DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DATA ENGINEERING AND MANAGEMENT IN TEXTILE SECTOR

by

Pelin YILDIRIM

January, 2019

İZMİR

DATA ENGINEERING AND MANAGEMENT IN TEXTILE SECTOR

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of
Philosophy in Computer Engineering

by

Pelin YILDIRIM

January, 2019
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **“DATA ENGINEERING AND MANAGEMENT IN TEXTILE SECTOR”** completed by **PELİN YILDIRIM** under supervision of **ASSOC. PROF. DR. DERYA BİRANT** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Derya BİRANT

Supervisor

Prof. Dr. Alp KUT

Thesis Committee Member

Assoc. Prof. Dr. Tuba ALPYILDIZ

Thesis Committee Member

Doç. Dr. Ayşegül Alaybeyoğlu

Examining Committee Member

Doç. Dr. Tuğba ÖZACAR ÖZTÜRK

Examining Committee Member

Prof. Dr. Kadriye ERTEKİN
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Assoc. Prof. Dr. Derya BİRANT, for her immense knowledge, continuous support, supervision and useful suggestions throughout this study. Her guidance helped me in all the time of my research and writing of this thesis. It has been an honor to be her Ph.D. student. I could not have imagined having a better advisor and mentor for my Ph.D. study. I am also thankful to Prof. Dr. Alp KUT for his valuable advice and encouragement throughout this study. I would also like to thank my committee member, Assoc. Prof. Dr. Tuba ALPYILDIZ for her intimate knowledge about the field of this study and precious contributions in this thesis.

Also, I would like to offer my special thanks to my family and my fiance for supporting me spiritually throughout writing this thesis and my life in general. It would not have been able to complete this thesis without their support and help.

Pelin YILDIRIM

DATA ENGINEERING AND MANAGEMENT IN TEXTILE SECTOR

ABSTRACT

Recently, enormous amounts of data are generated every day in textile industry. These multivariable and nonlinear data include raw material characteristics, machine settings, process parameters and quality attributes of the textile product. Deriving useful patterns and valuable knowledge from these raw data provides making right decisions to increase quality and productivity for textiles. To serve the purpose, this thesis focuses on the application of the data mining and machine learning techniques in the textile sector.

Data engineering is a discipline that concerns with data mining techniques for data processing and analysis. Data mining techniques can be grouped in three main categories: classification, clustering, and association rule mining. In this thesis, several case studies were conducted for each category. In the classification-based case studies, ensemble learning methods were proposed to improve prediction performance in textile sector (i.e. to determine stab resistance performances of knitted structures) as well as deep learning methods for textile object identification. As a clustering-based study, a novel hierarchical clustering approach, named *k*-Linkage, was proposed that calculates resemblance between pair of clusters considering *k* samples from two clusters. In the association rule mining study, an extended FP-Growth algorithm was used to discover the relationships between yarn and fabric properties.

In the thesis, several experimental studies were performed for each study to demonstrate the performances of the proposed methods. In each experiment, the proposed approaches were applied on real-world textile data and compared with the existing approaches in terms of different evaluation measures. In general, the results obtained from each experiment indicate that the proposed approaches in this thesis achieve more accurate results than the conventional solutions.

Keywords: Data engineering, textile sector, data mining, machine learning, classification, clustering, association rule mining

TEKSTİL SEKTÖRÜNDE VERİ MÜHENDİSLİĞİ VE YÖNETİMİ

ÖZ

Son dönemlerde, tekstil endüstrisinde her gün devasa miktarda veri üretilmektedir. Bu çok değişkenli ve doğrusal olmayan veriler, tekstil ürününün hammadde özelliklerini, makine ayarlarını, işlem parametrelerini ve kalite niteliklerini içerir. Bu ham verilerden faydalı örüntülerin ve değerli bilginin elde edilmesi, tekstil ürünlerinin kalite ve verimliliğini arttırmak için doğru kararlar almayı sağlar. Bu amaca istinaden, bu tez, tekstil sektöründe veri madenciliği ve makine öğrenmesi tekniklerinin uygulanmasına odaklanmaktadır.

Veri mühendisliği, veri işleme ve analizi için veri madenciliği teknikleri ile ilgilenen bir disiplindir. Veri madenciliği teknikleri üç ana kategoride gruplandırılabilir: sınıflandırma, kümeleme ve birliktelik kuralı analizi. Bu tezde, her kategori için çeşitli durum çalışmaları gerçekleştirilmiştir. Sınıflandırma temelli durum çalışmalarında, tekstil nesnelerinin ayırt edilmesi için derin öğrenme yöntemlerinin uygulanmasının yanı sıra, tekstil sektöründe tahminleme performansını iyileştirmek (örn. örgü yapıların delinme performansını belirlemek) için de topluluk öğrenmesi yöntemleri önerilmiştir. Kümelemeye dayalı bir çalışma olarak ise, *k*-Linkage isimli, küme çifti arasındaki benzerliği hesaplamak için iki kümeden de *k* tane örneklemi göz önünde bulunduran yeni bir hiyerarşik kümelenme yaklaşımı önerilmiştir. Birliktelik kuralı analizi çalışmasında, iplik ve kumaş özellikleri arasındaki ilişkilerin ortaya çıkarılması için genişletilmiş bir FP-Growth algoritması kullanılmıştır.

Tezde, önerilen yöntemlerin performansını göstermek adına her bir çalışma için çeşitli deneysel çalışmalar gerçekleştirilmiştir. Her bir deneyde, önerilen yaklaşımlar gerçek tekstil verileri üzerinde uygulanmış ve mevcut yöntemler ile farklı değerlendirme ölçütleri açısından karşılaştırılmıştır. Genel olarak, her bir deneyden elde edilen sonuçlar, bu tezde önerilen yaklaşımlarla geleneksel çözümlere göre daha doğru değerlere ulaşıldığını göstermektedir.

Anahtar kelimeler: Veri mühendisliği, tekstil sektörü, veri madenciliği, makine öğrenmesi, sınıflandırma, kümeleme, birliktelik kuralı analizi

*

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

1.1 General

The necessity of processing raw data and exploring valuable and potentially useful information obtained from them has arisen in many areas of engineering, science, business, medicine, and public service. Today's information technology applications analyze data and convert them to valuable knowledge in an efficient way. At this point, the Data Engineering concept plays an important role. *Data engineering* is a sub-branch of data science that transforms raw data to suitable for analyzing process and applies scientific methods on transformed data to discover potentially useful knowledge. *Data mining* (DM) is a specialized form of data engineering, which is used for discovering previously unknown, although potentially useful, patterns from raw data. DM is successful in analyzing situations when vast amount of data is available, when the data is complex with many variables and nonlinear relations, when there is the need to predict behaviors or outcomes, and when it is needed to find associations and relationships that are not currently understood.

In the textile industry, even when a simple product such as a basic t-shirt is considered, a large amount of data is generated and stored. These data include raw materials, machine settings, and quality parameters of the product. The data to be processed are multivariable and nonlinear if a relationship is sought among fiber properties, process parameters, and yarn properties or among yarn properties, machine settings, and fabric performance. In addition, significant innovations and improvements have occurred in the textile industry with the introduction of technical textiles with crucial performance expectations at extreme conditions, such as protective clothing against bullet, knives, microorganisms, impact, cold, sun, etc. Thus, there is the continuous demand for processing data and discovery of valuable and potentially usable knowledge from these data in the textile industry.

Many classical mathematical and statistical models have been used in numerous textile studies to process textile data. However, these traditional methods remain incapable of discovering overall and complex relations among features of data instances and predicting unknown feature values for a new instance. Because of this challenge, DM techniques that are implemented in wide range of engineering areas have also been used in textile engineering during the recent years.

1.2 Purpose

Data mining in textile industry (DMTI) is an interdisciplinary area, including but not limited to decision support systems, recommender systems, visual data analytics, information retrieval, database management system, domain-driven DM, and so on. DMTI can be drawn as a combination of three main areas: computer science, textiles, and statistics. The intersection of these three areas also forms other subareas closely related to DMTI, such as traditional software, traditional research, DM, and machine learning (ML) as shown in Figure 1.1. Traditional data analysis in the textile sector is assumption-driven, meaning that a hypothesis is formed against the data, whereas in contrast, DM is discovery-driven, meaning that patterns are automatically discovered from textile data. Because of this reason, the aim of this thesis is implementing novel data mining techniques for processing textile data and converting it to useful patterns to obtain valuable knowledge and making right decisions to increase quality and productivity.

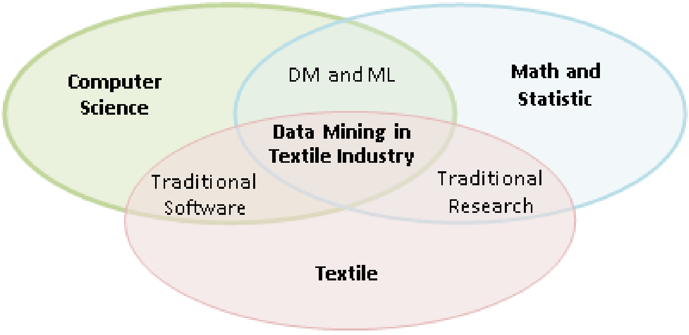


Figure 1.1 Main areas related to data mining in textile industry

Application of data mining techniques in textile sector for deriving useful patterns presents several advantages and these are given below:

- Prediction of parameters expected based on other parameters or under different cases in textile studies
- Construction of models to reduce the consumption of textile-related materials, such as fabrics, yarns, dyes, and sewing threads
- Discovering patterns that can be used to produce better textile end-products
- Analyzing textile data to achieve better customer satisfaction
- Recognition and classification of textile defects for quality control
- Discovering customers’ purchasing habits to increase the sales of textile products
- Identifying staff-related patterns in a textile factory
- Discovery of hidden, interesting, and meaningful rules and valuable correlations among textile data using association rule mining algorithms
- Determination of the most important factor that affects the performance of a yarn or fabric property using a DM technique such as decision tree
- Detection of anomalies in textile data using the clustering algorithm to identify bad values, changes, errors, noises, frauds, and abnormal activities to realize the purpose of giving an alarm
- Development of a model to manage resources effectively in textile industry
- Analyzing the records of financial transactions in the textile sector for better decision making
- Using DM as a pre-processing step before performing the essential textile study
- Text mining to extract interesting patterns and perform textile knowledge extraction from the unstructured textile documents that are obtained from different sources
- Clustering the items in textile data to describe the current situation more clearly and to plan different activities for different groups

- Usage of process mining in the context of workflow management and to improve the processes in the textile sector

Considering these motivations, in this thesis, five different data mining studies were performed on textile data to obtain valuable knowledge and making right decisions to increase quality and productivity for textiles. The purposes of the studies conducted within the scope of this thesis can be listed below:

- Determination of the most important parameters for stab performances of plain, plush, doubleface and doublefaceinlay knitted aramid samples at three energy levels
- Improving prediction performance of the parameters under ten different cases in textile studies
- Classification of 28x28 gray-scale images to fashion products
- Preventing both chaining and rounding effects of hierarchical clustering on textile items
- Discovering the relationships between yarn parameters and fabric properties

1.3 Novel Contributions of this Thesis

The main contributions of this thesis are on five levels;

First, we focused on the determination of the most important parameters for stab resistance performances of knitted structures using six different classification algorithms. In this study, stab performances of plain, plush, doubleface, and doublefaceinlay knitted aramid samples at three energy levels are analyzed to predict the model with six different classification algorithms; two of them which are ensemble learning algorithms were never used before in textile sector. According to experiment results, the decision tree is proposed as the most successful algorithm because it reveals the important parameters and also their critical values for perfect stab performances of knitted samples so that it will be possible to plan the knitted structure parameters before fabric manufacturing process according to these critical values for the intended level of protection.

Second, we proposed an ensemble learning approach that combines neural networks with different parameter values (the number of hidden layers, learning rate, and momentum coefficient) to improve prediction performance in textile sector. It is the first study that the proposed ensemble learner has been implemented in textile sector. This study also compares ensemble neural networks with a single neural network in terms of correlation coefficient and relative absolute error measures on different textile datasets.

Third, we developed a novel advanced neural network architecture that contains convolutional, max pooling, and fully connected layers to classify fashion products. This study also compares the proposed convolutional neural network (CNN) with ensemble learning methods (i.e. Bagging, Random Forest and AdaBoost) in terms of classification accuracy.

Fourth, we proposed a novel approach, named *k*-Linkage, which calculates the distance by considering *k* observations from two clusters separately. This study also introduced two novel concepts: *k*-min linkage and *k*-max linkage. While *k*-min linkage considers *k* minimum (closest) pairs from points in the first cluster to points in the second cluster, *k*-max linkage takes into account *k* maximum (farthest) pairs of observations.

The last, we uncovered relationships between yarn parameters and fabric properties using an extended FP-Growth algorithm in association rule mining. This study extracted different types of frequent itemsets (closed, maximal, top-*k*, top-*k* closed, top-*k* maximal) that have not been determined in textile sector before. It also proposed two novel concepts, closed frequent item and maximal frequent item, to identify significant items in data.

Consequently, in this thesis, (i) important parameters of knitted structures for stab performance were determined and ensemble learning algorithms were applied on textile sector for the first time, (ii) an ensemble learning approach that combines neural networks with different parameter values was proposed to improve prediction performance in textile sector, (iii) a novel convolutional neural network (CNN) architecture was developed to classify fashion products (iv) a novel hierarchical clustering approach, named *k*-Linkage was proposed and (v) an extended FP-Growth algorithm was introduced.

1.4 Organization of the Thesis

This thesis is organized in seven chapters and the remainder of the thesis is structured as follows:

In Chapter 2, existing data mining and machine learning studies implemented in textile industry were presented and explained in detail to provide an overview of how data mining and machine learning techniques can be applied in the textile industry.

In Chapter 3, data engineering, data mining and machine learning concepts with their methods; classification, clustering, and association rule mining were explained in detail.

In Chapter 4, three different classification case studies: (i) determination of the most important parameters for stab resistance performances of knitted structures, (ii) improving prediction performance on textile sector using a novel ensemble neural network model, and (iii) classifying fashion products using a novel convolutional neural network (CNN) were explained. This section also gave information about the application of the proposed models on the datasets and presented the obtained results with discussions separately.

In Chapter 5, a novel approach, named *k*-Linkage, and two novel concepts (*k*-min linkage and *k*-max linkage) were introduced. This chapter also gave background information on hierarchical clustering and the traditional linkage methods. Lastly, the experimental study was presented and the obtained experimental results were discussed in this chapter.

In chapter 6, brief background information on association rule mining and the types of frequent patterns were given firstly. Then, novel concepts with their definitions and the extended (proposed) version of the FP-Growth algorithm were introduced. This section also presented the obtained experimental results.

Finally, in Chapter 7, some concluding remarks and future directions were presented.

CHAPTER TWO
RELATED WORK

In this chapter, data mining studies, including classification, clustering and association rule mining techniques and machine learning algorithms, implemented in textile industry were presented and explained in detail to provide an overview of how these techniques can be applied in the textile industry to deal with different problems where traditional methods are not useful (Yildirim, Birant & Alpyildiz, 2018). The present studies clearly show that a classification technique has higher interest than both clustering and association rule mining techniques in the textile industry. This review also shows that the most commonly applied classification methods are artificial neural networks and support vector machines, and they generally provide high accuracy rates in the textile applications. For the clustering task of data mining, a *k*-Means algorithm was generally implemented in textile studies among the others that were investigated in this study.

2.1 Review of Classification Studies in Textile Industry

Classification is the process of analyzing the input data to develop an accurate model using the features present in the data and then using this model to assign new input data to predefined classes. A classification algorithm finds relationships between the values of the predictors and the values of the target. Applications of classification include document categorization, diagnostic prediction, price prediction, risk assessment, and sentiment analysis.

Classification is the commonly preferred DM technique in the textile industry because it provides predictions about unknown properties and parameters in textile studies (Kumar & Sampath, 2012). For example, Akyol, Tufekci, Kahveci & Cihan (2014) presented a predictive model for the estimation of the drying period of wool yarn bobbins and compared five categories of ML regression methods: functions, lazy-learning algorithms, meta-learning algorithms, rule-based algorithms, and tree-based learning algorithms. In their study, these methods were applied on datasets that consists of 20 parameters using Waikato environment for knowledge analysis (WEKA) DM tool. A reduced Error Pruning (REP) tree algorithm was utilized as the most successful algorithm among the others used in their study.

In the literature, there are five types of classification algorithms utilized in the textile industry: (i) artificial neural networks, (ii) support vector machines, (iii) Bayesian classifiers, (iv) decision trees and (v) *k*-nearest neighbors.

2.1.1 Neural Network in Textile Industry

Artificial Neural Networks (ANN) is a classifier inspired by the biological neural structure of the brain, which predicts unknown attribute values depending on input data, and it is a popularly used classification method in the textile industry. An ANN consists of a large number of highly interconnected processing elements (neurons) and weight values on the connections among them.

ANNs have been used in the textile industry for different purposes. Some studies were carried out for production planning (Jaouachi & Khedher, 2015) by evaluating and predicting the sewing thread consumption of jean trousers, automatic fabric fault detection (Eldessouki, Hassan, Qashqari & Shady, 2014; Su & Lu, 2011; Xin, Li, Qiu & Liu, 2012) by classifying faults via real time fabric images, performance development (Ahmad, 2016; Behera, 2006; Hu, Ding, Yu, Zhang & Yan, 2009; Mozafary & Payvandy, 2013; Ozkan, Kuvvetli, Baykal & Sahin, 2015) by maintaining predictive models for the relationship between textile parameters and performance. Another type of problem for which the NN technique was used in textile industry, was to predict utility properties of textile materials, such as the moisture and heat transfer rate in fabrics (Rahnama, Semnani & Zarrebini, 2013) and air permeability of fabrics (Matusiak, 2015). The use of neural networks for color science (Hamrouni, Kherallah & Alimi, 2011) and textile printing (Golob, Osterman & Zupan, 2008) was also investigated. For example, Golob et al. (2008) proposed to determine the correct pigment combinations of dyes for textile printing using the ANN technique. In their study, a collection of 1430 printed samples obtained from 10 dyes was used as a training dataset.

Quality control plays an important role in the textile industry. Traditional human inspection can result in mistaken judgments, an increase in costs, and low-speed manufacturing. Therefore, some researchers (Eldessouki et al., 2014; Su & Lu, 2011; Xin et al., 2012; Xin, J. Zhang, R. Zhang & Wu, 2017) used an ANNs to detect textile-related defects such as fabric defects (i.e., yarn, woven fabric, knitted fabric, dyeing defects) and garment defects (i.e., cutting, sewing, embroidery, and accessories defects). Su & Lu (2011) proposed an automated vision system for the detection of lycra spandex defects by extracting the features of fabric image textures using the gray level co-occurrence matrix and then applying a backpropagation neural network (BPNN) to establish flaw classifications of the fabric. In another textile study, Xin et al. (2012) proposed an expert system for the quality evaluation of fabric wrinkle appearance based on image analysis and using ANN. Xin et al. (2017) developed an artificial intelligence system based on a dual-side co-occurrence matrix and BPNN to classify color texture in their study. Eldessouki et al. (2014) proposed an automated fabric defect system using ANN, utilizing principal component analysis (PCA) to reduce the dimensionality of the features without losing the high variation embedded in the original data.

In textile studies, the most preferred type of ANN is the *feed-forward neural network* (FFNN) in comparison to the use of *Recurrent neural networks* (RNN) and *Competitive neural networks* (CNN) (Hamrouni, Kherallah & Alimi, 2011) (Figure 2.1). In FFNNs, the information moves in only one direction (forward) -from the input layer, through the hidden layers (if any) and to the output layer- while it is bidirectional in RNNs. As an example of FFNN application, Rahnama et al. (2013) developed an intelligent model that measures the moisture and heat transfer rate in light-weight nonwoven fabrics.

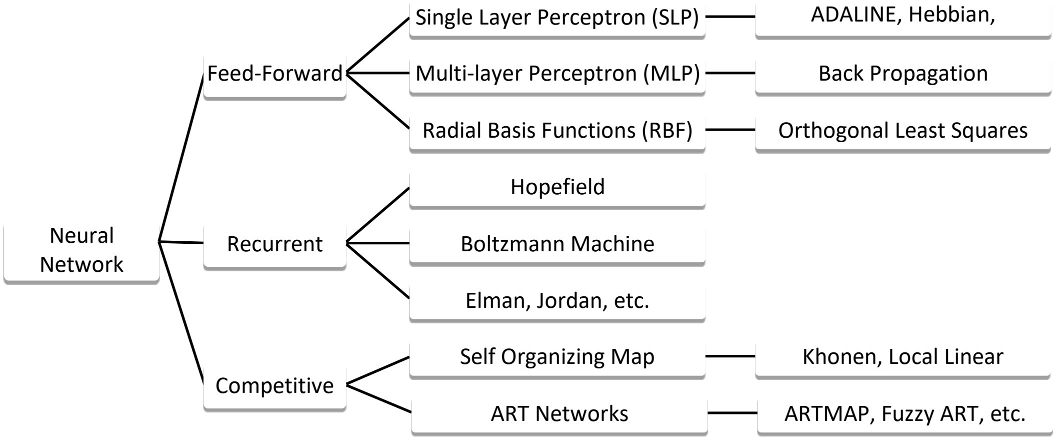


Figure 2.1 Types of neural networks commonly used in the textile industry

The most widely used model in FFNN for textile studies is *multilayer perceptron* (MLP), in comparison with *single-layer perceptron* (SLP). As an example textile study, SLP was applied on worsted woolen yarn samples to classify the unknotted joints of yarn ends over a dataset consisting of 1250 experiments (Lewandowski, 2011). An example study in the textile sector that used MLP was performed by Ozbek, Akalın, Topuz & Sennaroglu (2011). They aimed to estimate Turkey's denim trousers export during a year by considering 23 parameters, such as the minimum wage, the price of cotton, electricity, the credit usage of ready-made clothing enterprises, brands of denim trousers, and others. Their study clearly showed that MLP models predicted more successfully when compared with RNNs. Another textile study (Matusiak, 2015) that used MLP was performed to predict the air permeability of woven fabrics. A *radial basis function* (RBF) network is another type of FFNN that uses RBFs as its activation functions. As an example study that used RBF in textile sector, Behera (2006) introduced an expert system for the prediction of both construction and performance parameters of canopy fabrics. In experiments, predicted and actual values were compared, and the proposed system showed good prediction performance. Within the textile industry, a number of studies have been reported that compares MLP and RBF models. Z. Yildiz, Dal, Ünal & K. Yildiz (2013) used these techniques to predict the seam strength and elongation at break in garments of poplin and gabardine woven fabrics. They concluded that the best modeling results were obtained using MLP in the training process and RBF in the testing process; however, in general, there was a consistent similarity between the experimental results.

Although several types of NN algorithms have been proposed, the most popular one, which is used in textile studies, is the *back propagation algorithm* (BPNN). In the textile industry, this algorithm was used to develop a predictive model of a polyurethane-based coating process for forecasting the final characteristics of a coated fabric (i.e., thermal insulation, strength, dimensional stability) based on the process parameters (Furferi et al., 2012). Another textile study that used BPNN was performed by Farooq & Cherif (2008) to identify the leveling action point at the auto-leveling draw frame. One of the challenges facing in textile applications is that BPNN requires overly time-consuming trial and error to find the learning parameters. To overcome this problem, Su & Lu (2011) used the Taguchi method combined with BPNN. Another problem encountered in textile studies is the computational overhead of BPNN. To eliminate this challenge, the Levenberg–Marquardt algorithm was incorporated into the back propagation to accelerate the training, and Bayesian regularization was utilized to reduce the testing error for the practical textile applications (Farooq & Cherif, 2008).

When researchers compared ANN with regression analysis in textile studies (Haghighat, Johari, Etrati & Tehran, 2012; S. N. Ogulata, Sahin, R. T. Ogulata & Balci, 2006; Uçar & Ertuğrul, 2007), they proposed ANN to be able to get more accurate results. For example, Haghighat et al. (2012) used ANN and multiple linear regression to predict the hairiness of polyester-viscose blended yarns based on various parameters. The comparison results indicated that ANN had better performance, rather than multiple linear regressions, on the yarn hairiness prediction. Similarly, Uçar & Ertuğrul (2007) predicted the amount of fuzz, which was determined by image processing techniques on the fabric surface using ANN and regression analysis. According to experimental results, it was observed that NN provided more accurate results than regression analysis. Jaouachi & Khedher (2015) also proposed NN for the prediction of the amount of sewing thread consumption required to assemble jean trousers, rather than *linear regression* methods. In the case where ANN was checked against *multiple regression*, S. N. Ogulata et al. (2006) indicated that both of the models could be used to estimate the fabric or yarn properties accurately.

While some textile studies used a single activation function for all NN layers (e.g., sigmoid (S. N. Ogulata et al., 2006), linear (Lewandowski, 2011), most of the studies (Haghighat et al., 2012; Kumar & Sampath, 2012; Jaouachi & Khedher, 2015; Matusiak, 2015; Ozbek et al., 2011; Uçar & Ertuğrul, 2007; Xin et al., 2012; Yildiz et al., 2013) proposed the usage of different activation functions for different layers (i.e., sigmoid and linear activation functions in the hidden and output layers, respectively). Several textile studies (Bahadir, Kalaoglu, Jevsnik, Eryuruk & Saricam, 2015; Lewandowski & Drobina, 2008) aimed to perform the analysis of different activation functions to figure out the optimal function for a problem. For example, Bahadir et al. (2015) tried both sigmoid and hyperbolic tangent functions to show the performance of the model constructed to identify the fabric drape of woolen fabrics treated with different dry finishing processes. The results indicated that the hyperbolic tangent function presented better performance results than the sigmoid function for this problem, with lower mean square errors.

In the textile industry, hybrid systems based on NN have also been studied during the last decade. For example, Hu et al. (2009) developed hybrid systems for fit garment design based on both the NN and immune co-evolutionary algorithm (NN-ICEA), and the NN and genetic algorithm (NN-GA). The proposed approaches were applied on a dataset of 450 pairs of pants to demonstrate their prediction capabilities of them on the fit garment design.

The technical difficulty, when applying the ANN in a textile study, is to design the network correctly, especially in determining the type of ANN, its topology (number of hidden layers and processing elements in each layer), learning algorithm, stopping criteria, and activation function(s). Determining the value of the parameters (i.e., momentum, learning rate, bias value) is equally important for accurate work with ANNs to obtain reliable results. Some textile studies (Bahadir et al., 2015;

Farooq & Cherif, 2008; Haghighat et al., 2012; Matusiak, 2015; Ozbek et al., 2011; Su & Lu, 2011; Yildiz et al., 2013) focused on the design issues in the building of an artificial neural network classifier. For example, Bahadir et al. (2015) tested the neural network efficiency by changing the number of neurons for the problem of predicting the drape behavior of woolen fabrics treated with different finishing processes.

2.1.2 Support Vector Machine in Textile Industry

Support vector machine (SVM) (Salcedo-Sanz, Rojo-Alvarez, Martinez-Ramon & Camps-Valls, 2014) is a supervised learning model that is based on statistical learning theory, the concept of decision planes and structural risk minimization. SVM algorithm constructs a hyperplane in a high-dimensional space for the prediction of class labels. In order to avoid overfitting and high-dimensionality problems, SVMs choose the maximum margin separating the hyperplane and defined a kernel function to map the training data into a higher-dimensional feature space.

SVM has been applied to different textile problems, such as predicting fabric type (Ghosh, Guha & Bhar, 2015) and fabric parameters (Yap, X. Wang, L. Wang & Ong, 2009), predicting yarn properties (Abakar & Yu, 2013, 2014; Lü, Yang, Xiang & Wang, 2007), fiber identification (Lu, Zhong, Li, Chai, Xie, Yu & Naveed, 2018; Wan, Yao, Zeng & Xu, 2009), and color management (Zhang & Yang, 2014) in textile printing and dyeing. Similar to the ANN, SVM has also been applied for quality management (Nurwaha & Wang, 2012; Su, Yao, Xu & Bel, 2011) in textile sector, especially for defect classification (Li & Cheng, 2014) in textile textures. Lu et al. (2018) proposed an approach to identify microscopic images of cashmere and wool fibers using SVM. Features of the images are extracted and reformulated using a bag-of-words model and the identification process is performed by SVM classifier.

Surface inspection-based quality control studies in the textile industry have been analyzed in two steps. The SVM algorithm was applied as the second step to predict the class of that data, while fabric images were initially preprocessed and parameters were defined by a feature extractor. For example, Li & Cheng (2014) presented a yarn-dyed woven fabric defect model using combined feature extractors and modified SVM classifiers. The model was tested on 180 selected defect images of yarn-dyed fabrics, including different patterns, using a cross validation technique. According to the test results, more than 91% of the samples were accurately recognized and classified by the proposed method. Another textile study related to quality management developed a predictive model for the wrinkling appearance of fabric using modified wavelet coefficients and SVM classifiers. In the testing step, the developed model was applied on 300 images of five selected fabrics that had different weave structures, fiber contents, colors, and laundering cycles. Obtained validation results indicated that approximately 78% instances were correctly classified by the proposed model.

Several textile studies (Nurwaha & Wang, 2012; Lü et al., 2007; Zhang & Yang, 2014) compared SVM with other classification techniques. For example; Zhang & Yang (2014) tried both SVM and Naive Bayes algorithms for the prediction of color differences between the evaluated dyed fabrics. The results indicated that it was possible to increase the prediction accuracy by 9% with the SVM model. Another study by Lü et al. (2007), also proposed SVM instead of ANN for the prediction of worsted yarn properties, indicating that SVM provided more stability for predictive accuracy than ANN under real datasets and small population circumstances, such as worsted yarn spinning process being noisy and dynamic. Nurwaha & Wang (2012) developed an intelligent control system for estimating textile yarn quality by comparing six techniques: the General Neural Network (GNN), the Group Method of Data Handling Polynomial Neural Network (GMDHP), Gene Expression Programming (GEP), SVM, MLP, and RBF neural networks. When the estimation performances of these techniques were compared, the lowest error values were provided by the SVM model, followed by GEP.

