wed. Heartbleed
x1	x	x	x							
x2	x	x	x							
x3	x	x	x							
x4	x	x	x							
x5	x	x	x							
x6	x	x	x	x	x	x	x	x		x
x7	x	x	x							
x8	x	x	x							
x9	x	x	x	x	x	x	x	x		x
x10	x	x	x							x
x11	x	x	x							
x12	x	x	x		x					
x13	x	x	x		x					
x14	x	x	x							
x15	x	x	x					x		
x16	x	x	x							
x17	x	x	x		x	x	x	x		
x18	x	x	x		x	x	x	x		
x19	x	x	x	x		x				
x20	x	x	x	x	x	x	x			
x21	x	x	x			x	x	x		
x22	x	x	x	x	x	x	x	x		
x23	x	x	x	x	x	x	x	x		
x24	x	x	x			x		x		
x25	x	x	x	x	x	x	x	x		
x26	x	x	x	x	x	x	x	x		
x27	x	x	x	x	x	x	x	x		
x28	x	x	x	x	x	x	x	x		
x29	x	x	x			x	x	x		
x30	x	x	x	x	x	x	x	x		
x31	x	x	x			x	x	x		
x32	x	x	x	x	x	x	x	x		
x33	x	x	x	x	x	x	x	x		x
x34	x	x	x		x	x	x			
x35	x	x	x	x	x	x	x			
x36	x	x	x							x
x37	x	x	x							x
x38	x	x	x	x	x	x	x		x	
x39	x	x	x	x	x	x	x		x	
x40	x	x	x			x				
x41	x	x	x					x	x	
x42	x	x	x			x		x	x	
x43	x	x	x		x	x	x	x	x	
x44	x	x	x			x		x	x	

Continued on next page

Table 4.10: Variables in CVs of 1-to-1 associations in benign activities and attacks (Continued)

Variable	Mon. Benign	Tue. Benign	Wed. Benign	Tue. FTP-Patator	Tue. SSH-Patator	Wed. Slowhttp	Wed. Slowloris	Wed. Goldeye	Wed. Hulk	Wed. Heartbleed
x45	x	x	x						x	
x46	x	x	x	x	x	x	x			
x47	x	x	x							
x48	x	x	x	x	x	x	x	x	x	x
x49	x	x	x	x	x	x	x	x	x	x
x50	x	x	x		x	x	x	x	x	
x51	x	x	x							
x52										
x53	x	x	x			x		x	x	
x54	x	x	x			x				
x55	x	x	x			x		x	x	
x56	x	x	x							x
x57	x	x	x							
x58	x	x	x							
x59	x	x	x							
x60	x	x	x							x
x61	x	x	x	x	x	x	x	x	x	x
x62	x	x	x			x	x		x	
x63	x	x	x							
x64	x	x								
x65	x	x	x			x	x	x		
x66	x	x	x			x	x	x		
x67	x	x	x			x	x	x		
x68	x	x	x			x	x	x		
x69	x	x	x			x	x	x	x	
x70	x	x	x			x	x	x	x	
x71	x	x	x			x	x	x	x	
x72	x	x	x			x	x	x	x	
Total # of Variables in CVs	71	71	70	19	22	45	35	36	18	9

Such data associations are not present in benign network flows. In the above variables, only x38, x49 and x61 are among the variables extracted using the feature selection of the random forest technique (Sharafaldin *et al.*, 2017).

4.3.4 Summary

Based on the univariate and multivariate analyses of benign and attack network flow data from Canadian Institute of Cybersecurity, the following univariate and multivariate measures are established to detect network attacks/anomalies:

- Univariate measure: the number of variables with $dd \geq 0.5$ from the benign with the greater number indicating the greater likelihood of an attack/anomaly,
- Multivariate measures: the number of variables in CVs of 1-to-1 associations with a smaller number indicating the more likelihood of an attack/anomaly.

Two specific attacks, FTP-Patator and SSH-Patator attacks, are also characterized by the presence of x1 – x5, x7 – x8, x11, x14, x16, x47, x51, x57 – x59, and x63 in CVs of 1-to-1 associations.

4.4 Analysis of PVAD results on ASU Fall 2009 Engineering Freshmen Data

Many studies have investigated various factors of student retention and success in STEM (Sciences, Technologies, Engineering, and Math) undergraduate education, including demographics, financial aids, test scores and grades, courses and curriculums, intellectual skills and abilities, motivational factors, academic and social environments, and interventions. Existing findings on STEM retention and success are usually obtained from statistical analyses that do not model interactive, concurrent effects of multiple factors. As indicated in (Li *et al.*, 2009a), analytical techniques that can analyze and model interactive and concurrent effects of multiple factors are needed to produce a complete framework of STEM retention and success.

We obtained data of 890 undergraduate students who entered the engineering college at Arizona State University (ASU) in Fall 2009. Table 4.11 lists 51 data fields in this data set. These data fields are collected from college applications and ASU

academic records of these students. The data set has 890 instances for 890 students, respectively.

Table 4.11: Data fields in ASU 2009 Engineering Freshmen Data

x1:FA09 Entering Major	x18: SAT ACT Calculated Index Group	x35: Scholarship Dollar Amount
x2: SP10 Enrollment	x19: ALEKS Group	x36: FA09 Earned Hours
x3: FA10 University Enrollment	x20: AP Hours	x37: FA09 GPA
x4: FA10 College Enrollment	x21: Count of Any AP Hours	x38: Count of Grades D, E, and W
x5: Age	x22: Count of MatChmCse AP Hours	x39: Count of Grade A
x6: Gender	x23: Count of Mat27None AP Hours	x40: Count of Grades A and B
x7: Minority	x24: Application Earliness	x41: FA09 Math Course
x8: Underrepresented Minority	x25: Orientation Earliness	x42: FA09 Math Grade
x9: National Origin	x26: Admitted 1stProgram	x43: SP10 Earned Hours
x10: Target Market	x27: Admitted 1stPlan	x44: SP10 GPA
x11: 1st Generation	x28: Barrett Honors College	x45: E2 Camp
x12: High School	x29: ASU Residence Hall	x46: Camp
x13: High School Class Rank Percentage	x30: Local City	x47: EPICS
x14: High School ABOR GPA	x31: Live ON Campus	x48: Fulton Match
x15: High School City	x32: Live ES Community	x49: Tutor Visits
x16: High School Rating	x33: FinAid Dollar Amount	x50: Use of Tutor
x17: High School Charter	x34: Financial Need Dollar Amount	x51: Tutor Visit Group

The objective of analyzing engineering student data is to identify characteristics

of engineering students for engineering retention. That is, we want to find out what characteristics of engineering students are associated with student retention in engineering. The variable, x4, indicates engineering retention with x4 = 1 for a student enrolling in the engineering college in Fall 2010, one year after entering the engineering college in Fall 2009, or x4 = 0 for a student no longer enrolling in the engineering college in Fall 2010. By analyzing this data set, we want to examine and identify what specific values of variables other than x4 are associated with x4 = 1 and with x4 = 0. Hence, among all associations produced by the PVAD algorithm, we are interested in associations with x4 = 1 or x4 = 0 as AV.

In addition to parameter α, two other parameters, β and γ, are also needed in the PVAD algorithm. β is used to remove associations whose number of supporting instances (instances containing variable values in the numerator of Equation 2.1 is smaller than β. γ is used to remove an association with a common CV or AV that appears in more than γ of the data set. For the data set of engineering student data, α is set to 1 and 0.8, β is set to 10, and γ is set to 71%. We set $\gamma = 71\%$ because 625 students out of a total 890 students (70.22% of students) have x4 = 1 and we want to keep associations with x4 = 1 as AV.

We first examine 1-to-1 associations with x4 = 1 as AV when α is set to 1, there are no 1-to-1 associations with AV of x4 = 1.

The list below gives CVs of 1-to-1 associations with AV of x4 = 1 for $\alpha = 0.8$ with the number of supporting instances ≥ 27 (the number shown in parentheses for each CV). A student may be a supporting instance for several associations.

