ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

M.Sc. THESIS

Enver Can DORAN

DEVELOPMENT OF MODELS FOR THE PREDICTION OF PERFORMANCE OF COTTON/ELASTANE CORE YARN

DEPARTMENT OF INDUSTRIAL ENGINEERING

ADANA-2019

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEVELOPMENT OF MODELS FOR THE PREDICTION OF PERFORMANCE OF COTTON/ELASTANE CORE YARN

Enver Can DORAN

MSc THESIS

DEPARTMENT OF INDUSTRIAL ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Master of Science by the board of jury on 06/03/2019.

…….	……..	……...
Assoc. Prof. Dr. Cenk ŞAHİN	Prof.Dr. Ali KOKANGÜL	Prof.Dr. Onur BALCI
SUPERVISOR	MEMBER	MEMBER

This MSc Thesis is written at the Department of Institute of Natural and Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

ABSTRACT

MSc. THESIS

DEVELOPMENT OF MODELS FOR THE PREDICTION OF QUALITY PERFORMANCE OF COTTON/ELASTANE CORE YARN

Enver Can DORAN

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF INDUSTRIAL ENGINEERING

Supervisor : Assoc. Prof. Dr. Cenk ŞAHİN
Year: 2019 Pages: 129
Jury : Assoc. Prof. Dr. Cenk ŞAHİN
: Prof. Dr. Ali KOKANGÜL
: Prof. Dr. Onur BALCI

Core yarn is a type of yarn that has a filament fiber in the center with a different fiber wrapped around it. They have a raising importance in the textile industry. That's why error free production for cotton/elastane core yarns and the design of models that can correctly predict the product quality parameters from fiber quality and spinning parameters are needed more and more. Since the multivariate data size reduces the successful prediction chance, the ways to decrease the dimension of data matrix was investigated. Principal Component Analysis (PCA) and Analysis of Variance (ANOVA) were both used to reduce the dimension of input matrix. Since cotton/elastane yarn quality performance prediction problem is non linear, Artificial Neural Networks (ANN) and Support Vector Machines (SVM) were proposed to predict the quality control parameters of core yarns. Both inputs from PCA, ANOVA and the unreduced input matrix were used to compare the power of each size reduction. The predicting models were designed using the data from a textile plant. Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE) and R squared values were used as performance indicators. Neural Networks with PCA reduced input matrix was shown to have the best MSE and MAPE values. The best model has shown to have over 90% success rate in MAPE for most of the yarn quality characteristics.

Key Words: Prediction, spinning parameters, cotton/elastane blend core yarns, Support Vector Machines. Artificial Neural Networks

ÖZ

YÜKSEK LİSANS TEZİ

PAMUK/ELASTAN KARIŞIMLI ÖZLÜ İPLİĞİN KALİTE PERFORMANSINI TAHMİN EDEBİLEN MODELLER GELİŞTİRME

Enver Can DORAN

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Doç. Dr. Cenk ŞAHİN
Yıl: 2019 Sayfa: 129
Jüri : Doç. Dr. Cenk ŞAHİN
: Prof. Dr. Ali KOKANGÜL
: Prof. Dr. Onur BALCI

Özlü iplikler filaman çevresine farklı bir elyaf sarılmasıyla oluşturulan katma değerli ipliklerdir. Bu iplikler, tekstil endüstrisinde giderek artan bir öneme sahiptir. Bu yüzden hatasız üretim ve ürün kalite özelliklerini elyaf kalite ve eğirme parametrelerini kullanarak tahmin eden modellerin geliştirilmesine duyulan ihtiyaç da giderek artmaktadır. Çok değişkenli veri boyutu doğru tahmin olasılığını düşürdüğü için veri matrisinin boyutlarını küçültme yolları incelenmiştir. Temel Bileşenler Analizi (TBA) ve Varyans Analizi (VA) kullanılarak girdi matrisinin boyutu azaltılmıştır. Pamuk/elastin karışımlı özlü ipliklerin kalite özelliklerini tahmin etme problem lineer olmadığı için. özlü iplik kalite özelliklerinin tahmini için Yapay Sinir Ağları (YSA) ve Destek Vektör Makinaları'nın (DVM) kullanılması önerilmiştir. Boyut küçültme yöntemlerinin etkilerini daha rahat görebilmek amacıyla bütün TBA, VA ve boyutları küçültülmemiş girdi matrisleri ayrı ayrı modellenip sonuçları karşılaştırılmıştır. Tahmin eden modeler bir tekstil fabrikasından alınan veriler kullanılarak geliştirilmiştir. Ortalama Hataların Karesi, Mutlak Ortalama Hata Yüzdesi ve R^2 değerleri sistemin performans ölçütleri olarak belirlenmiştir. TBA girdilerini kullanan YSA en iyi Ortalama Hataların Karesi ve Mutlak Ortalama Hata Yüzdesi değerlerini vermiştir. Modellerin en iyisinin, özlü iplik kalite özelliklerinin birçoğunu, Mutlak Ortalama Hata Yüzdesi'ne göre yüzde 90 üzerinde bir başarıyla tahmin ettiği gösterilmiştir.

Anahtar Kelimeler: Tahminleme, Eğirme Parametreleri, Pamuk/Elastan Özlü İplik, Destek Vektör Makinaları, Yapay Sinir Ağları

EXPANDED ABSTRACT

Core spun yarns are used in a large variety of products ranging from elastic fabrics and even industrial clothing. They are shown to have higher strength, durability and stretching properties than their counterparts.

In this study, which can be defined as protective engineering, the aim is to predict the quality characteristics of the yarn before production and to prevent the faulty production that way the quality of the businesses, where this model can be used, will be increased and correct production concept can be developed from the beginning.

This study was focused on evaluating the performance of artificial neural networks and support vector machines. In order to get the most concurring data possible, 3 different learning methods for neural networks and support vector machines were used. Support vector machines were also used with multiple optimization methods. To make sure the data is valid, different number of nodes were used for the hidden layer on both artificial neural networks and support regressor machines.

However, the increase of multivariate in the data decreases the successful prediction chance of both ANNs and SVMs. Therefore, Principal Component Analysis (PCA) and Analysis of Variance (ANOVA) are implemented implemented to reduce the dimension of input data.

ANOVA was used to show the relationship between the fiber quality characteristics and core yarn spinning parameters and whether the change of a fiber parameter affects the core yarn parameter in a statistically significant way.

Principal Component Analysis showed the variables which changes the most or the variables which has the highest variance. This way input variables that has a less effect on the output variables can be removed.

The actual data provided by a textile factory contained 37 types of fibre quality and spinning parameters and 10 types of quality characteristics for cotton blend core yarns. They have 227 samples each without any missing data.

Len is for the length of fiber which is highly correlated with the breakage (strength) of the fiber. Unf is for fiber length uniformity index which effects manufacturing waste spinning potential and yarn strength. SFI is for Short Fiber Index which directly effects uniformity index changing yarn strength and spinning potential. Usually fibers shorter than 12.7 mm or 0.5 inch are called short fibers. Str is for fiber strength which directly effects yarn strength and can give a quick estimation of yarn strength and yarn twist. Elg is for elongation of fiber which directly effects yarn elongation which influences the performance of yarns during winding warping and weaving. Rd is for reflectance which is one of the main components that determine the colour grade of cotton. B is for yellowness also a main component of the colour grade of cotton. Fiber strength, fiber length, micronnaire, yellowness and uniformity all affect the price of the fiber. Tr Cnt is trash count and Tr Area is trash percentage area. Cotton trash impacts spinning and causes efficiency problems.

Nep Cnt/g is the nep count per gram. Nep (um) is the average size of neps in micrometers. Neps are small entanglement of textile fibers that can't be unraveled causing undyed spot sor unprinted spots. Immature fibers tend to cause more neps than mature ones. SCN/g is seed coat nep count per gram. SCN (um) is seed coat nep size in micrometers. SCNs effect spinning quality greatly which makes both their quantity and their size important aspects of the quality of the fiber. L(w) is the average fiber length by weight. L(w) CV% means length variation by weight. L(n) is the average fiber length in millimeters. L(n) 5% is 5% length of fibers in millimeters. UQL(w) is upper quartile fiber length by weight. It was previously mentioned that fiber length directly affects the strength of yarn. SFC(w) is short fiber content by weight. Short fibers change yarn strength and spinning potential. Total count/g is trash count per 1 g. Cnt/g is trash count per gram with a size larger than 500 micrometers. As mentioned above, trash count directly effects the spinning parameters and quality of yarn. Dust Cnt/g is the dust count per gram. Dusts are particles with a size shorter than 500 micrometers. Cotton dust causes pulmonary illnesses amongst workers. It increases the chance of bronchitis. Cotton dust also decreases the quality of yarn like trash count. Mean

size is the average size of fiber. VFM% shows the Visible Foreign Matter (dust and trash percentages) for sample comparison. Mat Ratio is the Maturity Ratio of the fiber. FinemTex is the fineness of the fiber in mTex. IFC is the percentage of immature fiber content.

The dataset was dropped from 37 to 29 by ANOVA and from 37 to 24 by PCA. For Principal Component Analysis, only components 1 and 2 were considered, since they represented 95% of the total variance.

For ANNs backpropagation algorithm was used with Levenberg Marquardt, Bayesian Regularization and Scaled Conjugate Gradient. For SVMs Gaussian Kernel, Linear Kernel and Polynomial Kernel algorithms were all used with Iterative Single Data Algorithm, Quadratic Programming and Sequential Minimal Optimization. All of them were compared to each other according to their Test MSE. Test MAPE and Test R values. This allowed a complete comparison including every possible algorithm from each side.

Later, both ANNs and SVMs were fed separately with ANOVA reduced input matrix and PCA reduced input matrix. To show their effect on the result, the whole input matrix was also fed into ANN and SVM both.

The results show that using PCA to reduce the input matrix increased the performance of both SVM and ANN. ANN was shown to have better performance rates with PCA and Scaled Conjugate Gradient learning algorithm than every other algorithm of SVM.

Neural networks fared better than SVMs on some of the outputs. Yet on some of the outputs SVM can be seen as the better choice. On some of the outputs. both ANN and SVM had high amount of errors. On some of the outputs. they had very similar and powerful results.

Unfortunately, the performance indicators for the different training algorithms stayed very close to each other. Even though Scaled Gradient had the best result. this was not conclusive.

Even though the results were close, on each of the tests PCA fared slightly better than ANOVA. Not only that the input matrix was reduced better by PCA

which is 24 from 37, but the reduction actually helped to get better results on the same model. We may conclude that PCA is the better size reduction technique.
Neural networks fared better than SVMs on some of the outputs like Um and ELG. Yet on some of the outputs like Neps 200 and H value SVM is shown to be the better choice. On the CRM, both ANN and SVM had high amount of errors. On B Force, they had very similar and powerful results. Therefore, it is impossible at this time to say one is better than the other. even though ANN was slightly better at the best model comparison.

Aside from these inputs. the spinning and core spun test results of the fibers are also in the dataset. Spinning the fiber is a part of the process to produce a yarn. so the fiber with better spinning properties produce higher quality yarns. The test with ringspun core yarns gives the thread number of the yarn and the winding of the said yarn. Since core fibers are used core spun test results were also used. They include elastical core spun fibers with lycra dtex results (Lycra Dtex). Take off value is the strength of the fiber.

The outputs were also given in the database for teaching and testing. Each of the 227 samples were tested before and after they got processed into yarn. The resulting quality characteristics are all shown in the database. ELG is the elongation of the produced yarn. RKM is the breaking point of yarn where the yarn will break under its own weight. RKM is short for Reisskilometer. RKM is basically the tenacity of the yarn. Um is the mass of the yarn and CVm is the mass variations along the yarn. B force is the breaking force of the yarn. The thinness and thickness of the yarn is shown by Thin and Thick. The rest are the neps which are between 140-200 micrometers (Neps 140) and the neps larger than 200 micrometers (Neps 200).

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude for my advisor Assoc. Prof. Dr. Cenk ŞAHİN for his guidance. His support and encouragement during my studies. I would also like to thank Asst. Prof. Dr. Yusuf KUVVETLİ for his immense support and guidance.

I want to thank Prof. Dr. Ali KOKANGÜL for his help in my studies and for his help during the preparation of this thesis.

I would also like to take this chance to show my gratitude to Prof. Dr. Onur BALCI for his help with providing the necessary data for my study and helping with textile related information.

I would also like to thank KİPAŞ company for providing the necessary data for my studies.

I am also very grateful to Mehmet Can PEKTAŞ for his help during my experiments.

I would also like to thank my family for their support and trust until now. I am forever indebted to them.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

NOMENCLATURE AND ABBREVIATIONS

a	: Activation Function
b	: Bias
C	: Cost Function
e	: Base of Natural Algorithm
h	: Hidden layer function
y	: Output Value
x	: Input Value
w	: Input Weight Value
k	: Synapse Weight Value
i	: Input Number
l	: Layer Number
j	: Output Number
z	: Weighted Input Value
p	: Significance Value
AFIS	: Advanced Fiber Information Systems
ANN	: Artificial Neural Networks
ANOVA	: Analysis of Variance
B	: Yellowness
B-Force	: Breaking Force of the Yarn
BRANN	: Bayesian Regularized Artificial Neural Networks
CVm	: Mass Variations Along Yarn
DTSVM	: Decision Tree Support Vector Machine
DustCnt	: Dust Count Per Gram
Elg	: Elongation of Fiber
ELG	: Elongation of Produced Yarn

FinemTex	: Fineness of Fiber
H	: Hairiness of the Yarn
HVI	: High Volume Index
IFC	: Percentage of Immature Fiber Content
ISDA	: Iterative Single Data Algorithm
L1QP	: Quadratic Programming
Len	: Length of Fiber
LikraDtex	: Dtex value of Lycra
Lnmm	: Average Fiber Length in Micrometers
LnCV	: Length Variation of Fiber
Lwmm	: Average Fiber Length by Weight
LwCV	: Length Variation by Weight
MAPE	: Mean Absolute Percentage Error
Mat	: Maturity Index
MatRatio	: Maturity Ratio
MeanSize	: Average Size of Fiber
Mic	: Micronnaire
Mm5	: 5% Length of Fibers in Milimeters
MSE	: Mean Squared Error
Nep	: Average Size of Neps
NepCnt	: Seed Coat Nep Count Per Gram
Neps140	: Neps Between 140-200 Micrometers
Neps200	: Neps Larger Than 200 Micrometers
NPCA	: Non-Linear Principal Component Analysis
PCA	: Principal Component Analysis
RBF	: Radial Basis Function
Rd	: Reflectance
RKM	: Tenacity of the Yarn

ROBPCA	: Robust Principal Component Analysis
SCI	: Spinning Consistency Index
SCG	: Scaled Conjugate Gradient
SCN	: Seed Coat Nep Size
SCNCnt	: Seed Coat Nep Count
SFCn	: Short Fiber Count
SFCw	: Short Fiber Count by Weight
SFI	: Short Fiber Index
Spinning	: Spinning Force of Fibre
Str	: Fiber Strength
SMO	: Sequential Minimal Optimization
SMOTE	: Synthetic Minority Over Sampling Technique
SNARC	:Stochastic Neural Analog Reinforcement Calculator
SVM	: Support Vector Machine
TakeOff	: Breaking Force of Fibre
Thick50	: Thick Yarn Count
Thin50	: Thin Yarn Count
TLN	: Threshold Logic Neuron
TrArea	: Trash Percentage Area
TrCnt	:Trash Count Per Gram Larger Than 500 micrometers
TotalCnt	: Trash Count Per 1 Gram
Um	: Mass of Yarn
Unf	: Length Uniformity Index
UQL	: Upper Quartile Fiber Length by Weight
VFM	: Visible Foreign Matter

1. INTRODUCTION

In order to maintain the high-quality standards established by the clothing industry, textile manufacturers have to monitor the quality of their product making textile quality a key factor in their competitiveness (Anagnostopoulos et al., 2001). This makes the evolution of quality control, a necessity for the sector. From human inspection to machines, the quality control of textile sector is a constantly evolving part of the industry.

Textile fibres are used by people since the beginning of their existence. New technologies and raw materials allow new types of fabrics to be produced. Fiber properties such as durability, strength and flexibility are emphasized in modern times (Vaskova and Hrano, 2017).

The garment industry is one of the main economic sectors which has an important role in everyday life and global economics. The pressure on the garment producers have increased due to the competition in global markets to provide the best quality in lower price levels. Since the quality of garments depends heavily on the fabric, the need to produce high quality fabric is higher than ever.

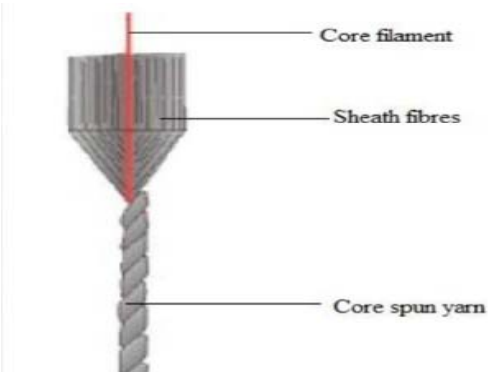

Figure 1.1. Core spun yarn structure (Akankwasa et al., 2015)

Core spun yarn is a type of yarn that has a filament fiber in the center with a different fiber wrapped around it as shown in Figure 1.1. They are yarns with

added value where the quality has more importance than the conventional yarns. They are produced mostly on the ring and the friction spinning machines. Core spun yarns take advantage of the qualities of both components. Filament provides strength and lower twist level while the sheath allows for a fibre yarn appearance and physical attributes (Akankwasa et al., 2015).

Core yarns are used in a large variety of products ranging from elastic fabrics to tents and even industrial clothing. They are shown to have increased strength, durability and stretching properties (Chandrasekaran et al., 2014).

The quality of core yarn is a highly important characteristic in textile industry. It has a great impact on the quality of fabric and the final product. Yarn has many important characteristics both physical and mechanical. The physical and mechanical properties of yarn (breaking force, tenacity, elongation, unevenness. thin places, thick places, neaps and hairiness) effect the resulting fabric so predicting these properties before the production of yarn is substantial (Demiryurek and Koc, 2009).

Elastane blend core yarns also known as cotton or lycra, was developed in 1959 by DuPont company. Although it was developed for sportswear, with its stretching ability this product was soon a hit in fashion.

A lycra core surrounded with cotton elastane/cotton blend core yarns add the flexible, stretching, rubber like characteristic to the durability of cotton as shown in Figure 1.2.

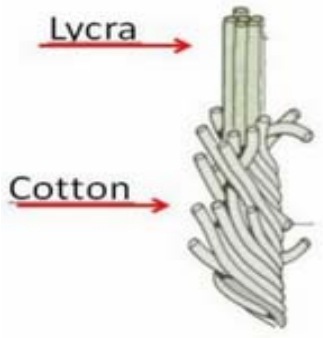

Figure 1.2. Elastane/cotton core yarn(Oztekin M., 2016)

Spandex has many strengths and few weaknesses as shown in Table 1.1. The most important strength is its ability to stretch 500% which allowed spandex to increase its market share and stay an important textile product.

Table 1.1. Strengths and weaknesses of elastane/cotton core yarns

Strengths	**Weaknesses**
Light and comfortable can be stretched 500% without breaking	Whites yellow with age
Light and comfortable can be stretched 500% without breaking	Whites yellow with age
Abrasion resistant retains original shape	Heat sensitive
Stronger than rubber yet soft and smooth	Harmed by bleach
No pilling or static	Nonabsorbent
Dries quickly	
Abrasion resistant retains original shape	Heat sensitive
Stronger than rubber yet soft and smooth	Harmed by bleach
No pilling or static	Nonabsorbent
Dries quickly	

It is important to predict the quality characteristics of the cotton/elastane core yarn before production, to prevent the faulty production of fabrics. The human mind can only do this based on experience and that way is not reliable or sustainable. Therefore the development of predicting models is a necessity in textile industry.

The predicting models in textile industry are used to get higher quality products using correct parameters and to decrease waste. By using correct estimations, the output (in this case product) efficiency can be increased and waste of fibers and power for each time the yarn doesn't meet the specifications, can be decreased. Furthermore, by knowing the relationships between the performance and characteristics of yarns and fibers the producer can make informed decisions

for the required quality of yarn. Mathematical models are generally based on assumptions so their success is limited (Turhan and Toprakci, 2013).

In the literature review, a lot of artificial neural network based predicting models such as feed forward neural networks and support vector machines for the textile industry can be found. Artificial neural networks can be seen to show much success in this area. Artificial neural networks have a past over 20 years in textile industry. Yet the use of ANNs for predicting properties of a product can only be seen much later (Demiryurek and Koc, 2009).

Artificial neural networks (ANN) are used inthis study since they can deal with a lot of variable inputs, can show the relationship between the result and input and can be used to develop a design that will predict the quality of yarn with a minimal error.

1.1. Aim of the Study

The objective of this thesis is predicting the quality characteristics of core yarns that can be determined by USTER and Tensorapid devices, from the fiber and some important spinning parameters using artificial neural network models and support vector machine models. During this study, while the design of the model with relevant inputs and outputs is the main objective, side objective is the determination of proper network parameters so that the optimum network can be achieved.

This study aims to answer several questions:

- Do artificial neural network models work as a predicting algorithm for quality characteristics of cotton/elastane blend core yarns?
- Do support vector machine models work as a predicting algorithm for quality characteristics of cotton/elastane blend core yarns?

- If both models work as a predicting algorithm for quality characteristics of cotton/elastane blend core yarns which one of them works as a better predictor?
- Does PCA help improve the performance of either artificial neural network models or support vector machine models?
- Does ANOVA help improve the performance of either artificial neural network models or support vector machine models?
- If PCA and ANOVA methods do improve the performance of artificial neural networks or support vector machines which one of them works as a better improvement?
- Is there a significant difference of performance between different training algorithms of artificial neural networks?
- Is there a significant difference of performance between different training algorithms of support vector machines?
- Is there a significant difference of performance between different solvers for support vector machines?
- Is there a significant difference of performance for increasing or decreasing the number of nodes in the hidden layer of either support vector machines or artificial neural networks?

The performance indicators for estimations are Mean Squared Errors (MSE), Mean Absolute Percentage Error (MAPE) and R squared values.

1.2. Scope and Original Contributions

This study uses data provided by KİPAŞ a Turkish textile plant. The dataset includes 37 types of fibre quality characteristics and spinning parameters and 10 types of core yarn quality characteristics. There are 227 samples for each of them without any missing data. The samples are collected from 4 different machines and

from 2 different types of core spun yarns. The quality characteristics and spinning parameters of the fibre are collected from both high volume instruments (HVI) and advanced fibre information system (AFIS) machines.

This study differs from the literature in three ways. First of all, the predicting models are focused on the quality characteristics of elastane core yarns. Second of all, this study will use actual business data. Finally, it acts as an original comparison between SVMs and ANNs using input data matrix dimension reduced by ANOVA and PCA.

1.3. Steps and Organization of the Study

This study will start with an extensive research of literature on ANN, SVM, PCA and ANOVA. Then in the materials and method section, it will show every method used during the study with an extensive explanation of the dataset given by KİPAŞ. After that, the results of the PCA and ANOVA will be discussed and the ANN and SVM outputs will be shown. The study will end with the comparison of the results and suggestions for future studies will be given.

The actual dataset from a textile plant includes High Volume Index (HVI) fibre characteristics along with Advanced Fibre Information Systems (AFIS) fibre characteristics and spinning parameters. The size of this dataset will be reduced by PCA and ANOVA separately. PCA results and ANOVA results will both be used as inputs for ANN and SVM models. The results will be compared against actual results and performance parameters which are MAPE, MSE and R^2 will be used to determine the best model. Figure 1.3 shows the required steps of this study.

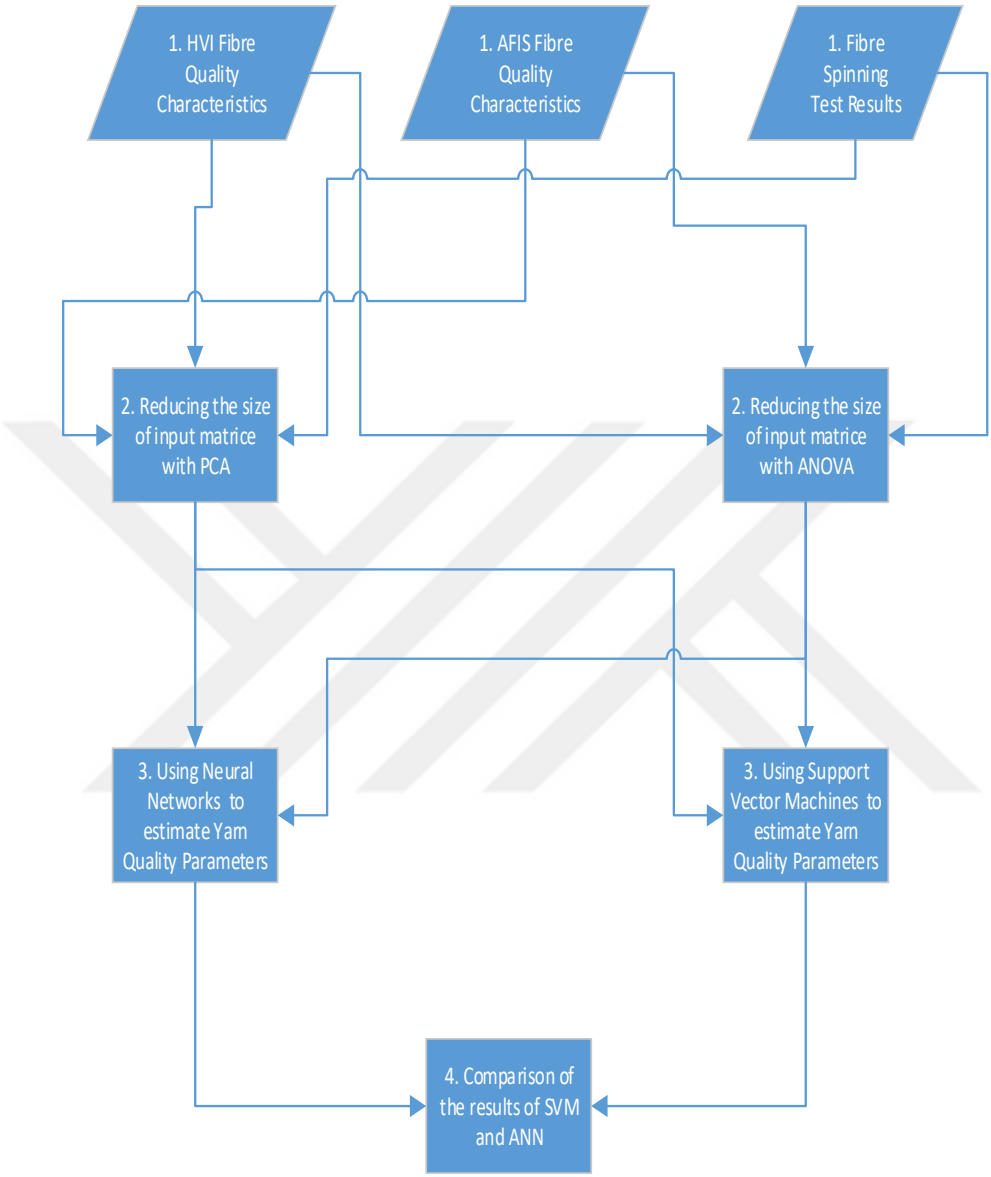

Figure 1.3. Steps of the study

2. LITERATURE REVIEW

In this chapter previous studies are divided into three groups for better understanding. Each part will begin with how the idea of the method started, how it enhanced into today's models. Then the previous studies in the textile industry will be shown for each of the methods. The previous studies of each method will end with a detailed explanation of the study with the highest amount of similarities to this study.

2.1. Artificial Neural Networks

Neural networks idea has been around for more than 60 years. The first neuron idea can be found in a Logical Calculus of Ideas Immanent in Nervous Activity, a paper by Warren S. McCulloch and Walter H. Pitts in 1943. (McCulloch and Pitts, 1943)

Since then it has continuously improved and its content was enriched. Turing (1950) proposed a learning machine idea. This machine was able to learn, therefore was an idea of artificial intelligence (Turing A. M., 1950). In 1951, based on Turing's proposal Marvin Minsky and Dean Edmonds build the first neural network machine. This machine, called SNARC (Stochastic Neural Analog Reinforcement Calculator) had 40 neurons and is still considered as one of the pioneering achievements for artificial intelligence. Kunihiko Fukushima (1980) discovered neocognitron, a hierarchical multi-layered artificial neural network (Fukushima K., 1980). John Hopfield (1982) found Hopfield networks which are a form of recurrent artificial neural network (Hopfield J., 1982).

Hu and Root (1964) used adaptive linear network for weather forecasting in his thesis. Therefore, it can be said that using ANNs for forecasting is quite an old practice (Hu M.J.C., Root H.E., 1964). David Rumelhart, Geoff Hinton and Ronald J Williams (1986) found back propagation (Rumelhart et al., 1986).