As in the other domains, one of the challenges faced in textile studies is to search for the optimal parameters of the SVM. To overcome this problem, some textile studies used genetic algorithms for optimization (Abakar & Yu, 2013; Zhang & Yang, 2014). Another technical difficulty in textile studies is to determine independent parameters from the training data to reduce the dimensionality of the feature vector, improving the speed of the evaluation and accuracy through this way. To solve this problem, several textile studies (Mustafic, Jiang & Li, 2016; Jing, Zhang, Kang & Jia, 2012; Wu, Chen, Wang, Sun & She, 2012; Zhang & Yang, 2014) used PCA technique.

SVMs can be classified into two categories, *linear* and *nonlinear* according to the kernel function they used. Some textile studies (Ghosh et al., 2015) performed linear classification. For example, Ghosh et al. (2015) presented a study for the identification of handloom and powerloom fabrics using a proximal support vector machine (PSVM). When the model was tested by applying a *k*-fold cross validation technique, the prediction results indicated that PSVM categorized handloom and powerloom fabrics efficiently and correctly, with 98.75% accuracy rate. In addition to performing linear classification, SVMs can efficiently perform a nonlinear classification by mapping input space into a high-dimensional, even infinite dimensional, feature space. As a nonlinear SVM study example, Wan et al. (2009) introduced an automated identification system for fiber cross-sectional shapes. Firstly, the shape features were characterized using a distance-based Skeletonization algorithm, and then, SVM with a nonlinear kernel function was implemented to classify shaped fibers; a total of 1200 samples have been evaluated. The fibers in the six testing groups were classified with 95.43% average accuracy, compared to human inspection. Some textile studies (Sun et al., 2011; Yap et al., 2009) performed both linear and nonlinear SVM to compare them. Yap et al. (2009) presented a study for the prediction of wool knitwear pilling propensity. While the linear kernel function showed an average prediction accuracy of 85%, the nonlinear kernel achieved the highest performance, with a 90% prediction accuracy.

Studies conducted in the textile industry showed that the kernel function played an important role in the accuracy of the SVM model forecast. For this reason, some textile articles (Abakar & Yu, 2014; Yap et al., 2009) compared the kernels (i.e., polynomial, sigmoid or RBF) with respect to their prediction performances. Abakar & Yu (2014) compared SVM models based on polynomial, radial basis, and Pearson functions in the prediction of yarn tenacity. When a k-fold cross validation technique was used as the evaluation method, the results showed that it is possible to provide correct predictions on the yarn properties using SVM with Pearson function as well as the SVM mode based on RBK kernel and ANN. The same researchers (Abakar & Yu, 2013) also proposed the prediction of yarn tenacity using SVMs for regression (SVMR). The genetic algorithm for feature selection was also used as the preprocessing stage to be able to select the best attributes related to the prediction of yarn tenacity. The proposed approach was compared with a noisy model of SVMR, and obtained results showed that the hybrid approach achieved higher predictive performance than the noisy model.

In the textile sector, the RBF (Jing et al., 2012) is the most commonly selected kernel function used in SVMs. This is mainly because of its accurate and reliable performances in applications. To obtain the highest possible classification accuracy, it is necessary to find a set of optimal parameters, including the regularization value (c) and the width of RBF kernel (σ) (Wu et al., 2012). As an example textile study that used RBF kernel, Wu et al. (2012) developed an intelligent clothing framework that recognized human daily activities using a single waist worn tri-axial accelerometer sensor. Six different physical activities obtained from 492 samples were recognized and classified into predefined categories using the SVM method. The proposed activity recognition method was validated, with a mean classification accuracy of 95.25%.

2.1.3 Bayesian Classification in Textile Industry

Although NN and SVM techniques offer reasonably good solutions for textile problems, researchers have also tried using the Naive Bayes algorithm. *Naive Bayes* (Cichosz, 2015) is one of the well-known and highly scalable classification algorithms, which uses Bayes Theorem to calculate conditional probabilities for the determination of unknown class values of samples.

As a well-established classification algorithm, the Naive Bayes algorithm has been used to overcome problems in various types of textile applications, such as the objective evaluation of surface roughness (Hu, Xin & Yan, 2002), color difference evaluation in fabric dyeing (Zhang & Yang, 2014), and fabric pilling evaluation (Kim & Kang, 2005).

In the textile industry, as a two-stage process, image analysis techniques have been generally used as the first step before Bayesian classification. Firstly, textile surface characteristics (i.e., surface roughness, color mean values, number, area, and density of pills) were extracted by image processing, and then, the classification algorithm was applied to evaluate fabric appearance, such as fabric pilling or fabric wrinkle. For example, Kim & Kang (2005) presented a study for fabric pilling evaluation using four classification algorithms: Naive Bayes, minimum distance, k-nearest neighbors, and neural network. The grades of fabric samples were predicted comprehensively, and high accuracy rates (≥90) were obtained using these algorithms. Hu et al. (2002) proposed a morphological fractal method to analyze the fabric surface objectively after abrasion, and the Naive Bayes method is applied to classify the pilling quality grades objectively.

The challenges arising in textile applications when Bayesian classification is used can be indicated as missing data, continuous variables, attribute independence, and zero observations. When proposing solution approaches, each challenge can be overcome in the following ways. If a textile data instance has a missing value for an attribute, it can be ignored when a probability is calculated for a class value or it can be filled with an appropriate value while preparing data. In the case when an attribute is continuous, it would either need to be converted to a discrete variable, or a probability density function (i.e., Gaussian), which describes the distribution of the textile data, should be used. If the attributes in a textile data are correlated or dependent, Bayesian networks can be used. If an attribute has a value that was not observed in training, the model assigns a zero probability, and thus, it cannot make a prediction. To overcome this problem, the Laplace correction technique can be used to assign arbitrarily low probabilities.

2.1.4 Decision Trees in Textile Industry

The *decision tree* (DT) is a supervised learning algorithm that classifies unknown attribute values by constructing a tree that is a conjunction of rules.

Attempts have been made to increase the accuracy of the knowledge discovery process by applying DT techniques to textile data. Agarwal, Koehl, Perwuelz & Lee (2011) aimed to reveal the effectiveness level of wash-ageing and fabric softener usage on the mechanical properties (such as tensile extension, shear and bending rigidity, compression energy, fabric roughness) of knitted fabrics using a DT algorithm over 104 types of different samples with different fiber types and fineness, yarn construction, and fabric structure. The same researchers (Agarwal, Koehl, Perwuelz & Lee, 2010) also investigated the softener pickup amount and its uniformity using the same dataset. In these studies, DT was indicated to be used for the better interpretation of the large set of variants.

The DT algorithm has also been implemented for the benefit of sizing determination problems of garment manufacture in the textile industry. The size of the garment with the best fitting can be a problem if not based on anthropometric data, which means a large set of data and variables. Thus, researchers indicated that the use of the DT algorithm provided the benefits for the use of anthropometric data, in that it allows for wider coverage of body shapes with fewer number of sizes and generates regular sizing patterns and rules. For example, Hsu & Wang (2004) introduced a new pants sizing system for the manufacture of garments to determine the pants sizes of army soldiers. The sample dataset consisted of 265 static anthropometric variables evaluated from 610 soldiers in Taiwan, resulting in 160,000 pieces of data. The classification and regression trees (CART) algorithm was utilized to discover body dimension patterns and categorize them into specific figure types. Researchers defended that their study as being useful for the specification of standard-sizing systems to produce military uniforms in Taiwan. In another study, Zakaria (2011) also proposed a sizing system for school-aged children in Malaysia using the DT technique. The sample dataset contained the body characteristics of 1001 randomly selected girls from 29 different schools in Malaysia.

In textile studies, the DT technique has been generally used to predict the probability of events of interest. For example, it was used to predict wash-ageing of fabrics (Agarwal et al., 2011) drying time periods of bobbins (Akyol et al., 2014), and the amount of softener picked up by the fabric (Agarwal et al., 2010) and to determine the most important factors/parameters among many variables for an effective sizing system (Hsu & Wang, 2004). For such studies, DT was preferred because it is capable of expressing the degree of relationships between output and input variables and selects the most important attributes when constructing the tree.

There are several challenges in using DTs effectively in textile studies, including overfitting, tree pruning, noisy data, and irrelevant attributes. To overcome these problems, from our point of view, future research is necessary on the use of the random forest (Ziegler & König, 2014) technique in textile studies.

2.1.5 K-Nearest Neighbor in Textile Industry

K-nearest neighbors (KNN) (Keller & Gray, 1985) is a lazy learning classifier that determines a class label by considering a majority vote of its closest *k* (a user-defined constant) neighbors using a similarity measure.

In the textile industry, KNN has been used for two types of problems: (i) as a classification technique (Mariolis & Dermatas, 2010; Yildiz, Buldu & Demetgul, 2016) and (ii) as a pre-processing step before applying a DM algorithm (Yu, Hui, Choi & Ng, 2010). In the first category, several applications have been reported. For example, Mariolis & Dermatas (2010) presented an automated seam quality control system based on surface roughness estimation with the help of the KNN algorithm. The experiment material used in their study included 211 seam specimens from two kinds of fabrics. The classification performance of textile seam quality of KNN was calculated with accuracy rate of 81.04%. Similarly, Yildiz et al. (2016)

also used KNN algorithms to classify defects in textile fabrics via the obtained properties of feature-extracted images using a thermal camera. As a preprocessing step before applying DM, Yu et al. (2010) proposed a model for fabric hand prediction using fuzzy NN. In their study, the KNN algorithm was used for the feature selection to reduce the computational cost by decreasing the number of input variables.

There are four challenges to using KNN effectively in textile studies: (i) finding the optimum value of parameter *k*, (ii) selecting the best distance measure, (iii) reducing features, and (iv) weighting features. A major challenge in KNN classification is how to choose the optimum value of the neighborhood parameter *k*. In order to overcome this problem in textile studies, the KNN algorithm is run for various *k*, and the value achieving the highest classification accuracy is then selected as the optimal value for *k* (Mariolis & Dermatas, 2010). Another challenge is to select the best distance measure from a set of alternative metrics (i.e., Euclidean, City-Block, Minkowski, etc.) in order to obtain the highest classification accuracy for the given data. As a solution, Mariolis & Dermatas (2010) used and compared different distance measures for the assessment of textile seam quality. The third problem is that textile data may have numerous features in dimension, and many of these features may be irrelevant or redundant. In this case, PCA (Glosh et al., 2015; Wu et al., 2012; Zakaria, 2011; Zhang & Yang, 2014; Jing et al., 2012) or filter-based feature selection (Akyol et al., 2014) methods can be used to reduce the dimensionality before applying the KNN algorithm. The last challenge is to identify the weights of features if the different importance of variables is take into consideration. Kim & Kang (2005) show that it is possible to increase classification accuracy by weighting features in textile studies.

2.1.6 Comparison of Classification Methods in Textile Industry

The classification based textile studies mentioned so far are compared in Table 2.1. In this table, the scopes of the studies, year they were performed, algorithms that were used in the studies, and success rate with the validation method are listed. In addition, if more than one algorithm is presented and compared with each other, the proposed one (the most successful one) is also indicated.

2.1.7 Classification Validation

The previous section elucidated different ways of obtaining knowledge from textile data using various classification techniques. One of the main questions when classification is performed in the textile industry is how to validate the classification results. Expert knowledge plays an important role in validating the result of these studies. Besides, some validation methods have been used, such as *correlation coefficient* (Bahadir et al., 2015; Haghighat, 2012; Lewandowski & Drobina, 2008; Matusiak, 2015; Ozbek et al., 2011; S. N. Ogulata, 2006; Uçar & Ertuğrul, 2007; Yildiz et al., 2013), *holdout validation* (Kuo & Juang, 2016), and *k-fold cross validation* (Ghosh et al., 2015; Nurwaha & Wang, 2012; Li & Cheng, 2014; Sun et al., 2011; Wu et al., 2012). For the k-fold cross validation method, it is possible to see different *k* values used in different textile studies, such as k=10 in the studies (Ghosh et al., 2015; Nurwaha & Wang, 2012), k=5 in the study (Sun et al., 2011) and k=4 in another work (Wu et al., 2012). In order to decide, different *k* values can be tried, and the average one or the best one that has a sufficiently small deviation can be chosen.

Table 2.1 Comparison of classification studies in textile industry (Reg, regression; MAE, Mean absolute error; MSE, Mean squared error; RMSE, Root mean squared error; R, Correlation coefficient)

Ref	Year	Type of Problem	Algorithms												Validation Method & Accuracy	Proposed Algorithm
			Artificial Neural Network (ANN)						Reg	SVM	Bayes	DT	KNN	Other		
			FFNN	RNN	SLP	MLP	RBF	BPNN								
Kuo & Juang	2016	Determination of embroidered textile defects	√			√		√							Train (70%) Test (30%) Accuracy=100%	ANN
Akyol et al.	2015	Prediction of drying time period of wool yarn bobbins				√	√		√			√	√	REP Tree, Kstar, etc.	MAE=0 RMSE=0	REP tree
Jaouachi & Khedher	2015	Prediction of the sewing thread consumption of jean trousers	√			√		√	√						R^2=0.973	ANN
Matusiak	2015	Prediction of the air permeability of woven Fabrics	√			√									R=0.9848 MAE=0.1714	ANN
Bahadir et al.	2015	Prediction of the drape behavior of woolen fabrics treated with dry finishing processes	√			√		√							R=0.92 MSE=0	ANN
Xin et al.	2017	Color texture classification						√							R=0.9726	BPNN
Eldessouki et al.	2014	Determination of plain woven fabric defects	√			√									correct classification rate (CCR) = 90%	ANN (and principal component analysis)
Mozafary & Payvandy	2013	Prediction of worsted spun yarn quality	√			√		√						K-Means	Regression=0.79177	K-means and ANN with Levenberg–Marquardt
Yildiz et al.	2013	Modelling of seam strength and elongation at break				√	√								R≈1 MSE=3.33e-05	MLP or RBF
Rahnama et al.	2013	Measurement of the moisture and heat transfer rate in light-weight nonwoven fabrics	√					√							error < 4.7% for heat < 7.9% for moisture	ANN

Table 2.2 continues

Kumar & Sampath	2012	Prediction of the dimensional properties of a weft-knitted double cardigan structure made from 100% cotton ring spun yarns	√			√		√							MSE=0.84 R^2 >0.98	ANN with Levenberg–Marquardt alg.
Xin et al.	2012	Quality evaluation of fabric wrinkle appearance based on image analysis	√		√			√							R^2 = 0.9798 MSE=0.004635	ANN
Furferi et al.	2012	Prediction of the final characteristics of a coated fabric, based on the process parameters	√			√		√							average error < 5.5%	ANN
Haghighat et al.	2012	Prediction of the hairiness of polyester-viscose blended yarns	√			√		√	√						R = 0.967 MSE=4.58 PF/3=7.059	ANN
Su & Lu	2011	Determination of lycra spandex defects	√			√		√							RMSE=0.000104 Accuracy=97.14 %	Taguchi-based BPNN
Hamrouni et al.	2011	Textile plant modeling		√		√									error = 0.013	ANN
Lewandowski	2011	Determination of the unknotted joints of yarn ends			√									Adaline	coefficient = 74.12%	Adaline
Ozbek et al.	2011	Estimation of Turkey's denim trousers export		√		√									R=0.9314 RMSE=0.0089	ANN (MLP version)
Hu et al.	2009	Fit prediction in garment design (pants)	√			√								ICEA, GA	error rate < 0.1	Hybrid (NN-ICEA)
Farooq & Cherif	2008	Prediction of the leveling action point at the drawframe auto-leveling	√			√		√							10-fold cross-validation R^2 = 0.9622	BPNN and Levenberg–Marquardt alg.
Ucar & Ertugrul	2007	Prediction of fuzz fibers on fabric surface	√			√		√	√						R=0.88	ANN

Table 2.3 continues

Behera	2006	Prediction of both construction and performance parameters of canopy fabrics					√								differences between actual and predicted values for all parameters	ANN
Ogulata et al.	2006	Prediction of elongation and recovery of woven bi-stretch fabric	√			√		√	√						R=0.992	ANN or Linear Regression

2.2 Review of Clustering Studies in Textile Industry

Clustering is the process of grouping a set of objects according to their similarities. It helps us better understand the characteristics of data because fewer groups are more easily interpreted. Several studies have focused on the application of the clustering process in the textile industry with the aid of algorithms named K-Means (Esfandarani & Shahrabi, 2012; Soltani, Shahrabi, Asadi, Hadavandi & Johari, 2013; Song & Ashdown, 2011; Yıldırım & Başer, 2011; Zhang, Xin, Fang & Cao, 2015), Fuzzy C-Means (Kuo, Lan, Dong, Chen & Lin, 2018; Kuo, Shih & Hsu, 2011; Shih, Kuo & Cheng, 2016) and Hierarchical (Nourani, Jeddi & Moghadam, 2011; Prada, Curran & Furton, 2014).

K-means is an easily implemented algorithm that divides the given dataset into k clusters by determining the centroids of each cluster. Zhang et al. (2015) used this algorithm to cluster interlaced, multi-colored, dyed, yarn-woven fabrics. Images captured from fabrics were divided into three sub images in red, blue and green and were then filtered in Lab color space; finally, the algorithm was processed for color clustering. In another study, Yıldırım & Başer (2011) also used the *K-Means* algorithm but for a different purpose -in order to determine cloth fell position. In their study, line laser lights were used as reference lines, and wavelet transform was utilized for the segmentation process.

Garment fitting problems in the textile industry are instances of clustering problems and were tackled via PCA and *K-means* algorithms (Esfandarani & Shahrabi, 2012; Song & Ashdown, 2011) for size charts with higher fitting. Esfandarani & Shahbari (2012) proposed a suit sizing system by segmenting the heterogeneous population to a more homogenous one, and the aggregate loss of fitness is used to evaluate the resultant sizing. Song & Ashdown (2011) presented a clustering application for the categorization of lower body shapes of adult females using PCA and K-Means algorithms. The samples, which include body shapes of 2488 women aged 18-35 with approximate body mass index of 34.14, were divided into three categories curvy shape, hip tilt shape and straight shape.

Several textile studies used clustering for subsequent analysis and further processing stages. For example, Soltani et al. (2013) used the *K-means* clustering algorithm as a preprocessing step before the actual analysis took place. They developed a three-stage hybrid model for the migration behavior of fibers: (i) key variables of samples were determined by stepwise regression analysis method; (ii) the samples divided into three clusters; and finally, (iii) yarn migration factor for each cluster was specified using the adaptive neuro-fuzzy inference system.

In terms of challenges while working with *K-means* algorithms in the textile industry that some textile researchers (Mozafary & Payvandy, 2013; Song & Ashdown, 2011) have attempted to overcome, two of them come to fore, the first of which is determining the optimal *k* value. Different cluster numbers were studied to determine the best grouping. Secondly, initial cluster centers are assigned randomly, and therefore, the final answer is dependent on the choice of initial centers. Thus, to be applied in future studies, the usage of *K-Means++* algorithm in the textile industry shall be preferred to improve both the speed and the accuracy of *K-Means*.

Recently, textile researchers have attempted to achieve high accuracy by combining and enhancing supervised techniques with clustering (Mozafary & Payvandy, 2013). Mozafary & Payvandy (2013) used both *K-means* clustering and ANN classification algorithms one after another to predict yarn quality parameters such as unevenness, nep, and thin and thick place. The model was validated using data obtained from more than 150,000 sources, including fiber, manufacturing process, and yarn quality parameters, and showed good performance for the prediction of worsted spun yarn quality. Another study that integrated clustering and classification methods into a single platform was performed by Kuo et al. (2011) They presented a texture simulation on embroidery fabrics, which uses probabilistic NNs and texture fitting methods based on fuzzy C-means. Firstly, regions and colors were separated using the clustering method, and then, texture patterns were categorized using a classification algorithm.

The main motivation for using clustering algorithms is to tackle the problem of grouping individual objects more efficiently and more accurately. To achieve this objective, some textile studies proposed combining two or more clustering algorithms. For example, Jiang, D. Zhang, Cong, A. Zhang & Gao (2014) combined multi-channel clustering and K-means clustering algorithms for the automatic identification of jacquard warp-knitted fabric patterns. According to observed experimental results, the proposed system gave exact segmentation results with high consistency and edge accuracy.

Fuzzy c-means (FCM) (Novak, Perfilieva & Dvorak, 2016) is a well-known clustering algorithm that partitions the dataset into *c* fuzzy clusters with respect to the distance between the cluster center and the data point. Xiao, Nie, Zhang, Geng, Wu & Li (2014a) used the FCM algorithm to recognize the yarn crossing areas of woven fabric. The same researchers also presented a study on the recognition of woven fabric patterns using gradient histogram and fuzzy C-means (Xiao, Nie, Zhang, Geng, Wu & Li, 2014b). Ammor, Lachkar, Slaoui & Rais (2008) conducted a similar study. They investigated a new formulation for providing the optimization of pattern recognition using fuzzy C-means in the textile field. Their study proposed a new cluster validity index that was based on the maximum entropy principle to determine the optimal number of clusters with a high degree of overlap. In another study, Kuo et al. analyzed the processing parameters for the quality characteristics of tensile strength and elongation by the fuzzy C-means algorithm. Shih et al. (2016) also proposed an automated analysis system for automated color, shape and texture analysis of Tatami embroidery fabric images using Fuzzy C-Means (FCM) clustering method.

In a hierarchical agglomerative clustering algorithm, there are three different types of inter-cluster distance measurement: *single-link*, *complete-link* and *average-link*. While some textile studies (Li, Yuen, Yeung & Sin, 2001; Prada et al., 2014) used only one approach (i.e., single link) to determine the effect of textile structural parameters on the performance, some of them (Nourani et al., 2011; Yoon & Park, 2002) used and compared all of the approaches to evaluate the relationship between the textile parameters and the properties. Li et al. (2001) combined both hierarchical and non-hierarchical algorithms with the aim to classify fabrics without noticeable color shade differences.

The clustering based textile studies performed until now are partitioning-based and hierarchical based. However, density-based clustering algorithms, such as density-based spatial clustering of applications with noise (DBSCAN), are capable of discovering clusters of any arbitrary shape and size in datasets, which even include noise and outliers. Because of these advantages, we believe that, in the future, the density-based clustering approach will begin to be used in the textile industry.

2.2.1 Comparison of Clustering Methods in Textile Industry

The clustering based textile studies mentioned so far are compared in Table 2.2. In this table, the scopes of the studies, years they were performed, and the algorithms that were used in the studies are listed.

Table 2.4 Comparison of clustering studies in textile industry

Ref	**Year**	**Type of Problem**	**Algorithms**			
			K-Means	**Fuzzy C-Means**	**Hierarchical**	**Other**
Zhang et al.	2015	Clustering interlaced multi-colored dyed yarn woven fabrics	√			
Mozafary & Payvandy	2014	Combining classification with clustering to predict yarn quality	√			ANN
Jiang et al.	2014	Identification of jacquard warp-knitted fabric patterns	√			multi-channel clustering
Soltani et al.	2013	Cluster analyses in modelling fiber migration	√			Adaptive Neuro-Fuzzy, ANN
Yildirim & Baser	2011	Clustering in order to determine the cloth fell position	√			
Song & Ashdown	2011	Categorization of lower body shapes for adult females	√			PCA
Xiao et al.	2014	Recognition for woven fabric pattern		√		Gray-level co-occurrence, gradient histogram
Kuo et al.	2011	Clustering and classification on embroidery fabric		√		probabilistic NN
Ammor et al.	2008	Optimization of pattern recognition in textile field		√		maximum entropy principle

Table 2.5 continues

Kuo et al.	2018	Analyzing the processing parameters for the quality characteristics of tensile strength		√		Taguchi method, stem cells
Shih et al.	2016	Automated color, shape and texture analysis		√		
Prada et al.	2014	Characteristic human scent compounds trapped on natural and synthetic fabrics			√	
Nourani et al.	2011	Determining the structural parameters and yarn type affecting tensile strength and abrasion of weft knitted fabrics			√	

"Classification" and "Clustering" methods are popularly preferred in textile industry. Figure 2.2 shows the number of publications that return from Elsevier's Scopus search engine when searching the terms '*Classification*' and '*Textile*' - '*Clustering*' and '*Textile*' on title / abstract / keywords parts, in each year from 2004 to 2018. As can be seen, both numbers continue to increase and classification topic has higher interest than clustering topic in textile industry.

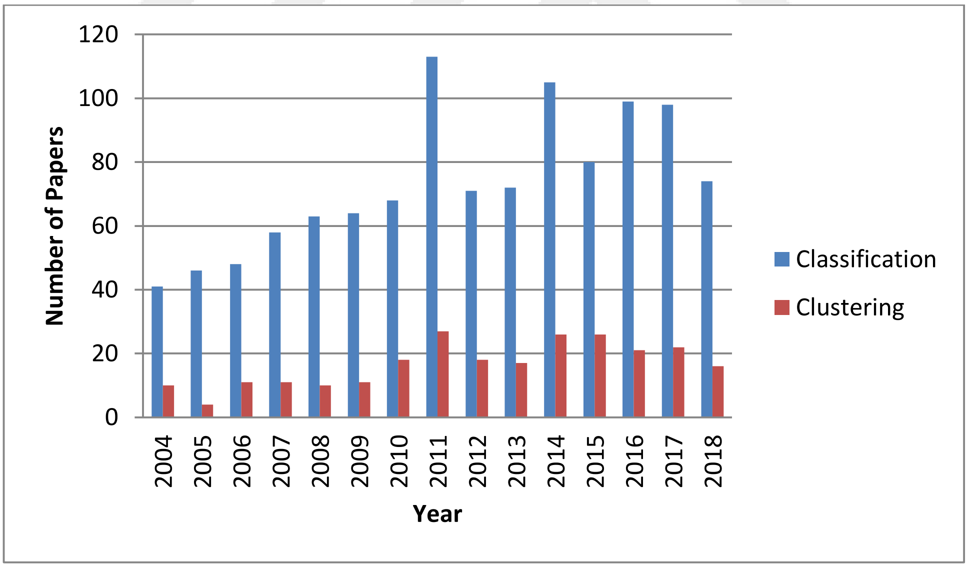


Figure 2.2 Number of publications related to classification and clustering in textile sector in Elsevier's Scopus by year

2.2.2 Clustering Validation

The validation of clustering remains an active topic in DM research (Derntl & Plant, 2016). This effort is also necessary for textile applications. Some textile studies (Jiang et al., 2014; Soltani et al., 2013) used the sum of squared error (SSE) to evaluate clusters regardless of the clustering algorithm that produced them.

SSE is commonly preferred clustering validation method which measures the variance of the clusters by calculating the sum of the squared differences between each observation and the mean value of the cluster. Being SSE value equals to zero means that each observation in the cluster is identical.

2.3 Review of ARM Studies in Textile Industry

Although ARM has a very long history in many areas, especially in marketing, only a few studies (Ciarapcia, Sanctis, Resta, Dotti, Gaiardelli, Bandinelli, Fani & Rinaldi, 2017; Huang, Qiu & Yang, 2009; Ingle & Suryavanshi, 2015; Logeswari, Valarmathi, Sangeethe & Masilamani, 2014) have been conducted so far in the textile industry. However, these studies do not aim to determine textile parameters; instead, they focus on different purposes such as textile marketing (Huang et al., 2009), ARM method verification by using a textile dataset (Ingle & Suryavanshi, 2015; Logeswari et al., 2014), and environmental strategies developed by fashion companies (Ciarapcia et al., 2017). One of the studies in the literature (Lee, Choy, Ho, Chin, Law & Tse, 2013) proposed a quality management system based on hybrid OLAP-association rule mining for extracting defect patterns in the garment industry. The experimental results presented that the proposed HQMS approach provided the quality improvement in the industry. Similarly, ARM can be used to discover useful patterns and interesting relationships among set of yarn and fabric parameters in a textile data.

CHAPTER THREE
DATA ENGINEERING AND MANAGEMENT

This chapter gives the background information about data engineering and management notions. It also explains data mining with its methods; classification, clustering, and association rule mining and machine learning approach in detail.

3.1 Data Engineering

As a result of technological innovations and developments, enormous amounts of data are generated every day in textile industry as well as in a wide range of areas such as business, education, healthcare, government, finance, social media. The increase in the amount of data that is generated creates the potential to discover valuable knowledge from it. However, it also generates a need to deal with data processing in an efficient and low-cost way. To overcome this necessity, data engineering and management concept was proposed.

Data engineering is the process of integrating raw data from various resources and managing it to prepare for the application of data mining and machine learning algorithms. Data pre-processing is the step which makes the raw data ready for the input demands of data mining and machine learning algorithms. This step consists of five steps: data integration, data transformation, data cleaning, data reduction and data discretization.

- **Data integration:** Data integration process combines data residing in different sources in a single format.
- **Data transformation:** This step converts raw data from one format to another validated format using normalization, aggregation and generalization methods.
- **Data cleaning:** It cleans raw data including incomplete, noisy and inconsistent records with the help of some techniques such as filling missing values, identifying outliers, and resolving inconsistent samples.

- **Data reduction:** Irrelevant attributes in the huge amount of data are eliminated in this step to reduce the complexity.
- **Data discretization:** Discretization is an essential task for the classification algorithms which accept only datasets that contains categorical valued records. This process divides numerical values into intervals called bins.

3.1.1 Data Mining

Data mining, which is a step of Knowledge Discovery in Databases process, is the process of extracting useful patterns or relations from raw data involving many disciplines such as machine learning, artificial intelligence, database systems and statistics. Data mining tasks can be categorized as classification, clustering, and association rule mining.

3.1.1.1 Classification

Classification is the most studied and commonly applied data mining task. It develops a model to assign new patterns into one of several predefined classes. Classification uses a training set $D = (R_1, R_2, \ldots, R_\mathrm{n})$ that has some records R, which consist of a number of attributes $\mathbf{R} = (a_1, a_2, \ldots, a_m)$ of which one (a_j) is a target outcome. Classification algorithms try to determine relationships between attributes in a training set to classify new observations.

In this study, three different classification case studies were performed: (i) determination of the most important parameters for stab resistance performances of knitted structures, (ii) improving prediction performance on textile sector using a novel ensemble neural network model, and (iii) classifying fashion products using a novel convolutional neural network (CNN).