Group 1: not poor ASU performance

- x38: Cnt_DEW = none (450)
  similar CVs in this group:

- x37 $\geq$ 3.5 (292)
- x39 = [4, 6] (280)
- x40 $\geq$ 6 (236)
- x42 $\geq$ B+ (216)
- x44 = [3.5, 4] (171)
- x43 = [17, 20] (89)
- x36 = [17, 20] (77)

Group 2: good high school performance

- x20: AP_Hours = [3, 38] (253, 53)
- x21: Cnt_AnyAP = 1 (253, 53)

  similar CV in this group:
- x22 = 1 (205)
- x23 = 1 (200)
- x41 = Calculus for Engineer II, Calculus for Engineer III, Modern Differential Equations (212)
- x28 = 1 (152)
- x19 = 120-130 (82)
- x33 $\geq$ 23000 (46)
- x35 $\geq$ 23000 (33)

Group 3: demographics and social-economic background

- x12 = Desert Vista High School, Dobson High School, Chandler High School, Desert Mountain High School, Mountain Pointe High School, Brophy College Preparatory, in other words, certain high schools in areas of families with social-economic advantages and/or engineering background (95, 9)
- x7 = Asian/Pacific Islander (73, 6)
- x34 $\leq$ 1000 (27, 1)

Group 4: student origin

- x15 = Chandler (45, 1)
- x49: Tutor_Visits = [2, 21] (71, 4)
- x50: Cnt_UseTutor = 1 (55, 5)
- x51 = 3 to 9 Visits (38, 4)

If the second number is included in parentheses for an association, it is the number of instances supporting this association only without supporting any previous group of associations. For example, x20: AP Hours = [3, 38] (253, 53), indicates that there is a total of 253 supporting instances for the association with this CV, and among 253 instances there are only 53 instances supporting this association without supporting any association in group 1.

There are 243 instances supporting both group 1 and group 2 of associations along with some associations in other groups, 296 instances supporting group 1 of associations along with some in other groups but not group 2, 10 instances supporting group 2 of associations along with some in other groups but not group 1, and only 29 instances supporting other groups but not group 1 or group 2. That is, group 1 of associations presents the most dominant CVs for engineering retention of AV: x4 = 1. In group 1, x38: Cnt_DEW = none, is the most dominant CV with the largest number of supporting instances, 450. Hence, not having any poor grade of D, E or W (which does not earn any credit) in the first year at ASU is the most dominant student characteristic for engineering retention.

All 1-to-1 associations for AV of x4 = 1 involve a total 578 supporting instances (students). Because 625 students have x4 = 1, there are 625-578=47 students with x4 = 1 but not among the supporting instances of 1-to-1 associations for AV of x4 = 1. To reveal uncommon characteristics of engineering retention for a small number of 47 students not covered by 1-to-1 associations for AV: x4 = 1, we computed

frequencies and percentages of variable values for these 47 students in the comparison with frequencies and percentages of variable values for 578 students. The percentage of 47 students for a given variable value is computed by dividing the frequency of 47 students for the given variable value by 47. The percentage of 578 students for a given variable value is computed by dividing the frequency of 578 students for the given variable value by 578. The percentage of 47 students is divided by the percentage of 578 students to give a ratio. Table 4.12 gives frequencies and percentages of variable values for these 47 students in comparison with frequencies and percentages of variable values for 578 students, for variable values with the frequency of 47 students ≥ 5 and the ratio of the 47 student percentage to the 578 student percentage ≥ 2. A large ratio ≥ 2 for a variable value indicates that the variable value is 2 times or more present among 47 students than 578 students.

Table 4.12: Frequencies and Percentages of 47 Students with x4 = 1

Variable Value	Frequency, Percentage of 47 Students	Frequency, Percentage of 578 Students	Ratio of 47 Student Percentage to 578 Student Percentage
x1:FA09 Entering Major=Civil Engineering	9, 19.15%	50, 8.65%	2.21
x1=Computer Systems Engineering	6, 12.77%	33, 5.71%	2.24
x7:Minority=International	6, 12.77%	17, 2.94%	4.34
x10:TargetMarket=CA	6, 12.77%	21, 3.63%	3.51
x10=International	6, 12.77%	16, 2.77%	4.61
x12:HighSchool=Foreign High School	5, 10.64%	13, 2.25%	4.73
x13:High School Class Rank Pct=31 to 60	11, 23.40%	53, 9.17%	2.55
x14:High School ABOR GPA=[2, 3)	12, 25.53%	27, 4.67%	5.47
x14=none	7, 14.89%	18, 3.11%	4.78
x15:High School City=None	23, 48.94%	122, 21.11%	2.32
x17: High School Charter=None	5, 10.64%	13, 2.25%	4.73

Continued on next page

Table 4.12: Frequencies and Percentages of 47 Students with x4 = 1 (Continued)

Variable Value	Frequency, Percentage of 47 Students	Frequency, Percentage of 578 Students	Ratio of 47 Student Percentage to 578 Student Percentage
x18: SAT ACT Index Group=103 – 107	6, 12.77%	27, 4.67%	2.73
x18=108 – 110	5, 10.64%	25, 4.33%	2.46
x18=94 – 102	5, 10.64%	8, 1.38%	7.69
x18=No Index	7, 14.89%	30, 5.19%	2.87
x19:ALEKS Group=70- 80	5, 10.64%	12, 2.08%	5.12
x19=90-100	9, 19.15%	46, 7.96%	2.41
x24:App Earliness=(5, 8]	19, 40.43%	81, 14.01%	2.88
x25:Orient Earliness=none	10, 21.28%	57, 9.86%	2.16
x27:Admit 1st Plan=ESCSEBSE	6, 12.77%	22, 3.81%	3.35
x30:Local City=TEMPE	6, 12.77%	17, 2.94%	4.34
x33:FinAid $=0	5, 10.64%	24, 4.15%	2.56
x33=none	13, 27.66%	50, 8.65%	3.2
x34:Need $ $\geq$ 15000	16, 34.04%	97, 16.78%	2.03
x34=none	13, 27.66%	51, 8.82%	3.13
x35:Scholarship $=none	35, 74.47%	118, 20.42%	3.65
x36:FA09 Earned Hrs=[6, 11]	15, 31.91%	22, 3.81%	8.38
x37:FA09 GPA $\leq$ 2.5	26, 55.32%	33, 5.71%	9.69
x38:Count of DEW=1	29, 61.70%	102, 17.65%	3.5
x38=2	10, 21.27%	20, 3.46%	6.15
x38=3	5, 10.64%	5, 0.87%	12.3
x39:Count of A=1	21, 44.68%	74, 12.80%	3.49
x39=none	12, 25.53%	29, 5.02%	5.09
x40:Count of AB=1	5, 10.64%	1, 0.17%	61.49
x40=2	10, 21.28%	28, 4.84%	4.39
x40=3	13, 27.66%	48, 8.30%	3.33
x41:FA09 Math Class=Precalculus	13, 27.66%	69, 11.94%	2.32
x42:FA09 Math Grade=D	13, 27.66%	19, 3.29%	8.41
x42=E	6, 12.77%	9, 1.56%	8.2
x42=W	13, 27.65%	35, 6.06%	4.57
x43:SP10 Earned Hours=[6, 11]	13, 27.66%	59, 10.21%	2.71
x44:SP10 GPA$\leq$2.5	23, 48.94%	86, 14.88%	3.29
x45:E2Camp=0	17, 36.17%	71, 12.28%	2.94

Continued on next page

Table 4.12: Frequencies and Percentages of 47 Students with x4 = 1 (Continued)

Variable Value	Frequency, Percentage of 47 Students	Frequency, Percentage of 578 Students	Ratio of 47 Student Percentage to 578 Student Percentage
x46:Camp=None	17, 36.17%	71, 12.28%	2.94

44 out of 47 students have x38: Count of D, E and W grades = 1, 2, or 3, in contrast to 450 out of 578 students who have x38 = none. Although most of those 47 students have x38 = 1, 2, or 3, they still continue in engineering after one year at ASU. Many of those 47 students are international students, students with average or below average high school performance, students who did not apply for ASU early, students who have little scholarship dollars, students who have either little financial need or financial need > \$15000, and students who have one grade of A and fewer than four grades of A or B, and students who have the grade of D, E or W in their math class. Hence, among students who have 1 to 3 grades of D, E and W in their first year of college, students with the above characteristics continue in engineering.

There are no 1-to-1 associations for AV of x4 = 0 when α is set to 1. CVs of 1-to-1 associations for AV of x4 = 0, α = 0.8 are listed below with the number of supporting instances in parentheses.

- x44 = none (48)
- x43: Spring10 Earned Credit Hours = none (47)
- x38: Count of DEW = 4 or 5 (35)
- x40 = 1 (32)
- x36 = none or [1, 5] (28)

These CVs of 1-to-1 associations for AV of x4 = 0 are all related to poor ASU grade/performance, including the count of D, E and W grades is 4 or 5. Therefore,

1-to-1 associations for AV: x4 = 1 tell us that not poor ASU performance is associated with engineering retention, whereas 1-to-1 associations for AV: x4 = 0 tell us that poor ASU performance is associated with students leaving engineering.