ANNs were used for modeling and prediction of nonlinear signal process. This study showed that ANNs can be used for modeling and forecasting of time series (Lapedes A.S., Farber R.M., 1987).

Irie and Miyake (1988) used three layered perceptron show that neural network can approximate a continuous function to a desired accuracy (Irie B., Miyake S., 1988). Also, P. J. Werbos (1988) showed that neural networks with back propagation can outperform traditional statistical methods like regression.(Werbos, P.J.,1988)

Hornik et al. (1989) showed that multilayered feedforward networks were universal approximators (Hornik et al., 1989). In same year Cybenko used sigmoidal functions to achieve the desired accuracy in his paper. (Cybenko G., 1989)

Hornik (1991 and 1993) showed that multilayer feedforward network can achieve any desired accuracy in any continuous function (Hornik K., 1991 and Hornik K., 1993)

Ever since 1990s, models that are based on the structure and function of biological neural networks are increased in number and in variety. According to Artificial Neural Networks, a book by B. Yegnanarayana (2005), the principles of neural network are closely related to such areas as pattern recognition, signal processing and artificial intelligence. In the same book it is said that artificial intelligence developments are while promising, show many inadequacies and brittlements. The need to redesign the problems for heuristic or rule-based approaches caused the development of artificial neural networks. These networks are built to provide new directions to solve problems of natural tasks or to extract the relevant fatures of input data and learn from examples and create a pattern recognition based on those examples (Yegnanarayana B., 2005).

In literature, artificial neural network studies of every sector can be found.

Dalton and Deshmane (1991) used a general neural network model with backward error propagation also known as back propagation to solve a physics problem (Dalton J., Deshmane A., 1991).

Tang et al. (1991) compared ANN models with Box Jenkins method in time series forecasting (Tang et al., 1991). Sharda and Patil (1992) and Tang and Fishwick (1993) also compared ANN models with Box Jenkins method (Sharda R., Patil R.B., 1992 and Tang Z., Fishwick P.A., 1993).

White H. (1992) analyzed single hidden layer feedforward networks' approximation and learning capabilities in his book titled "Artificial Neural Networks: Approximation and Learning Theory" (White H., 1992).

Widrow et al. (1994) studied ANN models' ability to learn and generalize from experience. Many examples were given for a variety of tasks in multiple different fields of businesses (Widrow et al., 1994).

Krogh and Vedelsby (1994) used ANN models as an ensemble to improve accuracy and to provide better active learning. It was shown that the ambiguity, the disagreement among the networks, can be used to select new training data (Krogh A., Vedelsby J., 1994).

Tokar and Johnson (1999) used an ANN model to forecast daily runoff from daily precipitation, temperature and snowmelt. The comparison of ANN model with statistical regression produced results in favor of ANN model (Tokar A.S., Johnson A.J., 1999).

Lek and Guegan (1999) used ANN models as a tool in ecological modeling, since the underlying data relationships were unknown. Both back propagation algorithm and Kohenen self-organizing mapping algorithm was used as learning algorithms (Lek S., Guegan J. F., 1999).

Ohta T. (2000) showed many examples about applications of ANN models in energy systems. (Ohta T., 2000)

Drew and Monson (2000) discussed usage of ANN models in diagnostic millieu. They proposed that the complex interactions of multiple

clinicopathological caused the introduction of analytic tools in diagnostics. ANN models were specifically used for their ability to analyze complex nonlinear datasets (Drew P.J., Monson R.T., 2000).

Khan et al. (2001) used ANN models with gene expression profiling to classify and diagnostically predict cancers. Small round blue cell tumors of four distinct categories were correctly classified by ANN models and the genes most relevant for the classification was identified (Khan et al., 2001).

Ogulata et al. (2006) used multilayer neural network models to classify subgroups of primary generalized epilepsy. The models were found to be successful in classifying multiple seizure epilepsy groups (Ogulata et al., 2006).

Aslan et al. (2009) researched neural network models' ability to predict the prognosis of epilepsy from the underlying etiologic factors. Multilayer neural network models were designed with satisfactory prediction rates (Aslan et al., 2009).

Bataineh et al. (2016) used ANN models to improve predictive capabilities of digital human models, especially in fields of posture and motion prediction (Bataineh H., 2016).

2.1.1 Artificial Neural Networks in Textile Science

The first study of artificial neural networks in the textile sector was the analyzing four different types of fabric and defect detection by ANNs. The defects were accurately found and classified in this study (Tsai et al., 1995).

Ogulata et al. (2006) used ANN and linear resgression models to predict elongation and recovery of bi-stretch fabric. Both of the models showed similar and accurate results (Ogulata et al., 2006)

Gharehaghaji et al. (2007) used ANN models and multiple regression analysis to predict nylon core cotton blend yarn's stress characteristics. Analyzing both of these methods they determined that ANNs gave more accurate results (Gharehaghaji et al., 2007).

Balci et al. (2008a) used ANN models with Levenberg-Marquardt algorithm to predict colorimetric values of stripped cotton fabrics. The number of nodes was changed to see the effect of nodes on the prediction (Balci et al., 2008a).

Balci et al. (2008b) again used ANN models with Lavenberg-Marquadt algorithm and linear regression models to predict CIELab data and wash fastness of Nylon 6,6. Even though ANN models were faster the linear regression models were found to be more accurate (Balci et al., 2008b)

Demiryurek and Koc (2009) studied the determination of yarn characteristics with statistical models and artificial neural networks. In this study, the process was followed from the production of braids including viscose and polyester fibers to the winding cone. The characteristics of these yarns were determined by USTER tool and entered as training and test datasets to artificial neural networks (Demiryurek O., Koc E., 2009).

Balci and Ogulata (2009) compared ANN models against linear regression models to predict changes of CIELab values of fabric after chemical finishing. ANN model was shown to give better performance than linear regression models (Balci O., Ogulata T.R., 2009).

Neural network models were used to predict dimensional properties of cardigan fabric made from cotton ring spun yarns. The prediction of neural network model was shown to be highly accurate (Kumar T.S., Sampath V.R., 2013).

Predicting the tensile strengths of cotton core-spun yarns was another subject of ANN studies. Yarn breaking strength, elongation and rupture of core spun yarns were predicted by both ANN model and linear regression model. Mean square error and coefficient of determination were both used as performance indicators. ANN models were shown to be more accurate in prediction of tensile strengths of cotton core spun yarns (Almetwally et al., 2014).

Another prediction study for neural network models was prediction of yarn strength. Back propagation neural network model was used to predict yarn strength

without physically spinning the yarn. The performance of the model was shown to be reliable and it was applied to a textile company in Italy (Furferi R., Maurizio G., 2010).

Ozkan et al. (2015) used ANN models to predict the intermingled yarn number of nips and nips stability. Feed forward neural network with multiple hidden layers was compared against regression neural network models. Multi-layered feed forward neural network was shown to have better accuracy (Ozkan et al., 2015)

The ANN study with most similarities to this one is the prediction of yarn characteristics by neural networks by Yıldıran Turhan and Ozan Toprakci in 2013. In this study, they compared the prediction performances of high volume instruments (HVI) and advanced fiber information systems (AFIS) by radial functioned neural networks. They developed a neural network model that can predict the tension and hairiness of yarns and this model was fed with the parameter outputs of HVI and AFIS. As a result of this study, it was actualized that the model was working with the same strength with HVI and AFIS and the prediction accuracy was also the same (Turhan Y., Toprakci O., 2013).

2.2. Support Vector Machines

Support Vector Machines although similar to artificial neural networks are quite newer than ANNs. SVMs were initially proposed in 1992 by Boser et al., to make the linear function suitable to the classification and for the situation where the training set can't be linearly separated. They mapped the input data into a Hilbert feature space by a non-linear feature map. This learning method was later called hard margin SVM (Boser et al., 1992).

Another one of their first appearance was in 1995 in a paper by Corinna Cortes and Vladimir Vopnik. They called it soft margin support vector machine. They added slack variables to relax constraints of hyperplanes to make no errors on training dataset. After this they had to add safeguards against getting too large

slack variables which led to quadratic optimization problem (Cortes C., Vopnik V., 1995).

Scholkopf et al. (1997) compared SVM model with a Gaussian kernel, a hybrid system with error back propagation and Radial Basis Functions (RBF) machine against each other for their ability of recognition accuracy of handwritten digits. SVM model was shown to have the superior practical application (Scholkopf et al., 1997).

SVM models just like ANN models were used in time series prediction. In that study, they were trained with 2 different training methods and compared against radial basis functions. It was shown that SVMs had far better performance than RBFs (Muller et al., 1997).

One of the first times SVM models were used for prediction, was for chaotic time series. It was implemented on a database and compared against polynomial techniques, RBFs and neural networks. It was shown that SVMs performance was better than all of the other approaches (Mukherjee et al., 1997).

It was also discussed to train an SVM model, a very large quadratic programming problem needs to be solved. To solve this model, Sequential Minimal Optimization (SMO) is used to break the large problem to smaller problems. Since it doesn't need much memory SMO can work with larger training sets (Platt J.C., Nitschke J., 1998).

SVM models were also used in 3D object recognition. Linear SVM models were used with a database of 7200 images. SVM models excelled at aspect-based recognition (Pontil M., Verri A., 1998).

Yet another use for SVM models was e-mail categorization as in spam or not spam. Three different classification algorithms were tested against SVM, with datasets of 1000 and 7000. It was shown in this study that SVM needed lesser training time and provided acceptable test performance (Drucker et al., 1999).

Since the posterior probabilities of SVMs can't be calibrated, causing post-processing problems, solutions were devised to treat this problem. One method was

to train a kernel classifier with a link function. Another was to train an additional sigmoid function to map SVM outputs (Platt J.C., 1999).

Image classification was another subject of SVM. High dimensionality causes traditional methods to generalize poorly. SVM models were shown to perform far better than RBF and with a simple input remapping they are a valid alternative to RBF kernels (Chapelle et al., 1999).

Kecman V. (2001), in the book entitled “Support Vector Machines”, suggested that SVMs have been developed in reverse order of the development of ANNs. ANNs were more heuristic and went from applications and experiments to theory. SVMs on the other hand started from theory and went to implementations (Kecman V., 2001).

Thorsten Joachims (2002), in his book titled “Learning To Classify Text Using SVM”, showed how to guarantee the lowest probability of error for a given training sample. A new approach to generating text classifiers was also presented (Joachims T., 2002).

Guyon et al. (2002), showed that SVM models were also quite multi-functional. SVM models were used for cancer classification by gene selection. The model proved to yield better and more compact gene subsets and was 98% accurate (Guyon et al., 2002).

Hsu and Lin (2002), chose two methods for multiclass SVM classification methods and compared these methods with three methods of binary classifications. They showed that large problems that took all the data at once need less support vectors (Hsu C.W., Lin C.J., 2002).

SVM models were used to detect faults in rotating machinery. Both SVM and ANN classifiers were compared and genetic algorithm based selection process was implemented to improve the performance of both techniques (Jack L.B., Andi A.K., 2002).

The algorithms behind an SVM application are ever changing; even algorithm comparison studies can be found. As an example, since SVMs have a

problem with learning from unbalanced datasets where negative data outnumbers positive data, there are a couple of options. These are called SMOTE (Synthetic Minority Over Sampling Technique) with different costs and undersampling. According to the study of Akbani et al., SMOTE algorithm outperforms the other two (Akbani et al., 2004).

In another image classification study, SVMs were used in classification of hyperspectral remote sensing images. They were compared with radial basis function neural networks and nearest neighbour classifier and with four different multiclass strategies. It was shown that no mateer what kind of strategy adopted SVMs were an effective alternative to other classifiers (Melgani F., Bruzzone L., 2004).

Even fuzzy memberships were applied to SVMs so that different input points can make different contributions for learning decision. SVM was reformulated into fuzzy SVM (Lin C.-F., Wang S.-D., 2005).

Finley and Joachims (2005) again proved that empirical results support the theoretical findings in text classifiers using SVM models. It was shown that SVM models get substantially better results than the statistical methods and they are quite robust with multiple different learning tasks (Finley T., Joachims T., 2005).

Time series prediction is another multipurpose technique used in a bunch of sectors ranging from financial market prediction to electric utility load forecasting. SVM models were compared in their prediction abilities against other non-linear prediction techniques including multi-layer perceptron neural networks. It was shown that SVM models outperformed other non-linear techniques (Sapankevych N.I., Sankar R., 2009).

SVM models were also developed for land cover classification. Otukei and Blaschke (2010), compared SVM models to popular classifiers at the time like maximum likelihood classifier, decision tree classifiers and neural networks. It was shown that SVMs outperformed the other classifiers at most of the cases (Otukei J.R., Blaschke T., 2010).

Maimon and Rokach (2010), in the book titled “Data Mining and Knowledge Discovery Handbook”, showed that SVM models are a set of supervised learning methods applicable to classification and regression (Maimon O., Rokach L., 2010).

The limits of SVM models were also considered in many articles. One such article investigated the problem of training a SVM on a very large database with a lot of support vectors. Alamili M. (2011) successfully implemented an SVM model with 10000 support vectors to foreign exchange rate problem (Alamili M., 2011).

Steinwart and Christmann (2008) in the book titled “Support Vector Machines”, said that SVM models started to gain popularity because of their solid theoretical foundations. Their emphasis on statistical learning theory increased their momentum causing SVMs to spread more and more to other disciplines such as statistics and mathematics (Steinwart I., Christmann A., 2008).

Support Vector Machines even started to replace artificial neural networks according to “Learning with Kernels” a book by Scholkopf and Smola. They actually described Support Vector Machines as a new class of theoretically elegant learning machines with a central concept of kernels (Scholkopf B., Smola A.J., 2001).

2.2.1. Support Vector Machines in Textile Science

SVMs were used in textile industry for a long time. SVM based framework was used for statistical classification of raw textile defects. Accurate analysis of multiclass classification schemes was achieved. SVM models results were very promising (Murino et al., 2004). Defect detection by SVM models was done with Gabor Wavelet features which are an adaptive selection method. SVM classifier was shown to classify pixels quite effectively (Hou Z., Parker J.M., 2005).

Later on, Single Value Decomposition Analysis with Vector Quantification was used to transform images into features and these features were used as

teaching data for both SVM and ANN. Then these two were compared against each other and it was shown that SVM had better results than ANN (Karras D.A., 2003).

Texture classification of a woven fabric was also a subject of SVMs. One of the studies is about texture analysis based classification. The classifiers were compared against each other. SVM had better results with some of the parameters like homogeneity than others like angular second moment (Salem Y.B., Nasri S., 2009).

SVM models were also used for classification and identification of foreign fibers in cotton. Decision Tree Support Vector Machine (DTSVM) was used to improve the training speed. Over 2% success rate was achieved (Ji et al., 2010).

Another subject that SVM models were used was fabric wrinkle categorization and classification. The study used wavelet transform to decompose images and gave the results to SVM models as parameters. In 300 images, 75% success rate was achieved (Sun et al., 2011).

Ghosh et al. (2011) developed an SVM classifier system to inspect common fabric defects (neps, oil stains). It was found that SVM classifier had a reasonable accuracy (Ghosh et al., 2011).

SVMs were also used in predicting the strength of rotor spun yarn.Nurwaha and Wang (2011) used HVI and advanced fiber information system (Uster AFIS) results as parameters. It was shown that SVM had a good performance compared to ANFIS model (Nurwaha D., Wang X., 2011).

Another subject of SVM studies was the prediction of yarn quality parameters. Combined with genetic algorithm SVM models were shown to be more accurate in small data sets and in real life production (Yang et al., 2012).

The SVM study with the most similarities with this one was the prediction of cotton yarn properties. SVMs were used to forecast properties of rotor and ring produced cotton yarns. Fiber properties from HVI and AFIS were used as parameters. The forecast performances of these models were compared against

ANN models. SVMs showed very high degree of accuracy and better results than their ANN counterparts (Ghosh A., Chatterjee P., 2010).

2.3. Principal Component Analysis

Principal Component Analysis (PCA) has been around for more than 100 years. According to Chatfield and Collins (1980) PCA was first mentioned in a study of Karl Pearson in 1901. It was further developed by Harold Hotelling in 1930 (Chatfield C., Collins A.J., 1980). Several interpretations into the use of PCA in applied research were made in the 1960s. PCA was used for the analysis of multiple measurements (Rao C.R., 1964).

PCA was also used in many different research fields. Size and shape analysis usage of PCA began as early as 1960s. In a study of Walker Museum, biometrical variation of size and shape of Painted Turtle was analyzed. The relationship between principle components and concepts of size and shape was thoroughly examined (Jolicoeur P., Mosimann J.E., 1960). Land cover change was one of the subjects of PCA. It was used on eight dimensional data array. The study shows the difference between overall radiation and atmospheric changes in major components and minor changes in land cover of minor components (Byrne et al., 1980).

The first use of PCA with neural networks was when a linear neuron model was analyzed causing limitless learning rules to be created. It was shown that neuron model extracted principal component from the input vector. Basically, neuron model was used as the analyst of principal component (Oja E., 1982).

In neurophysiology, PCA was used for correlational analysis of data from positron tomography. An intentional brain system was shown by two dependent principal components (Friston et al., 1993). For scheduling, it was used for monitoring batch processes. Multiway PCA was used to get information out of multivariate data. Principle components from that information were used to simplify monitoring charts tracking the progress of batch runs and detecting

observable upsets (Nomikos P., MacGregor J.F., 1994). In chemometrics, PCA was used for disturbance detection and isolation. In that study, PCA was used for statistical process control. Time-lag shift method was added to PCA producing dynamic PCA. It was shown to be quite simple to construct and quite effective (Ku et al., 1995).

Nonlinear PCA (NLPCA) was used to identify both linear and nonlinear correlations between variables. With feed forward neural network nonlinear PCA provided network inputs. It was shown that NLPCA was successful in dimension reduction and the production of feature space map (Kramer M.A., 1991). Since principle curved algorithm used in previous studies can't be used for predicting nonlinear PCAs. This led to integration of neural networks into non-linear PCA and it was shown to have excellent results (Dong D., McAvoy T.J., 1996).

For nonlinear form of PCA, Kernel Principal Component Analysis was developed. Using integral kernel operators, this method was shown to efficiently compute high dimensional spaces. Polynomial feature extraction was used as an example (Scholkopf et al., 1997).

Independent Component Analysis is a linear transformation method that is developed to minimize the statistical dependence of components. It was shown to capture the essential structure of data (Hyvarinen A., 1999).

In chemistry, the polyvinyl alcohol crosslinking induced by ultra-violet was evaluated by PCA. Fourier-transform spectroscopy data was used as original variables. It was shown that ultraviolet radiation causes crosslinking and insolubilization of polyvinyl alcohol (Miranda et al., 2001). In medicine, PCA was used to extrude information from clustering gene expression data. It was shown to degrade quality of the cluster (Yeung K.Y., Ruzzo W.L., 2001).

Different algorithms were included into PCA like expectation-maximizing algorithm. The study used Expectation-maximizing algorithm to predict the principal subspace and showed that there was a significant advantage (Buntine W., 2002).

As a facial recognition kernel PCA was used in nonlinear mapping. Adopting a polynomial kernel it was shown that PCA can deliver the correlations of pixels forming a facial image (Kim et al., 2002).

K-means clustering was also used for dimension reduction with PCA. The study proved that the dimension reduction was related to unsupervised learning. With effective usage of K-means, data clustering the PCA-based data reductions effectiveness was greatly increased (Ding C., He X., 2004).

Since principal components delivered by PCA are only linear combination of original variables to make the results easier to interpret, sparse PCA was developed. This sparse PCA was shown to get encouraging results using appropriate algorithms (Zou et al., 2006).

Gaussian Process was also used with non-linear PCA to embed high dimensional data into a low dimensional one. Called dual probabilistic PCA, this method can transform linear mappings from embedded space into nonlinearised mapping through Gaussian process. The method was compared with kernel PCA and showed satisfactory results (Lawrence N.D., Moore A.J., 2007).

Reflectance spectra of surface colours were yet another subject of PCA. Reconstructing the reflectance spectra required weighted PCA which allowed the user to increase the influence of samples closer to target. The results were shown to be promising improving on earlier methods (Agahian et al., 2008).

PCA can be called correspondance analysis while working with qualitative variables or multiple factor analysis while working with heterogenous or mixed set of variables. They are dependant to singular value decomposition (SVD) matrices (Abdi H., Williams L.J., 2010).

Later on, Principal Component Pursuit was further developed to recover both low rank and sparse components from a data matrix. Robust PCA was later on further developed with robust scatter matrix estimation and projection pursuit ideas. This PCA, called ROBPCA, produced better predicts with both contaminated and non-contaminated data sets. Algorithm was applied to

chemometrics and engineering applications (Hubert et al., 2005). This showed that a robust principal component analysis can be developed that can work even with corrupted data. Face recognition systems was used as an example and it was shown that PCA was able to remove shadows and specularities in face images (Candes et al., 2011).

The formulation of PCA is also upgraded over time. In one such study, authors developed a robust counterpart PCA that can be used to produce sparse solutions. The study also showed that even though variance maximization and minimum error formulations have different objectives, they end up with the same answer (Xanthopoulos et al., 2013).

2.3.1. Principal Component Analysis in Textile Science

In textile industry, PCA was used to transform feature vectors from the neighboring pixels to eigenspace. The transformed features were used to train both neural networks and support vector machine models. The developed models were able to successfully detect defects from images (Kumar A., Shen H.C., 2002).

Independent component analysis was used for defect detection with wavelet transforms. Using wavelet analysis prior to PCA was shown to be advantageous (Serdaroglu et al., 2006).

PCA was also used to predict correlations between fabric parameters and drape. PCA was used to prove the correlations between fabric parameters like rigidity, weight and thickness, and drape parameters. PCA was shown to be an effective way of finding similarities and contrasts between parameters (Thouraya et al., 2014).

PCA was used to calssify textiles based on UV diffuse spectroscopy. Using different natural and synthetic fibers UV diffuse reflectance spectra were analyzed. PCA showed that first components covered 99% of total variability which made UV spectroscopy a rapid real time identification process (Wang et al., 2017).

Predictive logistic modeling of a ring spun process was another subject of PCA. In the study of Alakent and Kunduracioglu Issever, actual data was used for

fiber/yarn quality parameters and process conditions. This was the PCA study with most similarities with the current study. PCA was used to determine subsets of variables which form clusters. Correspondence Analysis was used to determine the relationship between variables and machines. End breakage rate was used as the prediction target. Results show that this combination model was as good as ANN models (Alakent B., Kunduracioglu Issever R., 2016).

2.4. Evolution of Preliminary Work

The studies are shown to be quite close to each other especially in predicting algorithms. For neural networks the closest study to ours to this date is Ghosh and Chatterjee's 2010 study where they used ANNs and SVMs to predict the quality of the cotton yarn properties from fiber characteristics. Yet they didn't use PCA or any other algorithm to reduce the input matrix size. They also didn't study cotton/elastane core yarns. For PCA the study by Alakent and Kunduracioglu is quite close to ours and yet he did not use SVMs or ANNs and instead use Correspondance Analysis.

3. MATERIAL AND METHOD

This part starts with a detailed explanation of the materials used in this study. The materials were provided by KİPAS textile factory. 227 samples of 37 different fibre spinning and quality parameters were gathered with the product quality parameters. This part will also explain the methods used in this study specifically ANN, SVM and PCA. ANN methodology includes the 3 learning algorithms while SVM methodology includes both learning algorithms and solvers used in the study. Each of the methods were given a flowchart explaining the model step by step.

3.1. Materials

The data sets, that will be used in both ANNs and SVMs, are provided by KİPAS a textile factory. The datasets provided by the factory contains 227 samples and are divided into two columns representing different textile quality control machines HVI and AFIS. HVI or high volume instrument is an automated bundle testing fiber quality control machine. It is mainly used to classify cotton and the mixture of fibers in the spinning mill. HVI was developed in late 1960s and became the main source for the understanding of fiber quality and plant selection (Manickam, 2014).

HVI is capable of measuring micronaire, length, length uniformity, strength, colour, trash, maturity and even the sugar content. HVI has several specific parts for calculating these quality parameters. The stelometer is used to check the strength of fiber. Fiber strength, elongation and toughness are all mesured by stelometer. The Uster Evenness Tester checks the yarn evenness. The USTER 1000 which is the model that the factory used to produce the data is the global reference tool for cotton classification. It can measure micronaires, fiber length, uniformity, strength, colour, trash, cotton maturity and sample moisture (Chowdhury, M.,

2015). HVI values in this study were computed by USTER HVI machine shown in Figure 3.1.

Figure 3.1. USTER HVI 1000 (American Plant and Equipment, 2018)

AFIS or advanced fiber information system was produced to substantiate the need for additional information on fibers and for the potential role it will play in programming (Meredith et al. 1996). AFIS was not designed to compete with HVI but rather to provide extensive and unique data on the fiber (Shofner et al., 1998). By providing additional information on length characteristics and fiber maturity, it helps to diversify the variation in fiber lengths and the shape of distribution amongst the fibers therefore creating more uniform length distributions that reduce waste and produce better yarns (Zeidman and Sawhney, 2002). These quality characteristics were calculated using USTER AFIS PRO machine shown in Figure 3.2.

Figure 3.2. USTER AFIS PRO 2 (USTER Tech., 2011)

The quality characteristics of the yarn was calculated by USTER Tensorapid and USTER Tensojet. USTER Tensojet, shown in Figure 3.3. is a unique control system that gives an accurate forecast of yarn behavior in subsequent processing. especially on high performance weaving and knitting machinery.

Figure 3.3. USTER TENSOJET (USTER Tech., 2014)

USTER Tensorapid shown in Figure 3.4 is the most versatile instrument of its type on the market, able to meet all strength testing requirements for both staple and filament yarns. It is acknowledged worldwide as standard equipment for measuring tensile properties, with universal capability for all the bench mark strength and elongation tests required by the textile industry.

Figure 3.4. USTER TENSORAPID

In the dataset there is information about the sample number, the type of machine used to produce the yarn, the type of fiber that is tested (for this study the core spun fibers were used) and the name of both the blend of cotton and the plant aside from the quality characteristics of the fiber procured from both the HVI and AFIS type machines.

The quality characteristics of the fiber are classified with denotations. For HVI quality characteristics, SCI value is Spinning Consistency Index, which is a calculation for predicting the overall quality and spinnablity of the cotton fiber (Majumdar, 2004). Mic value is for Micronaire which is the main indicator of air

permeability, therefore the indicator of both density and the maturity of the fiber. Mat value is for maturity index which is an index for the development of the fiber. It varies even between fibers of the same seed. The maturity doesn't necessarily effect the fineness of the cotton yet the immature fiber will have a lower weight per length since it doesn't have enough cellulose in the fiber. It is hence necessary to check if the observed fineness of a product is due to the immaturity of the fiber or not (Dwivedi R., 2016). Len value is for the length of fiber which is highly correlated with the breakage (strength) of the fiber (Krifa M., 2006). Unf value is for fiber length uniformity index which effects manufacturing waste spinning potential and yarn strength. SFI is for Short Fiber Index which directly effects uniformity index changing yarn strength and spinning potential. SFCn is short fiber count. Usually fibers shorter than 12.7 mm or 0.5 inch are called short fibers (Ramey et al., 1989). Str value is for fiber strength which directly effects yarn strength and can give a quick estimation of yarn strength and yarn twist.(Pan et al., 2001). Elg value is for elongation of fiber which directly effects yarn elongation which influences the performance of yarns during winding, warping and weaving (Majumdar et al., 2004). Rd value is for reflectance which is one of the main components that determine the colour grade of cotton. B value is for yellowness also a main component of the colour grade of cotton (Matusiak et al., 2010). Fiber strength, fiber length, micronnaire, yellowness and uniformity all affect the price of the fiber (Hembree et al., 1986). TrCnt value is trash count and TrArea value is trash percentage area. Cotton trash impacts spinning and causes efficiency problems (Foulk et al., 2006).

Table 3.1 shows the descriptive statistics for HVI fiber characteristics. SCI, Rd and TrCnt all have high std. deviation compared to the rest. It can be said that these characteristics will not be eliminated by PCA. It should also be noted that there is no missing data since each and every characteristic was shown to have 227 different values.

