# CO364 - SOFT COMPUTING

## PROJECT REPORT

**TOPIC** : Textile Waste Damage Analysis (Project 4)


**MEMBERS** :

**TEAM 1** : Adya Kiran (15CO106)
Naladala Indukala (15CO230)

**TEAM 2** : Giffin Suresh (15CO119)
Shruti Tiwari (15CO149)

**TEAM 3** : Harita Reddy (15CO217)
Namratha Raj M (15CO132)

**TEAM 4** : Sahaya Raj Helan Agnes Sharan (15CO142)
Manali Sushil Gala (15CO214)

# Textile Waste Damage Analysis | Project Report (Project 4)

## CONTENTS


**Objectives** **3**

**Abstract** **3**

**Introduction** **3**

Feature Selection 3

Predictive Model using Artificial Neural Network 4

PSO 4

Predictive Model using Bayesian 4

**Literature Survey** **5**

Feature Selection using Fuzzy Set 5

Predictive Model using Artificial Neural Network 5

Particle Swarm Optimization 8

Predictive model using Bayesian 9

**Implementation Methods** **10**

Feature Selection using Fuzzy Set 10

Predictive Model using Artificial Neural Network 12

Optimization Using PSO 13

Predictive Model using Bayesian 14

**Experiment and Result Analysis** **14**

**Conclusion** **17**

**References** **17**

Feature Selection papers 17

ANN Papers 18

PSO Papers 18

Bayesian Papers 19

# Objectives

1) Feature Selection using Fuzzy set
2) Predictive Model using Artificial Neural Network
3) Optimize predictive model using Particle Swarm Optimization
4) Predictive Model using Bayesian

# Abstract

The textile industry has been condemned as being one of the world's worst offenders in terms of pollution because it requires a great amount of two components: 1) Chemicals - as many as 2,000 different chemicals are used in the textile industry, from dyes to transfer agents and 2) Water - a finite resource that is quickly becoming scarce, is used at every step of the process both to convey the chemicals used during that step and to wash them out before beginning the next step. The water becomes full of chemical additives and is then expelled as wastewater which in turn pollutes the environment by the effluent's heat, its increased pH, and because it is saturated with dyes, de-foamers, bleaches, detergents, and many other chemicals used during the process. This project aims to analyse the general trend of the waste produced by the textile industries, in terms of the amount of fabric wasted, and predict how much will be wasted in the future. Firstly, a thorough Feature Selection is carried out, using Fuzzy Sets, in MATLAB, to determine which factors contribute most to waste generation. Then, a basic Artificial Neural Network is designed using the Predictive Regression Model which gives a general idea about the amount of wastage. Thirdly, the predictive model is optimized using Particle Swarm Optimization to increase the accuracy of results. And finally, a predictive model is designed using the Bayesian technique which is particularly advantageous where some of the prior knowledge is lacking or vague, so that one is not forced to guess values for attributes that are unknown. The performance of each stage is tabulated and compared. This analysis enables the industries to make a smarter choice in terms of the raw materials they need and hence minimize textile wastes and subsequently, environmental pollution.

# Introduction

## Feature Selection

Modern day datasets are very rich in information with data collected from millions of IoT devices and sensors. This makes the data high dimensional and it is quite common to see datasets with hundreds of features and is not unusual to see it go to tens of thousands. Feature Selection is a very critical component in a Data Scientist's workflow. When presented data with very high

dimensionality, models usually choke. We use feature selection techniques to get insights about the features and their relative importance with the target variable.

## Predictive Model Using ANN

The textile industry is a significant contributor to many national economies, encompassing both small and large-scale operations worldwide. In terms of its output or production and employment, the textile industry is one of the largest industries in the world. Textile processing industry is characterised not only by the large volume of water required for various unit operations but also by the variety of chemicals used for various processes. There is a long sequence of wet processing stages requiring inputs of water, chemical and energy and generating wastes at each stage. The other feature of this industry, which is a backbone of fashion garment, is large variation in demand of type, pattern and colour combination of fabric resulting into significant fluctuation in waste generation volume and load. We attempt to predict the amount of waste generated using multiple linear regression.Regression techniques, being inherently predictive is nature is well-suited for this task.