These 1-to-1 associations for AV of x4 = 0 have 80 supporting instances among a total of 265 students who have x4 = 0. Hence, there are 265 – 80 = 185 students with x4 = 0 but not being covered by the 1-to-1 associations for x4 = 0. Table 4.13 gives frequencies and percentages of variable values for these 185 students in comparison with frequencies and percentages of variable values for 80 students, for variable values with the frequency of 185 students ≥ 10 and the ratio of the 185 student percentage to the 80 student percentage ≥ 2. A large ratio ≥ 2 for a variable value indicates that the variable value is 4 times or more present among 185 students than 80 students.

Table 4.13: Frequencies and Percentages of 185 Students with x4 = 0

Variable Value	Frequency, Percentage of 185 Students	Frequency, Percentage of 80 Students	Ratio of 185 Student Percentage to 80 Student Percentage
x1=Aerospace Engr (Astronautics)	6, 3.24%	1, 1.25%	2.59
x1=Chemical Engineering	17, 9.19%	3, 3.75%	2.45
x3=1	133, 71.89%	24, 30%	2.4
x6=F	133, 71.89%	24, 30%	2.4
x14=[3.7, 4.0]	76, 41.08%	11, 13.75%	2.99
x15=Gilbert	10, 5.41%	1, 1.25%	4.32
x15=Tempe	11, 5.95%	1, 1.25%	4.76
x18=129 - 146	44, 23.78%	4, 5%	4.76
x27=ESCHEBSE	15, 8.11%	3, 3.75%	2.16
x28=1	34, 18.38%	3, 3.75%	4.9
x29=Cereus Hall	28, 15.14%	3, 3.75%	4.04
x33=[7500, 9000]	51, 27.57%	10, 12.5%	2.21
x35=[9001, 23000)	27, 14.59%	1, 1.25%	11.68

Continued on next page

Table 4.13: Frequencies and Percentages of 185 Students with x4 = 0 (Continued)

Variable Value	Frequency, Percentage of 185 Students	Frequency, Percentage of 80 Students	Ratio of 185 Student Percentage to 80 Student Percentage
x36=[12, 16]	133, 71.89%	20, 25%	2.88
x37=[3.0, 3.5)	49, 26.48%	5, 6.25%	4.24
x37=[3.5, 4)	40, 21.62%	2, 2.5%	8.65
x38=1	61, 32.97%	8, 10%	3.3
x38=none	81, 43.78%	11, 13.75%	3.18
x39=2	42, 22.70%	7, 8.75%	2.59
x39=3	30, 16.22%	2, 2.5%	6.49
x39=4	22, 11.89%	1, 1.25%	9.51
x39=5	11, 5.95%	1, 1.25%	4.76
x40=4	40, 21.62%	3, 3.75%	5.77
x40=5	42, 22.90%	6, 7.5%	3.03
x40=6	33, 17.84%	1, 1.25%	14.27
x42=A	16, 8.65%	1, 1.25%	6.92
x42=B	31, 16.76%	3, 3.75%	4.47
x42=B+	9, 4.86%	1, 1.25%	3.89
x42=C+	10, 5.41%	1, 1.25%	4.32
x43=[12, 16]	118, 63.78%	13, 16.25%	3.93
x43=[17, 20]	11, 5.95%	1, 1.25%	4.76
x44≥4	13, 7.03%	1, 1.25%	5.62
x44=[2.5, 3.0)	35, 18.92%	4, 5%	3.78
x44=[3.0, 3.5)	45, 24.32%	2, 2.5%	9.73
x44=[3.5, 4)	34, 18.38%	2, 2.5%	7.35
x46=Camp 2	32, 17.30%	5, 6.25%	2.77

In addition to 80 students who left engineering with poor grades, we have 185 students who are mostly good students but left engineering for a non-engineering major at ASU. As shown in Table 4.13, among those 185 students who left engineering after one year at ASU, we have the following.

- 133 students continued for a non-engineering major in university

- 133 students are female
- Many are top students in high school
- Some are honor students
- Some have FA09 GPA in [3.0, 4)
- Many have good amounts of scholarship $
- 133 students have FA09 earned credit hours of [12, 16]
- 129 students have SP10 earned credit hours of [12, 16]
- 128 students have FA09 GPA $\geq$ 2.5
- 127 students have SP10 GPA $\geq$ 2.5
- 142 students have the count of D, E, and W grades = none or 1
- 105 students have the count of A grade $\geq$ 2
- 115 students have the count of A and B grades $\geq$4
- Many have FA09 Math grade $\geq$ C+.

Based on the above results for x4 = 1 and x4 = 0, we need to improve the retention of two types of engineering students: (1) students with poor ASU grades, and (2) students who had decent to good ASU grades but left engineering for a non-engineering major at ASU, especially female students.

The PVAD algorithm also produced 2-to-1, ..., 50-to-1 associations for AV of x4 = 1 and AV of x4 = 0. The examination of these p-to-1 associations with p = 2, ..., 50 gives us similar information about engineering retention to that from 1-to-1 associations.

4.5 Analysis of PVAD results on Common and Uncommon Characteristics of Engineering Student Retention after the First Year in University

Many studies (Gross *et al.*, 2015; Hieb *et al.*, 2015; Coletti *et al.*, 2014; Gross *et al.*, 2013; Shaw *et al.*, 2013; Tyson, 2011; Gross, 2011; Weatherton *et al.*, 2011;

Moses *et al.*, 2011; Teague *et al.*, 2018; Orr, 2019; Veletzos *et al.*, 2018; Chan-Hilton, 2019; Maccariella Jr *et al.*, 2019; Sithole *et al.*, 2017) investigated various factors of retention in STEM (Science, Technologies, Engineering, and Mathematics) education for undergraduates, including demographics (Geisinger *et al.*, 2013; Ackerman *et al.*, 2013a; Kokkelenberg and Sinha, 2010), financial aids (Gross *et al.*, 2015; Herzog, 2005; Sulaiman, 2016; Orr, 2019), test scores and grades in high school (Hieb *et al.*, 2015; Weatherton *et al.*, 2014; Ackerman *et al.*, 2013b; Hall *et al.*, 2013; Geisinger *et al.*, 2013; Ackerman *et al.*, 2013a; Shaw *et al.*, 2013; Kauffmann *et al.*, 2007; Herzog, 2005), test scores and grades in university/college (Hieb *et al.*, 2015; Coletti *et al.*, 2014; Tyson, 2011; García-Ros *et al.*, 2019; Orr, 2019; Veletzos *et al.*, 2018), courses and curriculums (Coletti *et al.*, 2014; Jones *et al.*, 2014b; Camacho and Lapuz, 2014; Ackerman *et al.*, 2013b), intellectual skills and abilities (Hieb *et al.*, 2015; Coletti *et al.*, 2014; Weatherton *et al.*, 2014; Moses *et al.*, 2011; Kauffmann *et al.*, 2008), motivational factors and self-efficacy (Hieb *et al.*, 2015; Jones *et al.*, 2014b; Ackerman *et al.*, 2013b; White and Massiha, 2016; Martin III, 2018), academic and social environments (Coletti *et al.*, 2014; Camacho and Lapuz, 2014; Geisinger *et al.*, 2013; Sithole *et al.*, 2017), and interventions (Desai and Stefanek, 2017; Fuesting, 2019; Uddin and Johnson, 2019). These studies identified factors that were commonly shared by a significant number of undergraduate students who achieved STEM retention and thus presented common student characteristics of STEM retention. However, there are uncommon/untypical undergraduate students including Type 1 students who do not have common student characteristics of STEM retention but still achieve STEM retention, and Type 2 undergraduate students who have common student characteristics of STEM retention but do not achieve STEM retention. There is little understanding of the characteristics of those uncommon/untypical students in STEM retention. Characteristics of Type 1 uncommon/untypical students will enable us to

work on attracting and recruiting such students to STEM fields although they do not fit into common profiles of students in STEM. Characteristics of Type 2 uncommon/untypical students will help us design intervention mechanisms to address and correct elements that drive them to leave STEM.

This study focuses on engineering retention and aims at identifying characteristics of uncommon/untypical students in engineering retention, called uncommon characteristics of engineering retention. Our study took two steps to first identify common student characteristics of engineering retention and then used common student characteristics of engineering retention to determine uncommon/untypical students and identify characteristics of uncommon/untypical students as uncommon student characteristics of engineering retention. The identification of uncommon student characteristics of engineering retention will guide us to carry out recruitment and interventions to help more students achieve engineering retention, thus broadening the participation of more students in engineering fields.

4.5.1 Data Sets and Data Analyses

This section describes the data sets and data analyses performed to identify common and uncommon characteristics of engineering retention.