Table 3.1. Descriptive statistics for HVI output of fiber characteristics

Descriptive Statistics						
	N	Minimum	Maximum	Mean	Std. Deviation	Variance
	Value	Value	Value	Value	Value	Value
SCI	227	117.3000	227.0000	140.5200	11.9693	143.2640
Mic	227	3.1800	5.6000	4.5510	.3863	.1490
Mat	227	.8000	.9300	.8880	.01718	.0000
Len	227	28.3000	37.6000	29.1610	.8719	.7600
Unf	227	79.2000	94.2000	83.3030	1.3389	1.7930
SFI	227	6.2000	10.7000	8.6980	1.1951	1.4280
Str	227	28.9000	43.4000	30.5720	1.3069	1.7080
Elg	227	3.5000	7.9000	6.0650	1.0673	1.1390
Rd	227	64.8000	83.6000	73.7110	4.6796	21.8990
B	227	6.6000	11.4000	9.1720	.8185	.6700
TrCnt	227	9.0000	166.5000	74.2980	31.0075	961.4640
TrArea	227	.1000	5.6700	1.1370	.9078	.8240

AFIS outputs are also used with denotations. NepCnt value is the nep count per gram. Nep value is the average size of neps in micrometers. Neps are small entanglement of textile fibers that can't be unraveled causing undyed spot sor unprinted spots. Immature fibers tend to cause more neps than mature ones. SCNCnt value is seed coat nep count per gram. SCN value is seed coat nep size in micrometers. SCN effects spinning quality greatly which makes both their quantity and their size important aspects of the quality of the fiber (Cao et al., 2013). Lwmm value is the average fiber length by weight. LwCV value means length variation by weight. Lnmm value is the average fiber length in millimeters. Mm5 value is 5% length of fibers in millimeters. UQL value is upper quartile fiber length by weight (Ulku et al., 1995). As mentioned above, fiber length directly affects the strength of yarn. SFCw value is short fiber content by weight. Short fibers change yarn

strength and spinning potential (Krifa and Ethridge, 2004). Total count/g value is trash count per 1 g. TrashCnt value is trash count per gram with a size larger than 500 micrometers. As mentioned above, trash count directly affects the spinning parameters and quality of yarn. DustCnt value is the dust count per gram. Dusts are particles with a size shorter than 500 micrometers. Cotton dust causes pulmonary illnesses amongst workers. It increases the chance of bronchitis. Cotton dust also decreases the quality of yarn like trash count (Kennedy et al., 1987). MeanSize value is the average size of fiber. VFM value shows the Visible Foreign Matter (dust and trash percentages) for sample comparison. MatRatio value is the Maturity Ratio of the fiber. FinemTex is the fineness of the fiber in mTex.

Table 3.2 shows the descriptive statistics for AFIS fibre quality characteristics. Without any missing data it is shown that TotalCnt, DustCnt and SCN all have higher standard deviation values than the rest of the characteristics. From this information it can be deduced that these characteristics will probably be chosen by PCA.

Table 3.2. Descriptive statistics for AFIS fibre outputs

Descriptive Statistics						
	N	Minimum	Maximum	Mean	Std. Deviation	Variance
	Value	Value	Value	Value	Value	Value
NepCnt	227	76.0000	622.0000	208.5190	77.7020	6037.6040
Nep	227	641.0000	868.0000	741.4670	46.1824	2132.8150
SCNCnt	227	4.0000	50.0000	26.7110	9.4638	89.5640
SCN	227	800.0000	1390.0000	1232.7290	104.3725	10893.6190
Lwmm	227	21.6000	29.2000	24.3770	1.0998	1.2100
LwCV	227	27.1000	43.8000	34.2170	2.4146	5.8300

Table 3.2. Continued

SFCw	227	4.2000	17.0000	7.7840	1.8789	3.5300
UQL	227	27.2000	35.4000	29.3810	1.2075	1.4580
Lnmm	227	14.6000	22.4000	19.2800	1.2251	1.5010
LnCV	227	40.4000	69.1000	51.4850	3.8553	14.8640
SFCn	227	15.3000	45.4000	25.5420	4.1960	17.6060
Mm5	227	31.0000	40.0000	33.4010	1.3991	1.9580
FinemTex	227	122.0000	187.0000	151.0490	11.2222	125.9380
IFC	227	5.2000	23.9000	12.7910	3.6932	13.6400
MatRatio	227	.7000	1.1000	.8940	.0437	.0020
TotalCnt	227	217.0000	3020.0000	932.1170	456.1523	208074.9280
MeanSize	227	226.5000	401.0000	293.5340	29.5910	875.6260
DustCnt	227	193.0000	2734.0000	817.3030	406.8017	165487.5840
TrashCnt	227	24.0000	306.0000	114.7600	53.9908	2915.0040
VFM	227	.0000	7.8000	2.4690	1.2099	1.4640

Aside from these inputs, the spinning and core spun test results of the fibers are also in the dataset. Spinning the fiber is a part of the process to produce a yarn, so the fiber with better spinning properties produces higher quality yarns. Since ringspun core yarns are used in this study, the test gives the thread number of the yarn and the winding of the said yarn. Spinning value is the spinning ability of the yarn. The core spun test results are there, since core fibers are used. Elastical core spun fibers with lycra dtex results (Lycra Dtex). Take-off value is the strength of the fiber.

Table 3.3 shows descriptive statistics for spinning and core spun quality characteristics. It is clearly shown that Spinning has the largest standar deviation value. Therefore it can be deduced that this quality characteristic probably will not be eliminated by PCA.

Table 3.3. Descriptive statistics for spinning and core spun test results

Descriptive Statistics						
	N Value	Minimum Value	Maximum Value	Mean Value	Std. Deviation Value	Variance Value
Fibre Number	227	8.0000	50.0	17.51	9.5370	90.955
Spinning	227	501.0000	1253.0	718.4	174.707	30522.
Likra Dtex	227	22.0000	78.0000	66.3700	16.3320	266.7600
Take Off	227	3.0000	3.5000	3.3970	.1442	.0210
Ratio	227	2.3000	11.6000	5.3770	2.0381	4.1540

The outputs were also given in the database for teaching and testing. Each of the 227 samples were tested before and after they got processed into yarn. The resulting quality characteristics are all shown in the database. ELG value is the elongation of the produced yarn. RKM value is the breaking point of yarn where the yarn will break under its own weight. RKM is short for Reisskilometer. RKM is basically the tenacity of the yarn (Souid et al., 2012). Um value is the mass of the yarn and CVm value is the mass variations along the yarn. B force is the breaking force of the yarn. The thinness and thickness of the yarn is shown by Thin and Thick values. The rest are the neps which are between 140-200 micrometers (Neps 140) and the neps larger than 200 micrometers (Neps 200). H value is for yarn hairiness.

Table 3.4 shows descriptive statistics for quality characteristics of the cotton/elastane core yarn. Thick, Neps140 and BForce are all shown to have high standard deviation values. It can be deduced that these quality parameters will probably be harder to predict since they are all variable with large variances.

Table 3.4. Descriptive statistics for yarn quality characteristics (Predicted Values)

Descriptive Statistics						
	N	Minimum	Maximum	Mean	Std. Deviation	Variance
	Value	Value	Value	Value	Value	Value
Um	227	7.8600	16.42	10.6096	1.1581	1.3410
CVm	227	9.9000	20.90	13.4476	1.4902	2.2210
Thin	227	.0000	61.00	2.6511	5.8408	34.1160
Thick	227	4.0000	1518.00	121.5621	135.9647	18486.4240
Neps140	227	12.0000	945.00	217.1471	172.5251	29764.9150
Neps200	227	2.0000	190.00	31.5112	30.3832	923.1420
H	227	.0000	9.16	6.5619	1.4104	1.9890
BForce	227	148.6700	1203.00	580.6776	219.6588	48250.0110
ELG	227	5.2000	12.10	7.8816	1.3101	1.7170
RKM	227	10.0800	20.3000	14.3652	1.3484	1.8180

3.2. Neural Networks

This chapter will start with a brief introduction of neural networks. It will then explain the structural parts and types of neural networks. The chapter will end with a detailed description of each learning algorithm used in this study.

3.2.1. A Short Introduction to Artificial Neural Networks

There are problems that can't be formulated as an algorithm. Yet our brains are able to compute such problems with ease. One such problem is predicting quality parameters of fabric from yarn. There isn't a definitive algorithm behind it. yet anyone working in the sector can approximately calculate it. The reason behind this is that humans learn. Computers don't have this capability, even though they are much more powerful than our brain.

The motivation behind neural networks is this capability to learn. The idea was formed so there is no need to explicitly program a neural network, it can learn

from training data or encouragement. This ability to associate data allows neural networks to solve similar problems without explicitly training for them. This increases the degree of tolerance against noise.

Table 3.5. The advantages and disadvantages of neural networks (Tu J.V., 1996)

Advantages	Disadvantages
Compared to other solvers it is easier to use	Can require a lot of training data or cases
Works better with complex problems like image recognition	Sometimes simpler solutions (linear regression) can provide better answers
Works with linear and non-linear functions	Its inner workings are that of a black box. It can't be gleaned or seen.
Mimics human brain to a certain degree	Increasing its accuracy even by a few percent may require a very large amount of data.
Structurally sound	Greater computational burden

As it is shown in the Table 3.5, there are numerous advantages and disadvantages for using a neural network system. The black box means that the inner workings or structure are unknown or unssen, only thing it shows are its responses, also known as practical behaviour.

Exemplary inputs and outputs called training samples are used to show the machine. the desired outputs for the related inputs. These samples are given to neural networks with a learning procedure which is a mathematical formula or algorithm.

There are several aspects of neural network model:

- Neurons: These are the processors of the system
- Activation State: These are the neuron activators equivalent to the output of the neurons. Each neuron with an output has an activator
- Links or Connections: The connections between units of the model generally have their own weight. These weights decide the effect of one unit's signal on another.
- Rule of Propagation: These rules decide the effective input of a unit from external inputs.
- Activation Function: It sets the new level of activation based on effective input.
- External Input: These set the bias of the system.
- Learning Rule: The rule which determines what kind of learning method the system will use to achieve the desired output.
- Environment: The system should operate in an environment providing the necessary inputs. If desired the error signals can also be added to the environment (Rumelhart and McClelland, 1986).

There are three types of processing units in artificial neural networks: Inputs which obtain data from the environment, Outputs which transmit data to environment, Hidden units whose signals remanin within the system (Kröse and Smagt, 1996).

The activity of a neuron is an all or none process which means that either the whole neuron is working and producing an output or it doesn't produce anything at all. If one part of neuron stays inactive the whole neuron stays inactive. It is also important to know that once a structure is set for the network. it does not change no matter the time length for the process (MCCulloch and Pitts, 1943).

Neural networks are classified as feed-forward and recurrent networks. In feed-forward networks, data flow is always in forward direction from inputs to

outputs. As the name suggests there is no feedback, connections from outputs to inputs, from any part of the system.

Recurrent networks have feedback links. These links make a dynamical property which is the structural basis of recurrent networks. Sometimes the activation values of the process go through a relaxation period which doesn't allow the activations to change which makes them stable and unaffecting to the system. Yet in other problems the activation values impact the dynamical behaviour of the whole system (Pearlmutter, 1990).

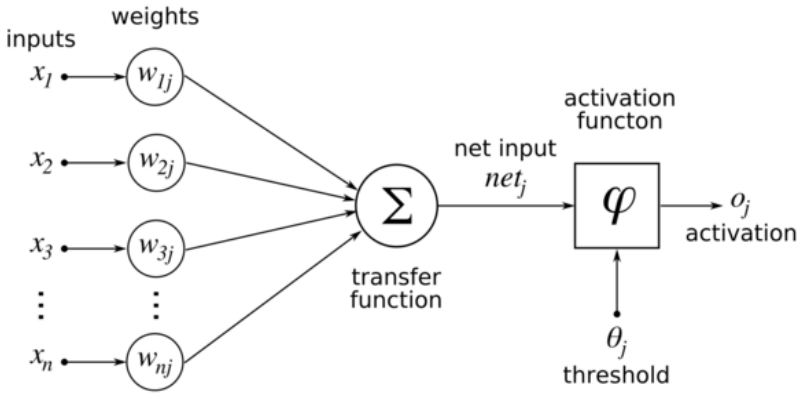

Figure 3.5. The basic structure of a neural network neuron (Kumar et al., 2014)

The basic processor of a neural network system is called a neuron as shown in Figure 3.5. Inputs are connected with weighted link sor synapses to transfer function. Basically an input of a synapse is connected to the neuron with a specific weight. The weight of a synapse in an artificial neuron can be positive and negative. These weights determine the effect of the inputs. Transfer function or the adder then basically summarizes the weighted inputs into a net input. This function is actually a linear combiner which can be effected externally. This effect, called a bias, is meant to increase or decrease the net input. The net input is then put into the activation function or the squashing function. This function basically squeezes the amplitude range of the output to a bounded value. If necessary the lower limit

of an activation requirement can be determined. This limit called threshold is compared against the net input. If the net input is not higher than threshold the neuron will not activate. If it is higher than neuron's activation function will transform the net input into an output (Haykin, 2009).

The oldest activation function is the sigmoid activation function. Sigmoid refers to a mathematical function with an "S" shaped curve defined with the formula:

$$S(t) = \frac{1}{1 + e^{-t}}$$ Eq. (3.1)

The equation 3.1 takes the value of a variable and squeezes it between 0 and 1. This method has several weaknesses:

- If the gradient is very small then the output of the gradient is almost non-existent. Since the sigmoid function squashes the value between 0 and 1, the activation will be concentrated at 0 or 1. If back propagation is used then the local gradient of each neuron would also be multiplied with the output causing an even larger saturation towards 0.
- The inputs are not usulaay processed so they are not zero-centered. Therefore, even if the data is always positive based on the gradient weights of back propagation the results will be all positive or all negative. This will maket he gradient updates difficult to handle.
- The sigmoid function is harder to compute. Since it uses the exponential function in its formula sigmoid function is quite harder to compute than other activation functions.

Since the convergency of the sigmoid function or its overlap is very slow, the initial values of the weights have a larger importance. If they are picked too large the network won't be able to learn (Karpathy A., 2018).

A layer in a neural network includes the weight matrix, the transfer functions, biases, activation functions and the output vector. In some studies inputs are also considered a layer (Hagan et al., 2014).

All in all it can be said that, networks have three elements which have the outmost importance:

- Structure of neurons (nodes)
- The network topology
- Learning algorithm used to train the network (Rojas, 1996)

3.2.2. Artificial Neural Network Types

There are 2 types of artificial neural networks. These are called feed forward and back propagating ANN. In this section both of these ANN models will be explained in detail beginning with feed forward ANN.

3.2.2.1. Feed-Forward Neural Networks

The most common neural network consists of an input level or layer, a hidden layer and an output level. Input level is the place where external data enters the network. Hidden layer is the network's calculation and transformation layer. A network can have many hidden layers, every neural network needs at least one. Output level is where information exits the network.

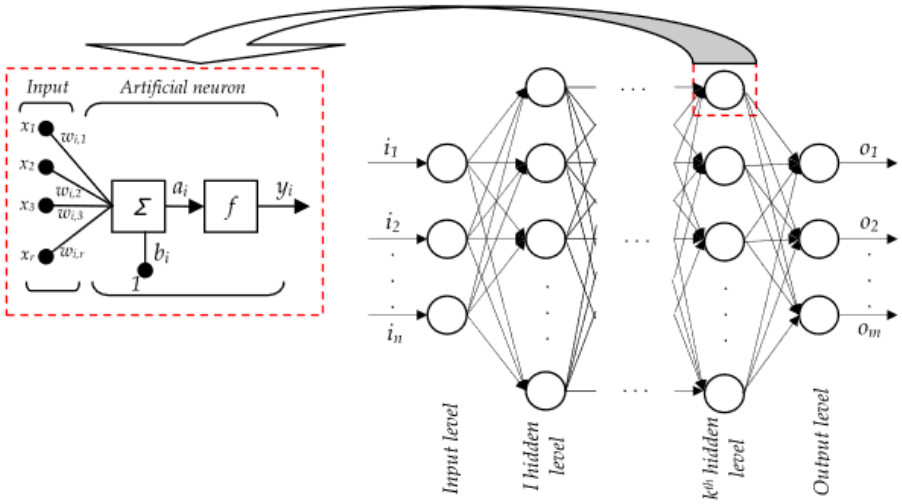

Figure 3.6. Artificial neuron and the structure of feed-forward artificial neural network (Tanikic et al., 2010)

There are two types of neural networks available feed-forward and back propagation neural networks.

Feed-forward neural networks like the one in Figure 3.6 have input, output and hidden layers. The data from the input layer is processed and forwarded to the output layer by the hidden layer. In equation 3.2, If x_i (i=1…..n) are the inputs and w_{jl} are the weights of synapses between hidden neuron *j* and input *l* and k_{jp} are the weights of synapses between hidden neuron *j* and output *p*, then the hidden layer activations h_{xj} and outputs y_{jp} can be found by equations 3.2 and 3.3.

$$h_j(x) = \left(\sum_{l=1}^{nI} w_{jl} x_l + w_{j0} \right) \qquad \text{Eq. (3.2)}$$

$$y_i(x) = \left(\sum_{j=1}^{nH} k_{ij} h_j(x) + k_{i0} \right) \qquad \text{Eq. (3.3)}$$

In equations 3.2 and 3.3 the w_{j0} and k_{i0} values are the biasses or thresholds (Larsen et al., 1996).

3.2.2.2. Back Propagation Neural Networks

Back propagation first found by Rumelhart and McClelland in 1986 was an update of feed-forward ANNs. The layers send their signals forward just like feed-forward ANNs. yet the errors are sent or propagated backwards. Input and output layers stand the same. Just as usual multiple hidden layers are allowed. This algorithm needs supervised learning, needing input and output examples to actualize what needs to be computed in the hidden layer. The difference between the expected value (calculated by ANN) and the actual value is calculated by the system. Back propagation was actually developed to reduce this error. It begins with its weights randomized and this method updates them gradually until it reaches the minimum error. The basic diagram is shown in Figure 3.7.

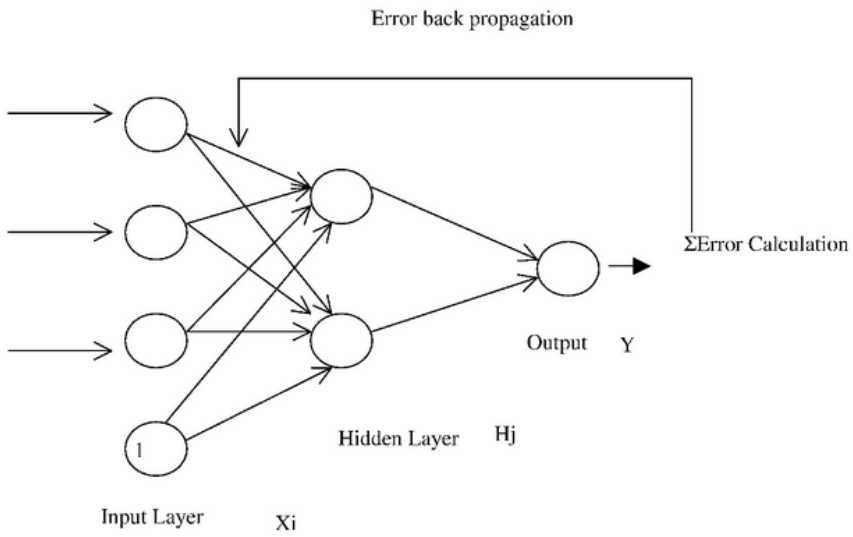

Figure 3.7. Back propagating neural network diagram (Medina. 2014)

To understand the back propagation algorithm, an understanding of the denotations of the algorithm is needed. w_{jk}^{l} is the weight of a synapse between the kth neuron from the layer *l-1* and the jth neuron from layer *l*. For example the weight of a synapse between the fourth neuron of the second layer and the second neuron of the third layer is denoted with w_{24}^{3}. The bias of the jth neuron from the layer *l* is shown with b_{j}^{l}. The activation of the jth neuron in the layer l will be denoted with a_{j}^{l}. Since the activation of a neuron in the layer l is related to the activation of neurons in the layer l-1 the formula can be shown as in equation 3.4.

$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right) \qquad \text{Eq. (3.4)}$$

Since the function σ is applied to every element of the addition the compacted formula can be written as equation 3.5.

$$a^l = \sigma(w^l a^{l-1} + b^l) \qquad \text{Eq. (3.5)}$$

It can be said that if $z^l = w^l a^l + b^l$ then $a^l = \sigma(z^l)$. This z^l is called the weighted input and just like a^l it can be opened as z_j^l.

The key of back propagation is understanding the relation between the weights of synapses and the cost function C. This was done by calculating $\partial C / \partial w$. The cost function is the quadratic formula is shown in equation 3.6.

$$C = \frac{1}{2n}\sum_x \left\| y(x) - a^L(x) \right\|^2 \qquad \text{Eq. (3.6)}$$

In equation 3.6, n is the number of training samples, x is the individual training sample, y is the desired output for the selected sample and L is the total number of layers in neural network. Two assumptions are needed to be made to implement back propagation. First assumption is the average cost function $C = \frac{1}{n}\sum_x C_x$ can be used for all of the individual samples. This is used so that the partial derivatives like $\partial C_x / \partial w$ (single sample derivative) can be shown as $\partial C / \partial w$. With this assumption x can be assumed as a fixed variable which can be dropped to simplify the formula.

Another assumption is to show the cost *C* as the output of the neural network. The quadratic formula of *C* actually can be shown as output of a neural system, so it can be written as an activation formula (Nielsen, 2015).

To find the partial derivatives of back propagation the error formula is used. Shown as ε_j^l, it gives the error (the difference between the expected value and produced value) of jth neuron in the layer *l*. This error changes the network's resulting cost by $\frac{\partial C}{\partial z_j^l}\Delta z_j^l$. The error of a single neuron is $\sigma(\Delta z_j^l)$. Therefore, the cost can be reduced by lowering error. Finally the error of the *j*th neuron in the layer *l* can be shown as in equation 3.7.

$$\varepsilon_j^l = \frac{\partial C}{\partial z_j^l} \qquad \text{Eq. (3.7)}$$

We can expand equation 3.7 to get equation 3.8.

$$\varepsilon_j^L = \frac{\partial C}{\partial a_j^L}\sigma'(z_j^L) \qquad \text{Eq. (3.8)}$$

The σ' shows the speed of reaction from the activation function for the change in z_j^L . The partial derivation calculates the effect of neuron's output on cost function. Therefore, if the output is heavily related to cost function the error will be higher. If the cost function is known there is little computational need will be minimum to calculate error. If the formula from equation 3.8 is rewritten in matrix form it becomes equation 3.9.

$$\varepsilon^L - \nabla_a C \therefore \sigma'(z^L) \quad \text{Eq. (3.9)}$$

$\nabla_a C$ is the vector of derivatives, showing the rate of change of cost against activation functions. If the quadratic formula is used again, since $\nabla_a C = (a^L - y)$ then equation 3.9 becomes equations 3.10 and 3.11.

$$\varepsilon^L = (a^L - y) \therefore \sigma'(z^L) \quad \text{Eq. (3.10)}$$

$$\varepsilon^l = \left((W^{l+1})^T \varepsilon^{l+1}\right) \therefore \sigma'(z^l) \quad \text{Eq. (3.11)}$$

"$\therefore$" is the elemental product of two vectors. This is the reverse calculation of an error. If the error in layer *l+1* is taken with the transpose of the vector of the weight of synapses in the same layer it is actually going backwards in the artificial neural network the resulting formula is shown in equation 3.12.

$$\frac{\partial C}{\partial w} = a_{in}\varepsilon_{out} \quad \text{Eq. (3.12)}$$

The equation is used for the shift in the cost according to the weight in artificial neural network. The *a* is the weighted input activation for the neuron. the

ε represents the error of the weighted neuron output. If the activation is too little the weight's effect on cost is also little which causes the shift in weight during the learning samples minimal. This shows that the neuron's with low activation learns slower than high activation neurons.

It is also shown that a weight in the last layer of artificial neural network becomes a slow learner if the output activation is very low or very high Since sigmoid function becomes very stable along those lines.

To use back propagation algorithm first the activation needs to be set for each training sample then the z for each neuron is calculated just like feed-forward algorithm. Then the output error for each neuron is calculated accordingly. After all this the error of each layer is calculated from output to input layers. The weights are updated using the formula shown in equation 3.13.

$$w^{l}_{new} = w^{l}_{old} - \frac{n}{m}\sum_{x} \varepsilon^{x,l} \left(a^{x,l-1}\right)^{T} \qquad \text{Eq. (3.13)}$$

The back propagation method is also called the gradient descent. The gradient vector points from any significant point of a function to the sharpest ascent showing the norm. It is an estimation of derivative for multi-dimensional functions. Negative gradient is the vector pointing to the sharpest descent. The operator "∇" is used for a specified point gradient of a function.

If the function has n number of dimensions then the gradient has n components. Gradient also means slowly therefore it can be said that this formula. the gradient descent will go to the lowest point o the steepest descent slowly. The speed of the formula is directly proportional to the size of its norm. If the norm is high the descent is sharper and the speed is higher. The gradient descent is very frequently used to solve multitude of optimization problems.

Yet this method has a very big weakness. The gradient can get stuck not in the sharpest descent but on the suboptimal descent (local minimum).

3.2.3. Learning Algorithms

In this study three different kinds of learning algorithms for neural networks are used. These all use the main backpropagation algorithm yet each one amplifies it in a different way. In this chapter all of these learning algorithms will be explained in detail.

3.2.3.1. Levenberg-Marquardt Method

Levenberg Marquardt method is the standard technique for solving non-linear least square problems. Least squares problems arise in the context of fitting a parameterized function to a set of measured data points by minimizing the sum of the squares of the errors between the data points and the function. If the fit function is not linear in the parameters the least squares problem is nonlinear. (Gavin H. P., 2011)

Levenberg-Marquardt (LM) algorithm is preferred due to providing fast convergence and stability in training of artificial neural networks (ANN) (Cavuslu et al..2012).

The Levenberg-Marquardt (LM) algorithm is an iterative technique that locates the minimum of a multivariate function that is expressed as the sum of squares of non-linear actual-valued functions [4. 6]. It has become a standard technique for non-linear least-squares problems [7]. Thickly adopted in a broad spectrum of disciplines. LM can be thought of as a combination of steepest descent and the Gauss-Newton method.

The advantages and the disadvantages of LM method was shown in Table 3.6.

Table 3.6. Advantages and disadvantages of LM

Advantages	Disadvantages
Since it has two directional options at each iteration LM is more robust than Gauss-Newton (Nelles, 2001).	LM is very weak in flat functions or smooth function with vanishing derivatives (Transtrum and Sethna, 2012).
It is faster to converge on the solution than Gauss-Newton and gradient descent methods	If the model has 10 or more parameters LM becomes slower to converge on the optimum value (Waterfall et al., 2006).
It can handle models with multiple free parameters (unknown parameters).	
It can find the optimal solution regardless of the distance from the initial guess.	

If the loss function which is the learning problem main function that is needed to be minimized is shown as Figure 3.8.

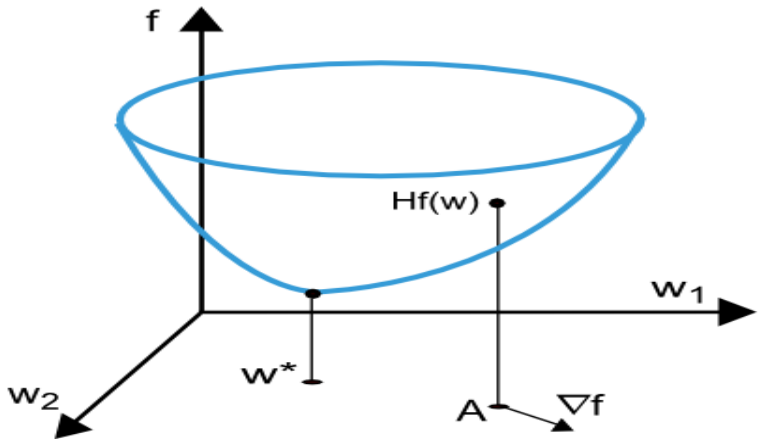

Figure 3.8. Loss function for learning algorithms of ANN (Quesada A., 2019)

Loss function needs adaptive parameters like biases and synaptic weights. As it is shown in Figure 3.8. w^* is the minimum point of the loss function. The first derivatives of the loss function can be shown as in eq 3.14.

$$\nabla_i f(w) = \frac{df}{w_i} \quad i = 1, \ldots, n \qquad \text{Eq 3.14}$$

After these. the derivatives of the loss function can be shown in the Hessian matrix as equation 3.15.

$$H_{ij} f(w) = d^2 f / dw_i . dw_j \quad (i,j = 1, \ldots n) \qquad \text{Eq 3.15}$$

LM which is also known as damped least squares method takes the loss function and turns it into sum of squared errors form. Since it doesn't compute the Hessian matrix but the gradient vector and Jacobian matrix it is slightly faster than other algorithms as shown in Table 3.6.

If the loss function is shown as in equation 3.16 with m as the sample size

$$f = \sum e_i^2 \quad i = 0, \ldots, m \qquad \text{Eq 3.16}$$

The Jacobian matrix can be shown as the matrix containing the derivatives of errors in accordance to the parameters.

$$J_{ij} f(w) = d^2 f / de_i . dw_j \quad (i = 1, \ldots m \ \& \ j = 1, \ldots n) \qquad \text{Eq 3.17}$$

In equation 3.17, n is the number of parameters in the network which makes the size of the Jacobian matrix to be mxn.