## Particle Swarm Optimization on ANN

We use particle swarm optimization to optimize the ANN model that we have built in the previous phase. Particle Swarm Optimization (PSO) is an evolutionary algorithm modeled on the behavior of flock of birds flying towards their resources. It has been used in several other works for determining both architecture of neural network (layers) as well as the weights and biases. It is similar to genetic algorithms except for the fact that it doesn't have crossover and mutation. It is carried out over a number of iterations. Practically, we can imagine a flock of birds searching for food and the one closest to food chirps the loudest, and all other birds move in that direction. This cycle continues until one of the birds has come upon the food. Similarly, in the algorithm, initially, a random set of solutions is initialized. The algorithm searches for optima by updating generations, thus moving closer to the solution.

## Predictive Model using Bayesian

A Naive Bayes can be extended to real-valued attributes, most commonly by assuming a Gaussian distribution. This extension of naive Bayes is called Gaussian Naive Bayes. Other functions can be used to estimate the distribution of the data, but the Gaussian (or Normal distribution) is the easiest to work with because you only need to estimate the mean and the standard deviation from your training data. We calculated the probabilities for input values for each class using a frequency. With real-valued inputs, we can calculate the mean and standard deviation of input values (x) for each class to summarize the distribution. This means that in addition to the probabilities for each class, we must also store the mean and standard deviations for each input variable for each class, and using these, we predict the output values of the test input.

# Literature Survey

## Feature Selection using Fuzzy Set

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for four reasons:

- Shorter training times
- Simplification of models to make them easier to interpret by researchers/users
- To avoid the curse of dimensionality
- Enhanced generalization by reducing overfitting

The central premise when using a feature selection technique is that the data contains many features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information. Redundant or irrelevant features are two distinct notions, since one relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.

The dimension of the data is reduced by finding a small set of important features which can give good classification performance. The concept of fuzzy sets is used for the same.

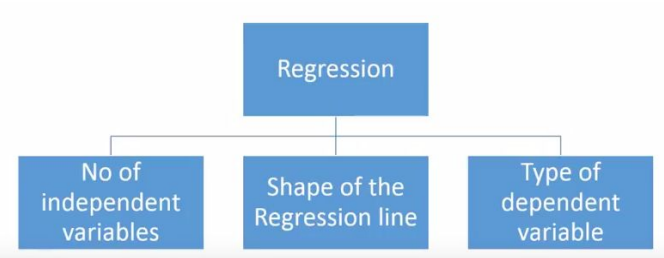
## Predictive Model using Artificial Neural Network

### ***1)Predictive Modelling***

***Predictive modeling*** is a process that uses data mining and probability to forecast outcomes. Each model is made up of a number of predictors, which are variables that are likely to influence future results. Once data has been collected for relevant predictors, a statistical model is formulated. The model may employ a simple linear equation, or it may be a complex neural network, mapped out by sophisticated software. As additional data becomes available, the statistical analysis model is validated or revised.

## ***2)Multiple Linear Regression***

***Regression analysis*** is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor). The features that classify regression into its many types is shown below.




*Multiple linear regression* attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable *x* is associated with a value of the dependent variable *y*. The population regression line for *p* explanatory variables $x_1$, $x_2$, ... , $x_p$ is defined to be $U_y = B_0 + B_1x_1 + B_2x_2 + ... + B_px_p$. This line describes how the mean response $U_y$ changes with the explanatory variables. The observed values for *y* vary about their means $U_y$ and are assumed to have the same standard deviation S. The fitted values $b_0$, $b_1$, ..., $b_p$ estimate the parameters $B_0$, $B_1$, ..., $B_p$ of the population regression line.