Data Sets

In this study, we collected and analyzed six large sets of engineering student data covering six classes of students who entered Ira A. Fulton Schools of Engineering at ASU in the Fall of 2009, 2011, 2014, 2015, 2016 and 2017, respectively. The following are the retention rates of engineering students in these six classes after the first year at ASU:

- 2009 engineering students: 70% of 890 students

- 2011 engineering students: 72% of 1522 students
- 2014 engineering students: 96% of 1931 students
- 2015 engineering students: 89% of 2535 students
- 2016 engineering students: 84% of 2833 students
- 2017 engineering students: 84% of 2717 students

Each data set has one data record for each student. In each data record, there is the retention variable indicating whether or not the student stayed in engineering after the first year, as well as other variables for student data covering demographics (e.g., gender, age, race, and home city), high school academic performance (e.g., SAT, ACT, GPA, rank percentile, and AP hours), academic performance at ASU (e.g., grades, credit hours, GPA, and major), financial aids, and survey data covering academic confidence, support, wellness, and university life concerning academics, social, and hours of various activities.

Data Analyses to Identify Common Characteristics of Engineering Retention

The first step of our data analysis was to identify common characteristics of engineering retention. For this purpose, we carried out both multivariate analysis and univariate analysis of data in each data set.

The PVAD (Partial-Value Association Discovery) algorithm (Ye, 2018, 2016, 2017a, 2013b) was chosen to perform the multivariate analysis of data in this study because each data set has both categorical data variables and numeric data variables and the PVAD algorithm was developed recently to handle both categorical data and numeric data together and uncover multivariate data associations from data. The PVAD algorithm was used to obtain data associations. Each association is in the form of X $\rightarrow$ retention = YES, where X represents the specific value(s) of one or multiple variables.

Hence, X in each data association reveals characteristics of students whose retention variable(s) indicates them staying in engineering after the first year at ASU. In this study, we looked into only 1-to-1 data associations with X containing one variable and its specific value, because p-to-1 data associations, $p > 1$, with X containing multiple variables and their specific value are often combinations of characteristics from 1-to-1 data associations. A supporting instance of a 1-to-1 data association is the data record of a student who stayed in engineering after the first year and has the specific value of the X variable in this data association. From 1-to-1 data associations which are supported by a majority of students who stayed in engineering, we reported variables and their values in those 1-to-1 data associations as the common characteristics of students who stayed in engineering.

If there were data records in the data set which have retention = YES but were not covered by 1-to-1 data associations of X $\rightarrow$ retention = YES from the PVAD algorithm, we performed the univariate data analysis of these data records by determining the frequency of each value for each variable in these data records and comparing the frequency of the same variable value in the entire data set of the student population. If there were a sufficient number of data records with the frequency of a variable value significantly higher than the frequency of the same variable value in the entire student population, we reported the variable value as a common characteristic of students who stayed in engineering.

Hence, common characteristics of students who stayed in engineering after the first year were obtained by using the PVAD algorithm of multivariate data analysis to obtain associations of student characteristics with retention = YES and the univariate frequency analysis of those data records that were not covered by data associations from the PVAD algorithm. There is no need to carry out the univariate frequency analysis if data associations from the PVAD algorithm cover all data

records of students with X → retention = YES.

The same multivariate data analysis using the PVAD algorithm and the univariate frequency analysis were also performed to obtain 1-to-1 data associations in the form of X → retention = NO and frequencies of variable values to extract common characteristics of students who left engineering after the first year at ASU.

Data Analyses to Identify Uncommon Characteristics of Engineering Retention

Common characteristics of engineering retention for retention = YES were then used to identify Type 1 uncommon/untypical students who did not have common characteristics of engineering retention but stayed in engineering after the first year and Type 2 uncommon/untypical students who had common characteristics of engineering retention but left engineering after the first year. Characteristics shared by a majority of Type 1 students and characteristics shared by a majority of Type 2 students were examined and identified as uncommon characteristics of engineering retention.

4.5.2 Results

Table 4.14 presents common characteristics of students who stayed in engineering after the 1st year at ASU. All the characteristics in Table 4.14 were obtained from 1-to-1 data associations of X → retention = YES with the largest numbers of supporting instances. The characteristics from data associations with small numbers of supporting instances are not listed in Table 4.14 because they are not considered as common characteristics.

Table 4.14 shows that not poor academic performance in the first year, in terms of GPA above $2.5 \pm \alpha$ ($0 \leq \alpha \leq 0.52$, varying for different data sets) and no D, E and W grades or no/few ASRs (Academic Status Reports), is the common characteristic

of students who stayed in engineering after the first year at ASU, consistently and dominantly among all data sets. ASU uses grades of A, B, C, D, and W (withdrawal) with A, B, and C giving the full course credits and producing 4.0, 3.0, and 2.0, respectively, in the GPA computation, and D, E, and W giving no course credits. ASU requires a minimum of 2.0 GPA to graduate. ASRs are used by course instructors when they want to give warnings to students who lag behind. The 2008 and 2011 data sets have data for the count of D, E, and W grades and the count of A and B grades, but do not have data for the count of ASRs. The 2014-2017 data sets do not have data for the count of D, E, and W grades or the count of A and B grades, but have data for the count of ASRs.

Among many variables measuring academic performance, GPA provides a comprehensive measure of academic performance, and is the most dominant indicator for retention as shown in Table 4.14, followed by the count of D, E, and W grades, the count of A and B grades, and the count of ASRs. The characteristic of GPA above $2.5 \pm \alpha$ ($0 \leq \alpha \leq 0.52$), e.g., 2.43 and 2.56 in the 2015 and 2016 data sets, 2.67 and 3.02 in the 2011 data set, and 2.29 and 2.67 in the 2017 data set, for retention = YES was found in five out of six data sets. In addition to the GPA characteristic, no D, E, or W grades (in the 2009 data set), the average count of A and B grades ≥ 4 (in the 2011 data set), or few or no ASRs (in the 2014-2017 data sets) were also found.

Table 4.14: Common characteristics of students who stayed in engineering after the first year at ASU

Data set	# of Students who stayed in engineering	Common student characteristics (# of students with characteristic)
2009	625	• Count of D, E, and W grades is 0 (450 students) • Fall GPA $\geq$ 3.5 (292 students), or • Have AP hours (253 students)

Continued on next page

Table 4.14: Common characteristics of students who stayed in engineering after the first year at ASU (Continued)

Data set	# of Students who stayed in engineering	Common student characteristics (# of students with characteristic)
2011	1101	• Fall GPA $\geq$ 3.02 (815 students), • Spring cumulative GPA $\geq$ 2.67 (951 students), or • Average count of A and B grades $\geq$ 4 (828 students)
2014	1859	• Spring cumulative GPA in (0, 4.24] (1841 students) • Number of ASR (Academic Status Report) in Fall Week 1 in [0, 3] (1854 students), or • Number of ASR in Spring Week 1 in [0, 2] (1858 students)
2015	2264	• GPA Fall GPA $\geq$ 2.56 (2043 students), • Spring GPA $\geq$ 2.43 (2033 students), • Fall ASR Count in [0, 3] (2194 students), or • Spring ASR Count in [0, 2] (2188 students)
2016	2387	• Fall GPA $\geq$ 2.56 (2147 students) • Spring cumulative GPA $\geq$ 2.55 (2057 students) • Spring GPA $\geq$ 2.43 (1985 students) • Spring ASR Count in [0, 2] (2233 students), or • Fall ASR Count is 0 (1616 students)
2017	2287	• Fall GPA $\geq$ 2.67 (1911 students) • Spring GPA $\geq$ 2.29 (1813 students) • Spring ASR Count is 0 (1859 students), or • Fall ASR Count is 0 (1750 students)

Table 4.15 presents the common characteristics of students who left engineering after their first year at ASU. The characteristics from the univariate frequency analysis were noted in Table 4.15. Other characteristics in Table 4.15 were obtained from 1-to-1 data associations of X $\rightarrow$ retention = NO with large numbers of supporting instances. The characteristics from data associations with small numbers of supporting instances are not listed in Table 4.15 because they are not considered as common characteristics.