The gradient vector of the loss function where e is the error vector can be shown as equation 3.18.

$$\nabla f(w) = 2J^T.e \qquad \text{Eq 3.18}$$

If λ is the damping factor which makes sure the Hessian matrix stays positive and I is the identity matrix then eq 3.19 can be used to approximate Hessian matrix.

$$Hf \approx 2J^T.J + \lambda I \qquad \text{Eq 3.19}$$

The weights of the loss function can be updated per the eq 3.20.

$$w_{i+1} = w_i - \left(J_i^T.J_i + \lambda_i I\right)^{-1}.\left(2J^T.e_i\right) \quad i = 0,1, \ldots \qquad \text{Eq 3.20}$$

If the damping parameter is zero this method is exactly the same as Newton's method yet if the damping parameter is large LM becomes gradient descent method with a smaller training rate.

The initial value of the damping parameter is always large so the first weight updates are in gradient descent. If the iteration fails the damping parameter value is increased. Otherwise. as the loss function value decreases the damping parameter also decreases. As the damping parameter decreases LM approaches Newton method. LM method can be shown as a flowchart as in Figure 3.9.

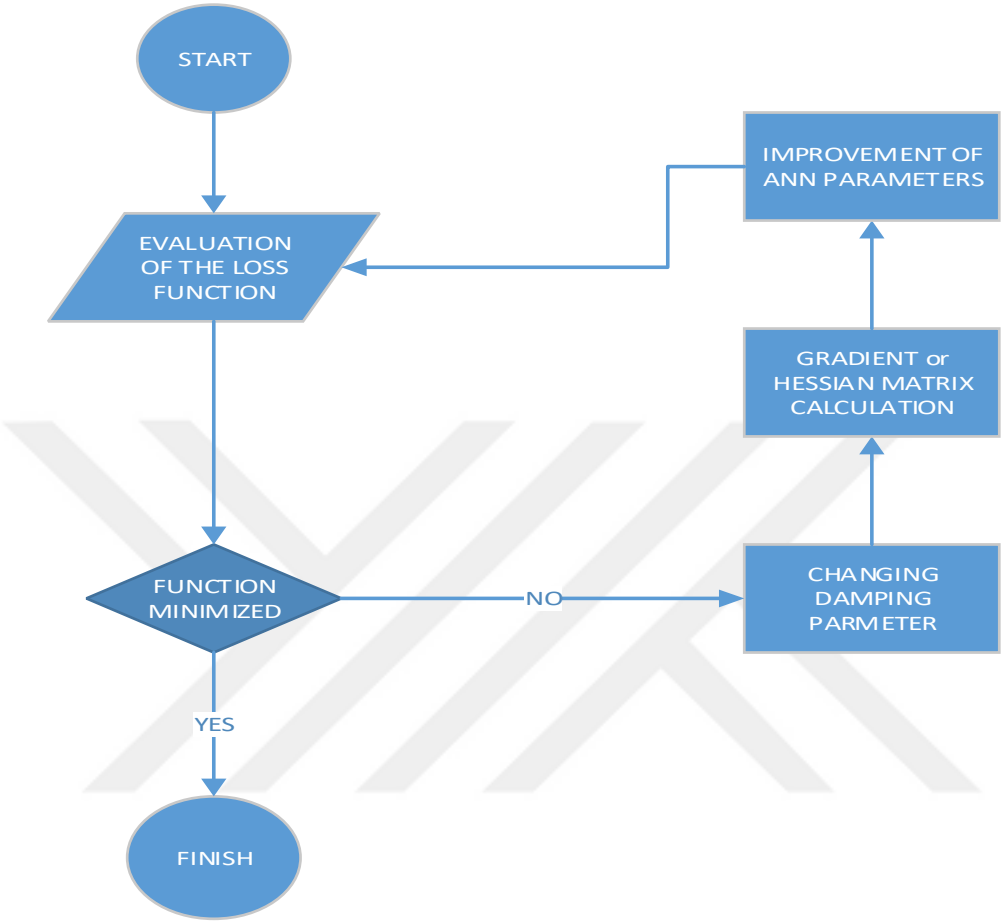

Figure 3.9. The basic flowchart of Levenberg-Marquardt Method

Levenberg-Marquardt memory shows better results than Newton and conjugate gradient methods yet it requires more memory than both of them (Quesada A., 2019).

3.2.3.2. Bayesian Regularization

Bayesian regularized artificial neural networks (BRANNs) are more robust than standard back-propagation nets and can reduce or eliminate the need for lengthy cross-validation. Bayesian regularization is a mathematical process that converts a nonlinear regression into a “well-posed” statistical problem in the

manner of a ridge regression. The advantage of BRANNs is that the models are robust and the validation process. which scales as $O(N^2)$ in normal regression methods. such as back propagation. is unnecessary (Burden and Winkler, 2009).

A BRANN refers to extending standard networks with posterior inference. Standard ANN training via optimization is (from a probabilistic perspective) equivalent to maximum likelihood estimation (MLE) for the weights.

For many reasons this is unsatisfactory. One reason is that it lacks proper theoretical justification from a probabilistic perspective. Using MLE ignores any uncertainty that exists in the proper weight values. From a practical standpoint. this type of training is often susceptible to overfitting.

The correct (i.e. theoretically justifiable) thing to do is posterior inference. though this is very challenging both from a modelling and computational point of view. BRANNs are neural networks that take this approach. (Gordon J., 2017).

Table 3.7. Advatages and disadvantages of Bayesian Neural Networks (Langford J., 2005)

Advantages of BANN	Disadvantages of BANN
Bayesian methods are of an engineering system meaning they build the model and the controller for that model themselves. Specification of a prior is more important than the calculation of a posterior. Other methods never show this kind of an advantage.	Specification of a prior can be a hard to solve problem. Basically a actual number must be used to specify every setting of model parameter.
It has an associated language for connecting priors to posteriors.	It needs a lot more computation so it may be quite slower than other models.
These methods just need the specification of a prior and the integration of that prior to the posterior. These two steps are useful for every method.	Unfortunately one of the advantages is also a disadvantage Since bayesian method needs a controller (engineer) to solve the learning problem.

A detailed description of advantages and disadvantages of a bayesian neural network model are shown in Table 3.7.

Bayesian algorithm starts with a belief called the prior. When the data is obtained. it is used to update the prior which becomes the posterior. As the data accumulates the old posterior becomes the new prior.

$$P(A | B) = P(B | A) * P(A) / P(B) \quad \text{Eq 3.21}$$

Equation 3.21 shows that the probability of A given B is equal to probability of B given A times probability of A over probability of B. This equation is also called the bayesian rule.

If the model parameters (k) and data (D) are included in this model then equation 3.21 comes out as:

$$P(k | D) = P(D | k) * P(k) / P(D) \quad \text{Eq 3.22}$$

All of the components of the equation are probability distributions. P(D) is a probability distribution that can't be calculated yet Since it is a normalizing constant it doesn't effect the model in a significant way. Therefore while calculating the equation the main interest is that of P(k).

P(k) is the prior suggesting what the model parameters might be.The method should converge on a P(k) if the P(k) is not zero. P(D | k) is the likelihood of data given model parameters. Specific to the model it is often used to evaluate the models(Zajac Z., 2016).

P(k | D) is the posterior or the distribution of model parameters from prior and obtained data.

There are two ways for fact checking bayesian regularization. Maximum likelihood estimation (MLE) and maximum a posteriori estimation (MAP). MLE is

generally used to calculate the likelihood of getting model parameter point predicts. MAP takes only the posterior into account.

Since the model is seperate from training the inference allows the model to learn parameters. The most well known inference method is Monte Carlo sampling yet it slows down learning considerably.

Probabilistic machine learning allows the classifier beliefs to be shown as probabilistic predictions.

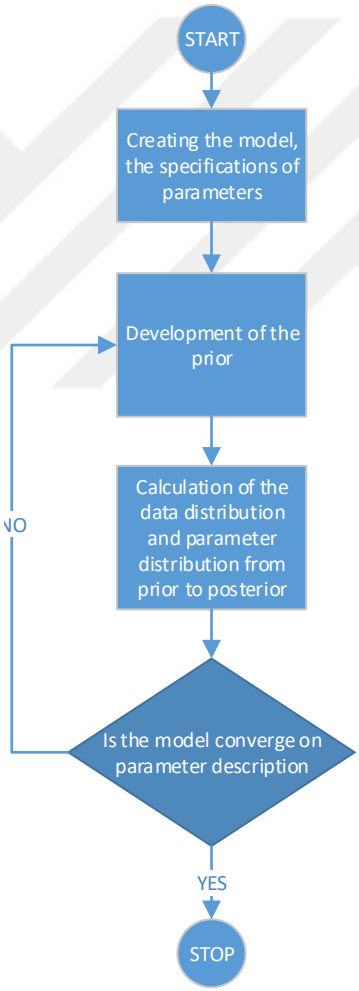

Figure 3.10. Flowchart for Bayesian neural networks

Figure 3.10 shows the basic flowchart for BRANNs. It is shown that the model is stopped when it converges on a single description for all parameters.

3.2.3.3. Scaled Conjugate Gradient

Most of the training algorithms use learning rate to find the extent of the weight update. The learning rate or constant behaves poorly on large scale problems. So the conjugate algorithms use a different step size for every iteration. The conjugate gradient direction effects the step size promptly (Moller M. F., 1993).

Conjugate gradient methods or algorithms are proven to be faster than standard backpropagation algorithms. Yet conjugate gradient algorithms increase the complexity of calculation Since they tend to perform a search amongst the line to calculate the step size. Even a simple search algorithm involves several computations which raises the complexity of the algoritm.

Scaled Conjugate Gradient Algorithm is the combination of Levenberg-Marquardt algorithm and conjugate gradient algorithm. It steers clear from search functions and use Levenberg-Marquardt like approach to find the step size. Even then the scaled conjugate method is shown to be faster yet less accurate than Levenberg-Marquardt (Mishra et al., 2016).

$$E_{qw}(y) = E(w) + E'(w)^T y + \frac{1}{2} y^T E''(w) y \qquad \text{Eq 3.23}$$

In Eq 3.23 the conjugate gradient algorithm is shown as a quadratic function. $E_{qw}(y)$ is the quadratic estimation of a point w. $E'(w)$ is the derivation of $E(w)$ which is the estimation of point w. T is used fot transpose of the matrices. $E''(w)$ is the second derivation of $E(w)$. The crucial points of E_{qw} are found with eq 3.24.

$$E'_{qw}(y) = E''(w)y + E'(w) = 0 \qquad \text{Eq 3.24}$$

After finding the crucial points for the linear system if the solution is considerably simplified it can be said that a conjugate solution is available.

$$y_* - y_1 = \sum_{i=1}^{N} \alpha_i p_i \,, \quad \alpha_i \in \Re \qquad \text{Eq 3.25}$$

In Eq 3.25, p_i is the vector of the conjugate system.

The flow of the sclaed conjugate algorithm can be seen in Figure 3.11.

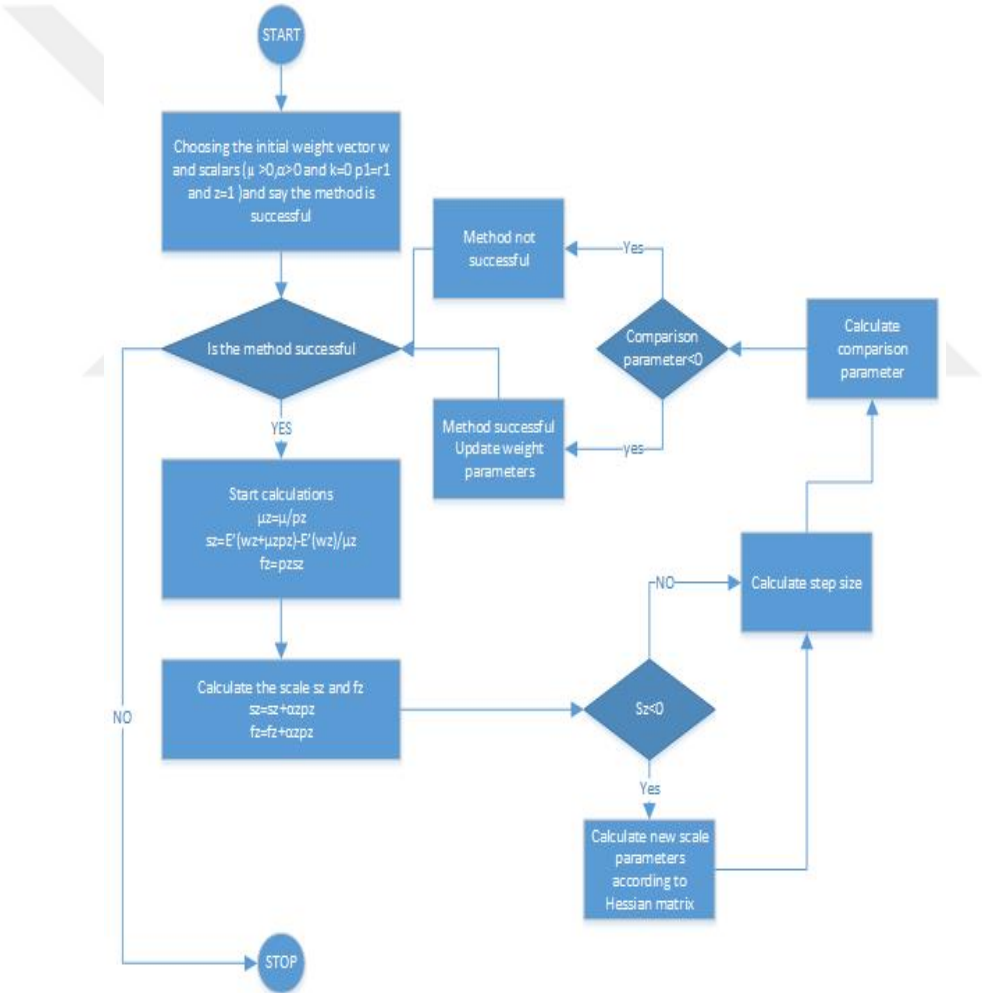

Figure 3.11. Scaled conjugate algoritm flowchart

In this study, back propagation learning neural networks will be compared against support vector machines.

3.3. Support Vector Machines

In this section a detailed explanation of the Support Vector Machines and theorems behind the application will be shown. This section will start with a short introduction to SVM and finish with detailed formulation of SVM models.

3.3.1. A Short Introduction to Support Vector Machines

Support Vector Machine (SVM) is a supervised learning algorithm for classification and/or regression problems. (Ray S., 2017) It is also described as a discriminative classifier defined by a separating hyperplane. This algorithm produces a hyperplane to categorize new examples according to the learning data. (Patel S., 2017)

The basic properties of SVM can be shown as:

- It has the principle basis of statistical learning theory.
- It is extremely practical.(It uses quadratic programming problem which has a single solution)
- In some special cases it may use heuristic algorithm. The choice of different kernel functions provides different structures. (Polynomial classifier, Radial Basis Function classifier, Three layered Artificial Neural Networks)

Support Vector Machine algorithm includes the roots of statistical theory with a large class of neural networks. It includes radial basis function (RBF) networks and polynomial classifiers. SVM is easy to analyze since it fits to a linear method in a multi-dimensional feature space that is related albeit non-linearly to the inputs.

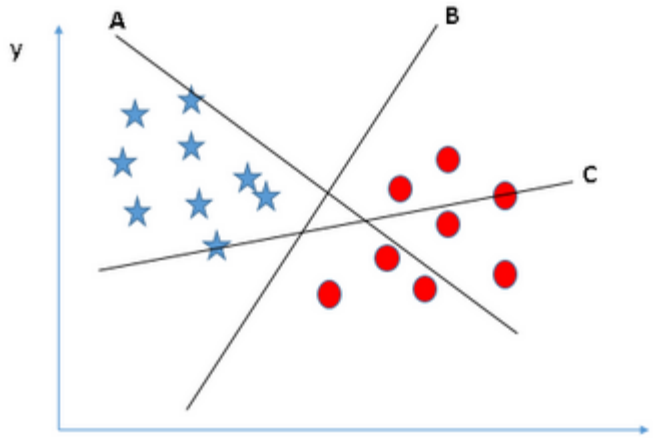

Figure 3.12. Hyperplane drawing (Ray S., 2017)

The algorithm starts by looking for the optimal seperating hyperplane between two classes as shown in Figure 3.12. If the margin between the closest points of the classes is maximized then the hyperplane's position is optimal. To better understand if there is a two dimensional data that needs to be classified the optimal seperating hyperplane is the line that divides the two classes in such a way that the distance between the closest points of these two classes are maximized. This way, the seperator is maximally far away from every data point as shown in Figure 3.13. The distance from the seperating hyperplane to the closest data point is called the margin. The points on the border of the margin are called support vectors. The purpose of this design is to decrease the decision factor parameters of the SVM to a small subset of data. If the margin is set too low than the data points near the margin has a high influence on the classification. Setting the margin high doesn't change the certainty of the system. Hence the classifier is robust, slight errors in measurement or documentation will not cause the system to miscalculate. Therefore SVM classifiers need a large margin around the decision boundary. The data points outside the support vector can be eliminated since they do not effect the model. It works empirically.

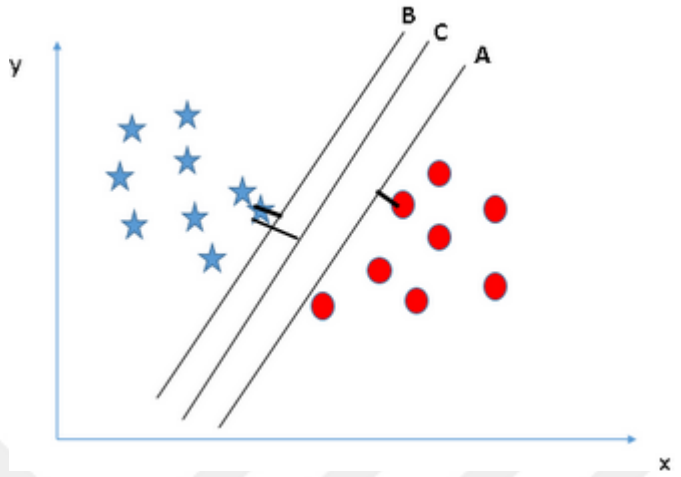

Figure 3.13. Choosing the correct hyperplane(Ray S..2017)

Table 3.8. Advantages and disadvantages of support vector machines against artificial neural networks

Advantages of Support Vector Machines	**Disadvantages of Support Vector Machines**
Artificial neural networks can miss the bigger picture Since they focus on the local (minima) rather than global as support vector machines.	Artificial neural networks are parametrical. their sizes are definite. support vector machines are non-parametrical. the smallest size includes as many support vectors as learning samples size.
If the learning is overdone artificial neural networks can show overfitting. a situation where the model can't tell the difference between the data and the noise. support vector machines can never show overfitting	In online training the feed forward artificial neural networks are faster than online support vector machine.

The advantages and disadvantages of SVM models are shown in Table 3.8. Support vector learning takes the basic ideas as learning from examples and show

their purpose easily. It shows high performance in practical use as well. It acts as a bridge between learning theory and practical use. In some basic applications, statistical learning theory can choose the necessary factors, yet in actual applications, more complex models and algorithms like artificial neural networks make the statistical analysis of the system harder (Hearst, 1998). Support machine algorithm succeeds at both. It constructs models complex enough to solve problems (large neural network classifiers. radial basis function networks and polynome classifiers), while being simple enough to be analyzed mathematically because it can be shown linearly in the feature space that is related non-linearly to the input space.

3.3.2. Formulation of Support Vector Machines

Support vector machines use kernels to directly operate on the input space. This is the characteristic crossness of support vector machines. They help the developers to act like they are using basic linear algorithm while in fact using complex algorithms like pattern recognition, regression or feature extraction.

x_i are the features and y_i are the class labels and N is the size of sample

$$(x_1, y_1), \ldots, (x_n, y_n) \in R^N$$

F function which was developed from the $P(x, y)$ that has the same probability distribution as the teaching data. will classify new (x, y) pairs accurately as $f(x) = y$. If there is no constraint on the class of function classes for an predictd f value. even a function that works accurately onthe teaching data can't generalizeon the data that wasn't shown on the teaching data. If the f function is a complete unknown where even the smoothness of the function is unknown than the features in the teaching data wouldn't have any information about new features.

This makes it nearly impossible to accurately teach the system where the decrease of errors in the learning wouldn't necesserily mean a decrease in the test.

Statistical learning theory (Vopnik-Chernovenkis theory) states that constraining the class of functions get the learning machines to work in a capacity that is appropriate for the learning data on the hand. To develop these algorithms it is needed to have a function class with a calculable capacity. The support vector classifiers in hyperspace are based on the class of equation 3.26.

$$(wx) + b = 0 \quad w \in R^N \; b \in R \qquad \text{Eq. (3.26)}$$

The decision function for this classifier can be shown as in equation 3.27.

$$f(x) = \text{sign}((wx) + b) \qquad \text{Eq. (3.27)}$$

Optimal hyperspace is the plane with the highest margin of distinction between two classes and it has the lowest capacity. In this example *x* are input sample features and *y* values are outputs or results. Each feature has a weight shown with *w*.

Optimal hyperspace is perpendicular to the shortest line connecting the convex shells of two classes and it cuts the line in half. w is the weight vector. b is the threshold as shown in equation 3.28.

$$y_i = ((wx_i) + b) > 0 \qquad \text{Eq. (3.28)}$$

If the formula $|(wx_i) + b| = 1$ in the closest points to the hyperspace is ensured then the hyperspace formula can be shown as $y_i((wx_i) + b) \geq 1$. The

margin is perpendicular to the hyperspace and can be calculated with $\frac{2}{|w||}$. |w| has to be minimized to maximize the margin.

It is needed to solve a constricted quadratic optimization problem to create the hyperspace. In the problem $w = \sum v_i x_i$ is the subset of patterns found on the margin. These patterns are called support vectors and they have all the information necessary for the classification problem.

If the quadratic function part is skipped. it can be shown that both quadratic programming and the last decision function are only related to the datapoints between templates. This is the reason behind the generalisation of nonlinear condition.

The dividing hyperspace with maximum margin is formed by nonlinear extraction of learning data to the high dimensional space by using φ. This forms a non-linear decision border in the input space. It is possible to calculate the hyperspace without extracting it from the feature space by using kernel functions as shown in equation 3.29.

$$\varphi: R^N \xrightarrow{\text{yields}} F$$

$$k(x, y) = (\varphi(x)\varphi(y)) \qquad \text{Eq. (3.29)}$$

The right hand side of this formula will be difficult to calculate if f function is multi-dimensional.

$$k(x, y) = (xy)^d \qquad \text{Eq. (3.30)}$$

Equation 3.30 shows the formula for Polynom Kernels. allowing the user to draw φ in a d dimensıonal plane of R^N. If d=2 and $x, y \in R^2$ then the formulas from equations 3.31 to 3.33 can be shown.

$$(xy)^2 = \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right)^2 \quad \text{Eq. (3.31)}$$

$$\left(\begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}\begin{pmatrix} y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{pmatrix}\right) = (\varphi(x)\varphi(y)) \quad \text{Eq. (3.32)}$$

$$\varphi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \quad \text{Eq. (3.33)}$$

Generally each kernel creates a $(k(x_i, x_j))_{ij}$ matrix and φ can be drawn from these. The resulting function is shown in equations 3.34 and 3.35.

$$k(x, g) - \exp\left(-\frac{|x-y|^2}{(2\sigma^2)}\right) \quad \text{Eq. (3.34)}$$

$$k(x, y) = \tanh(K(x, y + \theta) \quad \text{Eq. (3.35)}$$

Equation 3.34 shows the formula for Radial Basis Function Kernels. Equation 3.35 is the formula for Sigmoid Kernels. K is the gain vector and θ is the offset. In each training example $\varphi(x_i)$ will be changed with x_i and hyperspace algorithm will be implemented. Non-linear function is shown in Equation 3.36:

$$f(x) = \text{sign}\left(\sum_{i=1}^{l} v_i k(x, x_i) + b\right) \quad \text{Eq. (3.36)}$$

In Equation 3.36, v_i is the result of quadratic programming. In input space dividing hyperspace is shown with a non-linear function which has a formula that can only be shown by kernel functions. For estimation a $y \subset R$ can be created. In this situation algorithm creates a linear function in feature space. Just like pattern definition it uses quadratic problem solving algorithm shown in equation 3.37.

$$f(x) = \sum_{i=1}^{l} v_i . k(x_i, x) + b \qquad \text{Eq. (3.37)}$$

l or the upper bound is specified with a fraction of training data points allowed to lie outside the margin.

Support Vector Machines try to minimalize generalized error bound to acquire generalized performance. It is the calculation of linear regression function in the multi-dimensional space, organising the input data with a non-linear function. SVM uses time series, financial estimations and complex engineering analysises in quadratic programming. SVM tries to both minimize the empirical risk and reducing the Vopnik-Cherno (VC) dimension. It uses a learning algorithm that notices the hidden similarities in complex data sets (Ray S., 2017).

The size of VC dimension of most functions is equal to the size of the largest dat set. The largest data set sets the upper limit. Suggesting that $F = \{f(X, W)\}$ function set is put in $\{0, 1\}$ or $\{-1, 1\}$ of R^n points. If it is thought that each of Q points in R^n are assigned either to class zero or class one then there are 2^Q different assignments possible. R^2 has three points therefore eight different assignments. Threshold Logic Neuron (TLN) correctly assesses and classifies these configurations. This can only happen with a carefully positioned correctly oriented hyperplane that has the each point that will be classified as one on the positive side.

The SVM flowchart is shown in Figure 3.14.

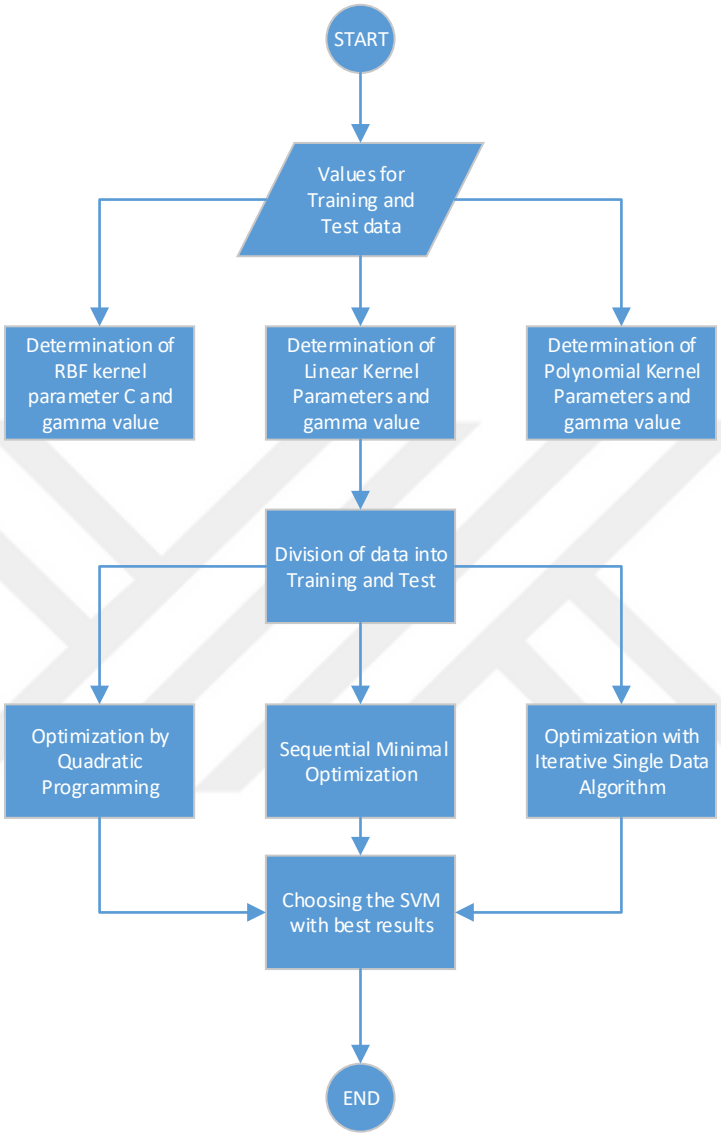

Figure 3.14. SVM Flowchart for the Study

3.4. Principal Component Analysis

In this section, PCA will be explained in detail. It will start with a brief introduction, where the basis of the theories will be explained. This section will end with detailed explanation of eigenvectors and their use in PCA.

3.4.1. A Short Introduction to Principal Component Analysis

Problems with large amounts of input data are hard to show in x. y. z axis. This requirement made the basis for the division of large input data into principal components. Principal components are the main constituents that made up the input data. They can also be described as directions that show the points in which the input data show the largest amount of variance.