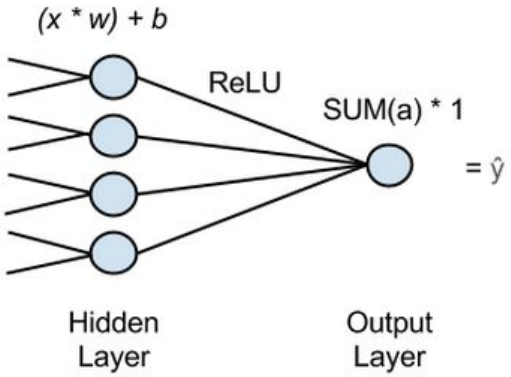
Since the observed values for *y* vary about their means $U_y$, the multiple regression model includes a term for this variation. In words, the model is expressed as DATA = FIT + RESIDUAL, where the "FIT" term represents the expression $B_0 + B_1x_1 + B_2x_2 + ... B_px_p$. The "RESIDUAL" term represents the deviations of the observed values *y* from their means $U_y$, which are normally distributed with mean 0 and variance S. The notation for the model deviations is E.

**Formally, the model for multiple linear regression, given *n* observations, is**
**$y_i = B_0 + B_1x_{i1} + B_2x_{i2} + ... B_px_{ip} + E_i$ for $i = 1,2, ... n$.**

In the least-squares model, the best-fitting line for the observed data is calculated by minimizing the sum of the squares of the vertical deviations from each data point to the line (if a point lies on the fitted line exactly, then its vertical deviation is 0). Because the deviations are first squared, then summed, there are no cancellations between positive and negative values. The least-squares estimates $b_0$, $b_1$, ... $b_p$ are usually computed by statistical software.
The values fit by the equation $b_0 + b_1x_{i1} + ... + b_px_{ip}$ are denoted $ybar_i$, and the residuals $e_i$ are equal to $y_i$ - $ybar_i$, the difference between the observed and fitted values. The sum of the residuals is equal to zero.
The variance $S^2$ may be estimated by $\boldsymbol{S^2 =}$ Summation$(e_i)^2$ / (n – p – 1), also known as the mean-squared error (or MSE).
The estimate of the standard error *s* is the square root of the MSE.

## ***3)Artificial Neural Network***

An ***Artificial Neural Network (ANN)*** is a computational model based on the structure and functions of biological neural networks. Information that flows through the network affects the structure of the ANN because a neural network changes - or learns, in a sense - based on that input and output.
ANNs are considered nonlinear statistical data modeling tools where the complex relationships between inputs and outputs are modeled or patterns are found.
ANN is also known as a neural network.
Neural networks are used for the purpose of clustering through unsupervised learning, classification through supervised learning, or regression. That is, they help group unlabeled data, categorize labeled data or predict continuous values.




Artificial Neural Network using Regression

In the diagram above, x stands for input, the features passed forward from the network's previous layer. Many x's will be fed into each node of the last hidden layer, and each x will be multiplied by a corresponding weight, w.
The sum of those products is added to a bias and fed into an activation function. In this case the activation function is a *rectified linear unit* (ReLU), commonly used and highly useful because it doesn't saturate on shallow gradients as sigmoid activation functions do.
For each hidden node, ReLU outputs an activation, a, and the activations are summed going into the output node, which simply passes the activations' sum through.
That is, a neural network performing regression will have one output node, and that node will just multiply the sum of the previous layer activations by 1. The result will be ŷ, "y hat", the network's estimate, the dependent variable that all x's map to.
To perform backpropagation and make the network learn, we simply compare ŷ to the ground-truth value of y and adjust the weights and biases of the network until error is minimized, much as we would with a classifier. Root-means-squared-error (RMSE) could be the loss function.

In this way, you we use a neural network to get the function relating an arbitrary number of independent variables x to a dependent variable y that you're trying to predict.

## Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an evolutionary algorithm modeled on the behavior of flock of birds flying towards their resources. A swarm of particles, where each of them represents a point in the solution space is the major component of this algorithm. In each iteration, the particles move towards a new position based on the updated velocities for each particle. The new position is expected to be a better solution.
The three things the algorithm keeps track of are:

- Target value or destination
- Global best (gbest) value indicating the particle which is the closest to the destination/target
- Stopping value to indicate when the algorithm should stop (in case the target isn't reached)

Each particle has the following components:

- Data (A possible solution of the problem)
- Velocity
- pbest or personal best indicating the closest the particle has approached the target

Updation of particle's position is done as follows:
v[i] = v[i] + c1 * rand() * (pbest[i] - present[i]) + c2 * rand() * (gbest[i] - present[i])
present[i] = present[i] + v[i]
Where, v[i] is the particle velocity, persent[i] is the current particle (solution). rand () is a random number between (0,1) and c1, c2 are learning factors.