Table 4.15 shows that poor academic performance in the first year, in terms of

GPA around or below $2.5 \pm \alpha$ ($0 \leq \alpha \leq 0.52$), e.g., 2.5 in the 2009 data set, 2.43 and 2.55 in the 2014 data set, 2.42 and 2.54 in the 2015 and 2016 data sets, and 2.85 in the 2017 data set, and non-zero counts of grades D, E and W or ASRs (e.g., 1, 4 and 5 counts of grades D, E and W in the 2009 data set, 1 to 2 and 4 to 7 counts of grades D, E and W in the 2011 data set, and 1 to 4 counts of ASRs in the 2017 data set), is the common characteristic of students who left engineering after the first year at ASU. Hence, the characteristics of students who stayed in engineering as shown in Table 4.14 are consistent with the characteristics of students who left engineering as shown in Table 4.15 as both tables show that the GPA of above versus around or below and zero versus $2.5 \pm \alpha$ ($0 \leq \alpha \leq 0.52$) non-zero count of D, E, and W grades or ASRs in the first year at ASU separate the majority of students who stayed from the majority of students who left engineering. This may be attributed to the fact that students earn no course credits from D, E, and W grades.

Among the common characteristics of students who stayed in engineering in Table 4.14, GPA below $2.5 \pm \alpha$ ($0 \leq \alpha \leq 0.52$), e.g., 2.29, 2.43, 2.5, 2.55, 2.56, 2.67 and 3.02 in different data sets, and non-zero count of D, E and W grades (see Table 4.16 and Table 4.17) were used to identify Type 1 students who had such poor GPA and grade counts but still stayed in engineering and Type 2 students who did not have such poor GPA or grade counts but left engineering. Table 4.16 shows the characteristics of Type 1 students. Table 4.17 shows the characteristics of Type 2 students.

Table 4.15: Common characteristics of students who left engineering after the first year at ASU

Data set	# of Students who left engineering	Common student characteristics (# of students with characteristic)
2009	265	• 35 students have the count of D, E, and W grades in [4, 5] • 185 students without the above characteristics have more of the following characteristics than the population (from the univariate frequency analysis): ◦ 95% have no AP hours, versus 69% of the population ◦ 33% have the count of D, E, and W grades is 1, versus 22% of the population ◦ 31% have Spring GPA < 2.5, versus 21% of the population
2011	421	• Average count of A or B grades is 0 (93 students) • Maximum count of A or B grades is [0, 1] (116 students) • Average count of D, E and W grades is [4, 7] (84 students) • Average count of C, D, E and W grades is [5, 7] (73 students) • Fall GPA is [0, 1.71] (99 students) • Spring cumulative GPA is [0, 1.04] (71 students) • 249 students without the above characteristics have more of the following characteristics than the population (from the univariate frequency analysis): ◦ 23% have the average count of A and B grades = 2, versus 11% of population ◦ 51% have the average count of D, E and W grades in [1, 2], versus 25% of the population ◦ 67% are white, versus 57% of population ◦ 39% are sophomores to start the current major, versus 24% of the population

Continued on next page

Table 4.15: Common characteristics of students who left engineering after the first year at ASU (Continued)

Data set	# of Students who left engineering	Common student characteristics (# of students with characteristic)
2014	72	• 72 students have more of the following characteristics than the population (from the univariate frequency analysis): ◦ 89% are not honor students, versus 74% of the population ◦ 65% are white, versus 49% of population ◦ 51% have the Spring probation, versus 9% of the population ◦ 21% have the Fall probation, versus 5% of the population ◦ 31% have Fall GPA < 2.55, versus 8% of the population ◦ 26% have Spring GPA < 2.43, versus 8% of the population
2015	271	• 216 students have more of the following characteristics than the population (from the univariate frequency analysis): ◦ 88% are not honor students, versus 74% of the population ◦ 58% do not live on campus, versus 41% of the population ◦ 29% have the Fall GPA in (0, 2.54], versus 12% of the population ◦ 37% have the Spring Semester GPA in (0, 2.42], versus 12% of the population

Continued on next page

Table 4.15: Common characteristics of students who left engineering after the first year at ASU (Continued)

Data set	# of Students who left engineering	Common student characteristics (# of students with characteristic)
2016	447	• Fall GPA is 0 (92 students) • Advisor Hold = Y (208 students) • 170 students without the above characteristics have more of the following characteristics than the population (from the univariate frequency analysis): ◦ 94% are not honor students, versus 77% of the population ◦ 42% have Fall GPA in (0, 2.54], versus 15% of the population ◦ 53% have Spring cumulative GPA in (0, 2.54], versus 16% of the population ◦ 45% have Spring GPA in (0, 2.42], versus 17% of the population
2017	430	• Spring cumulative GPA in [0, 1.21] (56 students) • 374 students without the above characteristics have more of the following characteristics than the population (from the univariate frequency analysis): ◦ 38% have Spring ASR Count in [1, 4], versus 21% of the population ◦ 31% have Spring cumulative GPA in [1.92, 2.85], versus 18% of the population ◦ 22% have Fall ASR Count in [2, 3], versus 11% of the population

Table 4.16 shows that students with the following characteristics had more tendency to stay in engineering even with GPA around or below $2.5 \pm \alpha$ ($0 \leq \alpha \leq 0.52$) and non-zero count of D, E and W grades at ASU:

- Students who are not white, in four out of six data sets
- Students who are not honor students, in five out of six data sets (this is related to the poor academic performance of those students)

- Male students, in three out of six data sets.

Hence, it appears that not-white students and male students had more tendency to stay in engineering even with poor academic performance.

Table 4.16: Characteristics of Type 1 students who had poor academic performance but still stayed in engineering

Data set	# of students who had measures of not poor academic performance	Characteristics of students
2009	Among 625 students who stayed in engineering, 56 students had: • Count of D, E, and W grades > 0, AND • Fall GPA < 2.5	• 95% male, versus 79% of population • 52% are not white, versus 39% of population • 32% are URM (Under Represented Minority), versus 22% of the population • 82% have no AP hours, versus 69% of the population • 96% are not honors students, versus 79% of the population
2011	Among 1101 students who stayed in engineering, 128 students had: • Fall GPA < 3.02, • Spring GPA < 2.67, AND • Average count of AB grades < 4	• 90% (115 students) are male, versus 82% of the population • 55% (70 students) are not white, versus 18% of the population • 33% (42 students) have transferred hours at admission > 0, versus 14% of the population
2014	Among 1859 students who stayed in engineering, 46 students had: • Fall GPA < 2.55, AND • Spring GPA < 2.43	• 91% are not honor students, versus 74 % of the population
2015	Among 2264 students who stayed in engineering, 64 students had: • Fall GPA < 2.56, AND • Spring GPA < 2.43	• 95% are not honor students, versus 75% of the population • 61% are not white, versus 52% of the population

Continued on next page

Table 4.16: Characteristics of Type 1 students who had poor academic performance but still stayed in engineering (Continued)

Data set	# of students who had measures of not poor academic performance	Characteristics of students
2016	Among 2387 students who stayed in engineering, 165 students had: • Fall GPA < 2.56, AND • Spring cumulative GPA < 2.55	• 94% are not honor students, versus 77% of the population • 62% are not white, versus 50% of the population
2017	Among 2287 students who stayed in engineering, 303 students had: • Fall GPA < 2.67, AND • Spring GPA < 2.29	• 94% are not honors students, versus 77% of the population • 46% are not resident, versus 35% of the population • 85% are male, versus 79% of the population

Table 4.17 shows that students with the following characteristics had more tendency to leave engineering even with a GPA above $2.5 \pm \alpha$ ($0 \leq \alpha \leq 0.52$) or zero counts of grades D, E and W in the first year at ASU:

- White students, in four out of six data sets
- Students who are not Arizona residents, in two out of six data sets
- Students who are not honor students, in two out of six data sets.

Type 1 students in Table 4.16 and Type 2 students in Table 4.17 are uncommon/untypical students – students who did not leave or stay in engineering based on the common characteristics of engineering retention with regards to GPA and counts of D, E, and W grades or ASRs in the first year at ASU. It is interesting that race/ethnicity in terms of white students versus not-white students is the major characteristic of those uncommon/untypical students, as Table 4.17 shows that white students had more tendency to leave engineering even with a GPA above approximately 2.5 (not-poor academic performance) at ASU, and Table 4.16 shows that not-white

students had more tendency to stay in engineering even with GPA around or below $2.5 \pm \alpha$ ($0 \leq \alpha \leq 0.52$) (poor academic performance) at ASU. Gender also plays a role among those uncommon/untypical students as male students had more tendency to stay in engineering even with GPA around or below $2.5 \pm \alpha$ ($0 \leq \alpha \leq 0.52$). Uncommon/untypical students who are not honor students tended to be less stable or to vary more in the retention outcome.