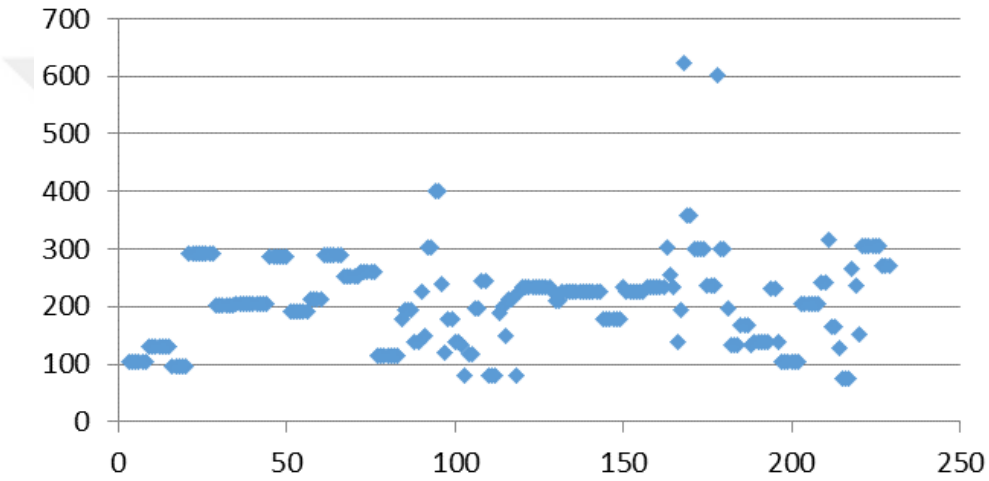

Figure 3.15. An Example Data Distribution

Let us suggest that the Figure 3.15 shows our data on x, y axis. To find the location with the maximum variance, it is needed to find the straight line upon which the data is placed with the largest spread.The vertical line in Figure 3.15, didn't spread the data enough therefore the variance isn't large enough. This suggests the vertical line is probably not a principal component. The horizontal line has a larger spread therefore a much larger variance. It can also be seen that the horizontal line has the largest spread of data. It is this line that is called the principal component.

There are several items that are necessary to produce principal components. They are called eigenvectors and eigenvalues. The data can be divided into eigenvalues and eigenvectors. Each eigenvector has an eigenvalue. The eigenvector

shows the direction of the line. Eigenvalue shows the amount of variance in the respective eigenvectors direction. It basically shows the spread of data on a particular line. The eigenvector with the highest eigenvalue is called the principal component.

The number of eigenvectors is equal to the size of the data. For example there is a research data of gender and internet usage of hours and the subject of the research is the relationship between them. The data set has two dimensions therefore has two eigenvectors.

Since the dataset has only two dimensions the second eigenvector can be placed perpendicularly to the first one. The eigenvectors need to cover the whole x. y area which is the reason behind the need for perpendicularity.

The data does not change at all. It is just shown in a different angle. The eigenvectors allow the users to change their point of view by changing axis. These new axis develop a much better perception about the shape of the data.

The largest variation shows the locations with the highest amount of information about the data. The locations without variation does not give any information about dataset.

The eigenvectors are primarily used for dimensionality reduction. The dimensionality reduction means to remove the unnecessary parts and focus on principal components.

If there is a dataset with 3 dimensions as shown in the graph below, it can be seen that the dataset is spreaded on a surface area. The first two eigenvectors give the length and width of the data yet the third eigenvector does not give any information about the data. It shows the height which can be seen as zero. If this eigenvector is removed than the data can be shown in 2 dimensions again, which reduced a third dimension problem into two dimensions.

The eigenvector does not need to be zero. If the three eigenvectors are 10, 8, 0.1 then it can be said that while 10 and 8 eigenvectors point to large information the 0.1 eigenvector does not contain enough information. Therefore this eigenvector can be removed.

3.4.2. Eigenvectors and Formulation of Principal Component Analysis

Suggest that patterns or features of facial recognition data are given as in above. If n is the number of features f_n is the feature. Since the number of features are too much, there are two questions that need to be answered:

- How many features are to be chosen?
- Which features are to be chosen?

The number of features affects the size of training sample. If too many features are chosen the training sample size should increase accordingly. If the training sample size is kept the same, while the number of features are increased artificial neural network performance drops severely. This is called peaking phenomena. Therefore, keeping the number of features as low as possible without hurting the main objective is the must use tactic.

As a rule, the multiplication of two matrices both of their rows should be equal. The resulting matrice is a new one transformed from the original position. If the resulting matrice is a product of the multiplier matrice this multiplier matrice is an eigenvector. Only square matrices can have eigenvectors yet, not every square matrix has an eigenvector. If there are eigenvectors of a square matrice of nxn then there are n numbers of eigenvectors. If the original matrice is multiplied with a product of eigenvector, the resulting matrice is also a product of same eigenvector Since the product of a vector only changes the intensity of a matrice not the direction. The eigenvectors of a single matrice are perpendicular to eachother no matter the size of a matrice.The eigenvector with a length of 1 is preferable to longer ones Since it is the direction not the length of an eigenvector that is important.

It should also be said that the eigenvectors are easier to find in smaller matrices like 3x3 which means three rows with three columns. Larger matrices need iterative models to find their eigenvectors.

The flowchart of PCA is given in Figure 3.16.

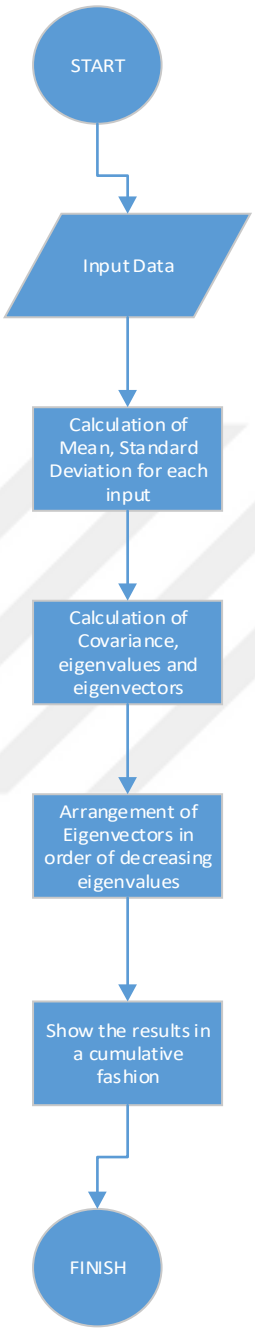

Figure 3.16. Flowchart of PCA

4. RESULT AND DISCUSSION

In this study, both ANN and SVM models were used to predict yarn quality parameters using fibre spinning and quality parameters. Input size was reduced using both PCA and ANOVA. Both reduced and non reduced input matrix was used for predictions, allowing for a comparison between reduced and non reduced input size and their effects on the performance of ANN and SVM. Mean Absolute Percentage Error (MAPE) was used as performance indicator for both ANN and SVM models.

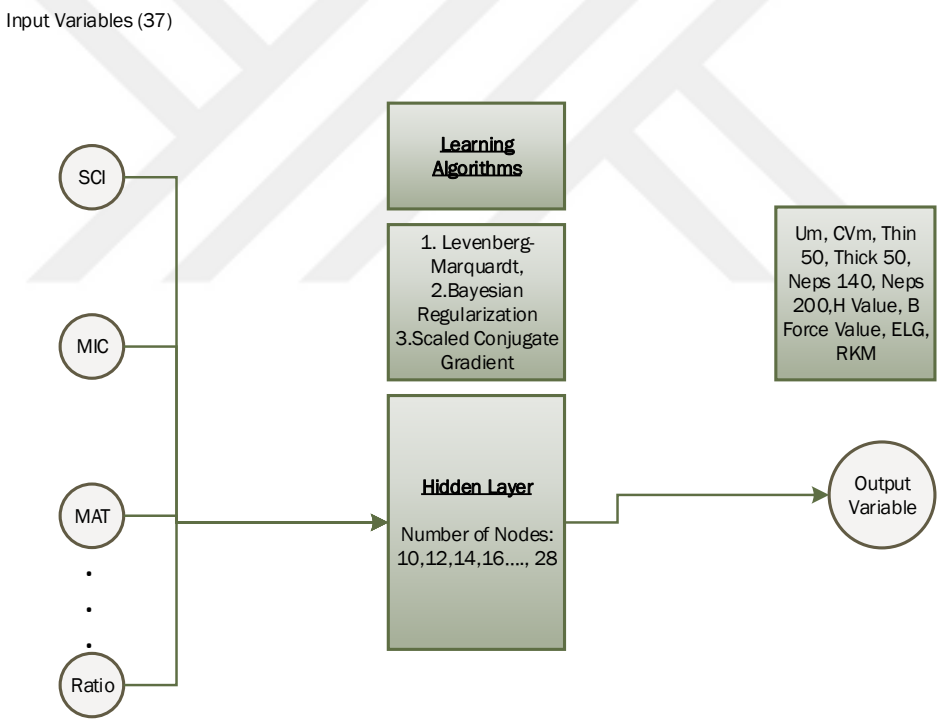

Figure 4.1. The experimental design for ANN (Output)

ANN models are trained with Levenberg-Marquardt, Bayesian Regularization and Scaled Conjugate Gradient learning algorithms. . For ANN models in each set there was 3 different learning algorithms each with 10 different

number of nodes. Since there were 3 sets of input data this makes a total of 90 different experiments just for ANN models. All of these learning algorithms used a single hidden layer with a range of nodes, starting from 10 nodes ending in 28 nodes. Different numbers of nodes were used to find the best result possible for both ANN and SVM models as shown in Figure 4.1 and Figure 4.2

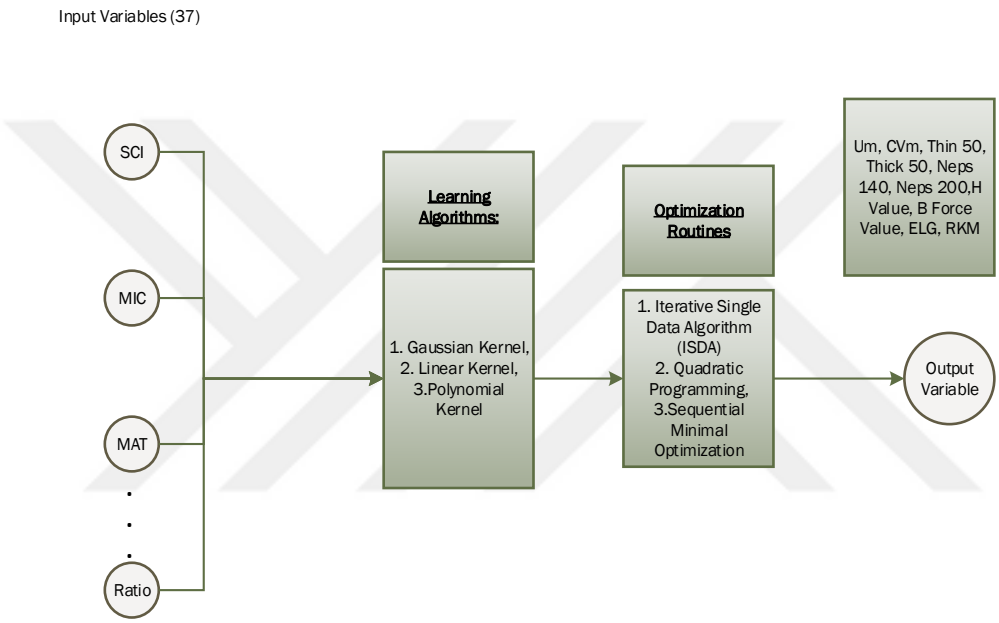

Figure 4.2. Experimental Design for SVM

For SVM there are 3 sets of input data with 3 different learning algorithm each with 3 different solvers. This makes 27 different models for SVM. In total 117 models are analyzed.

4.1. Principal Component Analysis Results

The PCA method allowed us to reduce 37 input variables to 24 input variables. As it was shown in Methodology Eigenvalue Matrix in Table 4.1 shows some of the 37 components for 37 input variables.

Table 4.1. PCA Eigenvalue Matrix

Component		Initial Eigenvalues[a]			Extraction Sums of Squared Loadings		
		Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
Raw	1	377735	88.1090	88.1090	377735.06	88.1090	88.1090
	2	32035	7.4720	95.5810	32034.973	7.4720	95.5810
	3	10088	2.3530	97.9340	10087.972	2.3530	97.9340
	4	4865.580	1.1350	99.0690	4865.580	1.1350	99.0690
	5	1639.157	.3820	99.4510	1639.157	.3820	99.4510
	6	1090.080	.2540	99.7050	1090.080	.2540	99.7050
	7	702.058	.1640	99.8690	702.058	.1640	99.8690
	8	193.993	.0450	99.9140	193.993	.0450	99.9140
	9	140.969	.0330	99.9470			
	10	117.825	.0270	99.9750			
	11	57.128	.0130	99.9880			
	..	..	..	..			
	37	3.525E-8	8.2E-12	100.0000			

Table 4.1 shows the main results of our principal component analysis. This result shows that the 95% of total variance of our input matrix can be explained by two components which means Components 1 and 2 will be used as the main components, 88% of total variance which is the variance of the whole input matrix was included in first component. Similarly, 7% was included in second component.

Table 4.2. PCA Component Matrix of Rescaled Data

Input Variable	Rescaled							
	Components and Coefficients of Variables							
	1	2	3	4	5	6	7	8
1.TotalCnt	**1.0000**	**.0030**	**-.0120**	**.0060**	**.0130**	**-.0100**	**-.0070**	**.0020**
2.DustCnt	**1.0000**	**.0010**	**-.0040**	**-.0010**	**-.0240**	**.0150**	**.0070**	**-.0040**
3.TrashCnt	**.9160**	**.0200**	**-.0670**	**.0570**	**.2920**	**-.1970**	**-.1110**	**.0470**
4.VFM	**.8880**	**.0000**	**-.0760**	**-.0400**	**.3610**	**-.0560**	**-.0520**	**.0090**
5.Rd	**-.4810**	**.1090**	**.1110**	**-.0650**	**-.0230**	**.3000**	**.0540**	**-.1110**
6.Spinning	**.0190**	**.9910**	**.1330**	**.0280**	**.0050**	**.0000**	**.0050**	**.0030**
7.Thread	**.0190**	**.9780**	**.1520**	**.0110**	**.0070**	**.0250**	**-.0080**	**-.0410**
8.Ratio	**.0500**	**.8800**	**.0540**	**.0200**	**.0500**	**-.0950**	**.0910**	**.3440**
9.Str	**.0140**	**.4120**	**.1140**	**-.0220**	**-.0450**	**.1960**	**-.0740**	**-.0380**
10.b	.0850	-.1830	.2150	.1280	-.2100	.0400	.1600	-.0360
11.Len	-.0140	.1930	.0870	-.1470	.0190	.3160	-.1330	.0150
12.Elg	**.1100**	**-.2240**	**.0900**	**.1830**	**-.0340**	**.0350**	**.0110**	**-.0790**
13.SCN	**.2100**	**-.4040**	**.8590**	**.2290**	**-.0360**	**-.0360**	**-.0170**	**.0010**
14.TrArea	**.1640**	**-.1320**	**-.5300**	**-.2310**	**-.0340**	**-.1410**	**.2640**	**.0920**
15.NepCnt	-.190	.027	-.436	.832	-.005	.050	.017	-.002
16.SFCn	**-.2850**	**.1270**	**-.2440**	**.6370**	**-.1780**	**.0230**	**.0740**	**.0510**
17LnCV	**-.2760**	**.1450**	**-.2140**	**.6350**	**-.2030**	**.0440**	**.0560**	**.0200**
18.SFCw	-.1860	.1080	-.2370	.5890	-.2390	.0240	.0750	.0750
19.LwCV	**-.2630**	**.1180**	**-.1250**	**.5480**	**-.4330**	**.0110**	**.0800**	**.0480**
20.Lnmm	.1880	.0140	.2230	-.5000	.2150	.1970	-.1580	-.0730
21.Mic	-.0890	.0350	.1580	-.4450	.3410	-.0680	.2590	-.2520
22.SFI	**.2010**	**-.1310**	**-.1140**	**.3920**	**-.2500**	**.0970**	**-.1120**	**.1270**
23.Mat	**-.0980**	**.0960**	**.2890**	**-.3600**	**.1400**	**.0220**	**-.0050**	**-.1900**
24.IFC	-.1990	-.0010	-.1720	.3420	-.3400	-.1770	.2250	.1830
25.MeanSize	-.1700	.0040	-.0700	.1470	.7770	-.3290	-.3050	-.1030
26.FinemTex	**.2020**	**-.1030**	**-.1050**	**-.4320**	**.5840**	**.1770**	**.0600**	**-.1370**
27.SCNCnt	**.3990**	**-.2370**	**.0810**	**.3410**	**.5680**	**.2740**	**.0140**	**.0610**
28.Nep	**.5040**	**-.2500**	**.3720**	**-.0400**	**.5360**	**.4650**	**.1960**	**.0190**
29.MatRatio	**.2970**	**-.0510**	**.0720**	**-.3520**	**.4000**	**-.0530**	**.3200**	**-.1330**
30.Unf	.1730	.0400	.0720	-.1920	.1990	-.0770	-.0970	.1070
31.UQL	**.1150**	**.2310**	**.0810**	**-.0470**	**.0430**	**.3550**	**-.2240**	**-.0640**

Table 4.2. Continued

32.Mm5	**.0970**	**.2390**	**.1130**	**-.0100**	**-.0730**	**.3180**	**-.2620**	**-.0580**
33.Lwmm	**.2160**	**.1180**	**.1660**	**-.2820**	**.1660**	**.2920**	**-.1690**	**-.0960**
34.TrCnt	**.3900**	**-.1310**	**-.0180**	**.0620**	**.1680**	**-.5630**	**.6880**	**-.0640**
35.SCI	**.2850**	**.0930**	**.1040**	**.1120**	**-.0530**	**.1860**	**-.3110**	**.1570**
36.LikraDtex	.1000	-.1580	-.1850	-.0030	.0300	-.2050	.1780	.7790
37.Take Off	.1000	-.1580	-.1850	-.0030	.0300	-.2050	.1790	.7780

Table 4.2. shows the effect of each fibre quality parameter on principal components. For this study any weights below 0.2 was disregarded. This allowed for an effective cut of input parameters. For negative weights any value above -0.2 was disregarded since it has little effect on the resulting component.

Rescaled data in Table 4.2 was used to analyze the effect of input variables on principle components. As a result, PCA cuts 13 input variables from the system allowing for a reduced input matrix which may help to reduce computational loads of ANN and SVM models. Input variables were included or excluded according to their weights in principal component 1 and 2. Weights less than 0.2 were excluded so PCA can reduce input dimension better. SCI, Unf, SFI, Elg, Rd, TrCnt, TrArea, Nep, SCN, FinemTex, MatRatio, TotalCnt, DustCnt, TrashCnt, VFM were included by Principal Component 1. Lwmm, LwCV, LnCV, SFCn, Mm5, Thread, Spinning and Ratio are included by Principal Component 2. The rest of the input variables are excluded by these two components.

4.2. ANOVA Input Matrix Reduction

Input dimension reduction study by ANOVA was based on the input variables effect on different output variables. Taking 99% confidence rate, any significance value (P Value) that is more than 0.01 was regarded as an input variable that doesn't have a statistically significant effect on the mentioned output variable. The significant input variables are highlighted in bold in all ANOVA tables.

Table 4.3. ANOVA results for core yarn characteristic Um

Output	Input	F Values	P Values	Input	F Values	P Values
Um	**SCI**	2.0063	0.0008	**SFCw %12.7**	1.8815	0.0019
	Mic	2.3916	6.23E-05	**UQL(w) (mm)**	1.7227	0.0039
	Mat	2.9278	0.0012	**L(n) (mm)**	10.3058	3.49E-26
	Len	5.9250	0.0031	**L(n) %CV**	1.7328	0.0047
	Unf	5.9250	0.0031	**SFC(n) %12.7**	1.7419	0.0034
	SFI	2.1015	0.0014	**5.0% (mm)**	1.7008	0.0059
	Str	1.9222	0.0049	**Fine mTex**	1.7682	0.0031
	Elg	2.2507	0.0005	IFC	1.2721	0.1464
	Rd	1.8433	0.0023	Mat Ratio	0.7313	0.5714
	B	1.3101	0.1419	Total Cnt/g	1.2212	0.2059
	Tr Cnt	1.9651	0.0005	Mean Size	1.5424	0.0242
	Tr Area	1.8413	0.0064	**Dust Cnt/g**	1.7136	0.0042
	Nep Cnt/g	1.6610	0.0068	**Trash Cnt/g**	1.6402	0.0089
	Nep (um)	1.7836	0.0026	**VFM %**	1.8914	0.0014
	SCN Cnt/g	1.6676	0.0097	Thread Number	1.5228	0.0269
	SCN (um)	1.6587	0.0067	Spinning	1.5327	0.0337
	L(w) (mm)	14.8202	2.13E-22	Lycra Dtex	1.5094	0.0465
	L(w) %CV	1.7211	0.0050	Take Off	1.6999	0.0167
				Ratio	8.4469	2.54E-17

Table 4.3 allows us to see the effect of each and every fibre quality characteristic on the core yarn characteristic Um. 28 out of 37 input variables were found significantly on the value of Um according to ANOVA results.

Table 4.4. ANOVA results for core yarn characteristic CVm

Output	Input	F Values	P Values	Input	F Values	P Values
CVm	**SCI**	2.0713	0.0005	**SFCw %12.7**	1.7524	0.0038
	Mic	2.4490	3.94E-05	**UQL(w) (mm)**	1.7800	0.0024
	Mat	3.0685	0.0007	**L(n) (mm)**	1.9259	0.0013
	Len	6.3648	0.0020	**L(n) %CV**	1.7701	0.0034
	Unf	1.5284	0.0417	**SFCn %12.7**	1.8012	0.0020
	SFI	2.1569	0.0009	**5.0% (mm)**	1.5884	0.0174
	Str	2.0027	0.0030	**Fine mTex**	1.8252	0.0019
	Elg	2.3151	0.0003	IFC	1.5616	0.0281
	Rd	1.9184	0.0012	Mat Ratio	0.7189	0.5797
	B	6.3648	0.0020	Total Cnt/g	1.2255	0.2020
	Tr Cnt	2.0307	0.0003	Mean Size	1.3056	0.1224
	Tr Area	1.9075	0.0041	**Dust Cnt/g**	1.7722	0.0025
	Nep Cnt/g	1.70946	0.0045	**Trash Cnt/g**	1.6896	0.0060
	Nep (um)	1.8427	0.0015	**VFM %**	1.9471	0.0009
	SCN Cnt/g	1.7336	0.0059	Thread Number	1.3056	0.1224
	SCN (um)	1.7070	0.0045	Spinning	10.316	3.30E-26

Table 4.4. Continued

	L(w) (mm)	1.5791	0.0180	Lycra Dtex	1.7712	0.0108
	L(w) %CV	1.7738	0.0032	Take Off	1.3589	0.1126
	SFC w	8.9749	2.06E-18	**Ratio**	16.3270	2.20E-24

Table 4.4 allows us to see the effect of each and every fibre quality characteristic on the core yarn characteristic CVm. 29 of the 37 input varables was statistically significant on the value of CVm according to ANOVA.

Table 4.5. ANOVA results for core yarn characteristic Thin-50

Output	Input	F Values	P Values	Input	F Values	P Values
Thin -50	SCI	1.0117	0.4605	UQL(w) (mm)	1.2075	0.2185
	Mic	0.8586	0.7048	L(n) (mm)	0.6409	0.9475
	Mat	0.7023	0.7357	L(n) %CV	0.9222	0.6235
	Len	0.8520	0.6936	SFCn %12.7	0.8130	0.8160
	Unf	1.1111	0.3219	5.0% (mm)	0.8582	0.7244
	SFI	1.1317	0.3016	**Fine mTex**	4.1451	7.67E-06
	Str	0.9901	0.4860	IFC	0.7201	0.9025
	Elg	0.8240	0.7295	Mat Ratio	0.1958	0.9403
	Rd	3.0683	2.09E-06	Total Cnt/g	0.8007	0.8352
	B	0.4891	0.9889	Mean Size	0.7009	0.9323

Table 4.5. Continued

	Tr Cnt	0.8262	0.7891	Dust Cnt/g	0.8008	0.8351
	Tr Area	0.7778	0.7983	Trash Cnt/g	0.7472	0.8933
	Nep Cnt/g	0.8447	0.7673	VFM %	0.7170	0.9117
	Nep (um)	0.8501	0.7550	Thread Number	0.7717	0.8656
	SCN Cnt/g	0.9653	0.5413	Spinning	0.8574	0.7279
	SCN (um)	0.7790	0.8637	Lycra Dtex	1.4346	0.2403
	L(w) (mm)	0.6667	0.9484	Take Off	1.4346	0.2403
	L(w) %CV	0.8638	0.7270	**Ratio**	3.5754	2.60E-06

Table 4.5 allows us to see the effect of each and every fibre quality characteristic on the core yarn characteristic Thin-50. Only 3 out of 37 variables were statistically significant effect on the value of Thin -50 according to ANOVA.

Table 4.6. ANOVA results for core yarn characteristic Thick+50

Output	Input	F Values	P Values	Input	F Values	P Values
Thick +50	SCI	1.3018	0.1213	SFCw %12.7	0.8732	0.6868
	Mic	1.3902	0.0793	UQL(w) (mm)	0.8954	0.6323
	Mat	2.4088	0.0076	L(n) (mm)	1.0631	0.3821
	Len	1.3439	0.1184	L(n) %CV	1.1815	0.2143
	Unf	1.0441	0.4107	SFCn %12.7	1.1795	0.2100
	SFI	42.5561	4.96E-50	5.0% (mm)	1.1342	0.2773
	Str	1.2847	0.1621	Fine mTex	1.1482	0.2511

Table 4.6. Continued

	Elg	1.4177	0.0842	IFC	1.0967	0.3292
	Rd	1.2742	0.1330	Mat Ratio	0.3194	0.8647
	B	20.0195	3.43E-44	Total Cnt/g	1.1544	0.2395
	Tr Cnt	1.2724	0.1262	Mean Size	1.2104	0.1822
	Tr Area	1.4427	0.0696	Dust Cnt/g	1.1497	0.2455
	Nep Cnt/g	6.4051	0.0019	Trash Cnt/g	1.1038	0.3124
	Nep (um)	6.4051	0.0019	VFM %	1.0824	0.3482
	SCN Cnt/g	1.2537	0.1508	Thread Number	1.5101	0.0520
	SCN (um)	1.1016	0.3131	Spinning	0.9927	0.4833
	L(w) (mm)	1.0173	0.4525	Lycra Dtex	1.1843	0.2044
	L(w) %CV	1.1356	0.2694	Take Off	1.2021	0.1871
				Ratio	10.5649	1.46E-21

Table 4.6 allows us to see the effect of each and every fibre quality characteristic on the core yarn characteristic Thick+50. Only 6 input variables out of 37 were found to have a statistically significant effect on the value of Thick +50.

Table 4.7. ANOVA results for core yarn characteristic Neps+140

Output	Input	F Values	P Values	Input	F Values	P Values
Neps +140	**SCI**	3.4390	4.61E-09	**UQL(w) (mm)**	2.2164	0.0005
	Mic	2.6625	7.01E-06	**L(n) (mm)**	2.2647	0.0001
	Mat	3.0547	0.000802	**L(n) %CV**	3.7550	4.33E-11
	Len	2.8437	6.52E-06	**SFCn %12.7**	4.6353	5.02E-15
	Unf	4.0906	3.16E-10	**5.0% (mm)**	3.1608	2.42E-08
	SFI	3.6142	3.08E-08	**Fine mTex**	4.8647	1.21E-15
	Str	2.1941	0.0008726	IFC	0.3194	0.864778
	Elg	4.3000	2.20E-10	Mat Ratio	0.8028	0.5244858
	Rd	1.2847	0.1621749	Total Cnt/g	1.1482	0.2511622
	B	2.4091	0.0001759	Mean Size	1.2724	0.1262368
	Tr Cnt	4.9646	6.39E-16	**Dust Cnt/g**	4.5988	5.83E-15
	Tr Area	4.0214	7.64E-10	**Trash Cnt/g**	3.2503	2.96E-09
	Nep Cnt/g	4.6478	4.49E-15	**VFM %**	2.6957	1.22E-06
	Nep (um)	4.8467	1.15E-15	Thread	1.4427	0.0696
	SCN Cnt/g	4.1278	4.59E-12	Spinning	1.1016	0.3131
	SCN (um)	3.9105	3.16E-12	Lycra Dtex	3.6736	0.0269
	L(w) (mm)	2.1687	0.0001	Take Off	3.6736	0.0269
	L(w) %CV	3.2257	5.00E-09	**Ratio**	4.8735464	2.22E-09
	SFCw %12.7	2.2557233	0.0001457			

Table 4.7 allows us to see the effect of each and every fibre quality characteristic on the core yarn characteristic Neps +140. 28 input variables were found to be statistically significant over the value of Neps +140.