3.1.1.2 Clustering

Clustering, an unsupervised learning technique, is the process of grouping a set of objects into meaningful clusters in such a way that similar objects are placed within a cluster. Clustering analysis is currently used in many areas such as image processing, pattern recognition, segmentation, machine learning, and information retrieval. The main task of clustering is to compare, measure, and identify the resemblance of objects based on their features by using a similarity measure such as the Manhattan, Euclidean, or Minkowski distance for numerical attributes and Jaccard's distance for categorical values. Clustering algorithms are categorized into five major types, as listed in Table 3.1.

Table 3.1 Categorization of well-known clustering algorithms

Cluster Models	**Clustering Algorithms**
Partitioning Methods	K-Means, C-Means, K-Medoids, CLARANS, ...
Hierarchical Methods	Single/Complete/AverageLink, BIRCH, ROCK, CAMELEON, ...
Density-Based Methods	DBSCAN, DENCLUE, OPTICS, ...
Grid-Based Methods	STING, CLIQUE, WaveCluster, ...
Model-Based Methods	SOM (Self Organizing Maps), COBWEB, EM (Expectation Maximization), ...

In this thesis, a novel approach, named *k*-Linkage, which calculates the distance by considering *k* observations from two clusters separately was proposed. This study also introduced two novel concepts: *k*-min linkage and *k*-max linkage. While *k*-min linkage considers *k* minimum (closest) pairs from points in the first cluster to points in the second cluster, *k*-max linkage takes into account *k* maximum (farthest) pairs of observations.

3.1.1.3 Association Rule Mining

Association Rule Mining (ARM), one of the most important and well researched techniques of data mining, is the extraction of interesting correlations, relationships, frequent patterns or associations, or general structures among sets of items in the transactions.

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of m distinct literals called items, T be transaction that contains a set of items such that $T \subseteq I$, D be a dataset $D = \{t_1, t_2, ..., t_n\}$ that has n transaction records T. An association rule is an implication of the form $X \Rightarrow Y$, where

$X \subset I$, $Y \subset I$ are sets of items called frequent itemsets, and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ can be interpreted as “if itemset X occurs in a transaction, then itemset Y will also likely occur in the same transaction”.

There are two important basic measures for association rules: support and confidence. Usually thresholds of support and confidence are predefined by users to drop those rules that are not particularly interesting or useful. In addition to these measures, additional constraints can also be specified by the users such as time, item, dimensional, or interestingness constraints.

Support of an association rule in the form of $X \Rightarrow Y$ is defined as the percentage of records that contain both X and Y itemsets to the total number of transactions in the dataset D. Support is calculated by the following formula in Equation (3.1).

$$Support(X \Rightarrow Y) = \frac{\text{Number of transactions contain both X and Y}}{\text{Total number of transactions in D}} \tag{3.1}$$

Suppose the support of rule $X \Rightarrow Y$ is 1%. This means that one percent of the transactions contain X and Y items together.

The *Confidence* of an association rule in the form of $X \Rightarrow Y$ is defined as the percentage of the number of records that contain both X and Y itemsets with respect to the total number of transactions that contain X, as in Equation (3.2).

$$Confidence(X \Rightarrow Y) = \frac{\text{Support}(\text{X} \cup \text{Y})}{\text{Support}(\text{X})} \tag{3.2}$$

Suppose the confidence of the association rule $X \Rightarrow Y$ is 85%. This means that 85% of the transactions that contain X also contain Y.

In this study, we uncovered relationships between yarn parameters and fabric properties using an extended FP-Growth algorithm in association rule mining. This study extracted different types of frequent itemsets (closed, maximal, top-*k*, top-*k* closed, top-*k* maximal) that have not been determined in textile sector before. It also proposed two novel concepts, closed frequent item and maximal frequent item, to identify significant items in data.

3.1.2 Machine Learning

Machine learning is a branch of artificial intelligence which provides computer systems to learn from experiences using statistical methods. In this discipline, models that trained by a dataset are generated using machine learning algorithms. The aim of the machine learning is to predict unknown target attributes of the datasets using these generated models. There are two main types of tasks in machine learning: supervised learning and unsupervised learning.

3.1.2.1 Supervised Learning

The predictive model is generated using input variables *x* and their output variable *y* in the supervised learning method. Classification and regression methods are examples of supverised learning.

3.1.2.2 Unsupervised Learning

In unsupervised learning paradigm, there is no prior information about output variables *y* in the datasets. It provides the discovering useful patterns based only on input variables *x*. This techniqe is used in clustering and association rule mining methods.

CHAPTER FOUR
CLASSIFICATION STUDIES IN TEXTILE SECTOR

This chapter presents three different classification case studies that performed in this thesis, (i) determination of the most important parameters for stab resistance performances of knitted structures, (ii) improving prediction performance on textile sector using a novel ensemble neural network model, and (iii) classifying fashion products using a novel convolutional neural network (CNN). In this chapter, implementation of the proposed approaches on the specific textile data were described and the obtained results from the experiments were discussed separately.

4.1 Material and Methods

In this thesis, classification, ensemble learning and deep learning methods were preferred for the three different case studies. In each study, different methods and their algorithms were used. In total, six different classification algorithms (K-Nearest Neighbor, AdaBoost, Naive Bayes, Neural Network, Random Forest, and Decision Tree) were applied on real-world textile data.

4.1.1 Classification Algorithms

Classification is one of the commonly used data mining task which categorizes new samples into predefined classes. Algorithms try to find out relationships between features of instances in training set and predict unknown target attribute value based on a given input. In this study, six classification algorithms: *k*-Nearest Neighbors , AdaBoost, Naive Bayes, Neural Networks, Random Forest and Decision Tree were implemented on the experimental data.

4.1.1.1 K-Nearest Neighbor

K-Nearest Neighbor (k-NN) is a type of instance-based learning to classify an instance with the same class label as a majority vote of its *k* neighbors that have certain class labels. It uses a distance metric (i.e. Euclidean distance) to find the *k* nearest neighbors of a data point. The optimal *k* value is determined empirically.

4.1.1.2 AdaBoost

AdaBoost (Adaptive Boosting) is a boosting based algorithm invented by Yoav Freund and Robert Schapire. It iteratively trains classifiers, each time reweighting the samples in the dataset to focus the next classifier on misclassified ones. In this method, all predictions obtained from each learner are combined using a weighted voting mechanism.

4.1.1.3 Naive Bayes

Naive Bayes is a well-known statistical classifier based on applying Bayes' theorem with naive independence assumptions between every pair of attributes to determine input samples’ classes by calculating unknown conditional probabilities.

4.1.1.4 Neural Network

Neural Network is a supervised learning technique which consists of interconnected multiple layers of nodes in a weighted directed graph that takes input data and transforms it into proper outputs.

4.1.1.5 Random Forest

Random Forest is a bagging based ensemble learning algorithm, introduced by Leo Breiman and Adele Cutler. It builds more than one decision trees using different subsets of the data and evaluating different subsets of features at each node. After training process, it classifies a new sample by applying a voting mechanism over all the trees in the forest.

4.1.1.6 Decision Tree

Decision Tree learning is a supervised learning classification method which uses a decision tree that is grown using depth- first strategy to predict input samples’ unknown class attribute values depending on input variables. The tree structure involves nodes to represent features, branches to represent values of features, and leaves for class labels. There are various decision tree algorithms to predict unknown values of instances, containing C4.5, C5, ID3, CART and CHAID. In this study, Quinlan's C4.5 algorithm was chosen for classification process (N. Bhargava, Sharma, R. Bhargava & Mathuria, 2013). To construct the tree, the attribute, which is beneficial for learning in training set and which gives high mutual information, should be specified. To determine order of features in the decision tree, information gain, which is based on entropy, is calculated for each feature as defined in Equation (4.1).

$$Gain(S,A) = Entropy(S) - \sum_{v \in A} \frac{|S_V|}{|S|} - Entropy(S_V) \tag{4.1}$$

where S_v represents subset of states S for which attribute A with v value. The entropy measures impurity of a particular feature in data set and is calculated as defined in Equation (4.2).

$$Entropy(S) = \sum_{i=1}^{m} -p_i \log_2 p_i \tag{4.2}$$

where p_i is the possibility of output state value i in the data set which has m instances and S subsets. The attribute which maximizes the information gain is selected as splitting point of the tree. The feature which has the highest information gain value among the others is chosen for the root of the tree. Each attribute has many possible split points value to branch out the tree and the optimal split point value is found by using information gain as defined in Equation (4.1). Information gain is evaluated for each split point between examples from different classes and the best split point is chosen. For example, the dataset shown in Figure 4.1 has nine samples consisting of fabric thickness (mm) and class parameters (*D* - dangerous, *M* - medium and *P* - perfect for stab resistance performances). Entropy values are calculated as defined in Equation 4.1 for each possible split points c_1, c_2, c_3 and c_4 as

0.666, 0.899, 1.267 and 1.387 respectively. Then these entropy values are subtracted from entropy value of whole dataset, so information gain values of possible split points are evaluated. According to information gain results, *c_1* is selected as the best split point which gives maximum information gain value among the other split points and calculated by the average of border values of *c_1* as 9.75 ((4+15.5)/2). This can be interpreted as the stab resistance performance will be dangerous (fatal) when the fabric thickness is less than 9.75 mm for the examined samples.

thickness	1.9	3.3	4	15.5	17.6	21	26.2	30.5	34
class	D	D	D	M	P	P	M	M	P
				c_1	c_2		c_3		c_4

c_1= 9.75
(the best splitting point value)

Figure 4.1 Finding split point of experimental dataset

The main advantages of C4.5 algorithm are that it is easy to understand, reveals good performance on large data sets and in lower training duration. C4.5 classifier shows successful results on many areas such as document classification, medical diagnosis, business and biomedical.

4.1.2 Ensemble Learning

Ensemble learning is the process that uses a set of learners and predicts an output by a voting mechanism over all learners. There are several voting mechanisms such as major class labels for categorical target attributes, and the output of an operation (average, weighted average, median, minimum, maximum) for numeric target attributes.

Many experiments (Che, Liu, Rasheed & Tao, 2011; Yu & Ni, 2014; Svetnik, Wang, Tong, Liaw, Sheridan & Song, 2005; Yu, Wang & Lai, 2008) found that ensemble learners often provide more accurate predictive results than an individual learner. Ensemble learning is generally used for constructing a strong classifier to improve prediction performance, feature selection and error-correcting output codes. In ensemble-based system, firstly multiple learners are trained and then these learners are generally compounded by taking a voting mechanism. Thus, this learning method reduces the risk of an unfortunate selection of a learner which shows poor prediction performance.

In the literature, four different approaches are proposed for constructing an ensemble of learners (Alpaydın, 2014):

- **Using different training sets:** Multiple training subsets are created over the original dataset and an individual learning algorithm is trained on each different training sets. Selecting different subsets from the original dataset can be performed randomly (i.e. bagging) or weighted (i.e. boosting and cascading).
- **Using different features:** The subsets of input features from the training set are selected and given to the learners as input.
- **Combining different learning algorithms:** Different learning algorithms are applied on the same dataset to get different models.
- **Using different parameters:** A single algorithm is used with different parameters such as varying *k* value in *k*-nearest neighbor classifier algorithm.

The ensemble methods can be grouped under four main types: Bagging, Boosting, Stacking and Voting. In the third case study, bagging and boosting methods by choosing C4.5 decision tree algorithm as base learner were applied on Fashion-MNIST dataset. To categorize sample fashion products, the Random Forest algorithm as a bagging based approach and AdaBoost algorithm as a boosting based approach were utilized.

Bagging: Bagging, also known as bootstrap aggregating, is a simple and commonly used ensemble method which creates multiple training sets by selecting samples randomly over the original dataset. Thus, it is provided that each learner in ensemble structure is trained with a different training subset. After training step, the predictions from each model are aggregated using a voting mechanism to obtain a single final output.

Boosting: Boosting is another ensemble technique which aims to convert weak learners to strong ones. In this approach, each sample in the training set is assigned a weight value and the algorithm increases the weights of misclassified samples in each iteration.

4.1.3 Deep Learning

Deep learning also known as deep neural networks is a specialized form of machine learning which uses multi layered neural network structures. The deep learning architecture needs large amounts of data with class labels for training phase of the model and requires high computing power (i.e. high-performance GPUs for parallelism) to reduce training time of the model. In the third case study, a type of convolutional neural networks (CNN) was developed for textile object classification.

CNN: A Convolutional Neural Network (CNN) is a category of deep feed-forward artificial neural networks consisting of hidden layers with learnable weights and biases. Each neuron in the architecture generally receives 2D image pixels as an input, performs a dot product and presents the mapping between input image pixels and their class labels. The structure of CNN comprises of a number of convolutional and subsampling layers followed by one or more fully-connected layers. The input image for the convolutional layer is denoted as m x m x c, where m refers pixels for height and width of the image and c symbolizes the number of channels in the image. The convolutional layer is utilized to extract features from the input image using a filter. Then, the pooling layer provides to reduce size of the image and the parameters of the model.

The main advantages of CNNs are that they sizably reduce the amount of parameters of the network model and create new features from the training sets. Therefore, the CNN paradigm shows a really good performance in areas such as image recognition and classification.

4.2 Case Study 1 - Classification of Stab Resistance Performances of Knitted Structures

Body armors are important equipments for the police, army or any security staff. Developments in the field of protective armor present stab and cut resistant materials as flexible armors which are comfortable, lightweight and invisible. Flexible body armors allow the wearer to move while providing protection against the stab attacks.

Previous studies examined evaluation of different stab and cut resistant materials' performance with the experiments performed by measurement devices using situation analysis and mathematical modeling methods and after these measurements the materials which has the highest performance among the compared were suggested. The search for better performing structures and fabrics continues; even though through previous experimental studies the important parameters have been outlined it will be helpful to be able to predict values for these important parameters so that selection of the raw materials, process types and parameters could be chosen more focused to the intended stab resistance performance. It is possible to have such a good start if important parameters and their values can be estimated and this can be done by using data mining techniques to decide which parameter and its value should be selected for a high stab resistance. While traditional data analysis forms a hypothesis against the data, "Data Mining" discovers the relations by converting raw data into useful patterns or relations and provides a model of the behavior.

In this study, it is aimed to determine important parameters and their critical values for stab resistance performances of knitted structures by using data mining techniques. Stab and cut resistance of body armor materials has been measured by experiments using situation analysis and mathematical modeling methods in all previous studies. To the best of our knowledge, this is the first study that estimates important parameters with their values for stab performances of knitted structures.

The main contributions of this study as follows; Firstly, important parameters of knitted structures for stab performance are determined from experimental data by using decision tree classification algorithm. Second, ensemble learning algorithms which haven't been used before in textiles, are implemented on the data of knitted structures. These algorithms benefit from multiple learning algorithms for classification task and shows better prediction performance for stab resistance of knitted structures. Lastly, applied algorithms are compared with each other in terms of accuracy rates and execution times on three different datasets, and the successful algorithm is suggested at the end of the experimental results. The proposed algorithm predicts the important parameters with their critical values. Thus, distinctly different from previous studies, where stab resistance performances were experimentally evaluated, the presented study focuses on prediction of the parameters with their critical values for stab resistance performances of knitted structures. In this way it will be possible to plan the knitted structure parameters before fabric manufacturing process for the intended stab resistance performance.

4.2.1 Experimental Study

Classification algorithms mentioned in material and methods section (k-NN, AdaBoost, Naive Bayes, Multilayer Perceptron, Random Forest, and C4.5 Decision Tree) were applied on the experimental dataset for three different stab energy levels using Weka open source data mining library. Decision tree algorithm differs greatly from the others in terms of providing the important parameters with their "critical" values.

4.2.2 Dataset Description

The experimental dataset, which is used to derive the pattern in this study, is obtained in a previous study (Alpyildiz, Rochery, Kurbak & Flambard, 2011), where stab resistance performances of plain, plush, doubleface and doublefaceinlay knitted single and multilayer aramid fabrics are examined in comparison for three different stab energy levels. Basic characteristics of the fabric set investigated are given in Table 4.1.

Table 4.1 Parameters of knitted samples

Knit structure	Thickness (mm) (1 Layer)	Mass/area (g/m2) (1 Layer)
Plain	1.94	504
Plush	4	1145
Doubleface	4.39	1565
DoublefaceInlay	4.41	1585

Knit structure, fabric thickness, fabric mass per unit area, number of fabric layers, side (face or back) of the fabric facing the stabbing and stab resistance performances for three energy levels are considered in this study and thus the dataset consists of 800 samples. “Sample” can be a single layer fabric or many fabrics layered; where all of the fabric layers have the same knit structure. The details of data set are given in two separate tables; nominal parameters in Table 4.2 and numerical parameters in Table 4.3. Numerical parameters of the knitted structures and their values were acquired from the results of the experiments realized in a previous study (Alpyildiz et al., 2011).

Table 4.2 Details of numerical attributes

Numerical Parameters	Min value	Max value
Number of layers	1	10
Mass per unit area (g/m2)	493.072	16963.449
Thickness (mm)	1.92	45.7

Table 4.3 Details of nominal attributes

Nominal Parameters	Label	Number of records
Knit structure	Plush	200
	Jersey	200
	DoubleFace	200
	DoubleFaceInlay	200
Front or back face	F	400
	B	400
Four joule perforation	Dangerous	200
	Medium	74
	Perfect	46
Six joule perforation	Dangerous	154
	Medium	90
	Perfect	118
Ten joule perforation	Dangerous	0
	Medium	6
	Perfect	134

Stab resistance of the fabrics is indicated (Alpyildiz et al., 2011) with the trauma values on the backing material and trauma is categorized based on the “perforation value (*p*) ” defining the depth of the knife in the backing material as being fatal when p > 10mm, medium when 10mm ≥ p > 7mm and perfect when p ≤ 7 mm (Croft & Longhurst, 2007). Thus class labels of samples regarding the stab performances were assigned according to perforation values; If the perforation value is less than or equal to 7 mm, it is regarded as “perfect”. Else if the perforation value of the sample is greater than 7 mm and less than or equal to 10 mm, sample shows “medium” perforation performance. Finally, the samples apart from these values (more than 10 mm) are labeled as “dangerous”.

4.2.3 Results

C4.5 decision tree algorithm was implemented on the experimental dataset of knitted structures to predict important parameters with their critical values for stab resistance performances at the following energy levels: 4, 6 and 10 J. To classify a new sample, the tree is traversed by starting from the root node to leaf by following branches appropriate for the attribute (parameter) values of sample. Reached leaf node at the end of the tree traversal will be sample’s class label (dangerous/medium/perfect stab performance). The trees generated for classification of perforation at different energy levels was pruned by selecting minimum number of instances at each leaf as 3. In the trees as the attributes; “thickness” indicates the thickness of the sample (sample can be a single layer fabric or many layers of fabric) in mm, “mass/Area” indicates the weight of the sample per unit area in g/m^2, “numOfLayers” indicate the number of fabric layers in the sample, knitStructure” indicates the knit stucture of the fabric layers of the sample, which can be plain, plush, doubleface or doublefaceinlay.

Figure 4.2 shows the pruned tree that is generated by applying C4.5 algorithm on the experimental data to discover significant parameters with their critical values for perforations at the energy level 4J. The tree (Figure 4.2) indicates that “thickness”, which has the highest information gain value, is the most determinant parameter for input sample to discover its stab resistance performance. Thus the tree starts with thickness parameter. The tree indicates that when the sample's thickness value is less than or equal to 9.75 mm, the sample's perforation value is more than 10 mm and thus stab performance of those samples is "Dangerous" at 4 J. Otherwise, if the "thickness" value is greater than 9.75 mm, the tree continues to branch out. At this point, "massArea" attribute value of sample shall be considered. If the "massArea" value is greater than 5089.03 gr/m^2, sample shows "Perfect" stab performance. Else, "numOfLayers" attribute becomes important to make the decision. For either cases when attribute of "numOfLayers" value is less than/equal to 3 and greater than 3, the tree branches with two different nodes of "knitStructure". If "numOfLayers" value is less than or equal to 3 and "knitStructre" value equals to "Plush", "thickness" and "massArea" values of the sample must be considered. Else if "numOfLayers" value is less than or equal to 3 but "knitStructre" value equals to "Jersey", "DoubleFace" or "DoubleFaceInlay", class labels of the sample will be "Medium", "Dangerous" or "Medium" respectively. In the other "knitStructure" node, "Plush", "DoubleFace" and "DoubleFaceInlay" fabrics shows "Medium" perforation performances. Feature of "massArea" identifies target value of sample which has "Jersey" structure. For perfect stab resistant performance with a perforation less than 7 mm at 4J, the tree (Figure 4.2) indicates that 3 layer plain knitted fabric shall be preferred with a sample mass per unit area less than 4661 g/m^2 and sample thickness more than 9.75 mm.

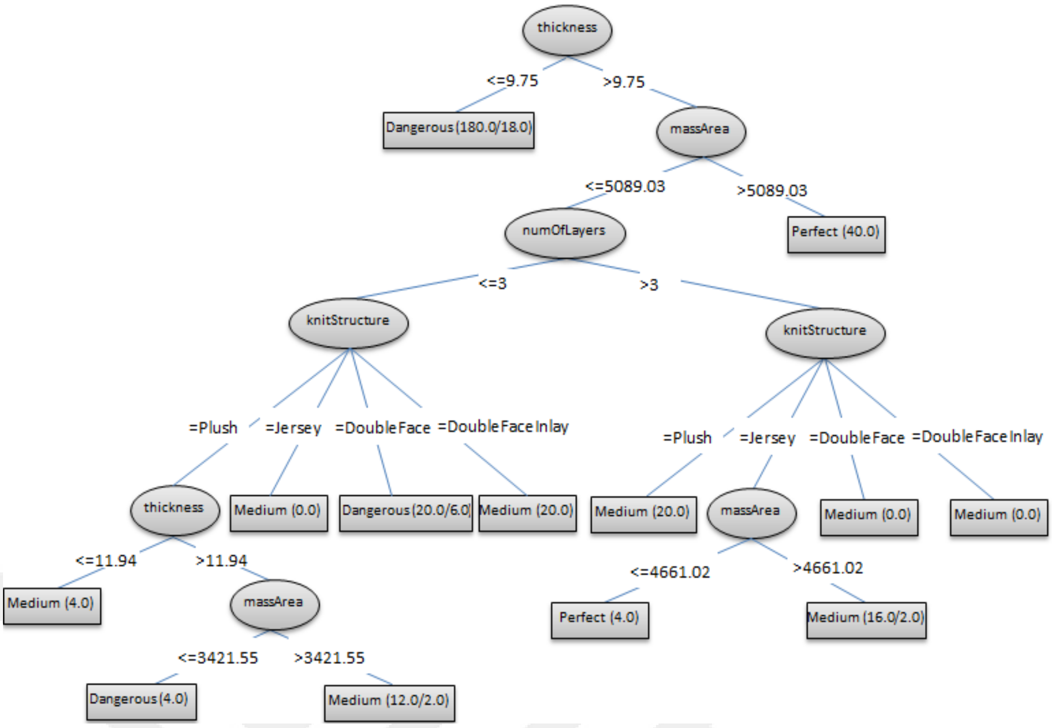


Figure 4.2 Decision tree for the dataset of stab performances at 4 J

C4.5 algorithm was implemented on the knitted structure data likewise as given in Figure 4.3, but this time for the stab performances at 6 J. “massArea” attribute of input sample determines the decision for classification in this tree. Target value of sample, which has “massArea” value that is less than or equal to 3883.39 g/m^2, is regarded to have stab resistance performance as “Dangerous”. If the “massArea” value is greater than 3883.39 g/m^2, attribute of “thickness” should be taken in consideration. Perforation performance of a sample will be “Perfect”, if “thickness” value is greater than 21.5 mm. Otherwise, “knitStructure” node branches according to fabric structure types. This process continues until reaching any leaf node. For perfect stab resistant performance with a perforation less than 7 mm at 6J, the tree (Figure 4.3) indicates that the sample shall have more than 3883 g/m^2 weight per unit area and 21.5 mm thickness regardless of the knit structure and number of fabric layers. If the sample thickness shall be preferred less than 21.5 mm then 3 layers of doubleinlay knitted fabric shall be preferred for perforation values less than 7 mm.

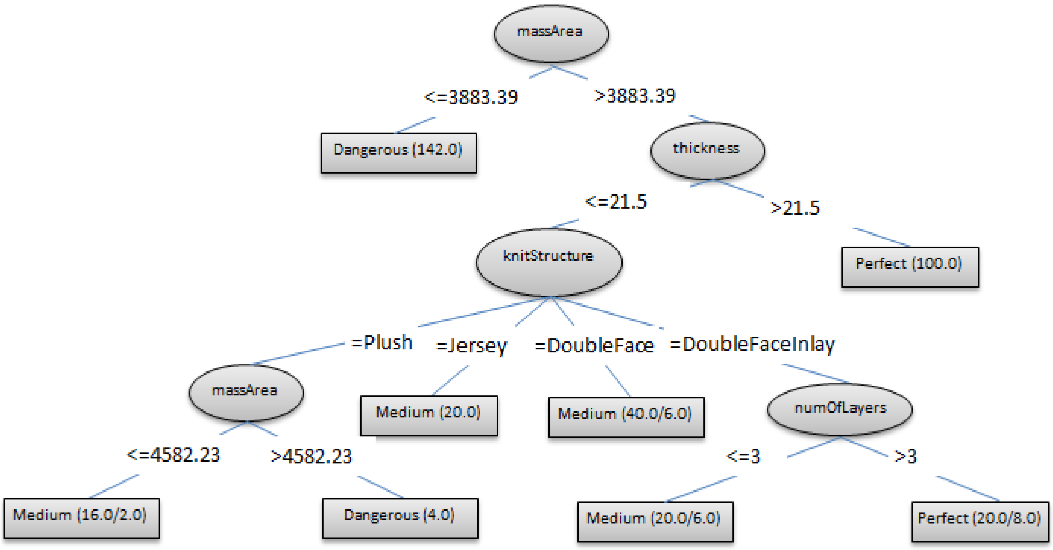


Figure 4.3 Decision tree for the dataset of stab performances at 6 J

Lastly, a new tree was constructed for classifying the stab performances of samples with doubleface and doublefaceinlay fabrics at 10 J using C4.5 algorithm as shown in Figure 4.4. Attribute of "massArea" was indicated as the most determinant feature for input sample. Distinctly from the trees mentioned above, class label of input sample depends on only "massArea" and "thickness" attributes.

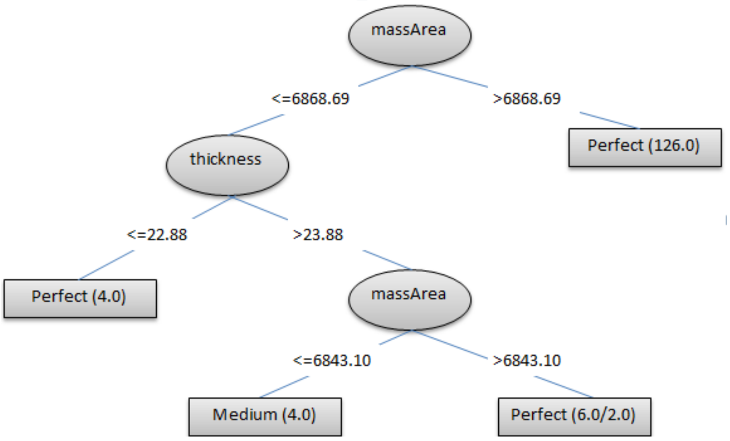


Figure 4.4 Decision tree for the dataset of stab performances under energy level at 10 J

For all of the energy levels, the model, proposed through decision tree algorithm, predicts the determinant parameters and their critical values but the accuracy rates shall be indicated in comparison with all of the other implemented algorithms.

All of the algorithms of the predictive model in this study were compared by using n-fold cross validation technique selecting n input value as 10 and success of the applied classification algorithms on the selected datasets were revealed. This technique divides dataset into 10 parts: 9 of them for training and the rest for test, and accuracy rates were evaluated by repeating it for each subset and taking mean of them as shown in Figure 4.5.

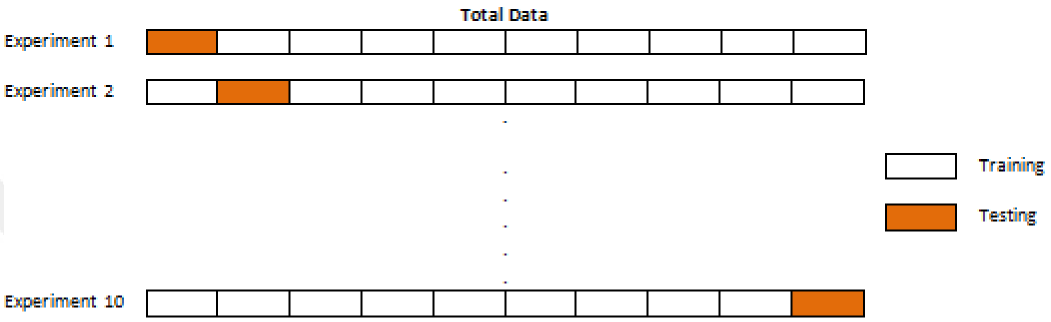


Figure 4.5 Evaluation of accuracy rate

Accuracy rate is defined as the ratio of sum of the correctly classified (true positive (*TP*)) and misclassified positive instances (true negative (*TN*)) to the number of test data including false positive (*FP*) and false negative (*FN*) instances and gives success of the applied algorithm on the selected data as shown in Equation (4.3).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.3}$$

F-measure which is a measurement of algorithm's accuracy on test set can be considered as harmonic mean of precision and recall was shown in Equation (4.4). F-measure value ranges between zero (0) for worst and one (1) for best.

$$F - measure = \frac{2 * precision * recall}{precision + recall} \tag{4.4}$$

Precision is the number of correctly classified positive instances divided by total number of positive elements shown in Equation (4.5).

$$Precision = \frac{TP}{TP + FP} \tag{4.5}$$

Recall is the number of correctly classified positive instances divided by sum of the correctly classified positive and misclassified false instances shown in Equation (4.6).

$$Recall = \frac{TP}{TP + FN} \tag{4.6}$$

Testing was performed on a personal computer with features of Intel Core 2 Duo CPU, 6 GB RAM. Tables 4.4, 4.5 and 4.6 present the precision, recall, f-measure values and accuracy rates of the classification algorithms, *k*-NN, AdaBoost, Naive Bayes, Multilayer Perceptron, Random Forest and C4.5 (decision tree), on the experimental datasets of stab performances of knitted structures at the energy levels: 4, 6 and 10 J. According to the results, C4.5 algorithm has the highest classification accuracy on the dataset of stab performances at 4 J. The Random Forest algorithm is the most successful algorithm for the dataset of perforation performances at 6 J energy level when compared to the other algorithms. For the last dataset which contains stab performances of knitted structures under energy level at 10 J, the AdaBoost and Random Forest algorithms show the best accuracy performance and reach the rate of 97.14%. Table 4.5 and 4.6 indicate that ensemble learning algorithms show great success on the datasets of perforation performances at 6 and 10 J because it uses multiple classifier models to get better accuracy rates when comparing with the other classification algorithms implemented in this study.