Student data in this study covers demographics, high school academic performance (e.g., GPA, rank percentile, AP hours, SAT, ACT, etc.), academic performance in university (e.g., GPA, grade counts, credit hours, ASRs, probations, etc.), scholarships and financial aids/needs, and survey data covering academic confidence, family and social support, wellness, university life related to academics, social, and hours of work and various activities. Based on the results of this study, the GPA measure of academic performance in university and certain demographics such as race/ethnicity and gender are most relevant to student retention after the first year at ASU. Most measures of high school academic performance including GPA, rank percentile, SAT, and ACT are not useful in determining student retention in engineering at ASU. At ASU, the majority of domestic students received scholarships and financial aid from ASU or other sources, which may be one reason why financial aid and needs are not related to student retention in engineering at ASU. Survey data, covering academic confidence, family and social support, wellness, university life concerning academics, social, and hours of work and various activities, is not relevant to student retention in engineering at ASU, likely because the survey data was collected at the beginning of the first semester when students had little experience yet to give a good assessment of their study and life. Moreover, it may be better to let students think and assess only one or a few survey questions at a time rather than having students complete many questions at a time, in order to get more accurate, robust survey data.

Table 4.17: Characteristics of Type 2 students who had not-poor academic performance but left engineering

Data set	Measures of not poor academic performance	Student characteristics
2009	Among 265 students who left engineering, 159 students have: • Count of D, E, and W grades = 0, OR • Count of D, E, and W grades > 0 & Fall GPA ≥ 2.5	• 91% have no AP hours, versus 69% of the population
2011	Among 421 students who left engineering, 220 students have: • Fall GPA ≥ 3.02, • Spring cumulative GPA ≥ 2.67, OR • Average count of A and B grades ≥ 4	• 29% are not white, versus 18% of the population • 65% (144 students) are not freshman when starting the current major, whereas 48% of the population
2014	Among 72 students who left engineering, 49 students have: • Fall GPA ≥ 2.55, OR • Spring GPA ≥ 2.42	• 86% are not honor students, versus 74% of the population • 59% are white, versus 49% of the population
2015	Among 271 students who left engineering, 185 students have: • Fall GPA ≥ 2.56, OR • Spring GPA ≥ 2.43	• 57% are not resident, versus 41% of population • 57% are white, versus 48% of the population
2016	Among 447 students who left engineering, 174 students have: • Fall GPA ≥ 2.56, OR • Spring cumulative GPA ≥ 2.55	• 91% are not honor students, versus 77% of the population • 60% are not resident, versus 37% of the population • 64% are white, versus 50% of the population
2017	Among 430 students who left engineering, 313 students have: • Fall GPA in ≥ 2.67, OR • Spring GPA ≥ 2.29	No difference from population

4.5.3 Conclusions, Implications, and Limitations

This study reveals the following common and uncommon characteristics of engineering student retention:

- Students stayed in engineering as long as their academic performance was not poor in terms of having a GPA above 2.5 $\pm$ α ($0 \leq \alpha \leq 0.52$) and zero counts of D, E, and W grades.
- There are two types of students who stayed in engineering after the first year at ASU: (1) students with GPA above 2.5 $\pm$ α ($0 \leq \alpha \leq 0.52$), and (2) not-white students as well as male students even with GPA around or below 2.5 $\pm$ α ($0 \leq \alpha \leq 0.52$).
- There are two types of students who left engineering after the first year at ASU: (1) students with GPA around or below 2.5 $\pm$ α ($0 \leq \alpha \leq 0.52$), and (2) students having GPA above 2.5 $\pm$ α ($0 \leq \alpha \leq 0.52$), whose reason to leave engineering varied individually, with especially white students having more tendency to leave engineering.

Hence, this study adds to the field of engineering education the above findings about the dominant role that a university GPA of 2.5 $\pm$ α ($0 \leq \alpha \leq 0.52$) played in engineering student retention, and relations of gender, race/ethnicity and non-honor students to the retention of uncommon/untypical students. Moreover, this study adds new findings about some factors of STEM retention investigated in the past. This study points out that measures of high school academic performance (e.g., GPA, rank percentile, SAT, and ACT) (Tyson, 2011; Weatherton *et al.*, 2011; Moses *et al.*, 2011; Kokkelenberg and Sinha, 2010; Li *et al.*, 2009b; Klopfenstein and Thomas, 2009; Kauffmann *et al.*, 2007; Herzog, 2005) are not useful in determining student retention in engineering at ASU. This study reveals that scholarships and financial aids (Gross *et al.*, 2015, 2013; Gross, 2011; Herzog, 2005) are not related to engineering student retention at ASU possibly because switching from engineering to non-engineering did not affect students' scholarships and financial aids.

The university has a minimum GPA requirement of 2.0 for graduation. Individual

degree programs may have additional GPA requirements for certain courses. Various scholarships may have different GPA requirements. It is not clear whether or not the minimum GPA requirement of 2.0 for graduation and additional GPA requirements for scholarships and certain courses are related to the GPA of $2.5 \pm \alpha$ as the dominant characteristic of engineering retention.

These findings have implications and can be used in the future to guide student recruiting and the development of new intervention mechanisms for engineering retention. For student recruiting, our efforts of recruiting more students from a diverse range of ethnicity backgrounds, especially non-white students, should be continued and expanded with the expectation of success in the retention of students from non-traditional ethnicity backgrounds. For the development of new intervention mechanisms, we should investigate intervention mechanisms to improve the grades and academic performance of students, especially with a focus on identifying poorly performing students early on and designing and using intervention mechanisms to move such students out of D, E, and W grades. We are investigating active learning methods to have students learn more in classes, as well as new grading methods with dynamical due dates and bonuses for early submission to change the procrastination habit that some students have and motivate and allow students with different abilities to study on different paces.

This study has limitations since we focused on common and uncommon characteristics of engineering student retention, did not analyze retention rates and characteristics of transfer students, did not look into elements which drove Type 2 students to leave engineering, as well as many other issues related to engineering student retention. Another limitation is that the findings of this study are based on data from only one university.

Chapter 5

CONCLUSIONS AND FUTURE WORK

This dissertation explores the common challenges faced by existing machine learning and data mining techniques in handling real-world data sets. It highlights three main difficulties: mixed-type data, variable relations across value ranges, and unknown variable dependencies.

To address these challenges, this dissertation introduces a novel algorithm called Partial-Value Association Discovery (PVAD). PVAD not only overcomes the drawbacks of existing techniques but also enables the discovery of partial-value and full-value variable associations. The advantage of PVAD is shown by comparing two other commonly used techniques: Association rule mining and Decision Tree. The comparison results demonstrate that the PVAD algorithm outperforms both Association rule mining and Decision Tree in terms of its ability to uncover meaningful associations in real-world data sets.

This research also investigates knee point detection on noisy data. A new mathematical definition of knee point on discrete data is introduced, and a deep-learning model, *UNetConv*, is developed that outperforms existing methods. The model outperforms existing methods and exhibits exceptional performance on synthetic data. However, there are limitations, including the use of only Gaussian noise in the training samples, limited curve shapes explored by the model, subjective determination of knee points in real-world data, and uncertainty in handling scenarios with more than five knee points. Future work should include incorporating a wider range of samples (both training and test data) with varying levels of noise to assess the model's robustness and sensitivity to noise.

In the final section of this study, the PVAD algorithm is applied to real-world data sets in various domains. We present the application and analyses of the PVAD algorithm to actual data sets. The primary objective of this section is to further illustrate the effectiveness of PVAD in capturing the variable relations. In all the applications, PVAD demonstrates its competence in capturing both variable's individual and interactive effects.

Specifically, in the application of network data, univariate and multivariate measures are established in detecting network attacks. Future work of this study includes comparison with other distribution difference measures in existing literature. This comparative analysis will enable us to gain a deeper understanding of the effectiveness of the measures

Overall, this dissertation addresses the limitations of existing machine learning and data mining techniques and introduces a novel algorithm that overcomes these limitations. It provides insights into handling real-world data sets and offers practical solutions for uncovering variable relationships in various domains. Future research in this study will focus on validating the PVAD findings. This may include conducting controlled experiments to explore and confirm the factors and the corresponding factor levels identified by PVAD, thereby verifying their significance.

REFERENCES

Ackerman, P. L., R. Kanfer and M. E. Beier, "Trait complex, cognitive ability, and domain knowledge predictors of baccalaureate success, stem persistence, and gender differences.", Journal of Educational Psychology **105**, 3, 911 (2013a).

Ackerman, P. L., R. Kanfer and C. Calderwood, "High school advanced placement and student performance in college: Stem majors, non-stem majors, and gender differences", Teachers College Record **115**, 10, 1–43 (2013b).

Agrawal, R., T. Imieliński and A. Swami, "Mining association rules between sets of items in large databases", in "Proceedings of the 1993 ACM SIGMOD international conference on Management of data", pp. 207–216 (1993).