Table 4.8. ANOVA results for core yarn characteristic Neps +200

Output	Input	F Values	P Values	Input	F Values	P Values
Neps +200	**SCI**	2.0696	0.0005	**UQL(w) (mm)**	2.8262	5.98E-06
	Mic	3.0218	3.68E-07	**L(n) (mm)**	2.6235	9.62E-06
	Mat	3.7861	1.27E-08	**L(n) %CV**	3.8895	1.27E-11
	Len	1.9796	0.0028	**SFC(n) %12.7**	4.0015	1.58E-12
	Unf	3.9824	7.17E-10	**5.0% (mm)**	3.7277	1.53E-10
	SFI	5.1366	6.11E-13	**Fine mTex**	4.3254	1.39E-13
	Str	2.4210	0.0001	IFC	1.1016	0.3131
	Elg	4.0154	1.69E-09	Mat Ratio	1.1218	0.3470
	Rd	3.1764	1.76E-08	Total Cnt/g	1.4427	0.0696
	B	2.5378	7.16E-05	Mean Size	3.6736	0.0269
	Tr Cnt	4.0929	1.37E-12	**Dust Cnt/g**	3.9586	2.01E-12
	Tr Area	4.3792	5.40E-11	**Trash Cnt/g**	2.1577	9.75E-05
	Nep Cnt/g	4.0404	1.10E-12	**VFM %**	2.3890786	1.98E-05
	Nep (um)	4.2070	3.35E-13	Thread Number	1.3101	0.1419

Table 4.8. Continued

	SCN Cnt/g	3.1700	2.23E-08	Spinning	2.2214	0.0144
	SCN (um)	3.6953	2.41E-11	Lycra Dtex	0.3272	0.7212
	L(w) (mm)	2.1556	0.0001	Take Off	0.3272	0.7212
	L(w) %CV	3.6253	1.19E-10	**Ratio**	4.4217	2.58E-08
	SFC(w) %12.7	2.0382	0.0008			

Table 4.8 allows us to see the effect of each and every fibre quality characteristic on the core yarn characteristic Neps +200. 29 of the input variables were shown to be statistically significant over the value of Neps +200.

Table 4.9. ANOVA results for core yarn characteristic H

Output	Input	F Values	P Values	Input	F Values	P Values
H	**SCI**	5.3948	4.61E-16	**UQL(w) (mm)**	6.7916	2.77E-18
	Mic	4.1293	4.28E-11	**L(n) (mm)**	5.0140	4.01E-14
	Mat	3.0333	0.0008661	**L(n) %CV**	6.7175	9.49E-22
	Len	4.5440	2.48E-11	**SFC(n) %12.7**	7.0401	1.92E-23
	Unf	5.0651	2.32E-13	**5.0% (mm)**	7.8992	1.26E-24
	SFI	5.5529	3.53E-14	**Fine mTex**	8.5483	1.20E-27
	Str	6.3998	2.46E-16	IFC	1.1016	0.3131
	Elg	4.05013	1.32E-09	Mat Ratio	0.5169	0.7233
	Rd	7.4784	1.51E-23	Total Cnt/g	0.8240	0.7295
	B	6.6898	1.97E-17	Mean Size	1.4177	0.0842

Table 4.9. Continued

	Tr Cnt	8.9724	1.04E-28	**Dust Cnt/g**	8.0914	9.66E-27
	Tr Area	7.1088	3.56E-19	**Trash Cnt/g**	8.7614	3.02E-28
	Nep Cnt/g	7.4375	1.10E-24	**VFM %**	9.5442	1.60E-29
	Nep (um)	7.9990	3.58E-26	Thread Number	0.9905	0.4860
	SCN Cnt/g	9.7274	1.20E-29	Spinning	3.6736	0.0269
	SCN (um)	8.1523	6.41E-27	Lycra Dtex	1.0824	0.3482
	L(w) (mm)	9.4648	3.84E-29	Take Off	1.0824	0.3482
	L(w) %CV	8.1392	3.15E-26	**Ratio**	21.007	2.50E-38
	SFC(w) %12.7	5.4615	5.89E-16			

Table 4.9 allows us to see the effect of each and every fibre quality characteristic on the core yarn characteristic H. 29 of the 37 input variables were shown to be statistically significant over the value of H.

Table 4.10. ANOVA results for core yarn characteristic B-Force

Output	Input	F Values	P Values	Input	F Values	P Values
B-Force	**SCI**	4.2097	6.62E-12	**UQL(w) (mm)**	4.9708	7.32E-13
	Mic	3.6584	1.96E-09	**L(n) (mm)**	3.8590	3.81E-10
	Mat	3.9721	2.83E-05	**L(n) %CV**	3.4273	8.90E-10
	Len	3.7655	7.25E-09	**SFC(n) %12.7**	4.3306	7.70E-14
	Unf	3.8267	2.34E-09	**5.0% (mm)**	3.4393	1.99E-09
	SFI	3.2246	5.20E-07	**Fine mTex**	4.4598	4.18E-14
	Str	4.5361	6.46E-11	IFC	1.1016	0.3131
	Elg	3.6990	1.66E-08	Mat Ratio	0.5697	0.6848
	Rd	3.6932	1.68E-10	Total Cnt/g	1.0441	0.4107
	B	3.8814	4.44E-09	Mean Size	0.8008	0.8351
	Tr Cnt	4.3000	2.12E-13	**Dust Cnt/g**	4.3122	7.71E-14
	Tr Area	4.9039	1.18E-12	**Trash Cnt/g**	4.5536	1.83E-14
	Nep Cnt/g	4.2241	2.03E-13	**VFM %**	4.4594	1.59E-13
	Nep (um)	4.5089	2.23E-14	Thread Number	0.0101	0.1419
	SCN Cnt/g	3.9814	1.64E-11	Spinning	2.2214	0.0144
	SCN (um)	4.3911	3.77E-14	Lycra Dtex	0.3272	0.7212
	L(w) (mm)	4.2965	8.33E-13	Take Off	0.3272	0.7212
	L(w) %CV	3.6466	9.80E-11	**Ratio**	94.9576	4.11E-91
	SFC(w) %12.7	3.3727	1.28E-08			

Table 4.10 allows us to see the effect of each and every fibre quality characteristic on the core yarn characteristic B-Force. 29 input variables are shown to be statistically significant over the value of B-Force.

Table 4.11. ANOVA results for core yarn characteristic ELG

Output	Input	F Values	P Values	Input	F Values	P Values
ELG	**SCI**	4.3586	1.92E-12	**UQL(w) (mm)**	7.2707	1.27E-19
	Mic	6.1975	6.05E-18	**L(n) (mm)**	4.2561	1.55E-11
	Mat	9.3609	1.09E-13	**L(n) %CV**	8.0749	6.57E-26
	Len	3.9019	2.66E-09	**SFC(n) %12.7**	7.0799	1.44E-23
	Unf	5.6729	3.04E-15	**5.0% (mm)**	9.1213	4.76E-28
	SFI	6.5689	4.27E-17	**Fine mTex**	7.5726	8.72E-25
	Str	4.9978	2.67E-12	IFC	1.0631	0.3821
	Elg	3.2052	5.98E-07	Mat Ratio	0.1157	0.5819
	Rd	5.8626	2.16E-18	Total Cnt/g	1.6570	0.0459
	B	4.4188	9.41E-11	Mean Size	1.0824	0.3482
	Tr Cnt	5.6153	2.86E-18	**Dust Cnt/g**	8.4691	7.82E-28
	Tr Area	5.5561	1.16E-14	**Trash Cnt/g**	7.1454032	1.82E-23
	Nep Cnt/g	8.2579	3.92E-27	**VFM %**	6.3064	5.19E-20
	Nep (um)	8.6510	4.75E-28	Thread Number	1.9236	0.0330
	SCN Cnt/g	7.1945	1.60E-22	Spinning	0.0166	0.0445

Table 4.11. Continued

	SCN (um)	8.5012	6.34E-28	Lycra Dtex	1.9519	0.1444
	L(w) (mm)	6.2167	1.42E-19	Take Off	1.9519	0.1444
	L(w) %CV	7.2307	1.69E-23	**Ratio**	8.5052	6.17E-28
	SFC(w) %12.7	4.4815	1.31E-12			

Table 4.11 allows us to see the effect of each and every fibre quality characteristic on the core yarn characteristic ELG. 29 input variables are shown to have a statistically significant effect on ELG value.

Table 4.12. ANOVA results for core yarn characteristic RKM

Output	Input	F Values	P Values	Input	F Values	P Values
RKM	**SCI**	4.2714	3.96E-12	**UQL(w) (mm)**	3.9394	1.41E-09
	Mic	3.1828	9.77E-08	**L(n) (mm)**	3.4803	8.44E-09
	Mat	8.0916	8.77E-12	**L(n) %CV**	2.1660	0.0001
	Len	3.1054	9.52E-07	**SFC(n) %12.7**	2.9539	3.53E-08
	Unf	4.7907	1.72E-12	**5.0% (mm)**	2.9407	1.76E-07
	SFI	3.6698	2.06E-08	**Fine mTex**	2.7531	3.46E-07
	Str	4.3361	2.61E-10	IFC	1.1544	0.2395
	Elg	2.6226	3.93E-05	Mat Ratio	1.2104	0.1822
	Rd	2.9430	1.47E-07	Total Cnt/g	1.1497	0.2455
	B	2.8424	8.21E-06	Mean Size	1.1038	0.3124
	Tr Cnt	3.6645	6.99E-11	**Dust Cnt/g**	3.2739	1.39E-09

Table 4.12. Continued

	Tr Area	3.6602	1.14E-08	**Trash Cnt/g**	3.1908	5.21E-09
	Nep Cnt/g	2.6339	8.06E-07	**VFM %**	3.0493	4.78E-08
	Nep (um)	3.1772	5.18E-09	Thread Number	0.0472	0.8933
	SCN Cnt/g	2.2472	8.44E-05	Spinning	0.7170	0.9117
	SCN (um)	3.2638	1.53E-09	Lycra Dtex	0.7717	0.8656
	L(w) (mm)	2.7858	6.14E-07	Take Off	0.8574	0.7279
	L(w) %CV	2.9922	4.53E-08	**Ratio**	8.1765	9.34E-17
	SFC(w) %12.7	3.1109	1.16E-07			

Table 4.12 allows us to see the effect of each and every fibre quality characteristic on the core yarn characteristic RKM. 29 input variables are shown to have a statistically significant effect on RKM value.

4.3. Input Size Reduction

Table 4.13. The results of ANOVA and PCA

Input Variable	PCA	ANOVA
SCI	Included	Included
Mic	Excluded	Included
Mat	Excluded	Included
Len	Excluded	Included
Unf	Included	Included
SFI	Included	Included
Str	Included	Included
Elg	Included	Included
Rd	Included	Included

Table 4.13. Continued

B	Excluded	Included
TrCnt	Included	Included
TrArea	Included	Included
NepCnt	Excluded	Included
Nep	Included	Included
SCNCnt	Excluded	Included
SCN	Included	Included
Lwmm	Included	Included
LwCV	Included	Included
SFCw	Excluded	Included
UQL	Excluded	Included
Lnmm	Excluded	Included
LnCV	Included	Included
SFCn	Included	Included
Mm5	Included	Included
FinemTex	Included	Included
IFC	Excluded	Excluded
MatRatio	Excluded	Excluded
TotalCnt	Included	Excluded
Mean Size	Included	Excluded
Dust Cnt	Included	Included
TrashCnt	Included	Included
VFM	Included	Included
Thread	Included	Excluded
Spinning	Included	Excluded
LycraDtex	Excluded	Excluded
Take Off	Excluded	Excluded
Ratio	Included	Included

As it is shown in Table 4.13, ANOVA reduces the input variables to 29, while PCA reduces the input variables to 25. The differences between PCA and ANOVA are Mat, Len, b, SCNCnt, Mat Ratio. Mean Size are all caused by the choice of weights, which was 0.2. Table 4.2 clearly shows that the weights of these variables are very close to 0.2 in either Principal Component 1, or Principal Component 2. Thread and Spinning are excluded in ANOVA, yet excluded in PCA. The difference of Thread, caused by 0.01 significance rate, which eliminated Thread from B-Force, which had a very close significance rate to 0.01 as shown in Table 4.10. Spinning was eliminated in output ELG for the same reason shown in Table 4.11.

4.4. Artificia Neural Network and Support Vector Machine Results

ANN and SVM model predictions are compared against each other using MSE and MAPE values. First both models were compared against each other for their general predictions performance or their prediction success for all 11 output variables. Then they were compared on each of the output variable predictions. The prediction performance on test data was the main comparison for this study.

In this study a grand total of 117 models were designed and analyzed. The performance table of these models can be found in the appendix A and B. The best results of the models are shown in this section.

Table 4.14. Performance results of ANN and SVM based on input size

	ANN Approach				SVM Approach			
SETS	MSE	MAPE	TEST R value	Time (sec.)	MSE	MAPE	TEST R Value	Time (sec.)
1	12521.1990	0.4802	0.9441	8	3336	0.5111	0.9501	6
2	2957.1880	0.290025	0.9568578	6	3238	0.5460	0.9514	4
3	4575.9998	0.49232	0.9332179	6	6885	0.5642	0.8963	4

Both ANN and SVM results are very close to each other. Set 1 shows the non reduced input size with 37 input variables. Set 2 shows input size reduced by PCA with 24 input variables. Set 3 shows input size reduced by ANOVA with 29 input variables. The best R value ANN result was with PCA input and scaled gradient descent training algorithm. The best R value SVM result was again with PCA input, linear kernel algorithm with quadratic programming solver. The best results for each type of input can be seen in Table 4.14.

Table 4.15. MAPE values for CVM

CVM	ANN model	SVM model
MAPE values %	9.9600%	9.0600%

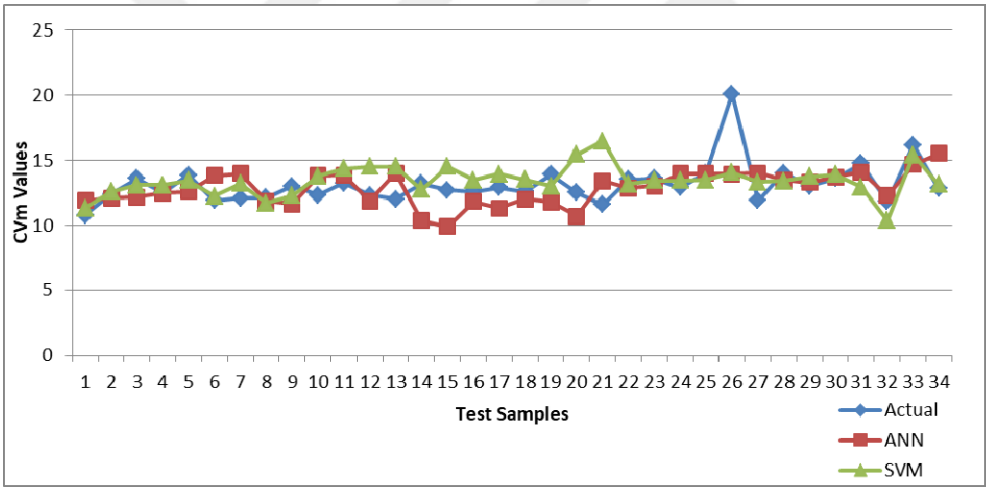

Figure 4.3. The test results for CVM output by ANN and SVM against actual CVM output

Table 4.15 shows that for MAPE both ANN and SVM are quite close to each other. And yet SVM MAPE value is slightly less than the ANN model MAPE value. Figure 4.3 shows that both ANN and SVM results are close to actual output except for one outlying point. For CVM the results are close to each other. Both of the models have over 90% success rate. On the graph X axis shows the test samples

Y axis shows the CVM values. Both ANN and SVM missed sample 26 with large margins. Sample 26 can be an outlier of the sample.

Table 4.16. MAPE Values for UM

UM	ANN Model	SVM Model
MAPE values (%)	13.8265%	9.0801%

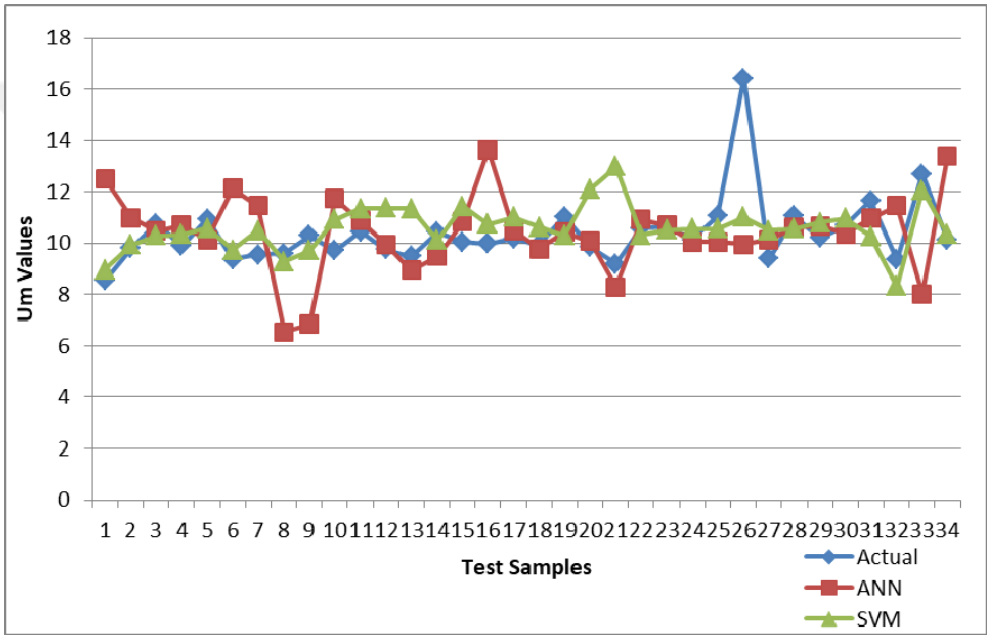

Figure 4.4. The test Um outputs of ANN and SVM against actual Um outputs

Table 4.16 shows that for Um value prediction MAPE values of SVM model is lesser than the MAPE values of ANN model. This result shows us that SVM model predictions were closer to actual values than ANN model predictions. Figure 4.4 shows that on the Um value both ANN model and SVM model has outlying predictions, yet ANN model prediction has the most noticeable outliers. SVM model predictions are shown to be closer to actual values. On the graph X axis shows the test samples Y axis shows the Um values. ANN model is shown to

give predictions with large errors on samples 8 and 9. ANN model also missed samples 33 and 34 by same margins. Again sample 26 was missed by both ANN and SVM models with large margins. Sample 26 can be seen as the outlier of the actual values.

Table 4.17. MAPE values for Thin-50

THIN-50	ANN	SVM
MAPE Values (%)	110.2916%	225.8153%

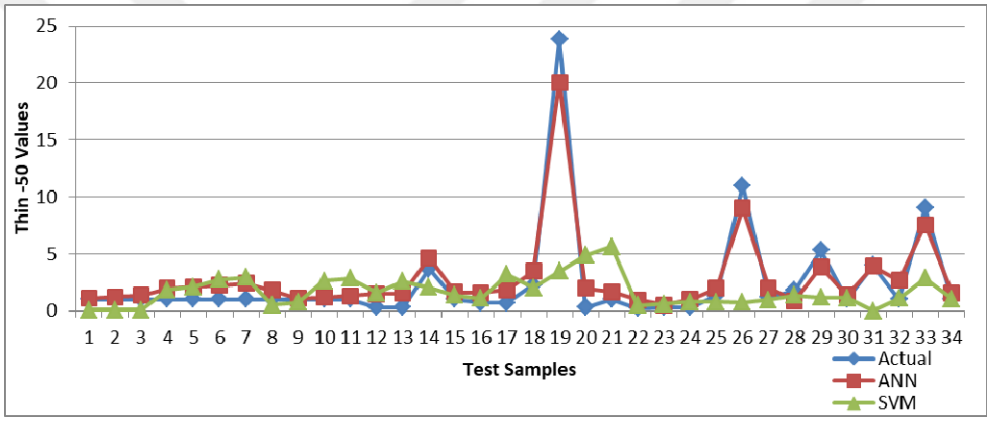

Figure 4.5. The test Thin-50 outputs of ANN and SVM against actual Thin-50 outputs

Table 4.17 shows that for Thin -50, both SVM and ANN models rarely get the correct prediction. Most of the time, these values are not even close to the actual value. Figure 4.5 shows that SVM and ANN prediction results have huge outliers and rarely conform to the actual values. For the most part on the THIN-50 values, none of the models give predictions close to actual values. On the graph X axis shows the test samples Y axis shows the Thin -50 values. SVM is shown to have a large margin of error in sample 19 and sample 26. The reason that both models fail to correctly predict Thin-50 model can be that. Thin-50 output didn't have many inputs that were statistically significant on it, shown in Table 4.5.

Table 4.18. MAPE values for Thick +50

THICK +50	ANN	SVM
MAPE Values (%)	46.3068%	148.9018%

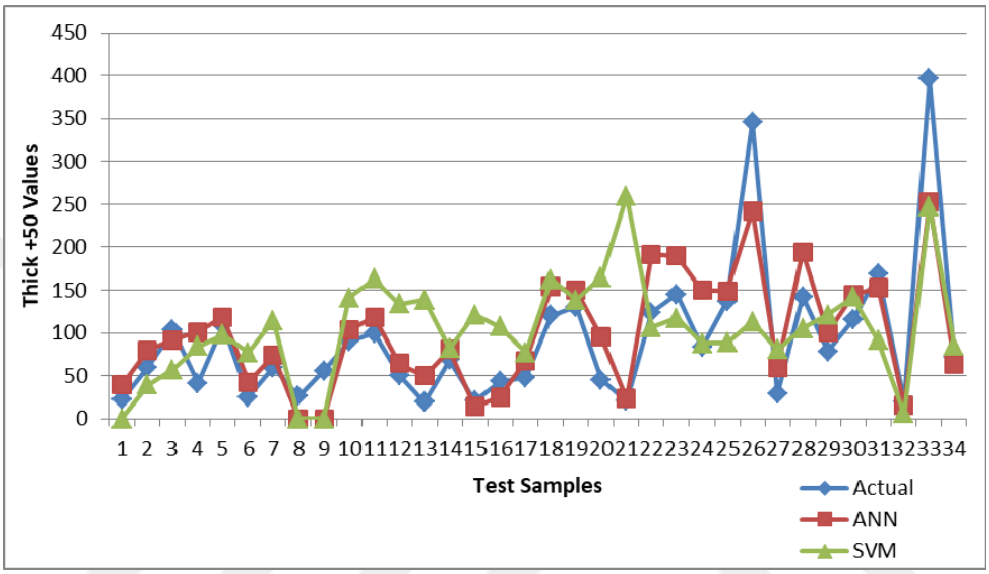

Figure 4.6. The test Thick +50 outputs of ANN and SVM against actual outputs

Table 4.18 shows that ANN models have the upper hand when it comes to predicting the Thick +50 value. Even though their MAPE value is high, it is not nearly as high as SVM model MAPE values. ANN can be the better choice for Thick +50 but the model's performance is not good enough. Figure 4.6 also shows that for the most part on the Thick +50 values ANN results fare better than SVM results when it comes to Thick +50 values. On the graph X axis shows the test samples. Y axis shows the Thick +50 values. On samples 20 and 21 SVM seems to have a large margin of error while on samples 22 and 23. ANN has the larger margin. On sample 26 both models fail with large error margins again. The reason Thick+50 is harder for the models to correctly predict can be that, Thick+50 output didn't have many inputs that were statistically significant on it as shown in Table 4.6.

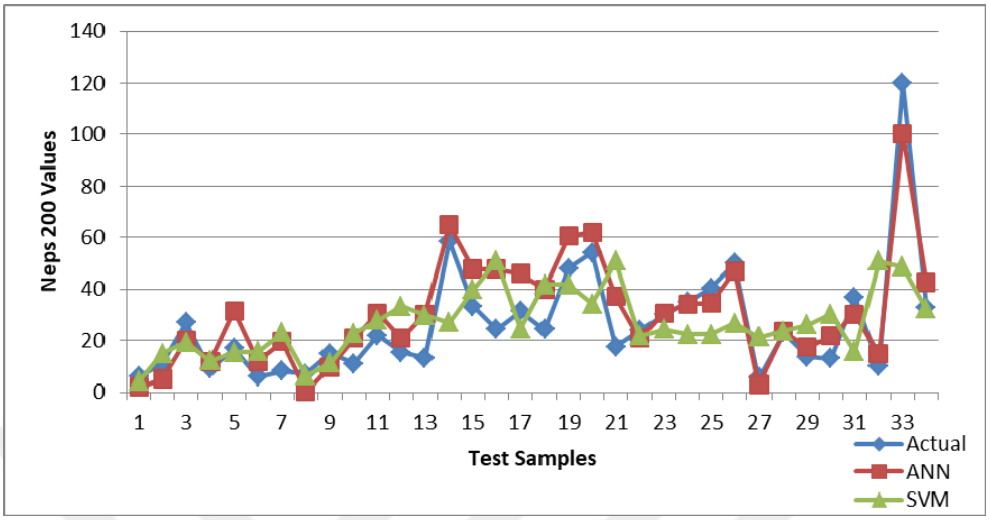

Figure 4.7. The test Neps 200 outputs of ANN and SVM against actual outputs

Table 4.19. MAPE values for Neps 200

Neps 200	ANN	SVM
MAPE Values (%)	48.2164%	76.7906%

Table 4.19 shows that both ANN and SVM models did not give good results according to performance indicator MAPE values. ANN model gave slightly closer results to actual values than SVM model.Figure 4.7 shows that for the most part on the Neps 200 values ANN model predictions are closer to the actual values than SVM model predictions. Both ANN and SVM models gave bad performances on Neps 200 predictions. On the graph X axis shows the test samples. Y axis shows the Neps 200 values. SVM model is shown to have large margins of error on samples 14, 16, 21, 24, 25 and 26, while ANN model has troubles on samples 5, 15, 16, 17 and 26. It can be said that sample 26 is the outlier of the system.

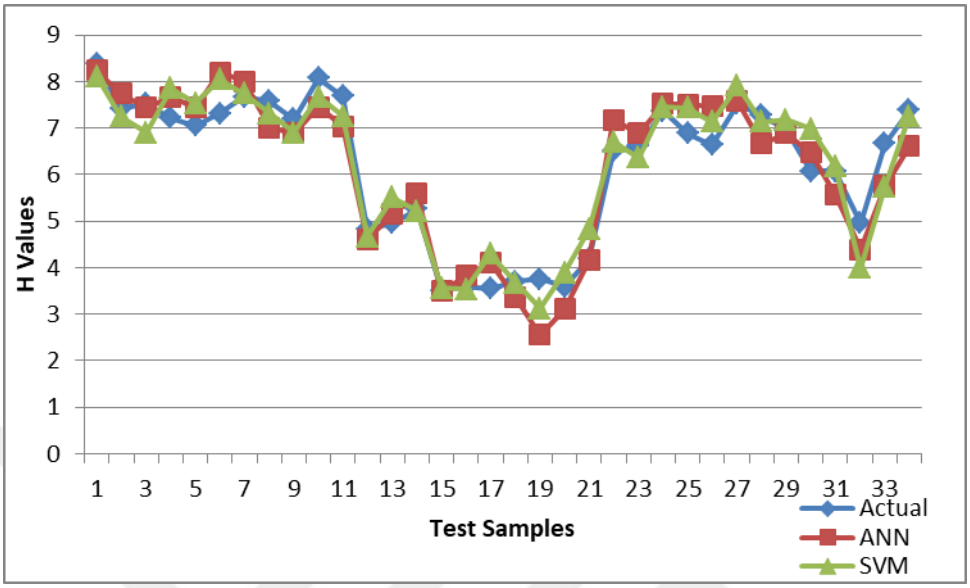

Figure 4.8. The test H outputs of ANN and SVM against actual outputs

Table 4.20. MAPE values for H Value

H Value	MAPE of ANN	MAPE of SVM
MAPE values (%)	7.4555%	6.5893%

Table 4.20 shows that both ANN and SVM model predictions were very close to actual values on H value predictions. The SVM model is shown to give better results than ANN model. Both models give over 90% success on predicting the H value, yet SVM model has a slightly lesser MAPE value showing that SVM was the better model for predicting H value. Figure 4.8 shows that both ANN and SVM are very close to the actual values of H. It also shows that SVM model has smaller outliers. On the graph X axis shows the test samples. Y axis shows the H values.