If the experimental results are considered in general, C4.5 is the most appropriate algorithm for the datasets of each energy level because it is the only algorithm among the others, which discovers significant parameters with their values that effects stab resistance of knitted structures at each energy level. While C4.5

algorithm shows the highest accuracy rate for the dataset of perforation performances at 4 J, it also gives successful rates for the other two energy levels.

Table 4.4 Comparison of the algorithms for the dataset of perforation performances at 4 J

Algorithms	**Precision**	**Recall**	**F-measure**	**Accuracy Rate (%)**
k-NN	0.79	0.77	0.74	76.88
AdaBoost	0.94	0.92	0.92	91.88
Naive Bayes	0.90	0.89	0.90	89.38
Multilayer Perceptron	0.94	0.94	0.94	93.75
Random Forest	0.94	0.94	0.94	94.38
C4.5	0.95	0.95	0.95	**94.69**

Table 4.5 Comparison of the algorithms for the dataset of perforation performances at 6 J

Algorithms	**Precision**	**Recall**	**F-measure**	**Accuracy Rate (%)**
k-NN	0.88	0.88	0.88	87.57
AdaBoost	0.91	0.90	0.90	89.78
Naive Bayes	0.90	0.87	0.88	87.02
Multilayer Perceptron	0.91	0.91	0.91	91.16
Random Forest	0.92	0.91	0.92	**91.44**
C4.5	0.91	0.90	0.90	90.33

Table 4.6 Comparison of the algorithms for the dataset of perforation performances at 10 J

Algorithms	**Precision**	**Recall**	**F-measure**	**Accuracy Rate (%)**
k-NN	0.93	0.92	0.92	92.14
AdaBoost	0.97	0.97	0.97	**97.14**
Naive Bayes	0.97	0.91	0.93	91.43
Multilayer Perceptron	0.92	0.96	0.94	95.71
Random Forest	0.97	0.97	0.97	**97.14**
C4.5	0.96	0.96	0.96	96.43

To show estimation success of the algorithms on selected dataset, confusion matrices can be used as another method in addition to accuracy rates. Confusion matrix is a matrix representation of closeness of the predicted values to the actual values which classified by a selected algorithm.

In Table 4.7, 4.8 and 4.9, confusion matrices of C4.5 algorithm which present number of instances as predicted and actual values on the datasets with three different energy levels, 4J, 6J, and 10J were shown. For example, in the confusion matrix given in Table 4.7 it is easily seen that 197 of 200 samples were correctly labeled as “Dangerous” and the rest three of them were misclassified as “Medium”. For the other class values, while 64 and 42 samples of the dataset were correctly classified as "Medium" and "Perfect" respectively. In this way, amount of experimental data can be shown with these confusion matrices.

Table 4.7 Confusion matrix of C4.5 algorithm for the dataset of perforation performances at 4J

4J		**Predicted Class**		
		Dangerous	Medium	Perfect
Actual Values	Dangerous	197	3	0
	Medium	8	64	2
	Perfect	0	4	42

Table 4.8 Confusion matrix of C4.5 algorithm for the dataset of perforation performances at 6J

6J		**Predicted Class**		
		Dangerous	Medium	Perfect
Actual Values	Dangerous	144	10	0
	Medium	3	79	8
	Perfect	0	14	104

Table 4.9 Confusion matrix of C4.5 algorithm for the dataset of perforation performances at 10J

10J		**Predicted Class**		
		Dangerous	Medium	Perfect
Actual Values	Dangerous	0	0	0
	Medium	0	3	3
	Perfect	0	2	132

Execution times of applied algorithms are measured when running on datasets under three different energy levels, 4, 6, and 10 J and compared as shown in Table 4.10. According to results, it is clearly understood that even though k-NN algorithm has the best time performance and multilayer perceptron shows the highest execution time for the datasets with all energy levels, C4.5 algorithm also shows good performance in terms of execution times in addition to successful accuracy rates.

Table 4.10 Execution times of the algorithms for the datasets of stab performances at 4, 6 and 10J

Algorithms	**Execution Times (sec.)**		
	4 Perforation	**6 Perforation**	**10 Perforation**
k-NN	0.034	0.033	0.033
AdaBoost	0.136	0.140	0.103
Naive Bayes	0.040	0.038	0.036
Multilayer Perceptron	0.464	0.613	0.340
Random Forest	0.245	0.329	0.198
C4.5	0.066	0.072	0.048

4.3 Case Study 2 - Improving Prediction Performance using Ensemble Neural Networks

Neural network technique has been recently preferred in textile sector for the prediction task because the traditional mathematical and statistical methods can be inadequate to derive complex relations within textile datasets. A neural network is an interconnected group of nodes that are constructed to identify underlying relationships in a set of data for classification and prediction. A multilayer perceptron (MLP) is the most utilized model in neural network connecting multiple layers in a directed graph. A MLP has many parameters to be concerned such as the number of hidden layers, learning rate, momentum coefficient, the type of activation function, hidden layer size, stopping criteria (the number of epochs or target error rate), and learning algorithm. In this study, different MLP models with different setting parameters have been used to construct an ensemble of neural networks (Yıldırım, Birant & Alpyıldız, 2017).

Meanwhile ensemble learning has become a popular machine learning approach in recent years due to the high prediction performance it provides. Ensemble learning is a type of machine learning that merges multiple base learning models to make final prediction (Alpaydın, 2014). The base learners can be any classification and prediction algorithms such as neural network (Yu, Wang & Lai, 2008), Bayesian networks (Alessandro, Corani, Mauá & Gabaglio, 2013), regression (Budka & Gabrys, 2010), and decision tree (Dietterich, 2010; Che, Liu, Rasheed & Tao, 2011). It has been observed that ensemble learners show a great success in many fields (Dietterich, 2010). Considering this motivation, this study focuses on the application of ensemble neural networks on real-world textile datasets.

The novelty and main contributions of this study are as follows: (i) it proposes an ensemble learner which consists of combination of multilayer perceptron models with three different initialization parameters (the number of hidden layers, learning rate and momentum coefficient) to improve prediction performance, (ii) it is the first study that the proposed ensemble learner has been implemented in textile sector, and

(iii) it compares ensemble neural networks with a single neural network in terms of correlation coefficient and relative absolute error measures on different textile datasets.

4.3.1 Experimental Study

This study proposes an ensemble learning approach that combines neural networks with different parameter values. The MLP models in our study were constructed with three different parameters: the number of hidden layers (*hl*), learning rate (*lr*) and momentum coefficient (*mc*). This specific approach was proposed because of two reasons. First, the target attribute in the dataset that will be predicted has numerical values, instead of categorical values; so a neural network algorithm can be used for numerical prediction, in contrast to categorical classification algorithms like decision tree. Second, the number of records in the dataset is not sufficient for selecting multiple training subsets or choosing feature subsets.

In the proposed approach, each of three parameters (*hl*, *lr*, *mc*) has three different values. Only one parameter is changed at each time, whereas other parameters are kept constant. So, the permutation of three parameters with three values forms 27 different MLP models. While *learning rate* parameter is used for fine-tuning the change of bias values and weight size of the algorithm, *momentum coefficient* stabilizes the weight change and provides to reach global minima. With learning rate η and momentum coefficient α, the weight update from unit i to unit j by Δw_{ji} at t iteration becomes

$$\Delta w_{ji}(t) = \eta \delta_j x_{ji} + \alpha \Delta w_{ji}(t-1) \tag{4.7}$$

where δ_j is the error gradient and x_{ji} refers the input.

Choosing too low learning rate causes the network learn very slowly and so it takes a long time to get good prediction performance. When the learning rate is too high, the objective function and weights diverge. Likewise, too low momentum coefficient slows down the network model and prevents to reach global minima from local minima credibly. Conversely, too high momentum coefficient may give rise to overshooting the minimum.

The general structure of the proposed ensemble model is presented in Figure 4.6. In this model, input *x* is given to the 27 different neural networks and the outputs y_i that are obtained from each model are averaged to find final prediction output $\hat{O}$. A unipolar sigmoid activation function is used in each node. Table 4.11 shows the parameter values of 27 different neural networks. The number of hidden layers ranges from 2 to 6 by increment 2. Learning rate and momentum coefficient parameters take values 0.1, 0.2 and 0.3 at each time.

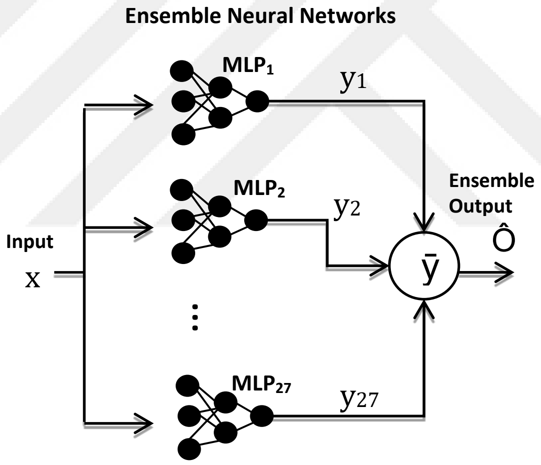


Figure 4.6 General structure of the proposed model

Table 4.11 Parameter-dependent multilayer perceptron model combinations

Number of hidden layers (hl)	Learning rate (lr)	Momentum coefficient (mc)	Neural Network Model
2	0.1	0.1	MLP_1
2	0.1	0.2	MLP_2
2	0.1	0.3	MLP_3
2	0.2	0.1	MLP_4
2	0.2	0.2	MLP_5
2	0.2	0.3	MLP_6
2	0.3	0.1	MLP_7
2	0.3	0.2	MLP_8
2	0.3	0.3	MLP_9

Table 4.12 continues

4	0.1	0.1	MLP_{10}
4	0.1	0.2	MLP_{11}
4	0.1	0.3	MLP_{12}
4	0.2	0.1	MLP_{13}
4	0.2	0.2	MLP_{14}
4	0.2	0.3	MLP_{15}
4	0.3	0.1	MLP_{16}
4	0.3	0.2	MLP_{17}
4	0.3	0.3	MLP_{18}
6	0.1	0.1	MLP_{19}
6	0.1	0.2	MLP_{20}
6	0.1	0.3	MLP_{21}
6	0.2	0.1	MLP_{22}
6	0.2	0.2	MLP_{23}
6	0.2	0.3	MLP_{24}
6	0.3	0.1	MLP_{25}
6	0.3	0.2	MLP_{26}
6	0.3	0.3	MLP_{27}

The outcomes obtained from each multilayer perceptron model are averaged as defined in Equation (4.8) and the result is selected as a final prediction.

$$\hat{O} = \frac{1}{N}\sum_{i=1}^{N} MLP_i(x) = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{4.8}$$

where *x* is input vector, y_i is the output of each model, $\hat{O}$ is the ensemble output (final output) and *N* is the number of neural networks.

The algorithm that used in this research is given below. The algorithm accepts two inputs: training dataset *D* and the number of neural networks *N*. It finds the average of outputs obtained from each MLP model as a final result.

Algorithm for ensemble neural networks

Input: *D*: training dataset, *N*: the number of neural networks

Output: $\hat{O}$: ensemble output

```
Step 1. Get input training $t$ samples ($x_{11}$, $x_{12}$, ..., $y_{1m}$), ..., ($x_{t1}$, $x_{t2}$, ..., $y_{tm}$) with categorical / numeric inputs $x$ and numeric outputs $y$
Step 2. Loop while $i <= N$ ensemble members
        a. Initialize parameters: learning rate $lr$, momentum coefficient $mc$, the number of hidden layers $hl$
        b. Train $MLP_i$
        c. Get hypothesis $ht$ from $MLP_i : X \rightarrow Y$
        d. Set $total = total + Y$
        e. Set $i = i + 1$
        End of loop
Step 3. Calculate final ensemble output $\hat{O} = (total / N)$
```

The proposed ensemble neural networks (NNs) model was tested on ten different real-world textile datasets. The application was developed by using Weka open source data mining library. Ensemble NNs model was compared with individual NN in terms of correlation coefficient and relative absolute error measures.

4.3.2 Dataset Description

In this experimental study, ten different datasets that are available for public use were selected to demonstrate the capabilities of the proposed ensemble NNs. The datasets were obtained from the data archive in Statistics Department of University of Florida. Basic characteristics of the investigated textile datasets are given in Table 4.12. These datasets are on different types of textile end-products (fiber, yarn, fabric or garment), clothing categories (towels, jeans, thermal clothing), fiber types (cotton, silk, wool), fiber properties (i.e. length), spinning methods (ring, mule), yarn parameters (i.e. count), fabric structural parameters (warp and weft density, mass per unit area), fabric quality parameters (i.e. hairiness, color difference, shrinkage), processes (i.e. dye, dry), and treatments.

Table 4.13 Description of textile datasets

Dataset ID	Dataset Name	Attributes | Values
Dataset 1	Color Among 3 Fabrics exposed to 4 soils, washed at 2 temperatures with 2 surfactants	-Soil type | { 1=tea, 2=coffee, 3=wine, 4=charcoal } -Fiber type | { 1=cotton, 2=silk, 3=wool } -Temperature | { 1=40C, 2=60C } -Surfactant | {1=anionic, 2=non-ionic } -Color difference | numeric
Dataset 2	Cotton Output by Yarn Count for Mule and Ring Spinning in New England Early 1900s	-Spinning type | { 1=mule, 2=ring } -Yarn count | numeric -Output (lbs/week) | numeric
Dataset 3	Effects of Impact, Specimen Layers, and Quilting on Energy Absorption of Body Armour	-Impact | { 1=slow, 2=fast, 3=edge } -Specimen layers | { 1, 2, 3,5 } -Quilting | { 1=none, 2=square, 3=diamond } -Energy absorbed | numeric
Dataset 4	Fabric Treatment and Cycle Effects on Wool Shrinkage	-Run | numeric -Treatment | { 1=untrt, 2=ac(15s), 3=ac(4m), 4=ac(15m) } -Number of revolutions | numeric -Top shrinkage | numeric
Dataset 5	Hairiness of Yarns of Various Twist Levels, Test Speeds, and Bobbins	-Twist level | { 1=373tpm, 2=563, 3=665 } -Test speed | { 1=25m/min, 2=100, 3=400 } -Bobbin number | numeric -Hairiness index (x100) | numeric
Dataset 6	Fraction of Wool Dye in Bath by Treatment and Observer	-Treatment | { 1=Ether extracted, 2=Ether and alcohol-extracted, 3=Alcoholic potash (15 seconds), 4=Alcoholic potash (4 min), 5=Alcoholic potash (15 min) } -Observer | numeric -Replicate | numeric -Proportion of dye in bath | numeric
Dataset 7	Energy Effectiveness of 4 Dryer Types on 3 Clothing Categories	-Clothing category | { 1=towels, 2=jeans, 3=thermal clothing } -Dryer type | {1=Electric dryer, 2=Bi-directional electric dryer, 3=Town gas-fired dryer, 4=LPG-fired dryer } -Energy effectiveness | numeric
Dataset 8	Air Permeability of Woven Fabrics as Function of Warp, Weft, Mass per Unit Area	-Warp density | numeric -Weft density | numeric -Mass per unit area | numeric -Average air permeability | numeric

Table 4.14 continues

Dataset 9	Comparison of 2 Fabrics and 7 Levels of Layers for Ballistic Tests on Bullet-proof Fabric	-Ply number | numeric -Material | { 1=twaron 2=k-flex } -Condition | { 1=dry 2=wet } -Bullet velocity mean | numeric -Bullet velocity SD | numeric -Trauma Depth Mean | numeric
Dataset 10	Fabric Treatment and pH Effects on Wool Shrinkage	-Run | numeric -Treatment | { 1=untrt, 2=ether, 3=ether/alcohol } -pH level | { 2,4,6 } -Top Shrinkage | numeric

4.3.3 Results

First, each neural network was trained with different initial values for network parameters using a training set and then target values of each sample were predicted and compared with original values in the test set. In order to evaluate the model, 90% of entire dataset was selected for training phase and the rest of them were used for testing step. To show success of ensemble learning model proposed in this study, the results obtained from ensemble NNs were compared with individual NN in terms of correlation coefficient and relative absolute error measures.

- **Correlation coefficient:** Correlation coefficient (r) is a measure which gives a degree ranging from -1.0 to 1.0 to indicate the relationship between the predicted outputs and actual outputs. While correlation value -1.0 shows a strong negative correlation, correlation value of 1.0 means that there is a strong positive relationship between two outputs. The correlation between the predicted and actual values is calculated as defined in Equation (4.9).

$$r = \frac{\frac{\sum_i (p_i - \bar{p})(a_i - \bar{a})}{n-1}}{\sqrt{\frac{\sum_i (p_i - \bar{p})^2}{n-1}\frac{\sum_i (a_i - \bar{a})^2}{n-1}}} \tag{4.9}$$

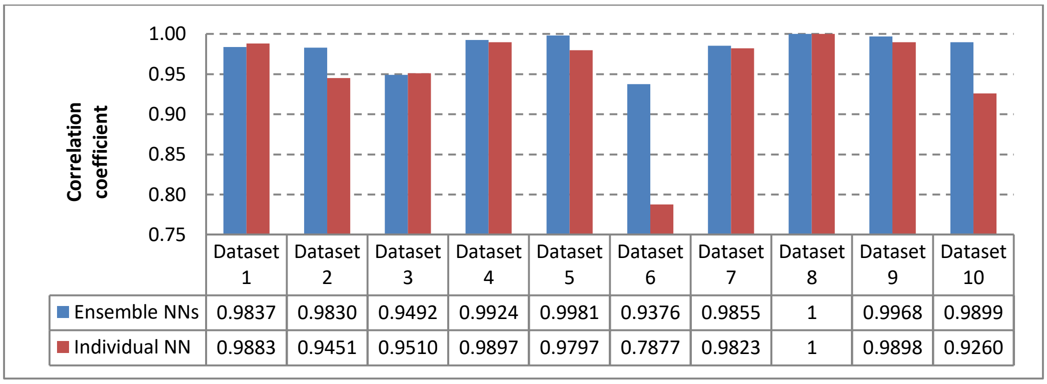
where p refers to predicted target value, a is actual target value and n is the number of samples.

- **Relative absolute error:** Relative absolute error (RAE) gives the ratio of magnitude of difference between the actual and predicted values to the mean value of the measured quantity as shown in Equation (4.10).

$$RAE = \frac{\sum_i |p_i - a_i|}{\sum_i |\bar{a} - a_i|} \tag{4.10}$$

where p refers to predicted target value, a is actual target value.

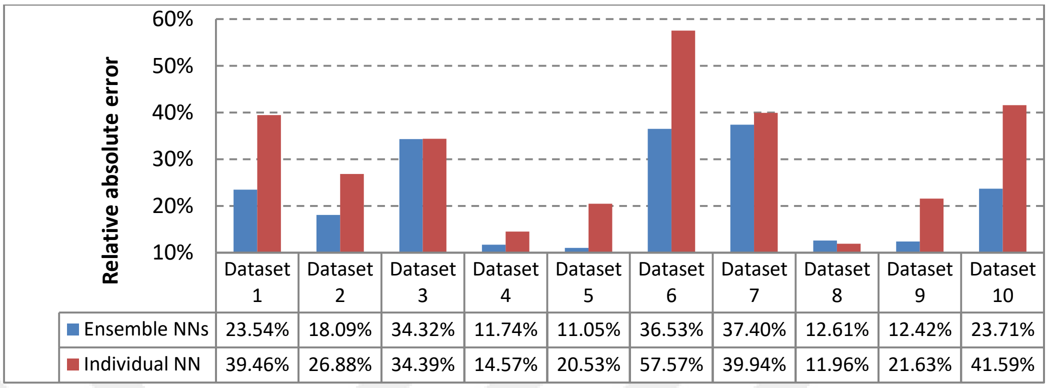
Correlation coefficient and relative absolute error were computed on ten textile datasets for ensemble NNs and individual NN with default settings to compare them. Figure 4.7 presents the comparative results in terms of correlation coefficient measure. The results indicate that the proposed ensemble model gives more accurate prediction results than the individual model in seven of ten datasets. In some cases, the difference is small; however, sometimes ensemble NNs has a significantly higher than individual NN.



	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8	Dataset 9	Dataset 10
🟦 Ensemble NNs	0.9837	0.9830	0.9492	0.9924	0.9981	0.9376	0.9855	1	0.9968	0.9899
🟥 Individual NN	0.9883	0.9451	0.9510	0.9897	0.9797	0.7877	0.9823	1	0.9898	0.9260

Figure 4.7 Experimental results based on coefficient correlation

The graph given in the Figure 4.8 illustrates the relative absolute errors obtained by ensemble versus individual NN models. There is an inverse ratio between the success of the model and the RAE value. The model with lower error rate means that it is more successful than the other. From this point of view, it is clearly seen that the proposed ensemble NNs model has lower RAE values than the individual one on almost all datasets.



	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8	Dataset 9	Dataset 10
🟦 Ensemble NNs	23.54%	18.09%	34.32%	11.74%	11.05%	36.53%	37.40%	12.61%	12.42%	23.71%
🟥 Individual NN	39.46%	26.88%	34.39%	14.57%	20.53%	57.57%	39.94%	11.96%	21.63%	41.59%

Figure 4.8 Experimental results based on relative absolute error

When the experimental results are considered in general, it is possible to say that the proposed ensemble model improves prediction performance in textile sector.

4.4 Case Study 3 - Comparison of Deep Learning and Ensemble Learning for Textile Object Classification

Object classification is one of the essential tasks in machine learning. The goal of object classification is to assign a class label to each unlabeled object using the existing labeled objects. Object classification is a difficult task in many domains, especially in the presence of large numbers of classes, due to the high dimensionality of data and the large variations between images belonging to the same class. This study focuses on object classification in textile domain.

Machine learning has proven to be powerful for object classification. It presents several classification algorithms which use the features of objects to specify the class of each object. Nowadays, ensemble learning has become one of the most active fields in machine learning and commenced to be used in many areas to produce accurate results. Ensemble learning is a machine learning technique which merges a set of individual learning models and then aggregates them to obtain single final prediction by a voting mechanism over all learners. These individual learning models can be any classification algorithms like decision tree, Naive Bayes, neural network, *k*-nearest neighbor and regression. The ensemble methods can be grouped under four main types: bagging, boosting, stacking and voting. In the literature, it is indicated that ensemble methods usually show better classification results than an individual model would.

In the last decide, deep learning approach has showed remarkable results on big data problems by running in reasonable time scale. Deep learning is a new field of machine learning research which provides computers to learn from experiences inspired by how the human brain works. The deep learning models include multi-layered neural network architecture that is trained by large-scale datasets. This paradigm achieves state-of-the-art performances in a considerable number of domains such as image recognition, computer vision, speech recognition, robotics and natural language processing.

The novelty and main contributions of this study are as follows: (i) it provides a brief survey of deep learning and ensemble learning, which has been revealed to improve the performance of learning models for textile object classification, (ii) it proposes a novel advanced network architecture which was designed as an example of a "deep neural network" including convolutional, max pooling, and fully connected layers, and (iii) it presents experimental studies conducted on a new benchmark dataset named Fashion-MNIST to demonstrate that the proposed advanced network architecture gives better classification results than ensemble learning methods in terms of accuracy.

4.4.1 Experimental Study

In this study, an advanced network architecture was designed as an example of a “deep neural network”, that it has several layers, including convolutional, max pooling, and fully connected layers. As given in Figure 4.9, the network consists of two convolutional layers, where each layer is followed by one max-pooling layer, and finally one fully connected layer.

```
input Picture [28, 28];
// first convolutional layer parameters
hidden C1 [5, 28, 28] from Picture convolve {
  InputShape  = [28, 28];
  KernelShape = [5, 5];
  Stride  = [1, 1];
  Padding = [T, T];
  MapCount = 5;
}
// first pooling layer parameters
hidden P1 [5, 14, 14] from C1 max pool {
  InputShape  = [5, 28, 28];
  KernelShape = [1, 2, 2];
  Stride  = [1, 2, 2];
}
// second convolutional layer parameters
hidden C2 [50, 14, 14] from P1 convolve {
  InputShape  = [5, 14, 14];
  KernelShape = [1, 5, 5];
  Stride  = [1, 1, 1];
  Sharing = [F, T, T];
  Padding = [F, T, T];
  MapCount = 10;
}
// second pooling layer parameters
hidden P2 [50, 7, 7]  from C2 max pool {
  InputShape  = [50, 14, 14];
  KernelShape = [1,  2, 2];
  Stride  = [1,  2, 2];
}
hidden H3 [100] from P2 all;
output Result [10] softmax from H3 all;
```

Figure 4.9 The design parameters of proposed advanced convolutional neural network architecture

In the experimental studies, we used Fashion-MNIST dataset (Xiao, Rasul & Vollgraf, 2017) that introduced in 2017 with the aim to provide a good benchmark dataset for deep learning models. Individual C4.5 decision tree algorithm, ensemble learning methods (Bagging, Random Forest and AdaBoost) by choosing C4.5 as base learner and the proposed CNN were compared on Fashion-MNIST dataset in terms of accuracy. Azure Machine Learning Studio was used to implement the deep neural network application. Ensemble learning applications were developed by using Weka open source data mining library (Frank, Hall & Witten, 2016).

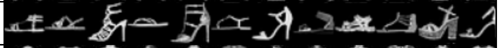
4.4.2 Dataset Description

Fashion-MNIST dataset (Xiao et al., 2017) consists of 70,000 examples where each sample is a 28x28 gray-scale image, associated with a label that belongs to 10 fashion product classes: t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. Those products come from different gender groups: men, women, kids and neutral. The number of examples for each class is equal (7,000 images per category), thus it is a balanced dataset. The training set has 60,000 front-look thumbnail images, where the test set has 10,000 images, both of which have been pre-split for bench-marking purposes. Each image in the training and test datasets is in the form of a vector with 28*28 = 784 elements, where each element corresponds to one pixel in the image. Each pixel-value is an integer between 0 and 255 that indicates the lightness or darkness of that pixel, where higher numbers correspond to darker colors. In other words, 0 is a pure white pixel and 255 is a pure black pixel. The first column of the dataset consists of the class labels (0-9) and the rest of the columns contain the pixel-values of the associated image. Table 4.13 gives a summary about the Fashion-MNIST dataset and Table 4.14 presents all class labels in the range [0, 9] with example images for each class.

Table 4.15 Dataset summary

Dataset	**Training**	**Testing**	**Classes**	**Features**
Fashion-MNIST	60,000	10,000	10	474

Table 4.16 Image classes and examples from Fashion-MNIST dataset

Class	**Type of Product**	**Example Images**
0	T-shirt/top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandal	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boot	

4.4.3 Results

The first experiment in this study was performed by applying three ensemble learning techniques (Bagging, Random Forest and AdaBoost) on Fashion-MNIST dataset to classify fashion products into their types. In Bagging and AdaBoost technique, C4.5 decision tree algorithm was preferred as base learner because of its popularity, low computational complexity and high classification performance. As a result of the experiment, the accuracy rates of each ensemble learning techniques were evaluated by checking whether the predicted and actual labels match in the test set.

In the second experiment, the proposed convolutional neural network, including two convolutional layers, one max pooling layer, and one fully connected layer, was applied on the same dataset. The CNN architecture in our study was constructed with modifying the hyper parameters to reach the most successful classification result. The input parameter values such as the number of convolution layers, filter size, padding mode, strides and the number of neurons in the fully-connected layer are given in Figure 4.9. The accuracy rate of the proposed method was calculated as the first experiment.

The graph given in Figure 4.10 shows the comparative results of implemented techniques on the dataset in terms of the accuracy rates. The results indicate that the ensemble based models (random forest 87.92%, bagging 89.09%, AdaBoost 89.36%) and the convolutional neural network model (90.56%) generally provide higher accuracy values than individual classification algorithm (83.05%). When the experimental results are considered in general, it is possible to say that the proposed convolutional neural network has the best accuracy score.

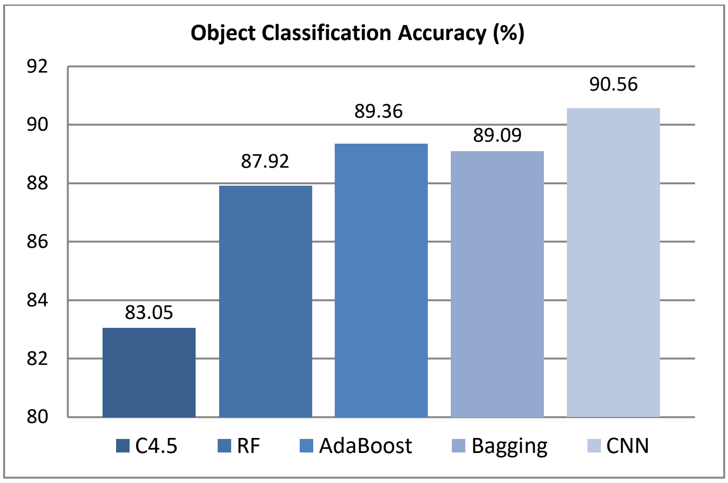


Figure 4.10 Comparison of ensemble learning and convolutional neural network techniques on Fashion-MNIST dataset

Table 4.15 presents the confusion matrix of the convolutional neural network model, which is a matrix representation of closeness of the predicted classes to the actual classes. Overall the results are very good for bag, trouser, and ankle boot that have accuracies 98.3%, 98.1% and 97.5%, respectively. All these categories have rather less false positives, giving a higher value along the diagonal of the confusion matrix. Furthermore, while the model achieved 97.2% accuracy for the sandal class, it has a score of 95.8% in the dress category.