Agrawal, R., H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo *et al.*, "Fast discovery of association rules.", Advances in knowledge discovery and data mining **12**, 1, 307–328 (1996).

AKUTSU, T., S. MIYANO and S. KUHARA, "Algorithms for inferring qualitative models of biological networks", in "Biocomputing 2000", pp. 293–304 (World Scientific, 1999).

Andersson, J., H. Ahlström and J. Kullberg, "Separation of water and fat signal in whole-body gradient echo scans using convolutional neural networks", Magnetic resonance in medicine **82**, 3, 1177–1186 (2019).

Antunes, M., H. Aguiar and D. Gomes, "Al and s methods: Two extensions for l-method", in "2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)", pp. 371–376 (IEEE, 2019).

Antunes, M., J. Ribeiro, D. Gomes and R. L. Aguiar, "Knee/elbow point estimation through thresholding", in "2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)", pp. 413–419 (IEEE, 2018).

Bazil, J. N., F. Qi and D. A. Beard, "A parallel algorithm for reverse engineering of biological networks", Integrative Biology **3**, 12, 1215–1223 (2011).

Breiman, L., "Bagging predictors", Machine learning **24**, 123–140 (1996).

Breiman, L., "Arcing classifier (with discussion and a rejoinder by the author)", The annals of statistics **26**, 3, 801–849 (1998).

Breiman, L., "Random forests", Machine learning **45**, 5–32 (2001).

Camacho, A. M. and D. R. Lapuz, "The stem center: Creating a model for success in community college stem education", in "2014 ASEE Annual Conference & Exposition", pp. 24–1246 (2014).

Cao, X., X. Fu, D. Hong, Z. Xu and D. Meng, "Pancsc-net: A model-driven deep unfolding method for pansharpening", IEEE Transactions on Geoscience and Remote Sensing **60**, 1–13 (2021).

Chan-Hilton, A., "Student success and retention from the perspectives of engineering students and faculty", (2019).

Chandola, V., A. Banerjee and V. Kumar, "Anomaly detection: A survey", ACM computing surveys (CSUR) **41**, 3, 1–58 (2009).

Coletti, K. B., E. O. Wisniewski, R. L. Shapiro, P. A. DiMilla, R. Reisberg and M. Covert, "Correlating freshman engineers' performance in a general chemistry course to their use of supplemental instruction", in "2014 ASEE Annual Conference & Exposition", pp. 24–323 (2014).

Desai, N. and G. Stefanek, "A literature review of the different approaches that have been implemented to increase retention in engineering programs across the united states", in "Proceedings of the 2017 ASEE Zone II Conference", pp. 2–5 (2017).

Ding, P. L. K., Z. Li, Y. Zhou and B. Li, "Deep residual dense u-net for resolution enhancement in accelerated mri acquisition", in "Medical Imaging 2019: Image Processing", vol. 10949, pp. 110–117 (SPIE, 2019).

Draper, N. R. and H. Smith, ""dummy" variables", Applied regression analysis pp. 299–325 (1998).

Ellis, B. and W. H. Wong, "Learning causal bayesian network structures from experimental data", Journal of the American Statistical Association **103**, 482, 778–789 (2008).

Frank, A. and A. Asuncion, "Uci machine learning repository. irvine: School of information and computer science, university of california, 2010", (2008).

Freund, Y. and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", Journal of computer and system sciences **55**, 1, 119–139 (1997).

Friedman, J. H., "Multivariate adaptive regression splines", The annals of statistics **19**, 1, 1–67 (1991).

Fuesting, M. A., *Engineering persistence: Designing and testing a communal strategies intervention to increase the retention of women in engineering*, Ph.D. thesis, Miami University (2019).

García-Ros, R., F. Pérez-González, F. Cavas-Martínez and J. M. Tomás, "Effects of pre-college variables and first-year engineering students' experiences on academic achievement and retention: a structural model", International Journal of Technology and Design Education **29**, 915–928 (2019).

Gasse, M., A. Aussem and H. Elghazel, "An experimental comparison of hybrid algorithms for bayesian network structure learning", in "Joint European Conference on Machine Learning and Knowledge Discovery in Databases", pp. 58–73 (Springer, 2012).

Geisinger, B., D. R. Raman and D. Raman, "Why they leave: Understanding student attrition from engineering majors", (2013).

Gross, J. P., "Promoting or perturbing success: The effects of aid on timing to latino students' first departure from college", Journal of Hispanic Higher Education **10**, 4, 317–330 (2011).

Gross, J. P., D. Hossler, M. Ziskin and M. S. Berry, "Institutional merit-based aid and student departure: A longitudinal analysis", The Review of Higher Education **38**, 2, 221–250 (2015).

Gross, J. P., V. Torres and D. Zerquera, "Financial aid and attainment among students in a state with changing demographics", Research in Higher Education **54**, 383–406 (2013).

Hall, C. W., K. A. De Urquidi, P. J. Kauffmann, K. L. Wuensch, W. W. Swart and O. H. Griffin, "Longitudinal study of entering students with engineering as their major: Retention and academic success", in "2013 ASEE Annual Conference & Exposition", pp. 23–875 (2013).

Han, J., J. Pei and M. Kamber, "Data mining: concepts and techniques. 2011", (1999).

Hastie, T., R. Tibshirani, J. H. Friedman and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2 (Springer, 2009).

Herzog, S., "Measuring determinants of student return vs. dropout/stopout vs. transfer: A first-to-second year analysis of new freshmen", Research in higher education **46**, 883–928 (2005).

Hieb, J. L., K. B. Lyle, P. A. Ralston and J. Chariker, "Predicting performance in a first engineering calculus course: Implications for interventions", International Journal of Mathematical Education in Science and Technology **46**, 1, 40–55 (2015).

Ho, T. K., "Random decision forests", in "Proceedings of 3rd international conference on document analysis and recognition", vol. 1, pp. 278–282 (IEEE, 1995).

Ho, T. K., "The random subspace method for constructing decision forests", IEEE transactions on pattern analysis and machine intelligence **20**, 8, 832–844 (1998).

Jones, B. D., J. W. Osborne, M. C. Paretti and H. M. Matusovich, "Relationships among students' perceptions of a first-year engineering design course and their engineering identification, motivational beliefs, course effort, and academic outcomes", International Journal of Engineering Education **30**, 6, 1340–1356 (2014a).

Jones, B. D., J. W. Osborne, M. C. Paretti and H. M. Matusovich, "Relationships among students' perceptions of a first-year engineering design course and their engineering identification, motivational beliefs, course effort, and academic outcomes", International Journal of Engineering Education **30**, 6, 1340–1356 (2014b).

Kauffmann, P., T. Abdel-Salam and J. D. Garner, "Predictors of success in the first two years: a tool for retention", in "2007 Annual Conference & Exposition", pp. 12–1171 (2007).

Kauffmann, P., C. Hall, G. Dixon and J. Garner, "Predicting academic success for first semester engineering students using personality trait indicators", in "2008 Annual Conference & Exposition", pp. 13–990 (2008).

Klopfenstein, K. and M. K. Thomas, "The link between advanced placement experience and early college success", Southern Economic Journal **75**, 3, 873–891 (2009).

Kokkelenberg, E. C. and E. Sinha, "Who succeeds in stem studies? an analysis of binghamton university undergraduate students", Economics of Education Review **29**, 6, 935–946 (2010).

Lee, D., J. Yoo, S. Tak and J. C. Ye, "Deep residual learning for accelerated mri using magnitude and phase networks", IEEE Transactions on Biomedical Engineering **65**, 9, 1985–1995 (2018).

Li, Q., H. Swaminathan and J. Tang, "Development of a classification system for engineering student characteristics affecting college enrollment and retention", Journal of Engineering Education **98**, 4, 361–376 (2009a).

Li, Q., H. Swaminathan and J. Tang, "Development of a classification system for engineering student characteristics affecting college enrollment and retention", Journal of Engineering Education **98**, 4, 361–376 (2009b).

Maccariella Jr, J., S. Pribesh and M. R. Williams, "An engineering learning community to promote retention and graduation for community college students", Journal of Professional Issues in Engineering Education and Practice **145**, 4, 04019013 (2019).

Martin III, S. E., *Engineering retention: Improving inclusion and diversity in engineering*, Ph.D. thesis, Murray State University (2018).

Mason, L., J. Baxter, P. Bartlett and M. Frean, "Boosting algorithms as gradient descent", Advances in neural information processing systems **12** (1999).