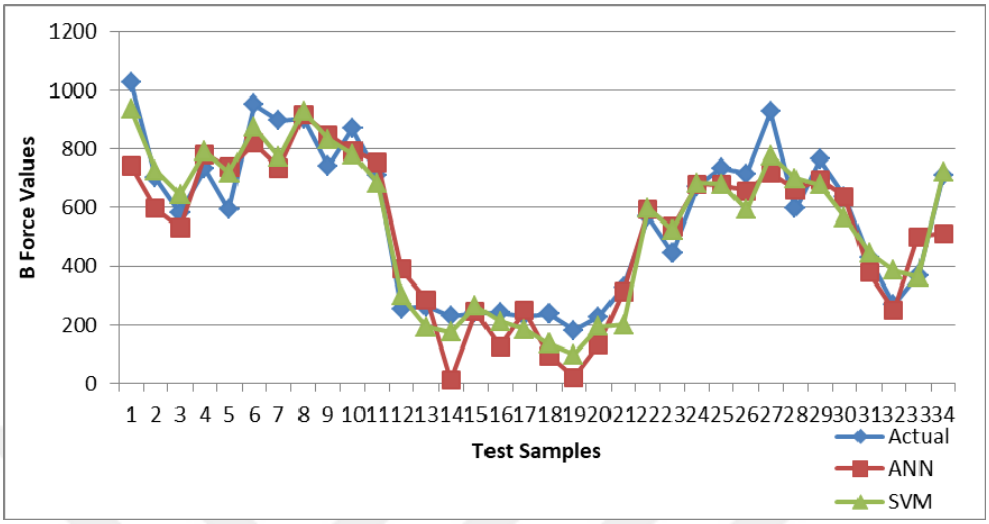

Figure 4.9. The test B Force outputs of ANN and SVM against actual outputs

Table 4.21. MAPE values for B Force

B-Force	MAPE of ANN	MAPE of SVM
MAPE values (%)	21.1695%	14.9037%

Table 4.21 shows that both ANN and SVM have high MAPE values on B-Force predictions. SVM model is shown to be the better predictor since it has a lesser MAPE value. Yet since the MAPE value is above 10% it is not a good predictor of B-Force values. Figure 4.9 shows that SVM model predictions stay close to the actual values than ANN values. Sample 1 has a large margin of error with ANN model, while SVM model stayed closer to the actual value. Sample 26 is again shown as the outlier for both models.

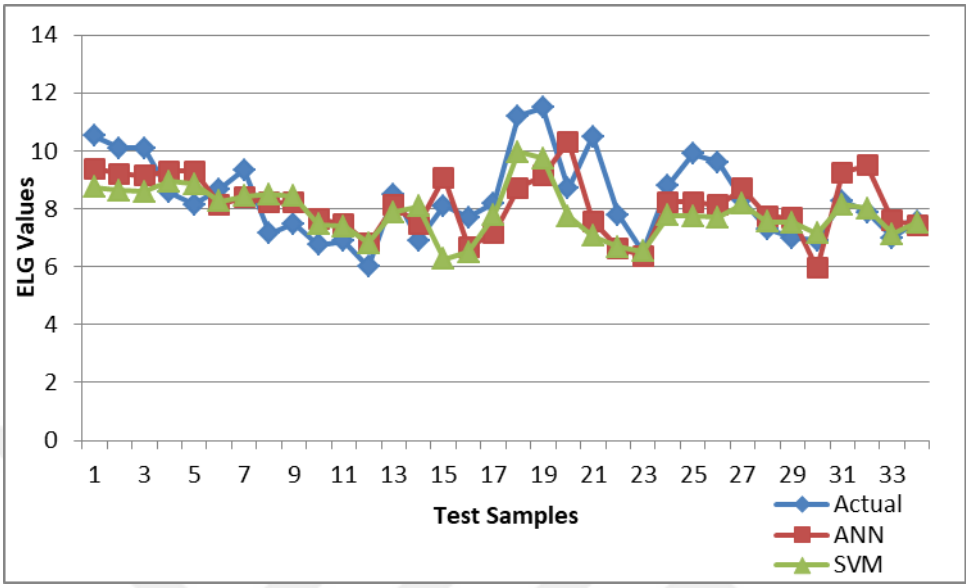

Figure 4.10. The test ELG outputs of ANN and SVM against actual outputs

Table 4.22. MAPE values for ELG

ELG	MAPE of ANN	MAPE of SVM
MAPE values (%)	11.7688%	10.6907%

Table 4.22 shows that both ANN and SVM models give close to actual value predictions. SVM is only slightly better according to the MAPE values. They are both close to 90% correct. Both ANN and SVM are able to predict ELG values with high performance. Figure 4.10 shows that both ANN and SVM model predictions stay close to the actual values. The SVM model stays closer to actual ELG values than ANN model. Samples 15, 16, 17 and 26 all have large margins of error for both models. Sample 26 is shown yet again as the outlier for the actual values.

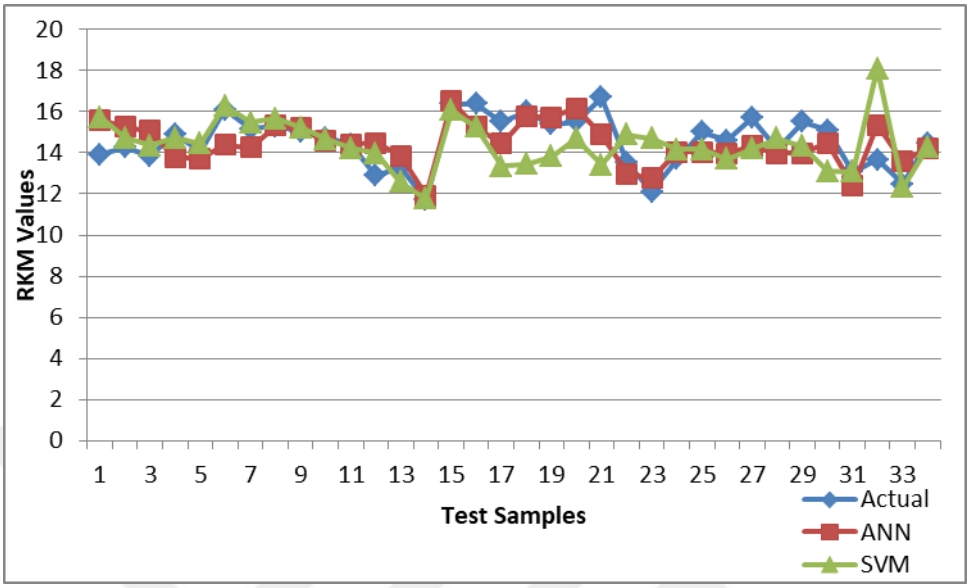

Figure 4.11. The test RKM outputs of ANN and SVM against actual outputs

Table 4.23. MAPE values for RKM

RKM	MAPE of ANN	MAPE of SVM
MAPE values (%)	5.4089%	6.9640%

Table 4.23 shows that both ANN and SVM model predictions are close to actual RKM values. Each model has less than 10% error which makes them good predictors of RKM values. It can also be seen that ANN stayed closer to the actual values in Figure 4.11. ANN model is shown to be the better predictor of RKM values.

Table 4.24. MAPE values for NEPS 140

NEPS 140	MAPE of ANN	MAPE of SVM
MAPE values (%)	54.2277%	31.0182%

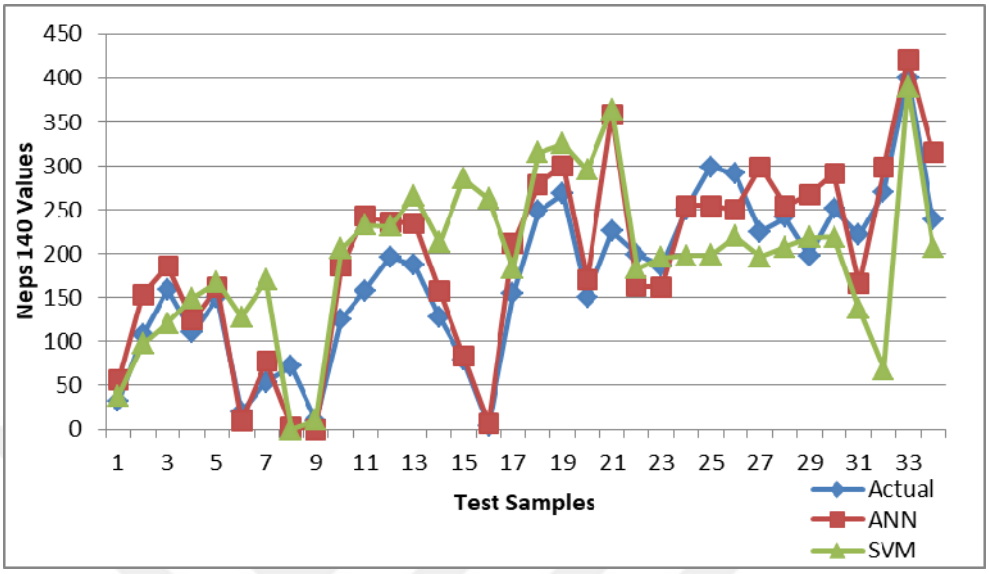

Figure 4.12. The test Neps 140 outputs of ANN and SVM against actual outputs

Table 4.24 shows that both ANN and SVM model predictions stay far away from actual Neps 140 values. Both models have large outliers, yet ANN has more errors than SVM model which shows in their MAPE values. Figure 4.12 shows the ANN and SVM test outputs against the actual outputs. Samples 6, 7, 28, 29 and 30 all have large margins of error for SVM.

5. CONCLUSION AND FUTURE STUDIES

In the present study, KİPAŞ plant data was used as the main input matrix. Principal Component Analysis (PCA) and Analysis of Variance (ANOVA) were both used to reduce the input matrix. These reduced matrices were then used as training, validation and test inputs for artificial neural networks and support vector machines. This study was focused on evaluating the performance of artificial neural networks and support regressor machines. In order to get the most concurring data possible 3 different learning methods for neural networks and support vector machines were used. Support vector machines also used several different optimization methods. To make sure the data is valid, different numbers of nodes were used for the hidden layer on both artificial neural networks and support vector machines.

The number of nodes started from 10 and ended in 28. Gaussian Kernel, Linear Kernel and Polynomial Kernel methods were used as learning algorithms for SVM, Levenberg-Marquardt, Scaled Conjugate Gradient and Bayesian Regularization training methods were used for ANNs. For SVMs Iterative Single Data Algorithm (ISDA), Quadratic Programming (L1QP) and Sequential Minimal Optimization (SMO) methods were used as solvers. The following conclusions can be derived:

1. **PCA looks like the better size reduction technique than ANOVA for this study**

   Even though the results were close, on each of the tests PCA fared slightly better than ANOVA. Not only that the input matrix was reduced better by PCA which is 24 from 37 but the reduction actually helped to get better results on the same model. We may conclude that PCA is the better size reduction technique. More detailed results were given in Table 4.16.

2. **Neural Network fared better than SVM yet not concurrently**
   Neural networks fared better than SVMs on some of the outputs like Um and ELG. Yet on some of the outputs like Neps 200 and H value SVM is the better choice. On the CRM, both ANN and SVM had high amount of errors. On B Force they had very similar and powerful results. Therefore it is impossible at this time to say one is better than the other, even though ANN was slightly better at the best model comparison shown in Table 4.14.
3. **Effect of Different Training Algorithms**
   Unfortunately the performance indicators for the diffenrent training algorithms stayed very close to each other. Even though Scaled Gradient had the best result, this was not conclusive.

Since this study couldn't decide on the training algorithms or the comparison between SVM and neural network, it is necessary to build more experiments including more types of fibres and yarns. This is necessary to get a conclusive result.

REFERENCES

Abdi, H., Williams L. J., 2010, "Principal Component Analysis", The Canadian Journal of Chemical Engineering, 2:433-459, doi:10.1002/wics.101.

Abraham, A., 2005, "Artificial Neural Networks", (edited by P. H. Sydenham and R. Thorn), Handbook of Measuring System Design, John Wiley & Sons, Ltd., New York, pp:901-908, doi:10.1002/0471497398.mm421.

Agahian, F., Amirshahi S. A., Amirshahi S. H., 2008, "Reconstruction of Reflectance Spectra Using Weighted Principal Component Analysis", Color Research & Application, 33(5): 360–371, doi:10.1002/col.20431.

Akankwasa, N. T., Wang J., Zhang Y., 2015, "Study of Optimum Spinning Parameters for Production of T-400/Cotton Core Spun Yarn by Ring Spinning", The Journal of The Textile Institute, 107(4):504–511, doi:10.1080/00405000.2015.1045254.

Akbani, R., Kwek, S., Japkowicz, N., 2004, "Applying Support Vector Machines to Imbalanced Datasets", (edited by Bolicaut J.F., Esposito F., Giznnotti F., Pedreschi D.), Machine Learning: ECML 2004 Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, volume:3201 pp:39–50, doi:10.1007/978-3-540-30115-8_7.

Alakent, B., Kunduracıoğlu Issever, R., 2016, "Exploratory and Predictive Logistic Modeling of a Ring Spinning Process Using Historical Data", Textile Research Journal, 87(13): 1643–1654, doi:10.1177/0040517516657063.

Alamili, M., 2011, "Exchange Rate Prediction Using Support Vector Machines", Master of Science in Management Technology Thesis, Delft University of Technology, 91 pages.

Almetwally, A. A., Idrees, H. M. F., Hebeish, A. A., 2014, "Predicting the Tensile Properties of Cotton/Spandex Core-Spun Yarns Using Artificial Neural Network and Linear Regression Models", Journal of The Textile Institute, 105(11):1221-1229.

Anagnostopoulos, C. N., Vergados, D., Kayafas E., Loumos V., 2001, "A Computer Vision Approach for Textile Quality Control", The Journal of Visualization and Computer Animation, 12(1):31–44, doi:10.1002/vis.245.

Aslan, K., Bozdemir H., Sahin C., Ogulata, S. N., 2009, "Can Neural Network Able to Estimate the Prognosis of Epilepsy Patients According to Risk Factors?", Journal of Medical Systems, 34(4):541–550, doi:10.1007/s10916-009-9267-8.

Balci, O., Ogulata, N. S., Sahin, C., Ogulata, T. R., 2008a, "An Artificial Neural Network Approach to Prediction of the Colorimetric Values of the Stripped Cotton Fabrics", Fibers and Polymers, 9(5):604–614, doi:10.1007/s12221-008-0096-z.

Balci, O., Ogulata, N. S., Sahin, C., Ogulata, T. R., 2008b, "Prediction of CIELab Data and Wash Fastness of Nylon 6,6 Using Artificial Neural Network and Linear Regression Model", Fibers and Polymers, 9(2):217–224, doi:10.1007/s12221-008-0035-z.

Balci, O., Ogulata, T. R., 2009, "Prediction of the Changes on the CIELab Values of Fabric after Chemical Finishing Using Artificial Neural Network and Linear Regression Models", Fibers and Polymers, 10(3):384–393, doi:10.1007/s12221-009-0384-2.

Bataineh, M., Marler, T. M., Abdel-Malek, K., Arora, J. S., 2016 "Neural Network for Dynamic Human Motion Prediction", Expert Systems with Applications, 48: 26–34, doi:10.1016/j.eswa.2015.11.020.

Boser, B. E., Guyon, I. M., Vapnik, V. N., 1992,"A training Algorithm for Optimal Margin Classifiers", (edited by Haussler, D.), 5th Annual ACM Workshop on COLT, ACM Press, Pittsburgh, PA, pages:144-152, doi:10.1007/springerreference_57934.

Brown, M. P., Grundy, W. N., Lin, D., Christianini, N., Sugnet, C. W., Furey, T. S., Ares, M. Jr., Haussler, D., 2000, "Knowledge-Based Analysis of Microarray Gene Expression Data by Using Support Vector Machines", Proceedings of the National Academy of Sciences, 97(1):262–267, doi:10.1073/pnas.97.1.262.

Bu, H., Wang, J., Huang, X., 2009, “Fabric Defect Detection Based on Multiple Fractal Features and Support Vector Data Description”, Engineering Applications of Artificial Intelligence, 22(2):224–235, doi:10.1016/j.engappai.2008.05.006.

Buntine, W., 2002, “Variational Extensions to EM and Multinomial PCA”, (edited by Elomaa, T., Toivonen, H., Mannila, H.) Lecture Notes in Computer Science, Machine Learning: ECML 2002, 2430:23–34, doi:10.1007/3-540-36755-1_3.

Burden, F., Winkler, D., 2009, “Bayesian Regularization of Neural Networks”, (edited by Clifton, N. J.) Methods in Molecular Biology, Humana Press, New York, pages:23-42, doi:10.1007/978-1-60327-101-1.

Byrne, G. F., Crapper, P. F., Mayo, K. K., 1980, “Monitoring Land-Cover Change by Principal Component Analysis of Multitemporal Landsat Data”, Remote Sensing of Environment, 10(3):175–184, doi:10.1016/0034-4257(80)90021-8.

Candes, E. J., Li, X., Ma, Y., Wright, J., 2011, “Robust Principal Component Analysis?”, Journal of the ACM (JACM), 58(3):1–37, doi:10.1145/1970392.1970395.

Cao, J. P., Li, Y. H., Sun, P. Z., 2013, “Effect of Humidity on Nep Movement in Electrostatic Field”, Advanced Materials Research, 821:393-397.

Cavuslu, M. A., Becerikli, Y., Karakuzu, C., 2012, “Hardware Implementation of Neural Network Training with Levenberg & Marquardt Algorithm”, TBV Journal of Computer Sciences and Engineering, 5:31-38.

Chandrasekaran, V., Senthilkumar P., Senthilkumar M., 2014, “Yarn Sample Preparation Techniques and Yarn Diameter Measurement for Analysing Cut-Cross Sectional View of Hollow Core Dref Spun Yarns”, Journal of The Institution of Engineers (India): Series E. 95(2):89–95, doi:10.1007/s40034-014-0041-1.

Chapelle, O., Haffner, P., Vapnik, V. N., 1999, “Support Vector Machines for Histogram-Based Image Classification”, IEEE Transactions on Neural Networks, 10(5):1055–1064, doi:10.1109/72.788646.

Chatfield, C., Collins, A. J., 1980, “Introduction to Multivariete Analysis”, Chapman and Hall, New York, 248 pages.

Chowdhury, M., 2014, “High Volume Instrument in QC Department of Textile Industry”, Last Visited at 12.03.2019, https://www.slideshare.net/mohiuddinchowdhuryrahat/high-volume-instrument-in-qc-department.

Cohen, J., 1973, “Eta-Squared and Partial Eta-Squared in Fixed Factor Anova Designs”, Educational and Psychological Measurement, 33(1):107–112, doi:10.1177/001316447303300111.

Cortes, C., Vapnik, V., 1995, “Support-Vector Networks”, Machine Learning, 20(3): 273–297, doi:10.1007/bf00994018.

Cybenko, G., 1989, “Approximations by Superpositions of a Sigmoidal Function”, Mathematics of Control, Signals and Systems, 2:303-314.

Dalton, J., Deshmane, A., 1991, “Artificial Neural Networks”, IEEE Potentials, 10(2): 33–36, doi:10.1109/45.84097.

Demiryurek, O., Koc, E., 2009, “The Mechanism and/or Prediction of the Breaking Elongation of Polyester/Viscose Blended Open-End Rotor Spun Yarns.” Fibers and Polymers, 10(5):694–702, doi:10.1007/s12221-010-0694-4.

Deng, N., Tian, Y., 2013, “Support Vector Machines Optimization Based Theory, Algorithms and Extensions”, CRC Press, Boca Raton, pages:313.

Ding, C., He, X., 2004, “K-Means Clustering via Principal Component Analysis”, Twenty-First International Conference on Machine Learning - ICML 2004, 1:29-37, doi:10.1145/1015330.1015408.

Dong, D., Mcavoy, T. J., 1994, “Nonlinear Principal Component Analysis-Based on Principal Curves and Neural Networks”, Proceedings of 1994 American Control Conference -ACC 94, 2:1284-1288, doi:10.1109/acc.1994.752266.

Drew, P. J., Monson, J. R. T., 2000 “Artificial Neural Networks”, Surgery, 127(1):3-11, doi:10.1067/msy.2000.102173.

Drucker, H., Wu, D., Vapnik, V. N., 1999, “Support Vector Machines for Spam Categorization”, IEEE Transactions on Neural Networks, 10(5):1048-1054, doi:10.1109/72.788645.

Dwivedi, R., 2016, “Fiber Maturity”, found in https://tr.scribd.com/doc/97265301/Fiber-Maturity. Last checked on 12.03.2019.

Finley, T., Joachims, T., 2005, “Supervised Clustering with Support Vector Machines”, Proceedings of the 22nd International Conference on Machine Learning - ICML 05, 1:217-224, doi:10.1145/1102351.1102379.

Foulk, J., Mcalister, D. D., Kulasekera, K. B., 2006, “Detecting Cotton Trash Particle Size Distribution in Mill Laydown Using HVI Trashmeter Software”, Journal of Textile And Apparel, Technology and Management, 5(2):1-11.

Friston, K. J., Frith, C. D., Liddle, P. F., Frackowiak, R., 1993, “Functional Connectivity: The Principal-Component Analysis of Large (PET) Data Sets”, Journal of Cerebral Blood Flow & Metabolism, 13(1):5–14, doi:10.1038/jcbfm.1993.4.

Fukushima, K., 1980, “Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”, Biological Cybernetics, 36(4): 193–202, doi:10.1007/bf00344251.

Furferi, R., Maurizio, G., 2010, “Yarn Strength Prediction: A Practical Model Based on Artificial Neural Networks”, Advances in Mechanical Engineering.

Gavin, H. P., 2011, “The Levenberg-Marquardt Algorithm for Nonlinear Least Squares Curve-Fitting Problems”, Duke University, 19 pages.

Gershenfeld, N. A., Weigend, A. S., 1993, “The Future of Time Series“, (edited by Gershenfeld, N. A., Weigend, A. S.), Time Series Prediction: Forecasting the Future and Understanding the Past, Westview Press, Santa Fe, pp:1-70.

Gharehaghaji, A. A., Palhang, M., Shanbeh, M., 2007, “Analysis of Two Modeling Methodologies for Predicting the Tensile Properties of Cotton-Covered Nylon Core Yarns”, Textile Research Journal, 77(8):565-571.

Ghosh, A., Chatterjee, P., 2010, “Prediction of Cotton Yarn Properties Using Support Vector Machine”, Fibers and Polymers, 11(1):84–88, doi:10.1007/s12221-010-0084-y.

Ghosh, A., Guha, T., Bhar, R. B., Das, S., 2011, "Pattern Classification of Fabric Defects Using Support Vector Machines", International Journal of Clothing Science and Technology, 23(2-3):142–151, doi:10.1108/09556221111107333.

Gibbons, J. D., Pratt, J. W., 1975, "P-Values: Interpretation and Methodology", The American Statistician, 29(1): 20–25, doi:10.2307/2683674.

Gordon, J., 2017, "What is Bayesian Neural Network?", KDNuggets, Last Visited:12.03.2019, https://www.kdnuggets.com/2017/12/what-bayesian-neural-network.html.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002, "Gene Selection for Cancer Classification Using Support Vector Machines", Machine Learning, 46(1-3):389-422, doi:10.1023/A:1012487302797.

Hagan, M. T., Demuth, H. B., Beale, M. H., De Jesus, O., 2014, "Neural Network Design", Martin Hagan, USA, 800 pages.

Haykin, S., 2009, "Neural Networks and Learning Machines- 3rd Edition", Pearson Education Inc., New Jersey, 906 pages.

Hembree, J. F., Ethridge, D. E., Neeper, J. T., 1986, "Market Values of Fiber Properties in Southeastern Textile Mills", Textile Research Journal, 56(2):140-144, doi:10.1177/004051758605600212.

Hopfield, J. J., 1982, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", PNAS, 79(8):2554-2558, doi:10.1073/pnas.79.8.2554.

Hornik, K., 1991, "Approximation Capabilities of Multilayer Feedforward Networks", Neural Networks, 4(2):251-257, doi:10.1016/0893-6080(91)90009-t.

Hornik, K., 1993, "Some New Results on Neural Network Approximation", Neural Networks, 6(8): 1069–1072, doi:10.1016/s0893-6080(09)80018-x.

Hornik, K., Stinchcombs, M., White, H., 1989, "Multilayer Feedforward Networks Are Universal Approximators", Neural Networks, 2(5):359-366, doi:10.1016/0893-6080(89)90020-8.

Hou, Z., Parker, J. M., 2005, “Texture Defect Detection Using Support Vector Machines with Adaptive Gabor Wavelet Features”, 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION05), 1:275-280, doi:10.1109/acvmot.2005.115.

Hsu, C. W., Lin, C. J., 2002, “A Comparison of Methods for Multiclass Support Vector Machines”, IEEE Transactions on Neural Networks13(2): 415-425, doi:10.1109/72.991427.

Hu, M. J. C., Root, H. E., 1964, “An Adaptive Data Processing System for Weather Forecasting”, Journal of Applied Meteorology, 3(5):513–523, doi:10.1175/1520-0450(1964)003<0513:aadpsf>2.0.co;2.

Hubert, M., Rousseauw, P. J., Branden, K. V., 2005, “ROBPCA: A New Approach to Robust Principal Component Analysis”, Technometrics, 47(1):64–79, doi:10.1198/004017004000000563.

Hyvarinen, A., 1999, “Fast and Robust Fixed-Point Algorithms for Independent Component Analysis”, IEEE Transactions on Neural Networks, 10(3): 626–634, doi:10.1109/72.761722.

Irie, B., Miyake, S., 1988, “Capabilities of Three-Layered Perceptrons”, IEEE International Conference on Neural Networks 1988, 1:641-648, doi:10.1109/icnn.1988.23901.

Jack, L. B., Nandi, A. K., 2002, “Fault Detection Using Support Vector Machines And Artificial Neural Networks, Augmented By Genetic Algorithms”, Mechanical Systems and Signal Processing, 16(2-3):373–390, doi:10.1006/mssp.2001.1454.

Jain, A. K., Mao, J., Mohiuddin, K. M., 1996, “Artificial Neural Networks: A Tutorial”, Computer, 29(3):31–44, doi:10.1109/2.485891.

Ji, R., Li, D., Chen, L., Yang, W., 2010, “Classification and Identification of Foreign Fibers in Cotton on the Basis of a Support Vector Machine”, Mathematical and Computer Modelling, 51(11-12):1433–1437, doi:10.1016/j.mcm.2009.10.007.

Joachims, T., 2002, “Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms”, Kluwer Academic Publishers, Norwell, MA, 179 pages.

Jolicoeur, P., Mosimann, J. E., 1960, “Size and Shape Variation in the Painted Turtle. A Principal Component Analysis”, Growth, 24:339-354, doi:10.1.1.454.279.

Karimi, H. A., 2009, “Handbook of Research on Geoinformatics”, Information Science Reference. USA, 518 pages.

Karpathy, A., 2017, “Convolutional Neural Networks for Visual Recognition.” CS231n Convolutional Neural Networks for Visual Recognition, 2017, cs231n.github.io/neural-networks-1/#nn. Last visited:12.03.2019

Karras, D. A., 2003, “Improved Defect Detection Using Support Vector Machines and Wavelet Feature Extraction Based on Vector Quantization and SVD Techniques”, Proceedings of the International Joint Conference on Neural Networks, 3:2322-2327, doi:10.1109/ijcnn.2003.1223774.

Kecman, V., 2001, “Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models”, A Bradford Book, Cambridge, 608 pages.

Kennedy, S. M., Christiani, D. C., Eisen, E. A., Wegman, D. H., Greaves, I. A., Olenchock, S. A., Ye, T. T., Lu, P. L., 1987, “Cotton Dust and Endotoxin Exposure-Response Relationships in Cotton Textile Workers”, American Review of Respiratory Disease, 135(1):194-200.

Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab M., Antonescu, C. R., Peterson, C., Meltzer, P. S., 2001, “Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks”, Nature Medicine. 7(6): 673–679, doi:10.1038/89044.

Kim, K. I., Jung, K., Kim, H. J., 2002, “Face Recognition Using Kernel Principal Component Analysis”, IEEE Signal Processing Letters, 9(2):40–42, doi:10.1109/97.991133.

Kramer, M. A., 1991, “Nonlinear Principal Component Analysis Using Autoassociative Neural Networks”, AIChE Journal, 37(2):233–243, doi:10.1002/aic.690370209.

Krifa, M., Ethridge, D., 2004, “A Qualitative Approach to Estimating Cotton Spinnability Limits”, Textile Research Journal, 74(7):611-616, doi:10.1177/004051750407400710.

Krifa, M., Gourlot, J. P., Drean, J. Y., 2001, “Effect of Seed Coat Fragments on Cotton Yan Strength”, Textile Research Journal, 71(11):981-986.

Krogh, A., Vedelsby, J., 1994, “ Neural Network Ensembles, Cross Validation and Active Learning”, NIPS’94 Proceedings of the 7th International Conference on Neural Information Processing Systems, 1:231-238.

Krose, B., Smagt, P., 1996, “An Introduction to Neural Networks”, University of Amsterdam, Amsterdam, 135 pages.