Table 4.17 Confusion matrix that shows the ratio of correct and incorrect predictions made by CNN model for each class

		Predicted Class (%)									
		T-shirt/top	**Trouser**	**Pullover**	**Dress**	**Coat**	**Sandal**	**Shirt**	**Sneaker**	**Bag**	**Ankle boot**
Actual Class (%)	**T-shirt/top**	**82.7**	0.1	1.8	2.5	0.4	0.2	11.7		0.6	
	Trouser	0.1	**98.1**	0.3	0.9	0.2	0.2		0.1	0.1	
	Pullover	1.9	0.2	**81.2**	1.3	9.3		6.0		0.1	
	Dress	1.3	0.6	0.1	**95.8**	1.1		1.0		0.1	
	Coat			3.1	3.8	**85.9**		7.1		0.1	
	Sandal				0.2		**97.2**		1.2	0.1	1.3
	Shirt	9.8	0.1	4.5	4.6	6.2		**74.4**		0.4	
	Sneaker						1.5		**94.5**		4.0
	Bag	0.1	0.1	0.2	0.1	0.3	0.2	0.3	0.2	**98.3**	0.2
	Ankle boot						0.3		2.1	0.1	**97.5**

However, shirt predictions (74.4%) occasionally were misclassified as a T-shirt/top. A similar issue occurred with the pullover class, which only had 81.2%

correct predictions with many being misclassified as a coat. The classifier seems to have trouble distinguishing between t-shirt/top and shirt, as well as pullover and coat, which are not surprising because these clothing items look quite similar. So, object classification is a difficult task, when near similarity among various types of textile products exist.

Figure 4.11 illustrates the changes in accuracy rates which were provided by the convolutional neural network model depend on number of iterations. The highest accuracy rate that is 90.56 was obtained when the number of iterations was 10. The classification accuracy stabilizes after approximately 10 epochs. So, the classification accuracy stabilizes in just a few iterations in this study.

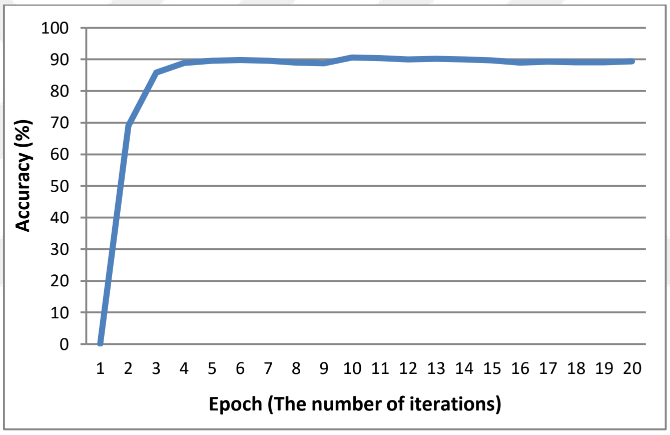


Figure 4.11 Test accuracy vs epoch (the number of iterations)

CHAPTER FIVE
A CLUSTERING STUDY: PROPOSED K-LINKAGE METHOD

Clustering, which is one of the data mining techniques, combines a set of objects into clusters based on a certain similarity measure. Clustering algorithms can be basically grouped under three categories: partitioning, hierarchical and density-based methods. *Partitioning clustering* is an iterative method which divides a dataset into disjoint clusters. *Hierarchical clustering* is characterized by the development of a hierarchy by either repeatedly merging small clusters into a larger one (agglomerative strategy) or splitting a larger cluster into smaller ones (divisive strategy). *Density-based clustering* is to discover clusters of arbitrary shape based on the density of the region surrounding the data point. This study focuses on hierarchical clustering problems on textile data.

Hierarchical clustering has been commonly used in many applications by applying either divisive or agglomerative method. *Divisive hierarchical clustering* is a top down approach which starts with a single cluster and splits the cluster into two dissimilar clusters recursively until specified condition is satisfied. *Agglomerative hierarchical clustering* is a bottom up approach and starts with clusters containing single observations and continuously merges them based on a similarity strategy until all clusters are merged into one big cluster, or a stopping criteria is met. The traditional strategies of computing cluster distances are *single*, *complete*, *average*, and *centroid linkages*. However, these strategies can remain incapable of merging correct clusters, because small perturbations in the data can lead to large changes in hierarchical clustering assignments. There is no guarantee that single linkage or complete linkage will individually give the optimal clusters, because they consider only a single distance between two clusters. The calculation of distances between clusters based on a single pair may not always reflect the true underlying relationship between clusters and so it returns clusters that are only locally optimal. The main aim of this study is to overcome this drawback by proposing a new approach for textile domain.

This study proposes a novel linkage method for hierarchical clustering, named *k*-Linkage (Yildirim & Birant, 2017). The proposed *k*-Linkage method evaluates the distance between two clusters by calculating the average distance between *k* pairs of observations, one in each cluster. This study also introduces two novel concepts: *k*-min linkage and *k*-max linkage. While *k*-min linkage considers *k* minimum (closest) pairs from points in the first cluster to points in the second cluster, *k*-max linkage takes into account *k* maximum (farthest) pairs of observations.

In the experimental studies, the proposed *k*-Linkage method was firstly tested on five well-known benchmark datasets to demonstrate its success. The results show that the proposed approach can often produce more accurate clustering results, when compared with the traditional linkage methods in terms of accuracy rate. The proposed approach was also applied on a real-world textile dataset to discover associations and obtain complex and nonlinear relations in textile domain. In addition to these, to determine the optimal number of pairs, we ran the algorithm several times using different *k* values, varying from 3 to 9 in increments of 2, and the optimal one was selected with the highest accuracy rate.

5.1 Agglomerative Hierarchical Clustering

Hierarchical clustering is one of the major cluster analysis techniques that construct hierarchical structure of clusters through a two-dimensional diagram known as dendrogram. The main steps in the agglomerative hierarchical clustering (AHC) are presented in Figure 5.1. Each observation in the dataset is assigned to one distinct cluster, then distances between each pair of the objects of the clusters are calculated and the closest pair of clusters according to the linkage criteria is merged into one cluster continuously.

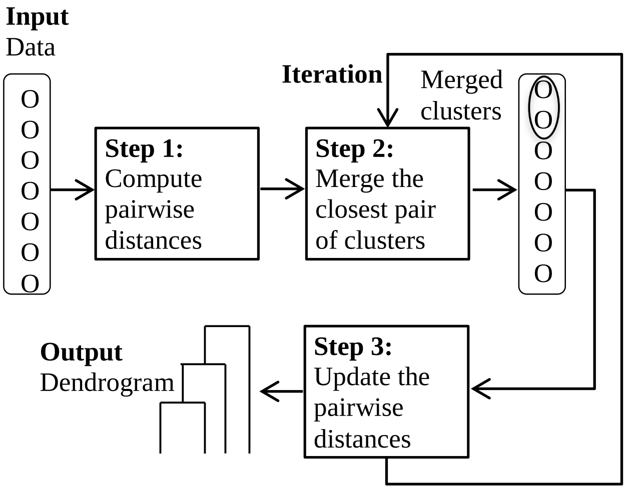


Figure 5.1 The step-by-step process of AHC algorithm

5.1.1 Classical Linkage Methods

While a hierarchical clustering algorithm is being computed on a given transactional dataset $T = \{t_1, t_2, ..., t_p\}$, there are p clusters such that $C = \{C_1, C_2, ..., C_p\}$, where $\bigcup_{i=1}^{p} C_i = T$ and $C_i \neq \varnothing$. A linkage method begins with p clusters and then the most similar clusters C_u and C_v are found and merged into one cluster. At the j-th step of the procedure, $j = 0, 1, ..., p\text{-}1$, the clustering procedure decides which of two clusters $C_u^{(j-1)}$ and $C_v^{(j-1)}$ are to be merged so that we get $C_w^{(j)} = C_u^{(j-1)} \cup C_v^{(j-1)}$, where $C_u^{(j-1)} \cap C_v^{(j-1)} = \emptyset$ for $u \neq v$.

Let $\{x_1, x_2, ..., x_m\}$ be a set of m observations from cluster C_u and $\{y_1, y_2, ..., y_n\}$ be a set of n observations from cluster C_v. The distance between clusters C_u and C_v is denoted by $d_{Cu,Cv}$ and it is formulated by $D(x,y)$ which is the distance between every possible observation x from C_u and observation y from C_v. To calculate $D(x,y)$, the Euclidian distance is usually used for numerical attributes, while Jaccard distance can be preferred for categorical variables.

Hierarchical clustering controls linkage strategies for iterative optimization, each of which repeatedly merges the most similar clusters. As an optimization problem, the main objective of our study is to minimize the differences within each cluster and maximize the differences between the clusters. It is possible to solve an optimization problem by using different techniques such as ant colony (Chen, Zhou & Luo, 2017), fuzzy models (Vaščák, 2012), particle swarm optimization (Precup, Sabau & Petriu, 2015; Vrkalovic, Teban & Borlea, 2017), and simulated annealing (Precup et al., 2015; Vrkalovic et al., 2017). The agglomerative hierarchical clustering is an example of greedy algorithms in that it does the locally optimal thing at each step, but this doesn't guarantee producing a globally optimal solution. The optimization problem in this study can be defined by objective functions, the variables and the constraints as follows.

Objective Functions: minimize d_{c_u,c_v}

Variables: $d_{Cu,Cv}$ is the distance between clusters C_u and C_v

$u,v = 1...p$ for p clusters

$C_u = \{x_1, x_2, ..., x_m\}$ for m items

$C_v = \{y_1, y_2, ..., y_n\}$ for n items

$D(x,y)$ is the distance between items $x \epsilon C_u$ and $y \epsilon C_v$

Constraints: $C_i \neq \varnothing$, $C_u \cap C_v = \emptyset$ for $u \neq v$

$D(x, y) \geq 0$, $D(x, x) = 0$ and $D(x, y) = D(y, x)$

The objective function of the optimization problem varies according to linkage method. There are mainly four linkage methods to evaluate the distances between clusters: single, complete, average and centroid. At each stage of the clustering process, two clusters that have the smallest linkage distance according to the selected linkage method are merged.

Single Linkage: (Figure 5.2a) Single linkage, also called nearest-neighbor technique, selects the distance between closest observations in clusters as shown in Equation (5.1).

Objective function for single linkage:

$$d_{C_u,C_v} = \arg\min_{(u,v)} \left(\min_{x \in C_u, y \in C_v} D(x,y) \right) \tag{5.1}$$

Complete Linkage: (Figure 5.2b) Complete linkage, also called furthest-neighbor technique, selects distance between farthest observations in clusters as shown in Equation (5.2).

Objective function for single linkage:

$$d_{C_u,C_v} = \arg\min_{(u,v)} \left(\max_{x \in C_u, y \in C_v} D(x,y) \right) \tag{5.2}$$

The main objective of this method is to minimize the maximum inter-cluster distance, as an optimization problem.

Average Linkage: (Figure 5.2c) Average linkage calculates distances between all pairs of observations in clusters and averages all of these distances as shown in Equation (5.3).

Objective function for average linkage:

$$d_{C_u,C_v} = \arg\min_{(u,v)} \left(\frac{1}{|C_u|} \frac{1}{|C_v|} \sum_{x \in C_u} \sum_{y \in C_v} D(x,y) \right) \tag{5.3}$$

where $\left|C_u\right|$ and $\left|C_v\right|$ are the number of objects in the clusters C_u and C_v respectively.

Centroid Linkage: (Figure 5.2d) Centroid linkage method finds the distance between two mean vectors of the clusters. As an optimization problem, the goal of centroid linkage method is to minimize the objective function given in Equation (5.4).

Objective function for centroid linkage:

$$d_{C_u,C_v} = \arg min\left(D(\bar{x},\bar{y})\right)$$

$$= \arg\min_{(u,v)}\left(D\left(\left(\frac{1}{|C_u|}\sum_{x\in C_u} x\right),\left(\frac{1}{|C_v|}\sum_{y\in C_v} y\right)\right)\right) \tag{5.4}$$

where $\bar{x}$ and $\bar{y}$ are the centroids (mean) of the clusters C_u and C_v respectively.

Theoretical properties of a distance measure *D(x,y)* between two objects *x* and *y* are as follows:

- *D(x, y)* ≥ 0. The distance between two objects must be strictly greater than 0.
- *D(x, x)* = 0. The distance between an object and itself must be 0.
- *D(x, y) = D(y, x)*. The distance between object *x* and *y* must be the same as the distance between *y* and *x*.

Given a dataset that consists of five instances, assume that there are two clusters $C_u = \{x_1, x_2\}$ and $C_v= \{y_1, y_2, y_3\}$. The calculation of distances between clusters in terms of four linkage methods is as follows:

- ***Single Linkage:***

$D(x, y) = \min\{D(x_1, y_1), D(x_1, y_2), D(x_1, y_3), D(x_2, y_1), D(x_2, y_2), D(x_2, y_3)\}$

- ***Complete Linkage:***

$D(x, y) = \max\{D(x_1, y_1), D(x_1, y_2), D(x_1, y_3), D(x_2, y_1), D(x_2, y_2), D(x_2, y_3)\}$

- ***Average Linkage:***

$$D(x,y) \ = \ \frac{\begin{gathered} D(x_1,y_1)+D(x_1,y_2)+D(x_1,y_3)+ \\ D(x_2,y_1)+D(x_2,y_2)+\ D(x_2,y_3) \end{gathered}}{6}$$

- ***Centroid Linkage:***

$$D(x,y) \ = \ D\left(\left(\frac{x_1+\ x_2}{2}\right),\left(\frac{y_1+\ y_2+\ y_3}{3}\right)\right)$$

To unify all of these methods, the Lance - Williams procedure provides a generalization in which all methods are special cases, as given in Equation (5.5).

Objective Function:

Minimizing

$$d_{C(AB)} = \alpha_A d_{CA} + \alpha_B d_{CB} + \beta d_{AB} + \gamma |d_{AC} - d_{BC} \tag{5.5}$$

Variables:

α_A, α_B, β, γ are parameters

A, B, C are clusters.

d_{ij} is the distance of cluster (or object) pairs.

$d_{C(AB)}$ is the distance between cluster *C* and the new cluster *AB*.

n_i refers to the number of items in cluster *i*, $i \epsilon \{A, B, C\}$.

Constraints:

$\alpha_A + \alpha_B + \beta = 1$

$\alpha_A = \alpha_B$

$\beta < 1$

Single Linkage:

α_A=1/2, α_B=1/2, β=0, γ=-1/2

Complete Linkage:

α_A=1/2, α_B=1/2, β=0, γ=1/2

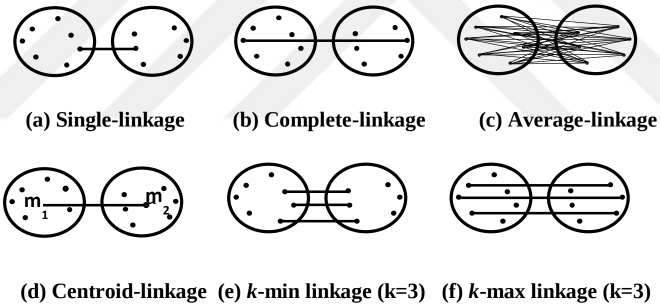
Average Linkage:

α_A=$n_A/(n_A+n_B)$, α_B=$n_B/(n_A+n_B)$, β=0, γ=0

Centroid Linkage:

α_A=$n_A/(n_A+n_B)$, α_B=$n_B/(n_A+n_B)$, β=$-n_{AnB}/(n_A+n_B)^2$, γ=0

In order to compensate the drawbacks of the current linkage schemes (given in the 5.1.2. section), we propose a new linkage criterion: *k*-Linkage. Instead of considering only one pair like single and complete linkage methods, the *k*-min linkage (Figure 5.2e) and *k*-max linkage (Figure 5.2f) methods take into account more than one pairs, i.e. *k* closest or *k* furthest pairs respectively.



(a) Single-linkage **(b) Complete-linkage** **(c) Average-linkage**

(d) Centroid-linkage **(e) *k*-min linkage (k=3)** **(f) *k*-max linkage (k=3)**

Figure 5.2 Classical and proposed linkage methods

5.1.2 Drawbacks of Classical Linkage Methods

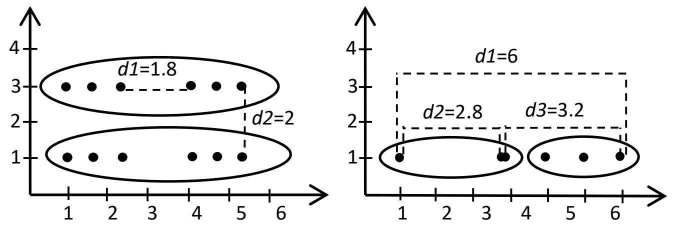
The classical linkage methods have the following drawbacks (Gagolewski, Bartoszuk & Cena, 2016; Chen, Zhou & Luo, 2017):

- The *single linkage* method suffers from a chaining effect and produces long chains and it has a tendency to produce clusters that are straggly or elongated. Figure 5.3a demonstrates chaining problem in single-link clustering. The single-link method only compares *d1* and *d2* distances, and the distance between left-right clusters (*d1*) is smaller than the distance between up-down clusters (*d2*). Since the merge criterion considers one pair and, a chain of points can be extended for long distances without regard to the overall shape of the emerging cluster (Manning, Raghavan & Schütze, 2012). In addition, the single linkage method tends to construct clusters of unbalanced sizes and produces highly skewed dendrograms. Furthermore, it has limitations on the detection of clusters that are not well separated. On the other hand, it might be useful to detect outliers in the dataset, because outliers often appear as clusters with only one member.

- The *complete-linkage* method in general tends to produce tightly bound clusters. Clusters tend to be compact and roughly equal in diameter. In addition, complete linkage method is sensitive to outliers which are points that do not fit well into the global structure of the cluster. Figure 5.3b demonstrates outlier problem in complete-link clustering. The outlier at the left edge splits the optimal cluster because the smallest furthest distance is *d2* among alternative distances. It tends to break large clusters, often resulting in a single large cluster and a number of singletons or ones with a very low cardinality.

- The *average linkage* method is somewhere between single linkage and complete linkage. However, it takes long time to calculate the distances between all pairs and average all of these distances. The time needed to apply a hierarchical clustering algorithm is most often dominated by the number of computations of a pairwise distance measure. Time constraint is an important issue for large datasets.

- In the *centroid linkage* method, the center will move as clusters are merged. As a result, the distance between merged clusters may actually decrease between steps, making the analysis of results problematic. In other words, clustering with centroid linkage is not monotonic and can contain an inversion, which means that similarity can increase during clustering, instead of monotonically decreasing from iteration to iteration. In the case of an inversion in a dendrogram, a horizontal merge line shows up lower than the previous merge line. Increasing similarity in clustering steps contradicts the fundamental assumption that small clusters are more coherent than large clusters (Manning et al., 2012). Therefore, the algorithm causes problems that some instances may need to be switched from their original clusters.



(a) **(b)**

Figure 5.3 (a) Chaining problem in single-link clustering (b) Outlier problem in complete-link clustering

Solution quality in hierarchical clustering may vary depending on how clusters are fused. There is no guarantee that single or complete linkage will collectively or individually give the optimal clusters. They sometimes do not reflect the true underlying data structure. Because of the greedy nature of the single and complete linkages in hierarchical clustering, the algorithm returns clusters that are only locally optimal. The presence of local minima leads to incorrect clustering results. To overcome the limitations of the current linkage methods, this study proposes a new linkage method, named *k*-Linkage.

5.1.3 Proposed K-Linkage Method

K-Linkage is a novel linkage method which aims to find similarity of clusters by considering *k* observations from a cluster C_u and *k* observations from another cluster C_v. In the subject of *k*-Linkage method, this study proposes two novel concepts, named *k-min linkage* and *k-max linkage*, to evaluate distances between clusters.

5.1.3.1 K-min Linkage Method

The *k*-min linkage method calculates the sum of distances between *k* closest observations in clusters and finds the average of them as a similarity measure. Definition 1 defines *k*-min linkage concept for the first time.

Definition 1. Let *{x_1, x_2, ..., x_m}* be a set of *m* observations from cluster C_u and *{y_1, y_2, ..., y_n}* be a set of *n* observations from cluster C_v. The distance between clusters C_u and C_v is denoted by $d_{Cu,Cv}$ and it is formulated by taking the average of *k* closest observation pairs *(x,y)*, where $x \in C_u$ and $y \in C_v$, as shown in Equation (5.6).

Objective function for k-min linkage:

$$d_{C_u,C_v} = \frac{1}{|k|} \sum_{i=1}^{k} \underset{(i)}{\arg} \min_{(u,v)} \left(\min_{x \in C_u, y \in C_v} D(x,y) \right) \tag{5.6}$$

On the basis of *k*-min linkage method, two clusters which have the most *k* similar members on average are merged at each stage of the process. For the case where more than one pairs of observations have the same similarity, some of them can easily be selected, because overall average distance doesn't change in the case of ties.

5.1.3.2 K-max Linkage Method

The *k*-max linkage method calculates the sum of distances between *k* farthest observations in clusters and finds the average of them as a similarity measure. Definition 2 defines *k*-max linkage concept for the first time.

Definition 2. Let *{x_1, x_2, ..., x_m}* be a set of *m* observations from cluster C_u and *{y_1, y_2, ..., y_n}* be a set of *n* observations from cluster C_v. The distance between clusters C_u and C_v is denoted by $d_{Cu,Cv}$ and it is formulated by taking the average of *k* farthest observation pairs *(x,y)*, where $x \in C_u$ and $y \in C_v$, as shown in Equation (5.7).

Objective function for k-max linkage:

$$d_{C_u,C_v} = \frac{1}{|k|} \sum_{i=1}^{k} \underset{(i)}{\arg} \min_{(u,v)} \left(\max_{x \in C_u, y \in C_v} D(x,y) \right) \tag{5.7}$$

On the basis of *k*-max linkage method, two clusters which have the most *k* dissimilar members on average are merged at each stage of the process.

The agglomerative hierarchical clustering algorithm has been improved in this research. The steps of the application of the proposed methods are given below.

Step 1. Assign each object in the dataset to a separate cluster so that for *n* objects we have *n* clusters each containing just one object.

Step 2. Calculate the distances between the clusters.

Step 3. Find the closest pair of clusters based on a similarity strategy and merge them into a single cluster.

- ***K*-min Linkage:** Select the average distance of *k*-closest objects between clusters.
- ***K*-max Linkage:** Select the average distance of *k*-farthest objects between clusters.

Step 4. Compute the distances between new cluster and each of other clusters.

Step 5. Repeat steps 3 and 4 until all objects are clustered into a single cluster.

5.1.4 Advantages of K-Linkage Method over Traditional Hierarchical Methods

Proposed *k*-Linkage method has several advantages over traditional linkage methods. First, considering *k* observations instead of single observation prevents the greedy nature of the single and complete linkages. It also achieves greater speed-up than average and centroid linkage, so as to reduce the number of computations of a pairwise distance measure, because a spatial index can be used for quick neighborhood lookup.

Another advantage of the proposed *k*-Linkage method is that it can be used to detect clusters with arbitrary shapes, because it prevents both chaining and rounding effects by considering several pairs. While the single linkage method produces elongated clusters and the complete linkage method tends to construct spherical clusters, *k*-Linkage method is more robust to local optimal decisions.

The main advantage behind the *k*-Linkage clustering lies in the fact that its solution is similar to the well-known technique *K*-nearest neighbor (KNN). How the KNN algorithm takes into account several objects when classifying a new instance, similarly, *k*-Linkage method also considers several pairs to make sure about the relationship among clusters.

5.1.5 Advantages of K-Linkage Method over Non-Hierarchical Methods

There are many advantages of *k*-Linkage method in comparison with non-hierarchical clustering algorithms such as k-Means, DBSCAN. First, *k*-Linkage method presents hierarchical structure of clusters, so it gives more information about the clusters than the unstructured set of clusters returned by non-hierarchical clustering. Thus, the output of the *k*-linkage method is easy to interpret and very useful in understanding the dataset.

Another significant advantage of *k*-Linkage method is that it is deterministic (non-random) which means that it does not include any random parameter or random initialization technique. Thus, it produces the same results when run several times on the same data. However, some non-hierarchical clustering algorithms (i.e. *k*-Means) depend on random initialization so that clustering results may vary across runs.

Another advantage of our *k*-Linkage method is that it is appropriate for clustering high-dimensional data. Besides these advantages, *K*-linkage method also supports different forms of similarity and distance, thus it can be used with many attribute types. It does not even require a distance, any measure can be used, including similarity functions, such as Euclidean distance for numerical data, Jaccard distance for categorical data, Levenshtein distance for strings, and Gower distance for time series and mixed type data, even semantic similarity measures. However, some non-hierarchical clustering algorithms are limited to Euclidean distance.

5.1.6 An Example for K-Linkage Method

In this section, the application of the proposed linkage metrics (*k*-min and *k*-max linkage) in agglomerative hierarchical clustering is illustrated by two datasets. Table 5.1 shows sample datasets that contain *X* and *Y* coordinates of the instances and consist of 18 records that are uniquely identified by an ID. Only *Y* value of the element "*N*" is different between two datasets. In this study, Euclidean distance was used as distance measurement between instances.

Table 5.1 Sample datasets

Dataset 1		
ID	**X**	**Y**
A	2	7
B	2	6
C	3	6
D	3	8
E	4	6
F	5	8
G	7	9
H	12	16
I	13	16
J	14	16
K	14	17
L	15	16
M	16	16
N	15	**18**
O	12	9
P	13	9
Q	15	9
R	16	8

Dataset 2		
ID	**X**	**Y**
A	2	7
B	2	6
C	3	6
D	3	8
E	4	6
F	5	8
G	7	9
H	12	16
I	13	16
J	14	16
K	14	17
L	15	16
M	16	16
N	15	**24**
O	12	9
P	13	9
Q	15	9
R	16	8

Table 5.2 shows the clustering steps of AHC algorithm with single and *k*-min linkage methods on dataset 1, while Table 5.3 shows the steps for complete and *k*-max linkage methods on dataset 2, where the user defined *k* parameter was selected

as 3. In the first step, each observation in the dataset is assumed as one distinct cluster. After that, the most similar pair of clusters according to the selected linkage criteria are merged into one cluster in each step. For example; the closest clusters in the first step are cluster {A} and cluster {B}, so clusters {A} and {B} are merged as {A, B} in step 2. This process is continued until all clusters are merged into one cluster. The first sixteen steps, each producing a new cluster by merging two existing clusters, are identical. At the step 17 in Table 5.2, single-link clustering joins clusters {A,B,C,E,D,F,G} and {O,P,Q,R} because on the maximum similarity definition of cluster similarity, those two clusters are closest. On the other hand, *k*-min linkage method joins clusters {H,I,J,K,L,M,N} and {O,P,Q,R} because those are the closest clusters according to top three similar pairs among them. Similarly, complete and *k*-max linkage methods make a difference at the step 17.