Moses, L., C. Hall, K. Wuensch, K. De Urquidi, P. Kauffmann, W. Swart, S. Duncan and G. Dixon, "Are math readiness and personality predictive of first-year retention in engineering?", The Journal of psychology **145**, 3, 229–245 (2011).

Neubauer, J. and A. Pesaran, "The ability of battery second use strategies to impact plug-in electric vehicle prices and serve utility energy storage applications", Journal of Power Sources **196**, 23, 10351–10358 (2011).

Orr, H., "Student retention in community college engineering and engineering technology programs", (2019).

Quinlan, J. R., "Induction of decision trees", Machine learning **1**, 81–106 (1986).

Quinlan, J. R., *C4. 5: programs for machine learning* (Elsevier, 2014).

Redmon, J., S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 779–788 (2016).

Ridler, T., S. Calvard *et al.*, "Picture thresholding using an iterative selection method", IEEE Trans. Syst. Man Cybern **8**, 8, 630–632 (1978).

Ronneberger, O., P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in "Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18", pp. 234–241 (Springer, 2015).

Salvador, S. and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms", in "16th IEEE international conference on tools with artificial intelligence", pp. 576–584 (IEEE, 2004).

Satopaa, V., J. Albrecht, D. Irwin and B. Raghavan, in "Finding a" kneedle" in a haystack: Detecting knee points in system behavior", pp. 166–171 (IEEE, 2011).

Schuster, S. F., T. Bach, E. Fleder, J. Müller, M. Brand, G. Sextl and A. Jossen, "Nonlinear aging characteristics of lithium-ion cells under different operational conditions", Journal of Energy Storage **1**, 44–53 (2015).

Sharafaldin, I., A. Gharib, A. H. Lashkari and A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset", Software Networking **2017**, 1, 177–200 (2017).

Sharafaldin, I., A. H. Lashkari and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.", ICISSp **1**, 108–116 (2018).

Shaw, E. J., J. P. Marini and K. D. Mattern, "Exploring the utility of advanced placement participation and performance in college admission decisions", Educational and Psychological Measurement **73**, 2, 229–253 (2013).

Sithole, A., E. T. Chiyaka, P. McCarthy, D. M. Mupinga, B. K. Bucklein and J. Kibirige, "Student attraction, persistence and retention in stem programs: Successes and continuing challenges.", Higher Education Studies **7**, 1, 46–59 (2017).

Sulaiman, M., "Effects of academic and nonacademic factors on undergraduate electronic engineering program retention", (2016).

Teague, S., G. J. Youssef, J. A. Macdonald, E. Sciberras, A. Shatte, M. Fuller-Tyszkiewicz, C. Greenwood, J. McIntosh, C. A. Olsson, D. Hutchinson *et al.*, "Retention strategies in longitudinal cohort studies: a systematic review and meta-analysis", BMC medical research methodology **18**, 1–22 (2018).

Tolsa, X., "Principal values for the cauchy integral and rectifiability", Proceedings of the American Mathematical Society **128**, 7, 2111–2119 (2000).

Tsamardinos, I., L. E. Brown and C. F. Aliferis, "The max-min hill-climbing bayesian network structure learning algorithm", Machine learning **65**, 31–78 (2006).

Tseng, F. and P.-Y. Chen, "Parallel association rule mining by data de-clustering to support grid computing", (2005).

Tso, G. K. and K. K. Yau, "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks", Energy **32**, 9, 1761–1768 (2007).

Tyson, W., "Modeling engineering degree attainment using high school and college physics and calculus coursetaking and achievement", Journal of Engineering Education **100**, 4, 760–777 (2011).

Uddin, M. and K. Johnson, "Faculty learning from the advisors for students' retention and persistence to graduation", in "2019 CIEC", (2019).

Veletzos, M., M. G. Noonan, M. W. Sakakeeny and C. McGowan, "Board 127: Implementing national best practices to improve stem retention in a liberal arts college setting", in "2018 ASEE Annual Conference & Exposition", (2018).

Weatherton, Y. P., A. P. Kruzic, B. R. Isbell, L. L. Peterson, C. Tiernan and V. V. Pham, "Mathematics performance and first year retention of students in engineering learning communities", in "2011 ASEE Annual Conference & Exposition", pp. 22–1047 (2011).

Weatherton, Y. P., A. P. Kruzic, S. P. Mattingly, Z. Rahman and H. L. Frost, "Is there a correlation between first-year critical thinking assessment test performance and retention among civil engineering students?", in "2014 ASEE Annual Conference & Exposition", pp. 24–829 (2014).

White, J. L. and G. Massiha, "The retention of women in science, technology, engineering, and mathematics: A framework for persistence.", International Journal of Evaluation and Research in Education **5**, 1, 1–8 (2016).

Williard, N. D., *Degradation analysis and health monitoring of lithium ion batteries* (University of Maryland, College Park, 2011).

Yang, F., D. Wang, Y. Xing and K.-L. Tsui, "Prognostics of li (nimnco) o2-based lithium-ion batteries using a novel battery degradation model", Microelectronics Reliability **70**, 70–78 (2017).

Yao, W., Z. Zeng, C. Lian and H. Tang, "Pixel-wise regression using u-net and its application on pansharpening", Neurocomputing **312**, 364–371 (2018).

Ye, N., *The handbook of data mining* (CRC Press, 2003).

Ye, N., *Secure computer and network systems: Modeling, analysis and design* (John Wiley & Sons, 2008).

Ye, N., *Data mining: theories, algorithms, and examples* (CRC press, 2013a).

Ye, N., "Reverse engineering: Mining structural system models from system data", Information Knowledge Systems Management **12**, 3-4, 205–208 (2013b).

Ye, N., "The partial-value association discovery algorithm to learn multilayer structural system models from system data", IEEE Transactions on Systems, Man, and Cybernetics: Systems **47**, 12, 3377–3385 (2016).

Ye, N., "A reverse engineering algorithm for mining a causal system model from system data", International Journal of Production Research **55**, 3, 828–844 (2017a).

Ye, N., "The partial value association discovery algorithm to identify engineering retention and success characteristics", in "EDULEARN17 Proceedings", pp. 5033–5041 (IATED, 2017b).

Ye, N., "Analytical techniques for anomaly detection through features, signal-noise separation and partial-value association", in "KDD 2017 Workshop on Anomaly Detection in Finance", pp. 20–32 (PMLR, 2018).

Ye, N. and T. Y. Fok, "Learning partial-value variable associations", in "Proceedings of the 4th International Conference on Big Data and Computing", pp. 24–28 (2019).

Ye, N., T. Y. Fok and O. Chong, "Modeling an energy consumption system with partial-value data associations", Advances in Science, Technology and Engineering Systems **3**, 6, 372–379 (2018a).

Ye, N., T. Y. Fok, J. Collofello and T. Coronella, "Common and uncommon characteristics of engineering student retention after the first year in university", in "2021 ASEE Virtual Annual Conference Content Access", (2021a).

Ye, N., T. Y. Fok, J. Collofello, D. Montgomery and K. Mills, "The partial-value association discovery algorithm and applications", in "2019 Amity International Conference on Artificial Intelligence (AICAI)", pp. 1–8 (IEEE, 2019).

Ye, N., T. Y. Fok, X. Wang, J. Collofello and N. Dickson, "Learning partial-value variable relations for system modeling", in "2018 4th International Conference on Control, Automation and Robotics (ICCAR)", pp. 368–372 (IEEE, 2018b).

Ye, N., T. Y. Fok, X. Wang, J. Collofello and N. Dickson, "The pvad algorithm to learn partial-value variable associations with application to modelling for engineering retention", IFAC-PapersOnLine **51**, 2, 505–510 (2018c).

Ye, N., T. Yan Fok and D. Montgomery, "Univariate distribution differences and conditional variables in multivariate data associations as network flow measures to detect network attacks", in "Proceedings of the 6th International Conference on Big Data and Computing", pp. 41–48 (2021b).

Zhang, H. and B. H. Singer, *Recursive partitioning and applications* (Springer Science & Business Media, 2010).

Zhao, Q., V. Hautamaki and P. Fränti, "Knee point detection in bic for detecting the number of clusters", in "Advanced Concepts for Intelligent Vision Systems: 10th International Conference, ACIVS 2008, Juan-les-Pins, France, October 20-24, 2008. Proceedings 10", pp. 664–673 (Springer, 2008).

Zhou, Y., R. Paul, P. Ding, R. Battle, A. Patel and B. Li, "Accelerated mri reconstruction using variational feedback u-net with transfer learning", in "MEDICAL PHYSICS", vol. 49, pp. E144–E145 (WILEY 111 RIVER ST, HOBOKEN 07030-5774, NJ USA, 2022).