Ku, W., Storer, R., Georgakis, C., 1995, “Disturbance Detection and Isolation by Dynamic Principal Component Analysis”, Chemometrics and Intelligent Laboratory Systems, 30(1): 179–196, doi:10.1016/0169-7439(95)00076-3.

Kumar, A., Shen, H. C., 2002, “Texture Inspection for Defects Using Neural Networks and Support Vector Machines”, Proceedings of International Conference on Image Processing, 3:353-356, doi:10.1109/icip.2002.1038978.

Kumar, S., Kumar, K., 2014, “A Comparative Study of Call Admission Control in Mobile Multimedia Networks Using Soft Computing”, International Journal of Computer Applications, 107(16):5-11, doi:10.5120/18833-0333.

Kumar, T. S., Sampath, V. R., 2013, “Prediction of Dimensional Properties of Weft Knitted Cardigan Fabric by Artificial Neural Network System”, Journal of Industrial Textiles, 42(4):446-458.

Kuo, J., Lee, C. –J., 2003, “A Back-Propagation Neural Network for Recognizing Fabric Defects”, Textile Research Journal, 73(2):147–151, doi:10.1177/004051750307300209.

Kuo, J., Lee, C. –J., Tsai, C. –C., 2003, “Using a Neural Network to Identify Fabric Defects in Dynamic Cloth Inspection”, Textile Research Journal, 73(3):238–244, doi:10.1177/004051750307300307.

Langford, J., 2005, “Tutorial on Practical Prediction Theory for Classification”, (edited by Schapire, R.) Journal of Machine Learning Research, 6:273-306.

Lapedes, A. S., Farber, R. M., 1987, “Nonlinear System Processing Using Neural Networks: Prediction and System Modelling”, Proceedings of First IEEE International Conference on Neural Networks, 1:1102-1158.

Larsen, J., Hansen, L. K., Svarer, C., Ohisson, M., 1996, “Design and Regularization of Neural Networks: The Optimal Use of a Validation Set”, Proceedings of the 1996 IEEE Signal Processing Society Workshop, 1:62-71, doi:10.1109/NNSP.1996.548336.

Lawrence, N. D., Moore, A. J., 2007, “Hierarchical Gaussian Process Latent Variable Models”, Proceedings of the 24th International Conference on Machine Learning - ICML 07, 1:481-488, doi:10.1145/1273496.1273557.

Lek, S., Guegan, J. F., 1999, “Artificial Neural Networks as a Tool in Ecological Modelling, an Introduction”, Ecological Modelling, 120(2-3):65–73, doi:10.1016/s0304-3800(99)00092-7.

Lin, C.-F., Wang, S. –D., 2005, “Fuzzy Support Vector Machines with Automatic Membership Setting”, (edited by Wang, L.) Support Vector Machines: Theory and Applications Studies in Fuzziness and Soft Computing, Springer, New York, pages: 233–254., doi:10.1007/10984697_11.

Maimon, O., Rokach, L., 2010, “Data Mining and Knowledge Discovery Handbook”, Springer, New York, 1285 pages.

Majumdar, A., Majumdar, P. K., Sarkar, B., 2004(a), “Selecting Cotton Bales by Spinning Consistency Index and Micronaire Using Artificial Neural Networks”, Autex Research Journal, 4(1):1-8.

Majumdar, P. K., Majumdar, A., 2004(b), “Predicting the Breaking Elongation of Ring Spun Cotton Yarns Using Mathematical, Statistical and Artificial Neural Network Models”, Textile Research Journal, 74(7):652-655, doi:10.1177/004051750407400717.

Manickam, S., Venkatakrishnan, S., 2014, “Research Article Studies on Fibre Quality of a Long Staple Cotton Variety Using High Volume Instrument and Advanced Fibre Information System for Fibre Quality Improvement”, Electronic Journal of Plant Breeding, 5:213-219.

Matusiak, M., Walawska, A., 2010, “Important Aspects of Cotton Colour Measurement”, Fibres and Textiles in Eastern Europe, 18(3):17-23.

McCulloch, W. S., Pitts, W. H., 1943, “A Logical Calculus of the Ideas Immanent in Nervous Activity”, Bulletin of Mathematical Biophysics, 5(4): 115–133, doi:10.1016/s0092-8240(05)80006-0.

Medina, G., 2014, "Drawing Back Propagation Neural Network", TeX, Accessed March 12, 2019, https://tex.stackexchange.com/questions/162326/drawing-back-propagation-neural-network.

Melgani, F., and Bruzzone, L., 2004, “Classification of Hyperspectral Remote Sensing Images with Support Vector Machines”, IEEE Transactions on Geoscience and Remote Sensing, 42(8):1778–1790, doi:10.1109/tgrs.2004.831865.

Meredith, W. R. Jr., Sasser, P. E., Rayburn, S. T., 1996, “Regional High Quality Fiber Properies as Measured by Conventional and AFIS Methods”, Proceedings of the Memphis Beltwide Cotton Conference, 2:1681-1684.

Miranda, T. M. R., Goncalves, A. R., Pessoa de Amorim, M. T., 2001, “Ultraviolet-Induced Crosslinking of Poly(Vinyl Alcohol) Evaluated by Principal Component Analysis of FTIR Spectra”, Polymer International, 50(10):1068–1072, doi:10.1002/pi.745.

Mishra, S., Prusty, R., Hota, P. K., 2015, “Analysis of Levenberg-Marquardt and Scaled Conjugate Gradient Training Algorithms for Artificial Neural Network Based LS and MMSE Estimated Channel Equalizers”, Proceedings of International Conference of Man and Machine Interfacing 2015 (MAMI 2015), 1:1-7, doi: 10.1109/MAMI.2015.7456617.

Moller, M. F., 1993, “A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning”, Neural Networks, 6(4):525-533, doi:10.1016/S0893-6080(05)80056-5.

Muhind, A., 2015, “Robust Classification with Convolutional Neural Networks”, University of Missouri-Columbia, Master of Science Thesis, USA, 53 pages.

Mukherjee, S., Osuna, E., Girosi, F., 1997, “Nonlinear Prediction of Chaotic Time Series Using Support Vector Machines”, Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop, 1:511-520, doi:10.1109/nnsp.1997.622433.

Muller, K. R., Smola, A. J., Ratsch G., Scholkopf, B., Kohlmorgen, J., Vapnik, V., 1997, “Predicting Time Series with Support Vector Machines”, (edited by Gerstner, W., Germond, A., Hasler, M., Nicoud, J. D.) Artificial Neural Networks ICANN’97, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 1327:999-1004, doi:10.1007/BFb0020283.

Murino, V., Bicego, M., Rossi, I. A., 2004, “Statistical Classification of Raw Textile Defects”, Proceedings of the 17th International Conference on Pattern Recognition, 2004, ICPR 2004, 4:311-314, doi:10.1109/icpr.2004.1333765.

Nelles, O., 2001, “Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models”, Springer, Berlin, 786 pages.

Nielsen, M. A., 2015, “Neural Networks and Deep Learning”, Determination Press, USA, 60 pages.

Nomikos, P., Macgregor, J. F., 1994, “Monitoring Batch Processes Using Multiway Principal Component Analysis”, AIChE Journal, 40(8):1361–1375, doi:10.1002/aic.690400809.

Nurwaha, D., Wang, X., 2011, “Prediction of Rotor Spun Yarn Strength Using Support Vector Machines Method”, Fibers and Polymers, 12(4): 546–549, doi:10.1007/s12221-011-0546-x.

Ogulata, S. N., Sahin, C., Erol, R., 2009, “Neural Network-Based Computer-Aided Diagnosis in Classification of Primary Generalized Epilepsy by EEG Signals”, Journal of Medical Systems, 33(2):107–112., doi:10.1007/s10916-008-9170-8.

Ogulata, S. N., Sahin, C., Ogulata, 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, 14(2):46-49.

Ohta, T., 2000, "Energy Systems and Adaptive Complexity", Applied Energy, 67(1-2):3-16, doi: 10.1016/S0306-2619(00)00004-0.

Oja, E., 1982, "Simplified Neuron Model as a Principal Component Analyzer", Journal of Mathematical Biology, 15(3):267–273, doi:10.1007/bf00275687.

Otukei, J. R., Blaschke, T., 2010, "Land Cover Change Assessment Using Decision Trees. Support Vector Machines and Maximum Likelihood Classification Algorithms", International Journal of Applied Earth Observation and Geoinformation, 12(1):27-31, doi:10.1016/j.jag.2009.11.002.

Ozkan, I., Kuvvetli, Y., Baykal, 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, doi:10.1080/00405000.2014.882041.

Oztekin, M., 2016, "Nowadays Textile in Fashion: Spandex, LYCRA & T400/PES", LinkedIn SlideShare, Last Visited on 12.03.2019, https://www.slideshare.net/minemithat/mdraft-dual-core-systems.

Pan, N., Hua, T., Qiu, Y., 2001, "Relationship Between Fiber and Yarn Strength", Textile Research Journal, 71(11):960-964, doi:10.1177/004051750107101105.

Patel, S., 2017, "Chapter 2: SVM (Support Vector Machine) – Theory", Machine Learning 101, Medium, Last Visited on 12.03.2019, https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72.

Pearlmutter, B. A., 1990, "Dynamic Recurrent Neural Networks", Carnegie Mellon University Computer Science, 29 pages.

Platt, J. C., 1999, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods”, (edited by Smola, A. J., Bartlett, P., Scholkopf, B., Schuurmans, D.) Advances in Large- Margin Classifiers, MIT Press, USA, pages: 61-74, doi:10.1.1.41.1639.

Platt, J. C., Nitschke, J., 1998, “Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines”, Technical Report, MSR-TR-98-14 available on www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/.

Pontil, M., Verri, A., 1998, “Support Vector Machines for 3D Object Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(6):637–646, doi:10.1109/34.683777.

Power, D. M., 1970, “Biometry. The Principles and Practice of Statistics in Biological Research”, Systematic Biology, 19(4):391-393, doi:10.2307/2412280.

Quesada, A., 2019, “5 Algorithms to Train a Neural Network”, Neural Designer, Accessed:12.03.2019, https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network.

Ramaswamy, G. N., Craft, S., Wartelle, L., 1995, “Uniformity and Softness of Kenaf Fibers for Textile Products”, Textile Research Journal, 65(12):765–770, doi:10.1177/004051759506501210.

Ramey, H. H., Beaton, P. G.,1989, “Relationships Between Short Fiber Content and HVI Fiber Length Uniformity”, Textile Research Journal, 59(2):101-108 doi:10.1177/004051758905900207.

Rao, C. R., 1964, “The Use and Interpretation of Principal Component Analysis in Applied Research”, Sankhya: The Indian Journal of Statistics, 26(4):329-358.

Ray, S., 2017, “Understanding Support Vector Machine From Examples (Along With Code)”, Analitics Vidhya, Last Visited on 12.03.2019, https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

Rojas, R., 1996, “Neural Networks A Systematic Introduction”, Springer-Verlag, Berlin, Hendelberg, 502 pages.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986, “Learning Representations by Back-Propagating Errors”, Nature,323(6088):533–536, doi:10.1038/323533a0.

Rumelhart, D. E., McClelland, J., Hinton, G. E., 1986, “A General Framework for Parallel Distributed Processing”, (edited by Rumelhart, D. E., McClelland, J. L.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, Cambridge, pages:45-76.

Salem, Y. B., Nasri, S., 2009, “Texture Classification of Woven Fabric Based on a GLCM Method and Using Multiclass Support Vector Machine”, 6th International Multi-Conference on Systems, Signals and Devices, 2009, 1:1-8, doi:10.1109/ssd.2009.4956737.

Sapankevych, N. I., Sankar, R., 2009, “Time Series Prediction Using Support Vector Machines: A Survey”, IEEE Computational Intelligence Magazine, 4(2):24–38, doi:10.1109/mci.2009.932254.

Scholkopf, B., Smola, A. J., Muller, K. R., 1997, “Kernel Principal Component Analysis”, (edited by Scholkopf, B., Smola, A. J., Muller, K. R.) Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, pages:327-352, doi:10.1007/BFb0020217.

Scholkopf, B., Smola, A. J., 2001, “Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond”, MIT Press, Cambridge, 644 pages.

Scholkopf, B., Sung, K. –K,, Burges, C. J. C., Girosi, F., 1997, “Comparing Support Vector Machines With Gaussian Kernels to Radial Basis Function Classifiers”, IEEE Transactions on Signal Processing, 45(11):2758-2765.

Serdaroglu, A., Ertuzun, A., Ercil, A., 2006, “Defect Detection in Textile Fabric Images Using Wavelet Transforms and Independent Component Analysis”, Pattern Recognition and Image Analysis, 16(1):61–64, doi:10.1134/s1054661806010196.

Sharda, R., Patil, R. B., 1992, “Connectionist Approach to Time Series Prediction: an Empirical Test”, Journal of Intelligent Manufacturing, 3(5):317–323, doi:10.1007/bf01577272.

Souid, H., Sahnoun, M., Baby, A., Cheikrouhou, M., 2012, “A Generalized Model for Predicting Yarn Global Quality Index”, The Open Textile Journal, 5:8-13.

Steinwart, I., Christmann, A., 2008, “Support Vector Machines”, Springer, New York, 601 pages.

Sun, J., Ming, Y., 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, doi:10.1177/0040517510391702.

Suryavathi, V., Sharma, Su., Sharma Sh., Saxena, P., Pandey, S., Grover, R., Kumar, S., Sharma, K. P., 2005, “Acute Toxicity of Textile Dye Wastewaters (Untreated and Treated) of Sanganer on Male Reproductive Systems of Albino Rats and Mice”, Reproductive Toxicology, 19(4):547–556, doi:10.1016/j.reprotox.2004.09.011.

Tang, Z., De Almeida, C., Fishwick, P. A., 1991, “Time Series Forecasting Using Neural Networks vs. Box- Jenkins Methodology”, Simulation: Transactions of the Society for Modeling and Simulation International, 57(5): 303–310, doi:10.1177/003754979105700508.

Tang, Z., Fishwick, P. A., 1993, “Feedforward Neural Nets as Models for Time Series Forecasting”, INFORMS Journal on Computing, 5(4):374–385, doi:10.1287/ijoc.5.4.374.

Tanikic, D., Manic, M., Devedzic, G., Stevic, Z., 2010, “Modelling Metal Cutting Parameters Using Intelligent Techniques”, Strojniski Vestnik-Journal of Mechanical Engineering, 56(1):52-62.

Thouraya, H., Ghith, A., Fayala, F., 2014, “A Principal Component Analysis (PCA) Method for Predicting the Correlation Between Some Fabric Parameters and Drape”, Autex Research Journal, 14(1):22-27.

Tokar, A. S., and Johnson, A. J., 1999, “Rainfall-Runoff Modeling Using Artificial Neural Networks”, Journal of Hydrologic Engineering, 4(3):232–239, doi:10.1061/(asce)1084-0699(1999)4:3(232).

Transtrum, M. K., Sethna, J. P., 2012, “Improvements to the Levenberg-Marquardt Algorithm for Nonlinear Least-Squares Minimization”, Cornell University, Computational Physics, arXiv:1201.5885.

Tsai, I-S., Lin, C. –H., Lin, J. –J., 1995, “Applying an Artificial Neural Network to Pattern Recognition in Fabric Defects”, Textile Research Journal, 65(3):123–130, doi:10.1177/004051759506500301.

Tu, J. V., 1996, “Advantages and Disadvantages of Using Artificial Neural Networks Versus Logistic Regression for Predicting Medical Outcomes”, Journal of Clinical Epidemiology, 49(11):1225-1231.

Turhan, Y., Toprakci, O., 2013, “Comparison of High-Volume Instrument and Advanced Fiber Information Systems Based on Prediction Performance of Yarn Properties Using a Radial Basis Function Neural Network”, Textile Research Journal, 83(2):130–147, doi:10.1177/0040517512445334.

Turing, A. M., 1950, “Computing Machinery And Intelligence”, Mind, 61(236):433–460, doi:10.1093/mind/lix.236.433.

Ulku, S., Ozipek, B., Acar, M., 1995, “Effects of Opening Roller Speed on the Fiber and Yarn Properties in Open-End Friction Spinning”, Textile Research Journal, 65(10):557-563, doi:2134/22053.

Uster Technologies, 2011, “Instruments”, Instruments :: Uster Technologies, found at www.uster.com/en/instruments/yarn-testing/uster-tensojet/. Last visited at:12.03.2019.

Vaskova, H., Hrano, J., 2017, “Spectroscopic Measurement of Textile Fibres”, Proceedings Proceedings of the 28th International DAAAM Symposium 2017, pages: 331–333, doi:10.2507/28th.daaam.proceedings.045.

Wang, C., Chen, Q., Hussain, M., Wu, S., Chen, J., Tang, Z., 2017, “Application of Principal Component Analysis to Classify Textile Fibers Based on UV-Vis Diffuse Reflectance Spectroscopy”, Journal of Applied Spectroscopy, 84(3):391-395.

Waterfall, J. J., Casey, F. P., Gutenkunst, R. N., Brown, K. S., 2006, “Sloppy-Model Universality Class and Vandermonde Matrix”, Physical Review Letters, 97(15):150601.

Werbos, P. J., 1988, “Generalization of Backpropagation with Application to a Recurrent Gas Market Model”, Neural Networks, 1(4):339–356, doi:10.1016/0893-6080(88)90007-x.

White, H., 1992, “Artificial Neural Networks: Approximation and Learning Theory”, Blackwell, New York, 329 pages.

Widrow, B., Rumelhart, D. E., Lehr, M., 1994, “Neural Networks: Applications in Industry. Business and Science”, Communications of the ACM, 37(3):93–105, doi:10.1145/175247.175257.

Xanthopoulos, P., Pardalos, P., Panos, M., Trafalis, T. B., 2013, “Robust Data Mining”, Springer, New York, 59 pages.

Yang, J. G., Lu, Z. J., Li, B. Z., 2012, “Quality Prediction in Complex Industrial Process with Support Vector Machine and Genetic Algorithm Optimization: A Case Study”, Applied Mechanics and Materials, 232:603-608.

Yegnanarayana, B., “Artificial Neural Networks”, Prentice-Hall of India, New Delhi, 476 pages.

Yeung, K. Y., Ruzzo, W. L., 2001, “Principal Component Analysis for Clustering Gene Expression Data”, Bioinformatics, 17(9):763–774, doi:10.1093/bioinformatics/17.9.763.

Zajac, Z., 2016, “Bayesian Machine Learning, Explained”, KDNuggets, Last visited: 12.03.2019, https://www.kdnuggets.com/2016/07/bayesian-machine-learning-explained.html.

Zeidman, M., Sawhney, P. S., 2002, “Influence of Fiber Length Distribution on Strength Efficiency of Fibers in Yarn”, Textile Research Journal, 72(3):216-220, doi: 10.1177/004051750207200306.

Zitzler, E., Thiele L., 1999 “Multiobjective Evolutionary Algorithms: a Comparative Case Study and the Strength Pareto Approach”, IEEE Transactions on Evolutionary Computation, 3(4):257–271, doi:10.1109/4235.797969.

Zou, H., Hastie, T., Tibshirani, R., 2006, “Sparse Principal Component Analysis”, Journal of Computational and Graphical Statistics, 15(2):265–286, doi:10.1198/106186006x113430.

CURRICULUM VITAE

Enver Can DORAN graduated in 2005 from Sainte Pulcherie French High School in Istanbul. He enrolled in Industrial Engineering Department of Çukurova University in 2010. He graduated from the department of Industrial Engineering in 2014 and began his Master of Science in Industrial Engineering Department in same year. Presently, he works as a research assistant in Industrial Engineering Department of Çukurova University.

APPENDIX

APPENDIX A (Results of Artificial Neural Network Models)

Input Type	Learning Algorithm	Number of Nodes	Test Reg	Test Mse
Standart	Lavenberg-Marquardt	10	0,846457	12521,2
Standart	Lavenberg-Marquardt	12	0,834601	13807,77
Standart	Lavenberg-Marquardt	14	0,893769	8446,153
Standart	Lavenberg-Marquardt	16	0,906406	6991,567
Standart	Lavenberg-Marquardt	18	0,922577	5850,712
Standart	Lavenberg-Marquardt	20	0,788421	19940,31
Standart	Lavenberg-Marquardt	22	0,92832	5319,52
Standart	Lavenberg-Marquardt	24	0,875118	9825,187
Standart	Lavenberg-Marquardt	26	0,917794	5745,671
Standart	Lavenberg-Marquardt	28	0,870645	10662,67
Standart	Bayesian Regularization	10	0,743443	25100,29
Standart	Bayesian Regularization	12	0,916922	6702,337
Standart	Bayesian Regularization	14	0,77355	20716,54
Standart	Bayesian Regularization	16	0,90337	7271,482
Standart	Bayesian Regularization	18	0,840967	13763,46
Standart	Bayesian Regularization	20	0,850945	12119,83
Standart	Bayesian Regularization	22	0,787762	23594,17
Standart	Bayesian Regularization	24	0,882888	9233,621
Standart	Bayesian Regularization	26	0,842148	13242,36
Standart	Bayesian Regularization	28	0,859736	10598,11
Standart	Scaled Conjugate Gradient	10	0,9374	4184,002
Standart	Scaled Conjugate Gradient	12	0,940512	3950,149
Standart	Scaled Conjugate Gradient	14	0,94412	3933,165

Standart	Scaled Conjugate Gradient	16	0,893049	7393,948
Standart	Scaled Conjugate Gradient	18	0,904034	6810,066
Standart	Scaled Conjugate Gradient	20	0,906355	6274,709
Standart	Scaled Conjugate Gradient	22	0,907911	6219,411
Standart	Scaled Conjugate Gradient	24	0,92906	5272,477
Standart	Scaled Conjugate Gradient	26	0,903757	6331,95
Standart	Scaled Conjugate Gradient	28	0,912155	5936,801
ANOVA	Lavenberg-Marquardt	10	0,864925	9941,364
ANOVA	Lavenberg-Marquardt	12	0,933218	4576
ANOVA	Lavenberg-Marquardt	14	0,890126	8011,599
ANOVA	Lavenberg-Marquardt	16	0,889434	7407,969
ANOVA	Lavenberg-Marquardt	18	0,919641	5429,093
ANOVA	Lavenberg-Marquardt	20	0,918225	5686,331
ANOVA	Lavenberg-Marquardt	22	0,61752	37075,04
ANOVA	Lavenberg-Marquardt	24	0,918971	5742,613
ANOVA	Lavenberg-Marquardt	26	0,524376	77719,58
ANOVA	Lavenberg-Marquardt	28	0,851777	10867,28
ANOVA	Bayesian Regularization	10	0,850945	12119,83
ANOVA	Bayesian Regularization	12	0,787762	23594,17
ANOVA	Bayesian Regularization	14	0,882888	9233,621
ANOVA	Bayesian Regularization	16	0,90337	7271,482
ANOVA	Bayesian Regularization	18	0,840967	13763,46
ANOVA	Bayesian Regularization	20	0,850945	12119,83
ANOVA	Bayesian Regularization	22	0,787762	23594,17
ANOVA	Bayesian Regularization	24	0,743443	25100,29
ANOVA	Bayesian Regularization	26	0,916922	6702,337

ANOVA	Bayesian Regularization	28	0,77355	20716,54
ANOVA	Scaled Conjugate Gradient	10	0,891336	7061,889
ANOVA	Scaled Conjugate Gradient	12	0,868515	8614,709
ANOVA	Scaled Conjugate Gradient	14	0,89537	6884,168
ANOVA	Scaled Conjugate Gradient	16	0,863289	8866,236
ANOVA	Scaled Conjugate Gradient	18	0,851752	9675,653
ANOVA	Scaled Conjugate Gradient	20	0,887274	7520,611
ANOVA	Scaled Conjugate Gradient	22	0,895776	6833,632
ANOVA	Scaled Conjugate Gradient	24	0,8574	9873,181
ANOVA	Scaled Conjugate Gradient	26	0,882124	7986,363
ANOVA	Scaled Conjugate Gradient	28	0,838848	11317,66
PCA	Lavenberg-Marquardt	10	0,92481	5177,961
PCA	Lavenberg-Marquardt	12	0,854333	12255,99
PCA	Lavenberg-Marquardt	14	0,933162	4862,266
PCA	Lavenberg-Marquardt	16	0,916068	6122,516
PCA	Lavenberg-Marquardt	18	0,918129	7694,095
PCA	Lavenberg-Marquardt	20	0,937878	4484,552
PCA	Lavenberg-Marquardt	22	0,833769	12487,38
PCA	Lavenberg-Marquardt	24	0,913538	6429,59
PCA	Lavenberg-Marquardt	26	0,840215	12358,29
PCA	Lavenberg-Marquardt	28	0,868949	10465,67
PCA	Bayesian Regularization	10	0,882888	9233,621
PCA	Bayesian Regularization	12	0,842148	13242,36
PCA	Bayesian Regularization	14	0,859736	10598,11
PCA	Bayesian Regularization	16	0,743443	25100,29
PCA	Bayesian Regularization	18	0,916922	6702,337

PCA	Bayesian Regularization	20	0,77355	20716,54
PCA	Bayesian Regularization	22	0,90337	7271,482
PCA	Bayesian Regularization	24	0,840967	13763,46
PCA	Bayesian Regularization	26	0,850945	12119,83
PCA	Bayesian Regularization	28	0,787762	23594,17
PCA	Scaled Conjugate Gradient	10	0,935472	4341,078
PCA	Scaled Conjugate Gradient	12	0,930723	4754,628
PCA	**Scaled Conjugate Gradient**	**14**	**0,956858**	**2957,188**
PCA	Scaled Conjugate Gradient	16	0,953628	3206,037
PCA	Scaled Conjugate Gradient	18	0,930319	4813,612
PCA	Scaled Conjugate Gradient	20	0,888719	7225,417
PCA	Scaled Conjugate Gradient	22	0,922894	5099,243
PCA	Scaled Conjugate Gradient	24	0,922532	5163,936
PCA	Scaled Conjugate Gradient	26	0,9187	5763,331
PCA	Scaled Conjugate Gradient	28	0,949082	3439,994

APPENDIX B (RESULTS OF SUPPORT VECTOR MACHINE MODELS)

Input Type	SVM Algorithm	Optimization Routine	MSE Test	R Test
Standart	Gaussian Kernel	Iterative Single Data Algorithm (ISDA)	8353,00	0,871424
Standart	Gaussian Kernel	Quadratic Programming (L1QP)	8351,11	0,869947
Standart	Gaussian Kernel	Sequential Minimal Optimization	8350,99	0,869949
Standart	Linear Kernel	Iterative Single Data Algorithm (ISDA)	5673,74	0,916966
Standart	Linear Kernel	Quadratic Programming (L1QP)	3336,45	0,950132
Standart	Linear Kernel	Sequential Minimal Optimization	3440,96	0,948474
Standart	Polynomial Kernel	Iterative Single Data Algorithm (ISDA)	23459,20	0,763782
Standart	Polynomial Kernel	Quadratic Programming (L1QP)	Not Availabel	Not Available
Standart	Polynomial Kernel	Sequential Minimal Optimization	348417,79	0,251064
ANOVA	Gaussian Kernel	Iterative Single Data Algorithm (ISDA)	6885,09	0,896365
ANOVA	Gaussian Kernel	Quadratic Programming (L1QP)	7136,27	0,889969
ANOVA	Gaussian Kernel	Sequential Minimal Optimization	7136,31	0,889971
ANOVA	Linear Kernel	Iterative Single Data Algorithm (ISDA)	10217,97	0,838504
ANOVA	Linear Kernel	Quadratic Programming (L1QP)	8961,09	0,859158
ANOVA	Linear Kernel	Sequential Minimal Optimization	8931,86	0,85967
ANOVA	Polynomial Kernel	Iterative Single Data Algorithm (ISDA)	39212,77	0,741037
ANOVA	Polynomial Kernel	Quadratic Programming (L1QP)	Not Available	Not Available

ANOVA	Polynomial Kernel	Sequential Minimal Optimization	9545,89	0,854558
PCA	Gaussian Kernel	Iterative Single Data Algorithm (ISDA)	8244,45	0,872862
PCA	Gaussian Kernel	Quadratic Programming (L1QP)	8273,65	0,871311
PCA	Gaussian Kernel	Sequential Minimal Optimization	8273,59	0,871312
PCA	Linear Kernel	Iterative Single Data Algorithm (ISDA)	7103,60	0,929063
PCA	**Linear Kernel**	**Quadratic Programming (L1QP)**	**3238,64**	**0,951497**
PCA	Linear Kernel	Sequential Minimal Optimization	3257,07	0,951211
PCA	Polynomial Kernel	Iterative Single Data Algorithm (ISDA)	934451,52	0,748947
PCA	Polynomial Kernel	Quadratic Programming (L1QP)	Not Available	Not Available
PCA	Polynomial Kernel	Sequential Minimal Optimization	415221,36	0,231336