Table 5.2 Clustering steps based on single and *k*-min linkage methods for dataset 1

Step	Clusters using Single Linkage	Clusters using k-min Linkage (k=3)
1	{A},{B},{C},{D},{E},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}	{A},{B},{C},{D},{E},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}
2	**{A,B}**,{C},{D},{E},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}	**{A,B}**,{C},{D},{E},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}
3	**{A,B,C}**,{D},{E},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}	**{A,B,C}**,{D},{E},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}
4	**{A,B,C,E}**,{D},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}	**{A,B,C,E},**{D},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}
5	{A,B,C,E},{D},{F},{G},**{H,I}**,{J},{K},{L},{M},{N},{O},{P},{Q},{R}	{A,B,C,E},{D},{F},{G},**{H,I}**,{J},{K},{L},{M},{N},{O},{P},{Q},{R}
6	{A,B,C,E},{D},{F},{G},**{H,I,J}**,{K},{L},{M},{N},{O},{P},{Q},{R}	{A,B,C,E},{D},{F},{G},**{H,I,J}**,{K},{L},{M},{N},{O},{P},{Q},{R}
7	{A,B,C,E},{D},{F},{G},**{H,I,J,K}**,{L},{M},{N},{O},{P},{Q},{R}	{A,B,C,E},{D},{F},{G},**{H,I,J,K}**,{L},{M},{N},{O},{P},{Q},{R}
8	{A,B,C,E},{D},{F},{G},**{H,I,J,K,L}**,{M},{N},{O},{P},{Q},{R}	{A,B,C,E},{D},{F},{G},**{H,I,J,K,L}**,{M},{N},{O},{P},{Q},{R}
9	{A,B,C,E},{D},{F},{G},**{H,I,J,K,L,M}**,{N},{O},{P},{Q},{R}	{A,B,C,E},{D},{F},{G},**{H,I,J,K,L,M}**,{N},{O},{P},{Q},{R}
10	{A,B,C,E},{D},{F},{G},{H,I,J,K,L,M},{N},**{O,P}**,{Q},{R}	{A,B,C,E},{D},{F},{G},{H,I,J,K,L,M},{N},**{O,P}**,{Q},{R}
11	**{A,B,C,E,D}**,{F},{G},{H,I,J,K,L,M},{N},{O,P},{Q},{R}	**{A,B,C,E,D}**,{F},{G},{H,I,J,K,L,M},{N},{O,P},{Q},{R}
12	{A,B,C,E,D},{F},{G},**{H,I,J,K,L,M,N}**,{O,P},{Q},{R}	{A,B,C,E,D},{F},{G},**{H,I,J,K,L,M,N}**,{O,P},{Q},{R}
13	{A,B,C,E,D},{F},{G},{H,I,J,K,L,M,N},{O,P},**{Q,R}**	{A,B,C,E,D},{F},{G},{H,I,J,K,L,M,N},{O,P},**{Q,R}**

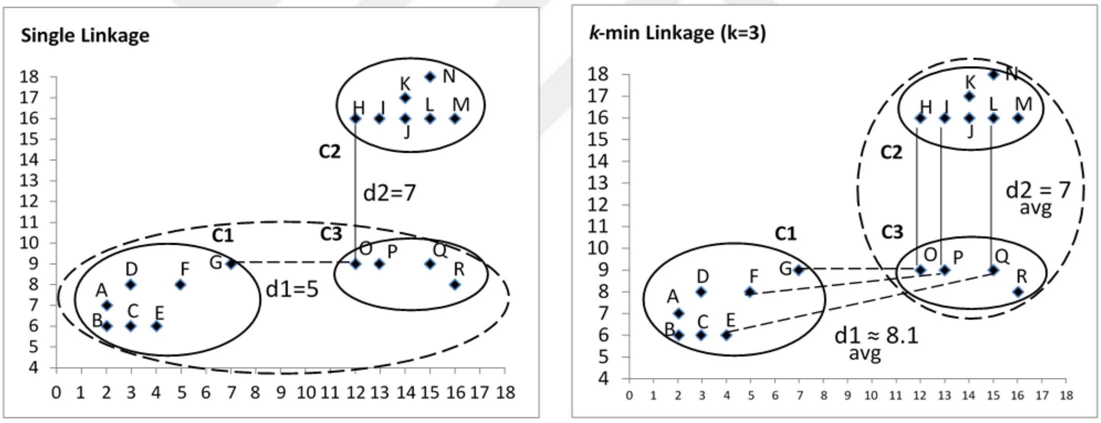
Table 5.3 continues

14	**{A,B,C,E,D,F}**,{G},{H,I,J,K,L,M,N},{O,P},{Q,R}	**{A,B,C,E,D,F}**,{G},{H,I,J,K,L,M,N},{O,P},{Q,R}
15	{A,B,C,E,D,F},{G},{H,I,J,K,L,M,N},**{O,P,Q,R}**	{A,B,C,E,D,F},{G},{H,I,J,K,L,M,N},**{O,P,Q,R}**
16	**{A,B,C,E,D,F,G}**,{H,I,J,K,L,M,N},{O,P,Q,R}	**{A,B,C,E,D,F,G}**,{H,I,J,K,L,M,N},{O,P,Q,R}
17	**{A,B,C,E,D,F,G,O,P,Q,R}**,{H,I,J,K,L,M,N}	{A,B,C,E,D,F,G},**{H,I,J,K,L,M,N,O,P,Q,R}**
18	{A,B,C,E,D,F,G,O,P,Q,R,H,I,J,K,L,M,N}	{A,B,C,E,D,F,G,O,P,Q,R,H,I,J,K,L,M,N}

Table 5.4 Clustering steps based on complete and *k*-max linkage methods for dataset 2

Step	Clusters using Complete Linkage	Clusters using k-max Linkage (k=3)
1	{A},{B},{C},{D},{E},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}	{A},{B},{C},{D},{E},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}
2	**{A,B}**,{C},{D},{E},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}	**{A,B}**,{C},{D},{E},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}
3	{A,B},**{C,E}**,{D},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}	{A,B},**{C,E}**,{D},{F},{G},{H},{I},{J},{K},{L},{M},{N},{O},{P},{Q},{R}
4	{A,B},{C,E},{D},{F},{G},**{H,I}**,{J},{K},{L},{M},{N},{O},{P},{Q},{R}	{A,B},{C,E},{D},{F},{G},**{H,I}**,{J},{K},{L},{M},{N},{O},{P},{Q},{R}
5	{A,B},{C,E},{D},{F},{G},{H,I},**{J,K}**,{L},{M},{N},{O},{P},{Q},{R}	{A,B},{C,E},{D},{F},{G},{H,I},**{J,K}**,{L},{M},{N},{O},{P},{Q},{R}
6	{A,B},{C,E},{D},{F},{G},{H,I},{J,K},**{L,M}**,{N},{O},{P},{Q},{R}	{A,B},{C,E},{D},{F},{G},{H,I},{J,K},**{L,M}**,{N},{O},{P},{Q},{R}
7	{A,B},{C,E},{D},{F},{G},{H,I},{J,K},{L,M},{N},**{O,P}**,{Q},{R}	{A,B},{C,E},{D},{F},{G},{H,I},{J,K},{L,M},{N},**{O,P}**,{Q},{R}
8	{A,B},{C,E},{D},{F},{G},{H,I},{J,K},{L,M},{N},{O,P},**{Q,R}**	{A,B},{C,E},{D},{F},{G},{H,I},{J,K},{L,M},{N},{O,P},**{Q,R}**
9	{A,B},{C,E},**{D,F}**,{G},{H,I},{J,K},{L,M},{N},{O,P},{Q,R}	{A,B},{C,E},**{D,F}**,{G},{H,I},{J,K},{L,M},{N},{O,P},{Q},{R}
10	**{A,B,C,E}**,{D,F},{G},{H,I},{J,K},{L,M},{N},{O,P},{Q,R}	**{A,B,C,E}**,{D,F},{G},{H,I},{J,K},{L,M},{N},{O,P},{Q},{R}
11	{A,B,C,E},{D,F},{G},**{H,I,J,K}**,{L,M},{N},{O,P},{Q,R}	{A,B,C,E},{D,F},{G},**{H,I,J,K}**,{L,M},{N},{O,P},{Q},{R}
12	**{A,B,C,E,D,F}**,{G},{H,I,J,K},{L,M},{N},{O,P},{Q,R}	**{A,B,C,E,D,F}**,{G},{H,I,J,K},{L,M},{N},{O,P},{Q,R}
13	{A,B,C,E,D,F},{G},**{H,I,J,K,L,M}**,{N},{O,P},{Q,R}	{A,B,C,E,D,F},{G},**{H,I,J,K,L,M}**,{N},{O,P},{Q,R}
14	{A,B,C,E,D,F},{G},{H,I,J,K,L,M},{N},**{O,P,Q,R}**	{A,B,C,E,D,F},{G},{H,I,J,K,L,M},{N},**{O,P,Q,R}**
15	**{A,B,C,E,D,F,G}**,{H,I,J,K,L,M},{N},{O,P,Q,R}	**{A,B,C,E,D,F,G}**,{H,I,J,K,L,M},{N},{O,P,Q,R}
16	{A,B,C,E,D,F,G},**{H,I,J,K,L,M,N}**,{O,P,Q,R}	{A,B,C,E,D,F,G},**{H,I,J,K,L,M,N}**,{O,P,Q,R}
17	**{A,B,C,E,D,F,G,O,P,Q,R}**,{H,I,J,K,L,M,N}	{A,B,C,E,D,F,G},**{H,I,J,K,L,M,N,O,P,Q,R}**
18	{A,B,C,E,D,F,G,O,P,Q,R,H,I,J,K,L,M,N}	{A,B,C,E,D,F,G,O,P,Q,R,H,I,J,K,L,M,N}

Figure 5.4 visualizes the clusters at the step 16 in Table 5.2. The figure shows that the application of AHC algorithm based on different linkage methods (single-link and *k*-min-link) can produce different clustering results at the next step. In the single-link clustering process (Figure 5.4a), the distances between “G-O” denoted by *d1* and “O-H” denoted by *d2* are compared, and then clusters *C1* and *C3* are merged, because *d1* is smaller than *d2*. However, this causes the chaining problem as discussed in Section 5.1.2. On the other hand, *k*-min linkage method merges clusters *C2* and *C3* according to the average of top three closest pairs: “O-H”, “I-P”, and “L-Q”. Thus, *k*-min linkage method avoids the construction of long chains and the production of clusters that are elongated as shown in Figure 5.4b. In addition, *k*-min linkage method reduces the sum of squared errors (SSE) from 3.32 to 2.85 in this example. So, the SSE results show that it is possible to get more optimal clustering results by using *k*-min linkage based agglomerative clustering algorithm.

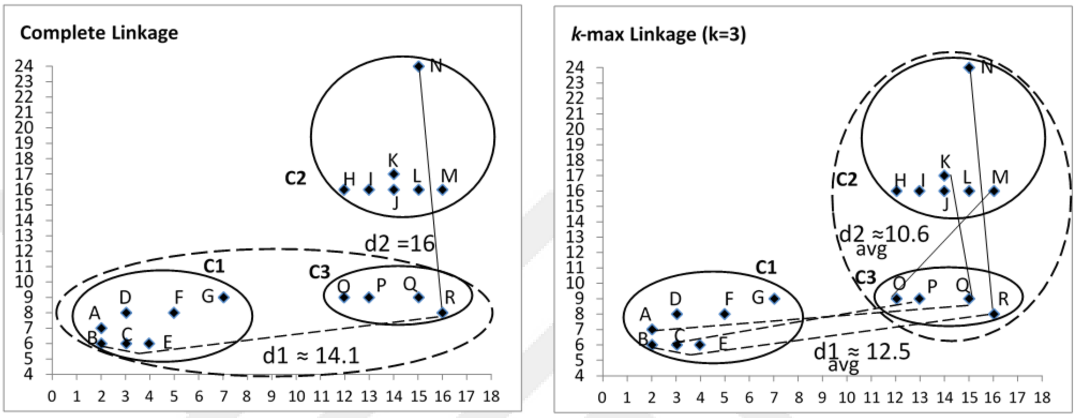


(a) Single Linkage merges clusters C1 and C3

(b) *K*-min Linkage merges clusters C2 and C3

Figure 5.4 Merging clusters with *k*-min linkage and single linkage methods

Figure 5.5 is an example of a complete linkage clustering of the set of points given in Table 5.1 and the *k*-max linkage clustering of the same set. It visualizes the clusters at the step 16 in Table 5.3. In complete-link clustering, the distances between “B-R” denoted by *d1* and “N-R” denoted by *d2* are compared, and then clusters *C1* and *C3* are merged, because *d1* is smaller than *d2*. On the other hand, *k*-max linkage method merges clusters *C2* and *C3* according to the average of top three farthest pairs: “N-R”, “O-M”, and “K-Q”. The sum of squared errors (SSE) of the clusters that are constructed by the complete and k-max linkages are 3.32 and 2.24 respectively. This means that complete-link clustering didn't find the most optimal cluster structure in this example, because it pays too much attention to outliers as explained in section 5.1.2. It can be affected by points at a great distance in a cluster where two merge candidates meet. However, *k*-max linkage method avoids the greedy nature of the complete-link by considering several pairs.

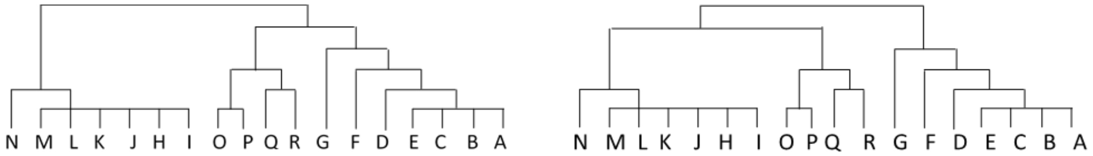


(a) Complete Linkage merges clusters C1 and C3

(b) *K*-max Linkage merges clusters C2 and C3

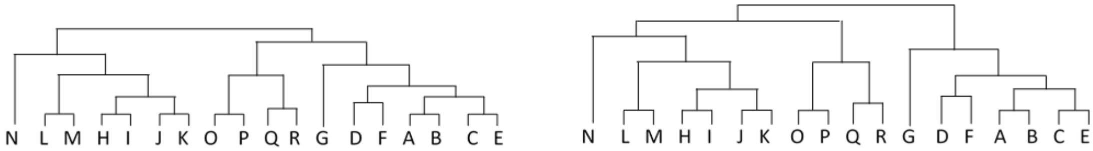
Figure 5.5 Merging clusters with *k*-max linkage and complete linkage methods

The clustering steps of four different methods (single, complete, *k*-min and *k*-max linkage) for the datasets given in Table 5.1 are summarized via dendrograms in Figure 5.6 and Figure 5.7. The first clusters are the same for single linkage and *k*-min linkage methods. However, the dendrogram differs in the last steps. Complete linkage and *k*-max linkage lead to the similar dendrogram pattern, but differs towards the end. The hierarchy needs to be cut at some point. A number of criteria can be used to determine the cutting point: (i) cut the dendrogram at a pre-specified level of similarity, (ii) cut the hierarchy where the gap between two successive similarities is largest, (iii) cut at the point that a target number of clusters is reached (Manning et al., 2012).


N M L K J H I O P Q R G F D E C B A
N M L K J H I O P Q R G F D E C B A

(a) Single linkage method **(b) *k*-min linkage method (*k*=3)**

Figure 5.6 Dendrograms of single-link and *k*-min link clustering


N L M H I J K O P Q R G D F A B C E
N L M H I J K O P Q R G D F A B C E

(a) Complete linkage method **(b) *k*-max linkage method (*k*=3)**

Figure 5.7 Dendrograms of complete-link and *k*-max link clustering

5.2 The Algorithm of *K*-Linkage Method

This study presents an improved version of agglomerative hierarchical clustering algorithm that has the ability to cluster objects in a dataset using *k*-Linkage similarity metric. The pseudocode of the proposed *k*-Linkage method is presented in Figure 5.8. This algorithm only finds the distance between two clusters *ClusterA* and *ClusterB*, so it should be called from a native AHC algorithm that executes the steps of merging the currently most similar clusters. In the pseudocode, a "*Link*" structure is constituted to define a pair of two objects; one belonging to the first cluster and the other belonging to the second cluster. The algorithm stores a list of pairs by calculating the distances between objects in the pairs using *findDistance()* function. If the linkage type is *k*-min, the list is sorted by ascending order considering *distance* values. On the contrary, in the case of *k*-max, the list is sorted by descending order. After that, the obtained top-*k* distances in the list according to user defined *k* value are averaged. The pairs including selected objects are removed from the list to consider distinct objects in the next step. The output value (*kLinkageDistance*) represents the distance between two clusters based on *k*-Linkage method.

```
struct Link
begin
    int startPoint
    int endPoint
   double distance
end

Algorithm K-Linkage
   Inputs: ClusterA: the first cluster,
           ClusterB: the second cluster,
           k: the number of object pairs,
           linkageType: the type of linkage (k_min or k_max)
   Output: kLinkageDistance
begin
    List list = new List()
    k = min(ClusterA.numberofObjects(), ClusterB.numberofObjects(), k)
    for i = 0 to ClusterA.numberofObjects() - 1
       for j = 0 to ClusterB.numberofObjects() - 1
          list.Add(new Link(i, j, findDistance(ClusterA[i], ClusterB[j])
       end for
    end for
    if linkageType == k_min
       list.OrderBy(d => d.distance)
    else
       list.OrderByDescending(d => d.distance)
    end if
    double totalDistance = 0
    for i = 0 to k-1
       totalDistance += list[0].distance
       int tempStart = list[0].startPoint
       int tempEnd = list[0].endPoint
       list.RemoveAll(s => s.startPoint == tempStart  ||  e => e.endPoint == tempEnd)
    end for
    double kLinkageDistance = totalDistance / k
    return kLinkageDistance
end
```

Figure 5.8 The pseudocode of the proposed *k*-Linkage method

AHC is one of the most commonly used hierarchical clustering algorithms but it needs a significant amount of time to cluster considerably large datasets. The complexity of the naive AHC algorithm is O(n^3), because it exhaustively requires to scan the *n*×*n* matrix to find the most similar clusters in each of *n*-1 iterations, where *n* is the number of instances (Manning et al., 2012). To handle this problem and to reduce time complexity to O(n^2), several improved algorithms are proposed, such as SLINK and CLINK for single-linkage and complete-linkage criterions respectively. Another study (Walter, Bala, Kulkarni & Pingali, 2008) uses kd-tree (k-dimensional tree) with locally-ordered and heap-based versions in which empirical performance is better than O(n^2) and closer to linear scaling with input size. The time complexity of the proposed *k*-Linkage method is also O(n^2) with a proper data structure and index-assisted searching mechanism, where *n* is the number of instances in the dataset.

5.3 Experimental Study

In this study, the proposed linkage types were compared with traditional linkage types such as single, complete, centroid and average linkages. We have expanded an application that can be accessed from GitHub repository: https://github.com/gyaikhom/agglomerative-hierarchical-clustering. The application was implemented for agglomerative hierarchical clustering in C programming language. Our expanded application reads data from a file and includes six different methods to cluster data: single, complete, average, centroid, *k*-min linkage and *k*-max linkage. First, the application was executed on five different benchmark datasets with varying *k* numbers to determine the optimal solution. In order to evaluate the cluster results and to compare our method with the existing methods, accuracy rate is calculated by comparing output cluster labels with previously known class labels. Then, the proposed approach was also applied on a benchmark textile dataset with selecting *k* number as 3.

5.3.1 Dataset Description

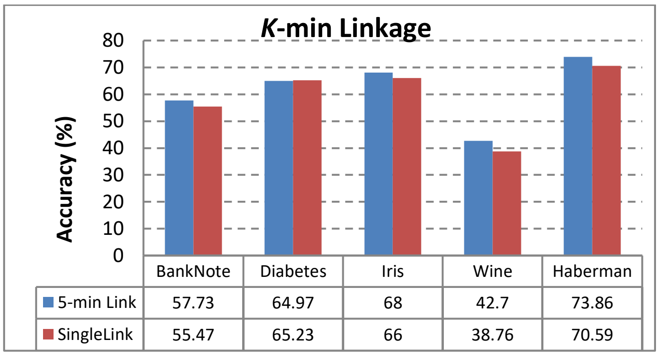
In the first experimental study, five different datasets which are well-known and broadly used in data mining were selected to demonstrate the capabilities ok *k*-min and *k*-max linkage methods. The datasets, named Iris, Wine, Haberman, Diabetes and Banknote were obtained from UCI Machine Learning Repository that can be accessed from the web site https://archive.ics.uci.edu/ml/datasets.html.

Then, a benchmark textile dataset was utilized in the second experiment to discover complex relations using *k*-min and *k*-max linkage methods. This dataset gives an information about comfort ratings for military fabrics.

5.3.2 Comparison of K-Linkage Method with Traditional Methods

In this experimental study, *k*-min and *k*-max linkage methods have been used for the first time to improve clustering validation and quality. To measure cluster validity in the first experiment, cluster labels that match externally previously known class labels are evaluated and regarded as accuracy rate of clustering result. In simple terms, *accuracy* is the ratio of the number of correctly clustered data points to the total number of data points. Accuracy is calculated using the formula, accuracy = (TP + TN)/ (TP + FN + TN + FP), where TP, FP, TN, and FN denote the number of true positives, false positives, true negatives, and false negatives, respectively.

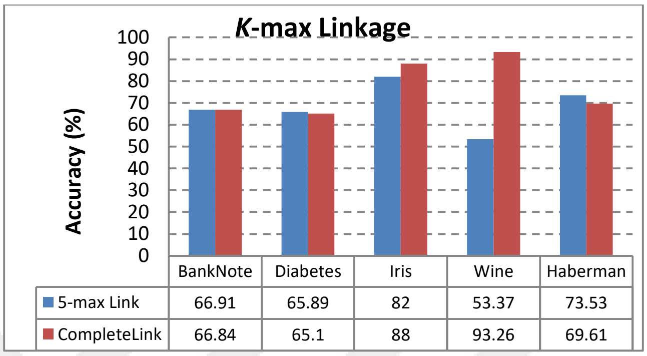
In order to evaluate the proposed *k*-linkage scheme, we tested it in various datasets. Figure 5.9 shows the comparative results of the *k*-min linkage method with single linkage method in terms of accuracy rate. After trying different alternatives, the best value for the *k* parameter was determined as 5. The obtained results show that the proposed *k*-min linkage method is generally more successful than the single linkage method in terms of accuracy rate. Even though single-link clustering may seem preferable at first, it is optimal with respect to the wrong criterion in many clustering applications. Single-link clustering reduces the assessment of cluster quality to a single similarity between a pair of observations. Since the merge criterion is strictly local, it cannot recognize the overall distribution of the clusters. On the other hand, *k*-min linkage method can reflect the true underlying relationship between clusters by considering several pairs and so it can find the better merge candidates.



	BankNote	Diabetes	Iris	Wine	Haberman
🟦 5-min Link	57.73	64.97	68	42.7	73.86
🟥 SingleLink	55.47	65.23	66	38.76	70.59

Figure 5.9 Comparison of *k*-min linkage with single linkage

Figure 5.10 shows a comparison between complete and *k*-max linkage methods, where *k* is equal to 5. The results show that the proposed *k*-max linkage method has a potential to outperform the complete-link approach. A measurement based on only one pair cannot fully reflect the distribution of instances in a cluster. It is therefore not surprising that complete-link algorithm can produce undesirable clusters. Considering *k* pairs in each step of clustering, instead of only one pair, can improve cluster validation. Taking into account our algorithm's accuracy performance, the proposed method may be recommended for practical use.



	BankNote	Diabetes	Iris	Wine	Haberman
🟦 5-max Link	66.91	65.89	82	53.37	73.53
🟥 CompleteLink	66.84	65.1	88	93.26	69.61

Figure 5.10 Comparison of *k*-max linkage with complete linkage

Table 5.4 gives comparison results of *k*-linkage metric with traditional similarity metrics (single-link, complete-link, average-link, and centroid linkage) on the datasets in terms of accuracy rate. According to the results, its performance is comparable with other linkage methods. The proposed *k*-linkage method outperforms the current linkage methods in three of the five datasets in terms of the clustering quality. Centroid clustering is not optimal for any dataset because inversions can occur. Rather than average-link, *k*-linkage method can be used, because its similarity measure is conceptually simpler than the average of all pairwise similarities. On the other hand, a spatial indexing mechanism can be used for *k*-linkage methods to determine top-*k* closest or farthest pairs faster.

Table 5.5 Comparison of *k*-linkage methods with classical methods

	Accuracy Rate (%)					
Dataset	**k-min Link (k=5)**	**k-max Link (k=5)**	**Single Link**	**Complete Link**	**Average Link**	**Centroid**
BankNote	57.73	**66.91**	55.47	66.84	64.5	63.78
Diabetes	64.97	**65.89**	65.23	65.1	65.23	65.23
Iris	68	82	66	88	**88.67**	66
Wine	42.7	53.37	38.76	**93.26**	38.76	38.76
Haberman	**73.86**	73.53	70.59	69.61	69.61	71.24

The proposed method *k*-linkage where *k* is equal to 3 was implemented on the textile dataset to compare this metric with traditional similarity metrics (single-link and complete-link) in terms of SSE. The results presented in Figure 5.11 indicate that the sum of squared errors (SSE) of the clusters that are constructed by the single, complete, 3-min and 3-max linkages are 3.44, 3.14, 3.09, and 2.80 respectively. This means that 3-max and 3-min linkages provide more accurate clustering results than the single-link and complete-link metrics. It is also possible to say that 3-max linkage method has the best clustering score among the other metrics.

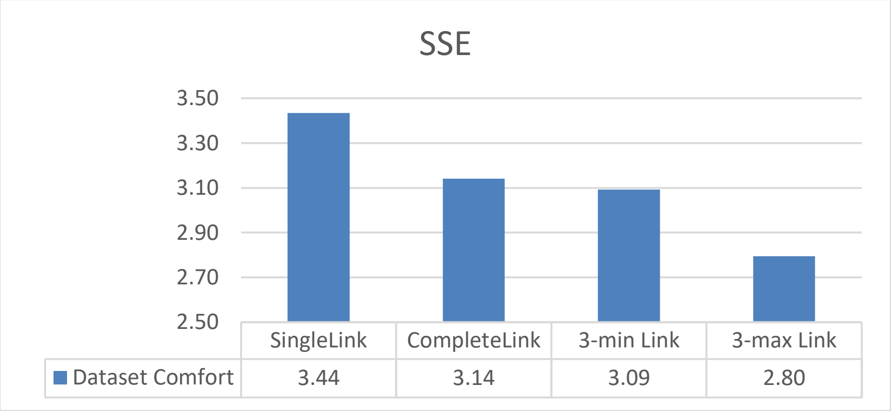


Figure 5.11 Comparison of *k*-linkage methods with single and complete linkage methods

5.3.3 The Effect of Parameter on K-Linkage Method

K-linkage method requires a user defined parameter *k* which is the number of pairs of instances between clusters. To achieve optimal *k* value, several experiments can be performed as trial-and-error approach and the value which gives the highest accuracy rate can be selected as *k*. The graphs in Figure 5.12 and Figure 5.13 show the accuracy rate changes for *k*-min and *k*-max linkage, where *k* is ranging from 3 to 9 in increments of 2. It is possible to see from the results that when the value of the *k* parameter increases, the accuracy rate remains the same or becomes a little bit higher. However, the rate of increase differs from dataset to dataset.

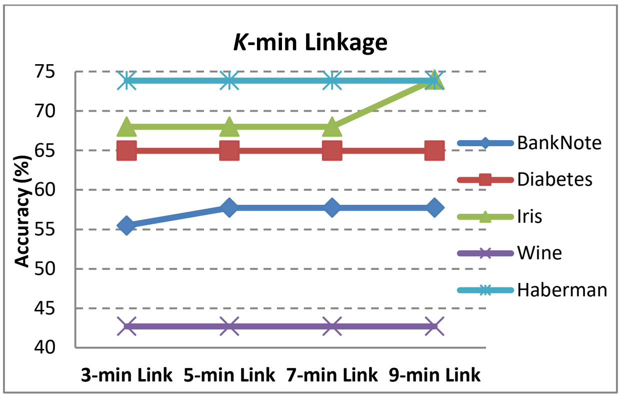


Figure 5.12 Parameter selection for *k*-min linkage method

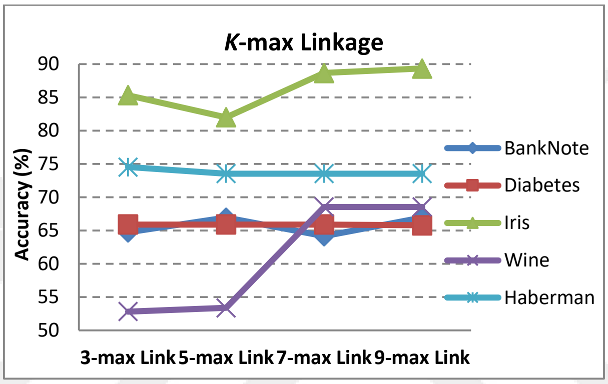


Figure 5.13 Parameter selection for *k*-max linkage method

CHAPTER SIX
AN ASSOCIATION RULE MINING STUDY: EXTENDED FP-GROWTH ALGORITHM

Textile datasets can be of any end-product in the textile industry, such as a fiber, yarn, fabric, or garment. Knowing what is expected from a raw material is important to both the supplier of raw material and the purchaser. A cotton grower and fiber manufacturer would like to know what sort of yarn quality can be produced from their crop so that they can ask the right price for the fiber. The buyer, a yarn manufacturer, would be interested in knowing whether the desired yarn properties can be obtained from a particular variety of cotton it intends to buy. The user of the yarn, the fabric manufacturer, will be interested in knowing the performance of the yarn from its physical and mechanical properties. Thus, one of the major concerns in the fabric-manufacturing process is to determine settings of design parameters that result in a satisfactory combination of quality characteristics. The fabric structure and properties are primarily influenced by fiber properties (length, fineness, etc.), spinning methods (ring, rotor, air jet, etc.), yarn parameters (count, twist, single and doubled, etc.), fabric structural parameters (warp and weft density, weave, etc.), and finally finishing treatments. The relationship between fabric structure and properties is complex and inherently nonlinear.

Despite many statistical and mathematical studies that predict and reveal specific properties of utilized yarn and fabric materials, a number of challenges continue to exist when evaluated in many perspectives, such as discovering complex relationships among material properties in data. Data mining plays an important role in discovering hidden patterns from fabric data and transforming it into knowledge. Therefore, the aim of the study is to uncover relationships between yarn parameters and fabric properties using an extended FP-Growth algorithm in association rule mining (Yildirim, Birant & Alpyildiz, 2017).

The novelty and main contributions of this study are as follows. First, it is the first study that proposes using ARM to identify the complex relationships between significant yarn parameters (i.e. hairiness, irregularity, diameter) and their effects upon fabric quality (i.e. pilling, abrasion). Second, it implements the FP-Growth algorithm in the textile domain for the first time. Third, it presents an extended FP-Growth algorithm that has the ability to find the different types of patterns such as closed, maximal, and top-k frequent itemsets. Fourth, this study also proposes two novel concepts, cf-item and mf-item, to identify significant items in data. In contrast to previous studies, the present study focuses on single-item based analysis because it can be used to solve different types of problems in different areas such as feature selection, the determination of important parameter values, and discretization.

An extended FP-Growth algorithm, which is proposed in this study, was executed on a real-world textile dataset with different support values to compare the different types of patterns and to demonstrate the applicability of ARM algorithms on yarn and fabric data. Experimental results show that proposed approach is very useful for discovering novel and interesting rules related to fabric quality.

6.1 Association Rule Mining (ARM)

ARM finds frequent patterns and interesting relationships among set of items in dataset. The most commonly used ARM algorithms are Apriori, FP-Growth, and Eclat. FP-Growth is a scalable algorithm that discovers large volumes of frequent itemsets efficiently using an extended prefix-tree structure (FP-tree).

6.1.1 Frequent Pattern Mining (FPM)

FPM discovers patterns from data that are more frequent than the specific threshold (Aggarwal, Bhuiyan & Hasan, 2014). An itemset I is called a frequent itemset (FI) if its support value, which is denoted by $\sigma(I)$, in the dataset D is greater than or equal to the user-defined minimum support, i.e. if $\sigma(I) \geq minsup$. Frequent pattern analysis on a large volume of data is a challenging process, since there is usually a large number of distinct single items, and their combinations may form a huge number of itemsets; thus, it requires a significant amount of time. In addition, the necessary storage capacity plays an important role. Due to the large amount of frequent itemsets that can be generated from a dataset, some studies (Zaki & Hsiao, 2002; Burdick, Calimlim & Gehrke, 2011) revealed the need for concise representations of all frequent itemsets such as closed and maximal frequent itemsets.

- **Closed Frequent Itemset (CFI)**

An itemset I is a closed itemset if there exists no itemset I' such that 1) $I \subset I'$ and 2) $\forall$ transaction T, $I \in T \longrightarrow I' \in T$.

CFI is a subset of frequent itemsets that has no superset with the same support, as shown in Equation (6.1) (Zaki & Hsiao, 2002).

$$\forall X \supset C : \sigma(X) < \sigma(C) \, \Lambda \, \sigma(X) \geq minsup \, \Lambda \, \sigma(C) \geq minsup, \tag{6.1}$$

where C is a closed frequent itemset whose supersets X have a strictly smaller support.

- **Maximal Frequent Itemset (MFI)**

A closed frequent itemset is an MFI if it is not a subset of any other frequent itemset, as shown in Equation (6.2) (Burdick et al., 2011).

$$\forall X \supset M : \sigma(X) < minsup \, \Lambda \, \sigma(M) \geq minsup, \tag{6.2}$$

where M is a maximal frequent itemset that has no frequent superset like X.

- **Top-k Frequent Itemsets (TFI)**

An itemset I is a TFI if I is the k most frequent itemset for a specified value k (Pietracaprina, Riondato, Upfal & Vandin, 2010; Cheung & Fu, 2004). Users will need to give the desired count of frequent itemsets, which is an easily understood parameter.

Given an itemset *I*, let *f*(*I*) be frequency of *I* in dataset *D*. Assume that the complete list of itemsets is denoted as δ, which is sorted in descending order according to their frequencies such that $\delta = \{I_1, I_2, I_3, ..., I_p\}$. For a given *k*, with $1 \leq k \leq p$ and $p = |\ \delta\ |$, $f(I_k)$ represents the frequency of *k*th itemset. Top-k frequent itemsets can be represented as shown in Equation (6.3).

$$TFI\ (k) = \{(I, f(I)) : I \in \delta, f(I) \geq f(I_k)\}, \tag{6.3}$$

where *TFI(k)* refers to the *k* most frequent itemsets in dataset *D*.

A closed itemset *I* is a top-k frequent closed itemset (TFCI) if there is no more than (*k – 1*) closed itemsets whose frequency is higher than that of *I*.

A maximal itemset is a top-k frequent maximal itemset (TFMI) if its frequency count is among the *k* highest frequencies of maximal itemsets, where *k* is the desired number of frequent maximal itemset.

6.1.2 Item Mining: cf-item and mf-item

Item mining is a part of traditional frequent pattern mining with the goal of identifying items that are essential for the analysis. It focuses on discovering frequent items whose length is equal to 1. In the subject of item mining, this study proposes two novel concepts, cf-item and mf-item, to distinguish the types of items in the dataset. Zaki & Hsiao (2002) indicated closed and maximal itemsets that are subsets of frequent itemsets and include more than one item. However, the present study introduces cf-item and mf-item concepts because single-item identification is particularly important where there is a need to investigate the significances of the attributes.

- **CF-item**

Cf-item is a single item that is both closed and frequent. Cf-items can be determined by finding the subset of closed frequent patterns whose lengths are equal to 1.

Definition 1. Let $\mathcal{I}$ = {i_1, i_2, ..., i_m} be a set of *m* items. Let *C* be closed and frequent, and ||C|| denotes its length. The item *C* is said to be cf-item if it has no superset with the same support value, its support count is higher than minimum support denoted by σ(*C*) ≥ *minsup*, *C* ∈ $\mathcal{I}$ and its length is equal to 1, so ||*C*|| = 1. Cf-item is defined in the Equation (6.4).

$$\forall X \supset C : \sigma(X) < \sigma(C) \land \|C\| = 1 \land \sigma(X) \geq minsup \land \sigma(C) \geq minsup, \qquad (6.4)$$

where *C* is a cf-item whose supersets *X* have a strictly smaller support and the maximum length is limited to 1.

- **MF-item**

Mf-item is a single item such that it is both maximal and frequent. Mf-items are the subset of maximal frequent patterns whose lengths are equal to 1. This study is the first study that proposes mf-item.

Definition 2. Let $\mathcal{I}$ = {i_1, i_2, ..., i_m} be a set of *m* items. Let *M* be maximal and frequent, and ||*M*|| denotes its length. The item *M* is said to be mf-item if it has no frequent superset, its support count is higher than minimum support denoted by σ(*M*) ≥ *minsup*, *M* ∈ $\mathcal{I}$, and its length is equal to 1 and so ||*M*|| = 1. Mf-item is defined in Equation (6.5).

$$\forall X \supset M : \sigma(X) < minsup \land \|M\| = 1 \land \sigma(M) \geq minsup, \qquad (6.5)$$

where M is an mf-item that has no frequent superset like X and the maximum length is limited to 1.

Figure 6.1 shows the relationship among the different types of frequent patterns. The general relationship is MFI ⊆ CFI ⊆ FI. Cf-items, mf-items, and top-k patterns are located in the related areas as a subset. In addition, all MFIs are closed because they have no frequent superset and so they cannot have the same support count as their supersets. Top-k patterns are located in the related areas as a subset.

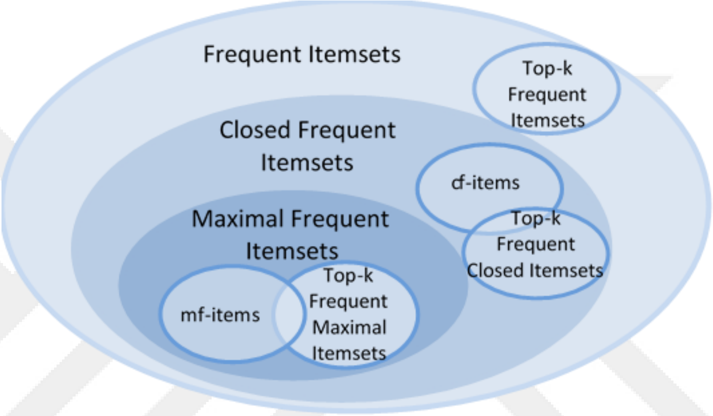

Frequent Itemsets
Top-k Frequent Itemsets
Closed Frequent Itemsets
cf-items
Top-k Frequent Closed Itemsets
Maximal Frequent Itemsets
mf-items
Top-k Frequent Maximal Itemsets

Figure 6.1 The relationship among the different types of frequent patterns

Table 6.1 shows an example of obtaining closed, maximal, top-k frequent itemsets, cf-item, and mf-item from a sample dataset. The example dataset contains five items $\mathcal{I}$ = {A, B, C, D, E} and consists of six transactions, which are uniquely identified by an ID. Minimum support threshold was chosen as 50% and so items or itemsets that occur in the dataset three or more times will be selected as frequent.

Table 6.1 A sample dataset and different types of frequent patterns obtained from it

ID	**Transactions**	**FI**	**CFI**	**MFI**	**Top-4 FI**	**Items**
1	A, C, D	1-itemset	1-itemset	1-itemset	1-itemset	cf-item
2	B, C, E	{B} : 4	{C} : 4	{D} : 3	{B} : 4	{C} : 4
3	A, B, C, E	{C} : 4	{D} : 3	3-itemset	{C} : 4	{D} : 3
4	D, B, E, C	{D} : 3	2-itemset	{B, C, E} : 3	{E} : 4	mf-item
5	D	{E} : 4	{B, E} : 4		2-itemset	{D} : 3

Table 6.2 continues

6	B, E	2-itemset {B, C} : 3 {B, E} : 4 {C, E} : 3 3-itemset {B, C, E} : 3	3-itemset {B, C, E} : 3	{B, E} : 4

Step 1: Frequent itemsets

To find frequent itemsets, all transactions are traversed and the support values of items are evaluated first. Items whose support values are greater than or equal to the minimum support are selected as frequent. In this example, all items, except item A, are frequent because they appear in more than three transactions. In the next iteration, candidate 2-itemsets are generated using only the frequent 1-itemsets and evaluated by counting their supports. For example, the candidate {B, D} is found to be infrequent after computing their support values. Next, the algorithm will iteratively generate new candidate k-itemsets using the frequent (k – 1)-itemsets found in the previous iteration.

Step 2: Closed frequent itemsets

Itemset {C}, with support 4, is a closed frequent itemset because its supersets ({B, C} : 3, {C, E} : 3, and {B, C, E} : 3) have a smaller support count (3). However, itemset {B, C} is not a closed frequent itemset because its superset {B, C, E} has the same support of 3. For the same reason, itemsets {B}, {E}, and {C, E} are also not closed frequent itemsets.

Step 3: Maximal frequent itemsets

Itemsets {D} and {B, C, E} are maximal frequent itemsets because none of their supersets are frequent. In contrast, itemset {B, E} is nonmaximal because one of its immediate supersets {B, C, E} is frequent. For the same reason, itemset {C} is also not a maximal frequent itemset.

Step 4: Top-k frequent itemsets

When *k* parameter is assigned to 4, the *k* most frequent itemsets in this example could be {B}, {C}, {E}, and {B, E} without any constraints.

Step 5: Top-k frequent closed itemsets

The frequency counts of the itemsets {C} and {B, E} are among the *k = 2* highest frequencies of closed itemsets.

Step 6: Top-k frequent maximal itemsets

When *k = 2* is the desired number of frequent maximal itemsets, {D} and {B, C, E} are selected in this example.

Step 7: Cf-item

The closed frequent itemsets whose lengths are equal to 1 are {C} and {D}. Cf-items in data can be found if there is a need to investigate the significances of the items.

Step 8: Mf-item

The item *D* is both closed and maximal, because none of the supersets of this item are frequent. Determining a mf-item is important when performing single item-based data analysis.

6.1.3 Advantages of Proposed Method

The novel concepts (cf-item, mf-item) and the extended FP-Growth algorithm can be used to solve different types of problems in different areas, such as feature selection, the determination of important parameter values, and discretization.

- **Feature selection**

Feature selection, one of the important data-preprocessing stages, is performed to choose a subset of relevant items in the dataset. This process decreases the number of features and increases the accuracy of the categorization. Cf-item and mf-item can be useful when identifying the most frequent single features in the data set.

- **Determination of parameter values**

Cf-item and mf-item specify the significant of the parameters and so they can be used to determine the important of items with their values. For example, a cf-item discovered from the dataset {YarnHairiness = (3–4]} and interpreted as *Yarn Hairness* is one of the significant parameters, with a range between 3 and 4.

- **Discretization**

Discretization converts numeric values of attributes to nominal/ordinal values by using a categorization strategy. The key point of discretization is the determination of a set of optimal split points and intervals. The presented method in this study discovers the patterns containing same attributes with different range of values. Suppose cf-items {YarnHairinessH = (3–5]} and YarnHairinessH = (8–10]} were obtained when the algorithm was executed. According to these patterns, the optimal split points can be found to define the interval boundaries in the discretization process.

6.2 Extended FP-Growth Algorithm

Pattern-mining algorithms have a wide range of applications, such as cross-marketing, website click stream analysis, and biomedical applications. The most commonly used algorithms in these applications are Apriori, FP-growth, and Eclat. FP-Growth stands for “frequent pattern growth” and was proposed for discovering sets of frequent patterns using an extended prefix-tree structure named FP-tree. FP-Growth was developed as an alternative for the Apriori algorithm to handle large volumes of frequent itemsets with high performance utilizing a divide-and-conquer strategy. The algorithm consists of two steps: building an FP-tree and obtaining frequent itemsets from this tree. An FP-tree has a compact prefix tree structure that stores and represents the transaction database horizontally and vertically. While horizontal representation of the tree indicates a prefix tree of transactions, vertical representation shows links between the prefix tree branches. The root of the tree is labelled as “null” and each node holds an item’s name, an item’s transaction count, and node links.

A simple FP-tree construction example with minimum support 50% is illustrated in Figure 6.2 by considering a sample dataset given in Table 6.2. To construct the tree, the algorithm passes over the dataset two times. First, the support count of each item is calculated and frequent items are sorted in decreasing order. At the next pass, each transaction is read and mapped to a path in the tree. Paths in the FP-tree overlap when different transactions share common items. Cf-items and mf-items are highlighted in gray on the tree.

Table 6.3 Transactions in a sample dataset and their frequent items

ID	Items	1-itemset	Ordered frequent items
1	A, C, D	{B} : 4	C, D
2	B, C, E	{C} : 4	B, C, E
3	A, B, C, E	{E} : 4	B, C, E
4	D, B, E, C	{D} : 3	B, C, E, D

Table 6.4 continues

5	D	D
6	B, E	B, E

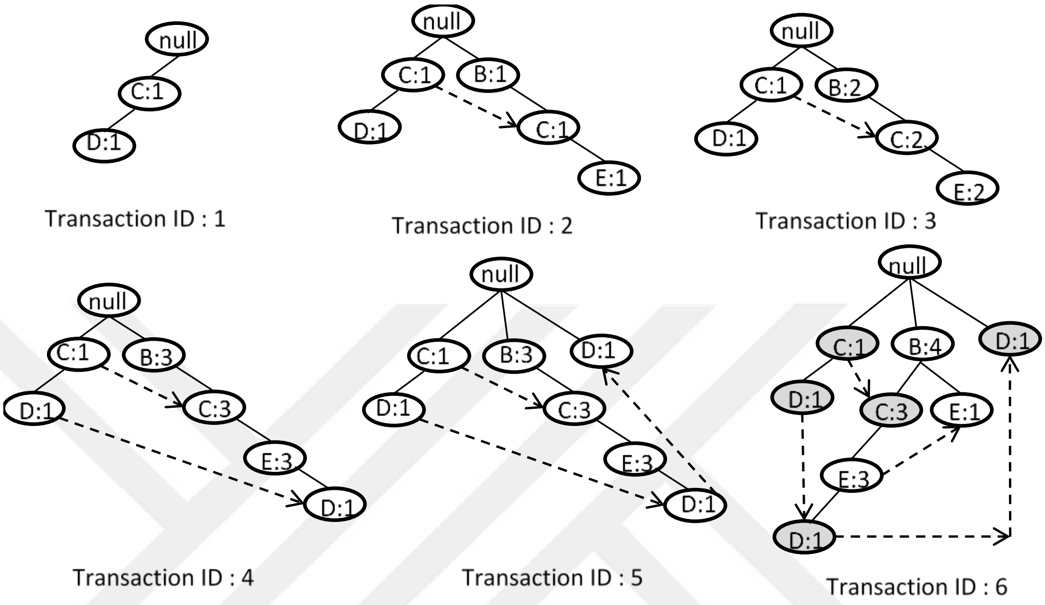


Figure 6.2 Illustrated FP-tree, cf-items and mf-items in the tree

This study presents an extended version of the FP-Growth algorithm that has the ability to find the different types of patterns, such as frequent, closed, maximal, top-k frequent, top-k closed, top-k maximal, cf-item, and mf-item. The pseudocode of the extended FP-Growth algorithm is presented in Figure 6.3. The algorithm first computes a list of frequent items sorted by frequency in descending order (*F[ ]*). After that, the FP-Tree is constructed by scanning each transaction in the dataset. Then the FP-Growth method (Li, Wang, D. M. Zhang & Chang, 2008) starts to mine the FP-tree for each frequent item by recursively building conditional trees. The algorithm also mines closed and maximal patterns on frequent itemsets. The lengths of each itemset in CFI and MFI are also controlled to determine cf-items and mf-items, respectively. In addition, it also determines the *k* most frequent, closed, and maximal itemsets. The time complexity of computing the list *F[ ]* is *O(n)*, where *n* is the number of the transactions in the dataset. However, the computational cost of procedure *Growth()* is at least polynomial.

```
Algorithm Extended_FP-Growth (D, minsup)
  Inputs:  D: dataset, minsup: minimum support, $\mathcal{I}$: set of items in D,
           k: the desired number of patterns (top-k)
  Outputs:  FI: frequent itemsets, CFI: closed frequent itemsets,
            MFI: maximal frequent itemsets, TFI: top-k frequent itemsets,
            TFCI: top-k frequent closed itemsets,
            TFMI: top-k frequent maximal itemsets, cf-item, mf-item
begin
    Define frequency list: F[ ] = {}
    foreach transaction $T_i$ in D
        foreach item $a_j$ in $T_i$
            F[$a_j$]++
    Sort F[ ]
    Define and clear the root of FP-tree: r
    foreach Transaction $T_i$ in D
        Make $T_i$ ordered according to F[ ]
        Call ConstructTree($T_i$, r)
    foreach item $a_i$ in $\mathcal{I}$
        if F[$a_i$] $\geq$ minsup
            FI = FI $\cup$ Growth(r, $a_i$, minsup)
    foreach frequent itemset $f_i$ in FI
        if $f_i$ has no superset in FI with same support
            CFI = CFI $\cup$ $f_i$
            if $|f_i|$ = 1
                cf-item = cf-item $\cup$ $f_i$
            if $f_i$ has no superset in FI
                MFI = MFI $\cup$ $f_i$
                if $|f_i|$ = 1
                    mf-item = mf-item $\cup$ $f_i$
    for i = 1 to k
        TFI = TFI $\cup$ FI.sort[i]
        TFCI = TFCI $\cup$ CFI.sort[i]
        TFMI = TFMI $\cup$ MFI.sort[i]
end
```

Figure 6.3 Pseudocode of extended FP-Growth algorithm

6.3 Experimental Study

In this study, the extended FP-growth algorithm was applied on a real-world fabric dataset (Yaşar, 2015) to discover the relationships between selected yarn parameters with selected fabric properties.

The algorithm was executed on the dataset with varying support threshold values to compare the different types of patterns. Most relevant yarn parameters with fabric properties that were obtained as the outcomes of this algorithm are explained and the number of frequent items/itemsets are shown with the help of charts. In this experimental study, cf-item and mf-item were utilized for the first time to perform single item-based data analysis.

6.3.1 Dataset Description

The dataset considered in this study contains selected yarn parameters (yarn manufacturing method, elongation at break, irregularity, hairiness, bending rigidity, and capillary properties) and selected fabric properties (pilling, abrasion resistance, and bending rigidity) that were experimentally obtained in a previous study (Yaşar, 2015). The raw dataset contains 1800 records and consists of fifteen attributes, including nominal and numerical values: nine of them are yarn parameters and the rest of them are fabric features. The statistical details of the dataset are presented in Table 6.3.

Table 6.5 Statistical details of the dataset and the categories in the attributes

Attributes	**# of records**	**Min value**	**Max value**	**Mean**	**Std. Dev.**	**Categories**
Yarn tenacity	100	16.98	21.87	18.97	1.6	[15.5–17.5], (17.5–19.5], (19.5–21.5], (21.5–23.5]
Yarn elongation at break (%)	100	14.54	18.47	16.66	0.99	[13.5–15.5], (15.5–17.5], (17.5–19.5]
Yarn irregularity	100	7.96	9.6	8.76	0.54	[7–9], (9–11]
Yarn hairiness (H)	100	3.63	5.5	4.16	0.69	[3–4], (4–5], (5–6]
Yarn hairiness (S3)	100	8	1947	594.08	784.27	[3–9], (9–27], (27–39], (39–462], (462–1419], (1419–2376]
Yarn capillary	200	2	4.3	3.18	0.56	[2–3], (3–4], (4–5]
Yarn bending rigidity	600	3.34	4.89	4.03	0.42	[3–3.5], (3.5–4], (4–4.5], (4.5–5]
Yarn diameter	200	0.51	0.7	0.58	0.06	[0.5–0.56], (0.56–0.63], (0.63–0.7]
Abrasion resistance	48	16.3	25.1	21.46	2.28	[16–18.75], (18.75–20.75], (20.75–23.125], (23.125–25.5]
Pilling resistance	36	3	5	4.17	0.49	[3–3.5], (3.5–4], (4–4.5], (4.5–5]
Wrinkle resistance	72	100.5	143.25	121.86	13.3	[100–111], (111–122], (122–133], (133–144]
Fabric bending rigidity	64	0.91	2.59	1.55	0.62	[0.5–1.3], (1.3–2.2], (2.2–3]

Table 6.6 continues

Capillary warp direction	40	1.6	3.6	2.44	0.55	[1–1.75], (1.75–2.5], (2.5–3.25], (3.25–4]
Capillary weft direction	40	1.2	3.4	2.1	0.61	[1–1.75], (1.75–2.5], (2.5–3.25], (3.25–4]
Yarn manufacturing method	MVS, RAJ, SIRO, RING					

ARM algorithms require categorical data; they cannot directly deal with numeric attributes. For this reason, in this study, numeric attributes were discretized into intervals by finding a set of significant split points of distribution changes that define the interval boundaries of the discretization. The discretization process is divided into different categories from different perspectives, such as supervised or unsupervised, top-down or bottom-up, static or dynamic, local or global, nominal or ordinal, univariate or multivariate, direct or iterative (Liu, Hussain, Tan & Dash, 2002). The discretization technique applied in this study was unsupervised, top-down, static, global, nominal, univariate, and direct. The split points were obtained by using both expert techniques approved by the textile community and an equal width binning method that divides numerical values into equal *n* intervals. Due to irregular distribution, three attributes (hairinessS3, yarn diameter, and abrasion resistance) were categorized by evaluating boundaries and binning widths by using a frequency table, while the rest of the numeric attributes in the dataset were discretized by equal width binning method. For example, attribute tenacity was discretized into four categories as follows: [15.5–17.5], (17.5–19.5], (19.5–21.5], (21.5–23.5] and each numeric value in this attribute was mapped in a category according to the range of value. The last column in Table 3 shows the categories of attributes with their interval values that were selected during the discretization process.

6.3.2 Comparison of Different Types of Patterns

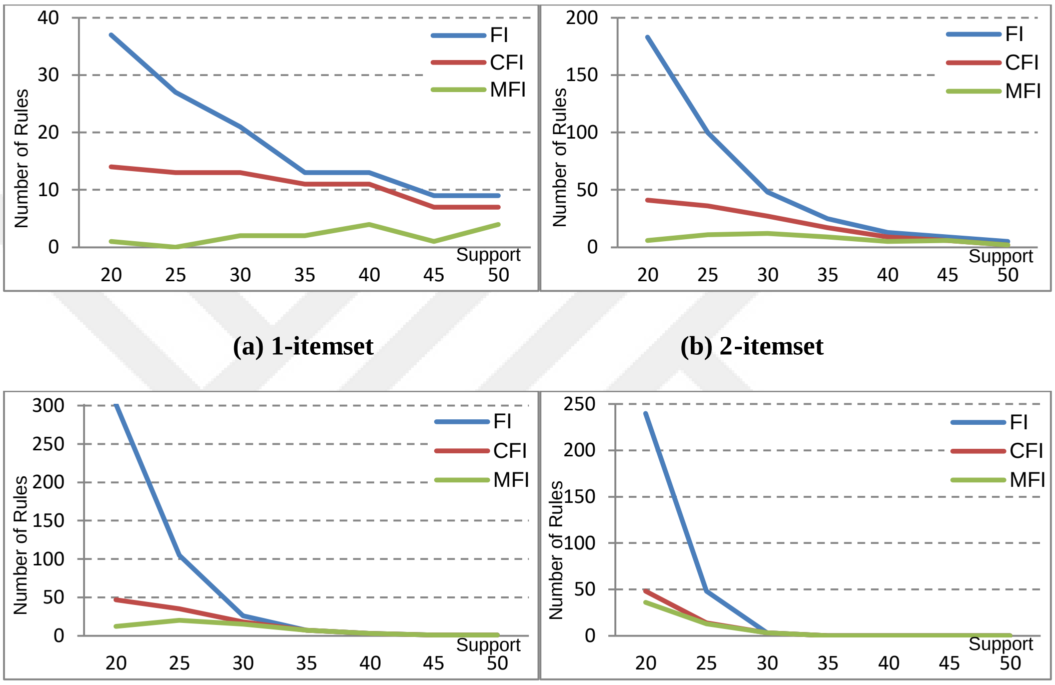
An extended FP-Growth algorithm was executed on the dataset with varying support thresholds from 10 to 60 in increments of 5. Only those items with support values greater than or equal to the threshold level were selected as frequent patterns and others were discarded. Table 6.4 shows the numbers of different types of patterns (i.e. frequent, closed, maximal, cf-item, and mf-item) separately varying from 1-itemset to 5-itemset. Results show that the number of frequent itemsets produced from dataset is large when the minimum support level is set to low, i.e. the number of 5-itemset patterns is 7166 when *minsup* = 10%. However, the algorithm produces a reasonable number of closed and maximal frequent itemsets, i.e. the number of 5-itemset closed patterns is 57 when *minsup* = 10%. Thus, it is possible to compress the collection of frequent itemsets in a more manageable size. In addition, it is also possible to determine the significance of the attributes with cf-item and mf-item concepts.

Table 6.7 The numbers of different types of patterns with different support thresholds

	Frequent itemsets					**Cf-item**	**Closed frequent itemsets**				**Mf-item**	**Maximal frequent itemsets**			
Support	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
10	48	502	2084	4772	7166	14	41	58	67	57	0	0	1	7	12
15	41	306	805	1084	891	14	41	57	60	45	0	0	10	12	35
20	37	183	302	240	113	14	41	47	48	10	1	6	12	36	9
25	27	100	105	48	13	13	36	35	14	1	0	11	20	13	1
30	21	48	26	3	0	13	27	18	3	0	2	12	15	3	0
35	13	25	7	0	0	11	17	7	0	0	2	9	7	0	0
40	13	13	3	0	0	11	9	3	0	0	4	5	3	0	0
45	9	9	1	0	0	7	6	1	0	0	1	6	1	0	0
50	9	5	1	0	0	7	2	1	0	0	4	2	1	0	0
55	6	1	0	0	0	6	1	0	0	0	4	1	0	0	0
60	3	1	0	0	0	3	1	0	0	0	1	1	0	0	0

Figure 6.4 shows the comparative results of the numbers of frequent, closed and maximal itemsets for support threshold levels ranging from 20 to 50 in increments of 5. The figure presents the results for different lengths of patterns, i.e. 1-itemset, 2-itemset, 3-itemset, and 4-itemset. Results show that when the minimum support value decreases, the number of FI patterns increases almost exponentially. This

means that a large amount of frequent itemset patterns are generated when the algorithm is executed with small support values. However, the algorithm produces a reasonable number of CFI and MFI patterns. When the minimum support value and size of the itemset increase (i.e. *minsup* = 25% and 4-itemset), the differences between the number of FIs, CFIs, and MFIs decrease. For this reason, the type of the pattern is not critical in the case of large parameter values.



(a) 1-itemset

(b) 2-itemset

(c) 3-itemset

(d) 4-itemset

Figure 6.4 Comparison of the numbers of FI, CFI and MFI patterns

The graph in Figure 6.5 shows the number of closed and maximal frequent items generated by the extended FP-Growth algorithm for varying support thresholds from 30 to 60 in increments of 5. From this graph, it is possible to see that the number of cf-items is always greater than or equal to the number of mf-item patterns because of the relationship MFI ⊆ CFI. The obtained results also show that while minimum support value increases, the number of cf-items decreases about linearly. However, the numbers of obtained mf-item patterns are irregular because the supersets of the patterns changes according to support threshold levels. When the support value increases, the differences between the number of CFI and MFI decrease. For this reason, the type of the pattern is not as critical as in the case of large support values.

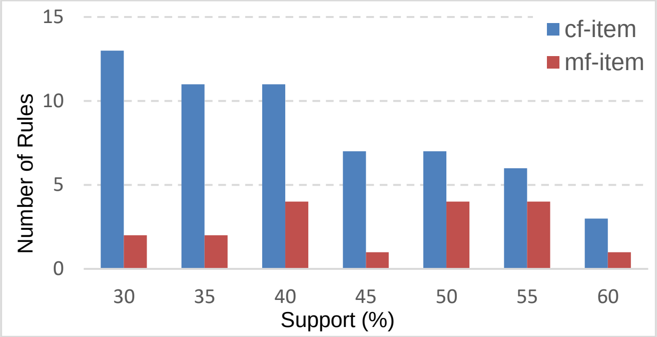


Figure 6.5 Comparison of the numbers of cf-item and mf-item

Figure 6.6 shows the lowest and highest support values of top-k patterns with *k* ranging from 1 to 10. It compares TFI, TFCI, and TFMI patterns with 2-itemset when the minimum support threshold is 35%. According to the results, the lowest and highest support values of TFI and TFCI patterns are generally close to each other, but the support values of TFMI patterns in top-k lists are lower than them. To select interesting relationships among data and determining frequent itemsets, the widely used parameter is minimum support value. However, specifying optimal minimum support threshold is a difficult and time-consuming task for users because selecting the threshold is somewhat unstable. For this reason, it is also possible to discover top-k frequent patterns without the minimum support specification. In this case, a specified itemset-length can also be used as a threshold to focus on the desired pattern size.

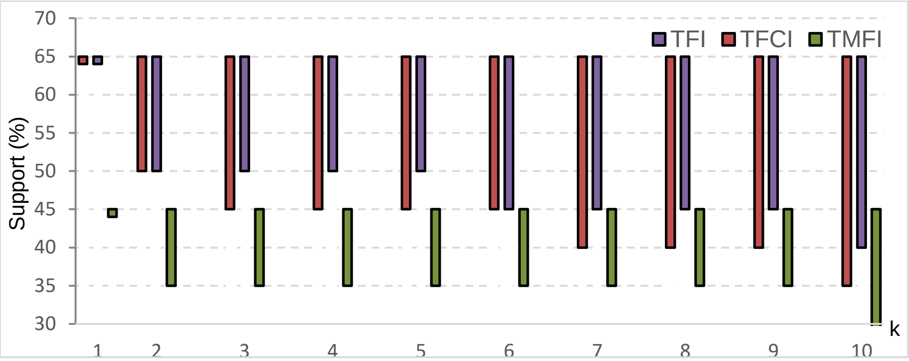


Figure 6.6 The lowest and highest support values of top-k patterns (2-itemset)

6.3.3 Association Rules

Table 6.5 shows some rules discovered by the algorithm. According to the results, the patterns {YarnElongationAtBreak = (15.5–17.5]} and {YarnIrregularity = [7–9]} are the most frequent 1-itemsets, which indicates that they are the most influential parameters among the ones considered. Following these two parameters, the important attributes and their range values were determined as cf-items and mf-items and given as, for example, {YarnHairinessH = (3–4]} and {YarnCapillary = (3–4]}, respectively. The results also express the relationships among yarn manufacturing methods (msv, raj, siro, ring), yarn parameters (i.e. hairiness, capillary, diameter), and fabric properties (i.e. pilling, wrinkle, abrasion), as some are indicated in Table 6.5 by example patterns of two or more itemsets. For 2-itemsets, the patterns {YarnHairinessH = (3–4], FabricPilling = (3.5–4]} and {FabricAbrasion = (20.75–23.125], YarnHairinessH = (3– 4]} indicate that when the yarn hairiness index *H* values are between 3 and 4, the fabric pilling performance can be expected to be from 3.5 to 4 and the fiber loss values due to abrasion of the fabric will lie between 20.75 and 23.12 mg. When the yarn irregularity values lie between 7 and 9 and yarn diameter is between 0.5 and 0.56 mm the fabric bending rigidity is expected to be between 0.5 and 1.3 with the rule {YarnIrregularity = [7–9], YarnDiameter = [0.5–0.56], FabricBendingRigidity = [0.5–1.3]}. If the yarns are manufactured by the Rieter Air Jet (RAJ) method and have elongation at break values between 15.75% and 17.75% and yarn hairiness H index between 3 and 4, then the pilling performance of the fabric from such yarn will be between 3.5 and 4, as indicated by the pattern {YarnManufacturing = RAJ, YarnElongationAtBreak =

(15.5–17.5], YarnHairinessH = (3–4], FabricPilling = (3.5–4]}. For the textile industry, where there are many parameters affecting the end-product performance, the relationships between the parameters of the yarn and the fabric is important to understand the product behavior so that the end-product can be shaped according to the customer. With these example patterns, the algorithm proves to be able to derive the relationships between the yarn and fabric parameters and also the significant values for the important parameters could be stated.

Table 6.8 Example of rules discovered by association rule mining

Length	Pattern	Support (%)	Minsup (%)	Type
1-Itemsets	{YarnElongationAtBreak = (15.5–17.5]}	75	55	TFCI
	{YarnIrregularity = [7–9]}	75		TFCI
	{YarnHairinessH = (3–4]}	70		cf-item
	{YarnBendingRigidity = (3.5–4]}	55		TFMI
	{YarnCapillary = (3–4]}	55		mf-item
2-Itemsets	{YarnHairinessH = (3–4], YarnIrregularity = [7–9]}	45	30	CFI
	{YarnHairinessH = (3–4], FabricPilling = (3.5–4]}	35		FI
	{YarnElongationAtBreak = (15.5–17.5], FabriWrinkle = [100–111]}	30		FI
	{YarnTenacity = (19.5–21.5], FabricBendingRigidity = [0.5–1.3]}	30		FI
	{YarnHairinessH = (3–4], FabricAbrasion = (20.75–23.125]}	30		MFI
3-Itemsets	{YarnIrregularity = [7–9], YarnDiameter = [0.5–0.56], FabricBendingRigidity = [0.5–1.3]}	50	30	MFI
	{YarnTenacity = (17.5–19.5], YarnBendingRigidity = (3.5–4], YarnIrregularity = [7–9]}	30		CFI
	{YarnBendingRigidity = (3.5–4], FabricCapillaryWarp = (1.75–2.5], YarnElongationAtBreak = (15.5–17.5]}	35		FI

Table 6.9 continues

4-Itemsets	{YarnManufacturing = SIRO, YarnIrregularity = [7–9], YarnDiameter = [0.5–0.56], FabricBendingRigidity = [0.5–1.3]}	25	20	CFI
	{YarnManufacturing = RAJ, YarnElongationAtBreak = (15.5–17.5], YarnHairinessH = (3–4], FabricPilling = (3.5–4]}	25		FI
	{YarnManufacturing = MVS, FabricCapillaryWarp = (1.75–2.5], YarnTenacity = (17.5–19.5], FabricBendingRigidity = (1.3–2.2]}	20		FI
	{YarnManufacturing = RING, YarnHairinessH = (5–6], YarnElongationAtBreak = (17.5–19.5], FabricWrinkle = (133–144]}	20		FI
	{YarnDiameter = [0.5–0.56], FabricAbrasion = (20.75–23.125] YarnIrregularity = [7–9], FabricBendingRigidity = [0.5–1.3]}	20		MFI

CHAPTER SEVEN
CONCLUSION AND FUTURE WORK

7.1 Conclusion

Discovering previously unknown and potentially useful knowledge from raw data and making right decisions based on this knowledge is a major need for textile engineering as well as in many areas, such as healthcare, finance, and marketing. The present textile studies in the literature that implements classical mathematical and statistical models to analyze raw data can be inadequate to derive complex relations within textile datasets. Because of this reason, DM-based information technology applications have been recently preferred in the textile industry during the past decade. The "related work" part of this document presents a survey on DM methods specifically designed for textile applications and also describes some experimental works in the literature. The second chapter demonstrates how clustering and classification techniques can be applied in textile sector to deal with a problem.

Data engineering is a sub-branch of data science which prepares raw data to suitable for analyzing process and implementing DM techniques on to discover potentially useful knowledge. The aim of this thesis is application of novel DM techniques on raw textile data to obtain valuable knowledge and making right decisions to increase quality and productivity. In this thesis, five novel DM studies including classification, clustering, and association rule mining methods were performed on well-known benchmark textile datasets.

As a summary, in this thesis, (i) important parameters of knitted structures for stab performance were determined and ensemble learning algorithms were applied on textile sector for the first time, (ii) an ensemble learning approach that combines multiple neural networks with different parameter values was presented to improve prediction performance in textile sector, (iii) a novel convolutional neural network (CNN) architecture was developed to classify fashion products (iv) a novel hierarchical clustering approach, named *k*-Linkage was proposed and (v) an extended

FP-Growth algorithm was introduced as a novel concept for association rule mining approach.

Experiments were performed for each study in the thesis to demonstrate the performance of the proposed methods. In each experiment, the proposed approaches were executed on real-world benchmark textile data and compared with the present algorithms in terms of different evaluation measures. According to the results, the proposed methods in this thesis produce more accurate results and so show better performance than the conventional solutions. When the experimental results are considered in general, we recommend the implementation of data science and engineering techniques in textile sector because it provides higher data processing ability than the classical mathematical and statistical models.

7.2 Future Work

In future work, ARM can be suggested for applying more frequently in textile industry, besides the classification and clustering methods. Similarly, the negative association rule could be addressed in future research to be able to describe the occurrences of some textile properties characterized by the absence of others. For example, it would be interesting to find out which factors are relatively and absolutely relevant and irrelevant that they may arise frequently or infrequently. Mining sequential patterns in textile data could also be one of the future research areas. There could be scope for research in determining time-related behavior in textile data.

An additional aspect related to current clustering studies in textile industry is that the K-Means++ algorithm could be addressed in future research to deal with the challenges of K-Means in a broad sense, i.e., to improve both the speed and the accuracy of K-Means. Even though partitioning and hierarchical clustering approaches have been generally proposed in textile studies, it is also possible to use density-based clustering techniques (i.e., DBSCAN algorithm) for future work because, DBSCAN is capable of forming arbitrarily shaped clusters and dealing with noise in the data.

Text mining, web mining, and process mining have been used in many engineering fields. However, there is very limited usage of them in textile industry. Future research can focus on: (i) *text mining* such as sentiment analysis to determine positive or negative textile related contents, (ii) *web mining* for building effective textile marketing strategies such as personalized recommendation, and (iii) *process mining* to improve performance of textile processes while reducing costs.

In our opinion small textile data are an important challenge that could be addressed in future research. In this case, different strategies (i.e., reducing the number of features) should be investigated.

Recently, several ontologies have been developed for the textile, fashion, and clothing domains. We believe that the future DM-based textile studies will be supported by the ontologies to extract semantic relationship, to improve accuracy, and to develop better decision support systems.

REFERENCES

Abakar, K. A. A., & Yu, C. (2013). Application of genetic algorithm for feature selection in optimisation of SVMR model for prediction of yarn tenacity. *Fibres and Textiles in Eastern Europe, 102* (6), 95-99.

Abakar, K. A. A., & Yu, C. (2014). Performance of SVM based on PUK kernel in comparison to SVM based on RBF kernel in prediction of yarn tenacity. *Indian Journal of Fibre and Textile Research, 39*, 55-59.

Agarwal, G., Koehl, L., & Perwuelz, A. (2010). The Influence of Constructional Properties of Knitted Fabrics on Cationic Softener Pick Up and Deposition Uniformity. *Textile Research Journal, 80* (14), 1432–1441.

Agarwal, G., Koehl, L., Perwuelz, A., & Lee, K. S. (2011). Interaction of textile parameters, wash-ageing and fabric softener with mechanical properties of knitted fabrics and correlation with textile-hand. I. Interaction of textile parameters with laundry process. *Fibers and Polymers, 12* (5), 670–678.

Aggarwal, C. C., Bhuiyan, M. A., & Hasan, M. A. (2014). Frequent pattern mining algorithms: A survey. In *Frequent Pattern Mining* (19-64). Switzerland: Springer International Publishing.

Ahmad, G. G. (2016). Using artificial neural networks with graphical user interface to predict the strength of carded cotton yarns. *Journal of the Textile Institute, 107* (3), 386-394.

Akyol, U., Tufekci, P., Kahveci, K., & Cihan, A. (2014). A model for predicting drying time period of wool yarn bobbins using computational intelligence techniques. *Textile Research Journal, 85* (13), 1367-1380.

Alessandro, A., Corani, G., Mauá, D., & Gabaglio, S. (2013). An ensemble of Bayesian networks for multilabel classification. *IJCAI International Joint Conference on Artificial Intelligence*, 1220-1225.

Alpaydin, E. (2010). *Introduction to Machine Learning* (2nd ed.). London: The MIT Press.

Alpyildiz, T., Rochery, M., Kurbak, A., & Flambard, X. (2011). Stab and cut resistance of knitted structures: A comparative study. *Textile Research Journal, 81* (2), 205-214.

Ammor, O., Lachkar, A., Slaoui, K., & Rais, N. (2008). Optimisation of pattern recognition in textile field. *Journal of the Textile Institute, 102* (1), 227-233.

Bahadir, S. K., Kalaoglu, F., Jeysnik, S., Eryuruk, S. H., & Saricam, C. (2015). Use of Artificial Neural Networks for Modelling the Drape Behaviour of Woollen Fabrics Treated with Dry Finishing Processes. *Fibres & Textiles in Eastern Europe, 110* (2), 90-99.

Behera, B. K., & Karthikeyan, B. (2006). Artificial neural network-embedded expert system for the design of canopy fabrics. *Journal of Industrial Textiles, 36* (2), 111-123.

Bhargava, & Sharma. (2013). Decision Tree Analysis on J48 Algorithm for Data Mining. *International Journal of Advanced Research in Decision Tree Analysis on J48 Algorithm for Data Mining, 3* (6), 1114-1119.

Budka, M., & Gabrys, B. (2010). Ridge regression ensemble for toxicity prediction. *Procedia Computer Science, 1* (1), 193-201.

Burdick, D., Calimlim, M., & Gehrke, J. (2001). MAFIA: A maximal frequent itemset algorithm for transactional databases. *Proceedings 17th International Conference on Data Engineering*, 443-452.

Che, D., Liu, Q., Rasheed, K., & Tao, X. (2011). Decision tree and ensemble learning algorithms with their applications in bioinformatics. In *Advances in Experimental Medicine and Biology* (191-199). New York: Springer.

Chen, Z., Zhou, S., & Luo, J. (2017). A robust ant colony optimization for continuous functions. *Expert Systems with Applications, 81*, 309-320.

Cheung, Y. L., & Fu, A. W. C. (2004). Mining frequent itemsets without support threshold: With and without item constraints. *IEEE Transactions on Knowledge and Data Engineering, 16* (9), 1052-1069.

Ciarapcia, F. E., Sanctis, I. D., Resta, B., Dotti, S., Gaiardelli, P., Bandinelli, R., Fani, V., & Rinaldi, R. (2017). Integrating sustainability in the fashion system using association rules. In *Lecture Notes in Electrical Engineering* (239-250), New York: Springer.

Cichosz, P. (2015). Naïve Bayes classifier. In *Data Mining Algorithms: Explained Using R* (1st ed.) (118-134). New Jersey: Wiley.

Croft, J., & Longhurst, D. (2007). HOSDB Body Armour Standards for UK Police Part 3: Knife and Spike Resistance. *Home Office Scientific Development Branch*.

Derntl, A., & Plant, C. (2016). Clustering techniques for neuroimaging applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 6* (1), 22-36.

Dietterich, T. G. (2000). Experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning, 40* (2), 139-157.

Eldessouki, M., Hassan, M., Qashqary, K., & Shady, E. (2014). Application of Principal Component Analysis to Boost The Performance of The Automated Fabric Fault Detector And Classifier. *Fibres and Textiles in Eastern Europe, 4* (106), 51-57.

Esfandarani, M. S., & Shahrabi, J. (2012). Developing a new suit sizing system using data optimization techniques. *International Journal of Clothing Science, 24* (1), 27-35.

Farooq, A., & Cherif, C. (2008). Use of Artificial Neural Networks for Determining the Leveling Action Point at the Auto-leveling Draw Frame. *Textile Research Journal, 78* (6), 502-509.

Frank, E., Hall, M. A., & Written, I. H. (2016). *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"* (4th ed.). San Francisco: Morgan Kaufmann.

Furferi, R., Governi, L., & Volpe, Y. (2012). Modelling and simulation of an innovative fabric coating process using artificial neural networks. *Textile Research Journal, 82* (12), 1282-1294.

Gagolewski, M., Bartoszuk, M., & Cena, A. (2016). Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm. *Information Sciences, 363,* 8-23.

Ghosh, A., Guha, T., & Bhar, R. B. (2015). Identification of handloom and powerloom fabrics using proximal support vector machines. *Indian Journal of Fibre and Textile Research, 40* (1), 87-93.

Golob, D., Osterman, D. P., & Zupan, J. (2008). Determination of pigment combinations for textile printing using artificial neural networks. *Fibres and Textiles in Eastern Europe*, *68* (3), 93-98.

Haghighat, E., Johari, M. S., Etrati, S. M., & Tehran, M. A. (2012). Study of the hairiness of polyester-viscose blended yarns. Part III - Predicting yarn hairiness using an artificial neural network. *Fibres and Textiles in Eastern Europe, 90* (1), 33-38.

Hamrouni, L., Kherallah, M., & Alimi, A. M. (2011). Textile plant modeling using recurrent neural networks. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 1580-1584.

Hsu, C. H., & Wang, M. J. J. (2005). Using decision tree-based data mining to establish a sizing system for the manufacture of garments. *International Journal of Advanced Manufacturing Technology*, *26* (5-6), 669–674.

Hu, J., Xin, B., & Yan, H. J. (2002). Classifying Fleece Fabric Appearance by Extended Morphological Fractal Analysis. *Textile Research Journal*, *72* (10), 879–884.

Hu, Z. H., Ding, Y. S., yu, X. K., Zhang, W. Bin, & Yan, Q. (2009). A Hybrid Neural Network and Immune Algorithm Approach for Fit Garment Design. *Textile Research Journal, 79* (14), 1319-1330.

Huang, Y., Qiu, X., & Yang J. (2009). The association rule mining technology in marketing decision-making and its application in logistics enterprises. *Proceedings of the 4th International ISKE Conference on Intelligent Systems and Knowledge Engineering*, 252-257.

Ingle, M. G., & Suryavanshi, N. Y. (2015). Association rule mining using improved Apriori algorithm. *International Journal of Computer Applications, 112* (4), 37-42.

Jaouachi, B., & Khedher, F. (2015). Evaluation of sewed thread consumption of jean trousers using neural network and regression methods. *Fibres and Textiles in Eastern Europe, 3* (111), 91-96.

Jiang, G. M., Zhang, D., Cong, H. L., Zhang, A. J., & Gao, Z. (2014). Automatic identification of jacquard warp-knitted fabric patterns based on the wavelet transform. *Fibres and Textiles in Eastern Europe, 104* (2), 53-56.

Jing, J., Zhang, Z., Kang, X., & Jia, J. (2012). Objective evaluation of fabric pilling based on wavelet transform and the local binary pattern. *Textile Research Journal, 82* (18), 1880–1887.

Keller, J. M., & Gray, M. R. (1985). A Fuzzy K-Nearest Neighbor Algorithm. *IEEE Transactions on Systems, Man and Cybernetics, 15* (4), 580-585.

Kim, S. C., & Kang, T. J. (2005). Fabric Surface Roughness Evaluation Using Wavelet-Fractal Method: Part II: Fabric Pilling Evaluation. *Textile Research Journal, 75* (11), 761–770.

Kumar, T. S., & Sampath, V. (2012). Prediction of dimensional properties of weft knitted cardigan fabric by artificial neural network system. *Journal of Industrial Textiles, 42* (4), 446-458.

Kuo, C. F. J., & Juang, Y. (2016). A study on the recognition and classification of embroidered textile defects in manufacturing. *Textile Research Journal, 86* (4), 393–408.

Kuo, C. F. J., Lan, W. L., Dong, M. Y., Chen, S. H. & Lin F. S. (2018). Development of disperse dyes polypropylene fiber and process parameter optimization Part II: Dyeable polypropylene fiber production and melt spinning process parameter optimizationColor texture classification of yarn-dyed woven fabric based on dual-side scanning and co-occurrence matrix. *Textile Research Journal, 88* (13), 1505–1516.

Kuo, C. F. J., Shih, C. Y., & Hsu, C. T. M. (2011). Pattern-making simulation on embroidery using probabilistic neural network and texture fitting method. *Textile Research Journal, 81* (20), 2082-2094.

Lee, C. K. H., Choy, K. L., Ho, G. T. S., Chin, K. S., Law, K. M. Y., & Tse, Y. K. (2013). A hybrid OLAP-association rule mining based quality management system for extracting defect patterns in the garment industry. *Expert Systems with Applications, 40* (7), 2435-2446.

Lewandowski, S. (2011). Neural network classification of the unknotted joints of Yarn ends. *Fibres and Textiles in Eastern Europe, 86* (3), 37-43.

Lewandowski, S., & Drobina, R. (2008). Prediction of properties of unknotted spliced ends of yarns using multiple regression and artificial neural Networks. Part I: Identification of spliced joints of combed wool yarn by artificial neural networks and multiple regression. *Fibres and Textiles in Eastern Europe, 70* (5), 33-39.

Li, H., Wang, Y., Zhang, D., Zhang, M., & Chang, E. Y. (2008). PFP: Parallel FP-Growth for Query Recommendation Haoyuan. *Proceedings of the 2008 ACM conference on Recommender systems,* 107-114.

Li, W., & Cheng, L. (2014). Yarn-dyed woven defect characterization and classification using combined features and support vector machine. *Journal of the Textile Institute, 105* (2), 163-174.

Li, Y. S. W., Yuen, C. W. M., Yeung, K. W., & Sin, K. M. (2001). Modifying an Existing Numerical Shade Sorting System Through Cluster Analysis. *Textile Research Journal, 71* (4), 287-294.

Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery, 6*, 393-423.

Logeswari, T., Valarmathi, N., Sangeetha, A., & Masilamani, M. (2014). Analysis of traditional and enhanced Apriori algorithms in association rule mining. *International Journal of Computer Applications, 87* (19), 4-8.

Lu, K., Zhong, Y., Li, D., Chai, X., Xie, H., Yu, Z., & Naveed, T. (2018). Cashmere/wool identification based on bag-of-words and spatial pyramid match. *Textile Research Journal, 88* (21), 2435–2444.

Lü, Z-J., Yang, J., Xiang, Q., & Wang, X. (2007). Support vector machines for predicting worsted yarn properties. *Indian Journal of Fibre and Textile Research, 32* (2), 173-178.

Manning, C. D., Raghavan, P., & Schütze, H. (2012). Hierarchical clustering. In *An Introduction to Information Retrieval* (377-402). New York: Cambridge University Press.

Mariolis, I. G., & Dermatas, E. S. (2010). Automated assessment of textile seam quality based on surface roughness estimation. *Journal of the Textile Institute, 101* (7), 653–659.

Matusiak, M. (2015). Application of artifcial neural networks to predict the air permeability of woven fabrics. *Fibres and Textiles in Eastern Europe, 109* (1), 41-48.

Mozafary, V., & Payvandy, P. (2014). Application of data mining technique in predicting worsted spun yarn quality. *Journal of the Textile Institute, 1*, 100-108.

Mustafic, A., Jiang, Y. & Li, C. (2016). Cotton contamination detection and classification using hyperspectral fluorescence imaging. *Textile Research Journal, 86* (15), 1574–1584.

Nourani, Gh., Jeddi, A. A. A., & Moghadam, M. B. (2011). Determining the structural parameters and yarn type affecting tensile strength and abrasion of weft knitted fabrics using cluster analysis. *Middle-East Journal of Scientific Research, 8* (6),1008-1017.

Novák, V., Perfilieva, I., & Dvořák, A. (2016). Fuzzy Cluster Analysis. In *Insight into Fuzzy Modeling* (1st ed.) (137-149). New Jersey: Wiley.

Nurwaha, D., & Wang, X. H. (2012). Using intelligent control systems to predict textile yarn quality. *Fibres and Textiles in Eastern Europe, 90* (1), 23–27.

Ogulata, S. N., Sahin, C., Ogulata, R. T., & Balci, O. (2006). The prediction of elongation and recovery of woven bi-stretch fabric using artificial neural network and linear regression models. *Fibres and Textiles in Eastern Europe, 56* (2), 46-49.

Ozbek, A., Akalin, M., Topuz, V., & Sennaroglu, B. (2011). Prediction of Turkey's denim trousers export using artificial neural Networks and the autoregressive integrated moving average Model. *Fibres and Textiles in Eastern Europe, 86* (3), 10-16.

Özkan, İ., Kuvvetli, Y., Baykal, P. D., & Sahin, C. (2015). Predicting the intermingled yarn number of nips and nips stability with neural network models. *Indian Journal of Fibre and Textile Research, 40* (3), 267-272.

Pietracaprina, A., Riondato, M., Upfal, E., & Vandin, F. (2010). Mining top-K frequent itemsets through progressive sampling. *Data Mining and Knowledge Discovery, 21* (2), 310-326.

Prada, P. A., Curran, A. M., & Furton, K. G. (2014). Characteristic human scent compounds trapped on natural and synthetic fabrics as analyzed by SPME-GC/MS. *Journal of Forensic Science & Criminology, 1* (1), 1-10.

Precup, R. E., Sabau, M. C., & Petriu, E. M. (2015). Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for Anti-lock Braking Systems. *Applied Soft Computing Journal, 27*, 575-589.

Rahnama, M., Semnani, D., & Zarrebini, M. (2013). Measurement of the moisture and heat transfer rate in light-weight nonwoven fabrics using an intelligent model. *Fibres and Textiles in Eastern Europe, 102* (6), 89-94.

Salcedo-Sanz, S., Rojo-Álvarez, J. L., Martínez-Ramón, M., & Camps-Valls, G. (2014). Support vector machines in engineering: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4* (3), 234-267.

Shih, C. Y., Kuo, C. F. J. & Cheng, J. H. (2016). A study of automated color, shape and texture analysis of Tatami embroidery fabrics. *Textile Research Journal, 86* (17), 1791–1802.

Soltani, P., Shahrabi, J., Asadi, S., Hadavandi, E., & Johari, M. S. (2013). A study on siro, solo, compact, and conventional ring-spun yarns. Part III: modeling fiber migration using modular adaptive neuro-fuzzy inference system. *Journal of the Textile Institute, 103* (9), 755-765.

Song, H. K., & Ashdown, S. P. (2011). Categorization of lower body shapes for adult females based on multiple view analysis. *Textile Research Journal, 81* (9), 914-931.

Su, T. L., & Lu, C. F. (2011). Automated vision system for recognising lycra spandex defects. *Fibres and Textiles in Eastern Europe, 1* (84), 43-46.

Sun, J., Yao, M., Xu, B., & Bel, P. (2011). Fabric wrinkle characterization and classification using modified wavelet coefficients and support-vector-machine classifiers. *Textile Research Journal, 81* (9), 902–913.

Svetnik, V., Wang, T., Tong, C., Liaw, A., Sheridan, R. P., & Song, Q. (2005). Boosting: An ensemble learning tool for compound classification and QSAR modeling. *Journal of Chemical Information and Modeling, 45* (3), 786-799.

Uçar, N., & Ertuğrul, S. (2017). Prediction of fuzz fibers on fabric surface by using neural network and regression analysis. *Fibres and Textiles in Eastern Europe, 61* (2), 58-61.

VašČák, J., & Pal'a, M. (2012). Adaptation of fuzzy cognitive maps for navigation purposes by Migration Algorithms. *International Journal of Artificial Intelligence, 41* (3), 429-443.

Vrkalovic, S., Teban, T., & Borlea, I. (2017). Stable Takagi-Sugeno fuzzy control designed by optimization. *International Journal of Artificial Intelligence, 15* (2), 17-29.

Walter, B., Bala, K., Kulkarni, M., & Pingali, K. (2008). Fast agglomerative clustering for rendering. *RT'08 - IEEE/EG Symposium on Interactive Ray Tracing 2008,* 81-86.

Wan, Y., Yao, L., Zeng, P., & Xu, B. (2010). Shaped Fiber Identification Using a Distance-Based Skeletonization Algorithm. *Textile Research Journal, 80* (10), 958-968.

Wu, Y., Chen, R., Wang, J., Sun, X., & She, M. F. H. (2012). Intelligent clothing for automated recognition of human physical activities in free-living environment. *Journal of the Textile Institute, 103* (8), 806-816.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). *Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms*. Retrieved May 15, 2018, from https://www.kaggle.com/zalando-research/fashionmnist/data.

Xiao, Z., Nie, X., Zhang, F., & Geng, L. (2014). Recognition for woven fabric pattern based on gradient histogram. *Journal of the Textile Institute, 102* (1), 744-752.

Xiao, Z., Nie, X., Zhang, F., Geng, L., Wu, J., & Li, Y. (2015). Automatic recognition for striped woven fabric pattern. *Journal of the Textile Institute, 107* (11), 409-416.

Xin, B., Li, Y., Qiu, J., & Liu, Y. (2012). Texture modelling of fabric appearance evaluation based on image analysis. *Fibres and Textiles in Eastern Europe, 2* (91), 48-52.

Xin, B., Zhang, J., Zhang, R. & Wu, X. (2017). Color texture classification of yarn-dyed woven fabric based on dual-side scanning and co-occurrence matrix. *Textile Research Journal, 87* (15), 1883–1895.

Yap, P. H., Wang, X., Wang, L., & Ong, K. L. (2010). Prediction of Wool Knitwear Pilling Propensity using Support Vector Machines. *Textile Research Journal, 80* (1), 77-83.

Yaşar, T. (2015). *Hava jetli iplik eğirme sistemleri ile üretilmiş ipliklerin performanslarının değerlendirilmesi*. MSc Thesis, Erciyes University, Kayseri.

Yildirim, B., & Başer, G. (2011). Measurement of cloth fell position using image analysis. *Journal of the Textile Institute*, *105* (6), 905-916.

Yildirim, P., & Birant, D. (2017). K-Linkage: A New Agglomerative Approach for Hierarchical Clustering. *Advances in Electrical and Computer Engineering, 17* (4), 77 – 88.

Yildirim, P., Birant, D., & Alpyildiz, A. (2017). Discovering the relationships between yarn and fabric properties using association rule mining. *Turkish Journal of Electrical Engineering & Computer Sciences, 25* (6), 4788 – 4804.

Yıldırım, P., Birant, D., & Alpyıldız, A. (2017). Improving prediction performance using ensemble neural networks in textile sector. *International Conference on Computer Science and Engineering (UBMK)*, 639 – 644.

Yildirim, P., Birant, D., & Alpyildiz, A. (2018). Data mining and machine learning in textile industry. *WIREs Data Mining and Knowledge Discovery, 8* (1), 1-20.

Yildiz, K., Buldu, A., & Demetgul, M. (2016). A thermal-based defect classification method in textile fabrics with k-nearest neighbor algorithm. *Journal of Industrial Textiles, 45* (5), 780-795.

Yildiz, Z., Dal, V., Ünal, M., & Yildiz, K. (2013). Use of artificial neural networks for modelling of seam strength and elongation at break. *Fibres and Textiles in Eastern Europe, 101* (5), 117-123.

Yoon, H. S., & Park, S. W. (2002). Determining the Structural Parameters That Affect Overall Properties of Warp Knitted Fabrics Using Cluster Analysis. *Textile Research Journal, 72* (11), 1013-1022.

Yu, H., & Ni, J. (2014). An improved ensemble learning method for classifying high-dimensional and imbalanced biomedicine data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 11* (4), 657-666.

Yu, L., Wang, S., & Lai, K. K. (2008). Credit risk assessment with a multistage neural network ensemble learning approach. *Expert Systems with Applications, 34* (2), 1434-1444.

Yu, L., Wang, S., & Lai, K. K. (2008). Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Economics, 30* (5), 2623-2435.

Yu, Y., Hui, C. L. P., Choi, T. M., & Ng, S. F. F. (2011). A new approach for fabric hand prediction with a nearest neighbor algorithm-based feature selection scheme. *Textile Research Journal, 81* (6), 574–584.

Zakaria, N. (2011). Sizing system for functional clothing - uniforms for school children. *Indian Journal of Fibre and Textile Research, 36*, 348–357.

Zaki, M. J., & Hsiao, C.-J. (2002). CHARM: An Efficient Algorithm for Closed Itemset Mining. *Proceedings of the 2002 SIAM International Conference on Data Mining,* 457-473.

Zhang, J., & Yang, C. (2014). Evaluation model of color difference for dyed fabrics based on the Support Vector Machine. *Textile Research Journal, 84* (20), 2184–2197.

Zhang, J., Xin, B., Shen, C., Fang, H., & Cao, Y. (2015). Novel colour clustering method for interlaced multi-colored dyed yarn woven fabrics. *Fibres and Textiles in Eastern Europe, 111* (3), 107–114.

Ziegler, A., & König, I. R. (2014). Mining data with random forests: Current options for real-world applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4* (1), 55–63