In the earliest works[1], PSO was introduced as a concept for optimizing non-linear functions. In the application of PSO to neural network, the position of the particle is nothing but the values of the network's weights and biases. Several works have shown that applying PSO for training the neural network's weights as well as for determining the architecture of the neural network is a valid alternative to the traditional method of using back propagation.
It has been observed by Gudise et al.[2] from the number of computations required by each algorithm that PSO requires less iterations to achieve the same error goal as with the Back Propagation Algorithm, making it suitable for applications that require fast learning algorithms. Similar comparison has been done by Garro et al. [8] in which they have they have observed that PSO presents a better accuracy than classical algorithm in linear problems and for the cases of non-linear problem they could not distinguish a clear winner. But PSO could be used to evolve certain parameters that BP cannot, for example parameters to find the best topology, the best transfer function and so on. Also, they have taken a note that PSO is more useful in there are several problems where the solution space is very complex due to non-differentiable spaces. Carvalho et al.[5] have used three benchmark classification problems of the medical

field (Cancer, Diabetes and Heart) to evaluate the various flavours of PSO including PSO-PSO and PSO-PSO:WD[6]. The PSO-PSO methodology proposed by them used a PSO algorithm to search for architectures and a PSO with weight decay (PSO:WD) to search for weights. They have observed that the PSO-PSO based approach provides a valid alternative to finding the weights using the traditional Back Propagation technique.
Garro et al.[4] have developed a methodology that automatically designs an ANN using particle swarm optimization algorithms such as Basic Particle Swarm Optimization (PSO), Second Generation of Particle Swarm Optimization (SGPSO), and a model of PSO called NMPSO.Their aim was evolve, at the same time, the three principal components of an ANN: the set of synaptic weights, the connections or architecture, and the transfer functions for each neuron.
The choice of PSO parameters can hugely impact the performance of the PSO algorithm. Selection of PSO parameters for good performance has been an active area of research[7].

The current research directions being pursued in particle swarm optimization are[9]:

- Theoretical aspects
- Matching algorithms (or algorithmic components) to problems
- Application to more and/or different class of problems
- Parameter selection

## Predictive model using Bayesian

A **Bayesian classifier** is based on the idea that the role of a (natural) class is to predict the values of features for members of that class. Examples are grouped in classes because they have common values for the features. Such classes are often called **natural kinds**. In this section, the target feature corresponds to a discrete **class**, which is not necessarily binary.
The idea behind a Bayesian classifier is that, if an agent knows the class, it can predict the values of the other features. If it does not know the class, Bayes' rule can be used to predict the class given (some of) the feature values. In a Bayesian classifier, the learning agent builds a probabilistic model of the features and uses that model to predict the classification of a new example.
A **latent variable** is a probabilistic variable that is not observed. A Bayesian classifier is a probabilistic model where the classification is a latent variable that is probabilistically related to the observed variables. Classification then become inference in the probabilistic model.
In probability theory and statistics, **Bayes' theorem** describes the probability of an event, based on prior knowledge of conditions that might be related to the event. For example, if cancer is related to age, then, using Bayes' theorem, a person's age can be used to more accurately assess the probability that they have cancer, compared to the assessment of the probability of cancer made without knowledge of the person's age.
Bayes' theorem is stated mathematically as the following equation:[2]

$P(A \mid B) = [P(B \mid A) P(A)] / P(B)$ ,where A and B are events and P(B) != 0.

P ( A | B ) is a conditional probability: the likelihood of event A occurring given that B is true.

P ( B | A ) is also a conditional probability: the likelihood of event B occurring given that A is true.

P ( A ) and P ( B ) are the probabilities of observing A and B independently of each other; this is known as the marginal probability.

In our implementation, we have used the **Gaussian Naive Bayes classifier.** A Gaussian Naive Bayes algorithm is a special type of NB algorithm. It's specifically used when the features have continuous values. It's also assumed that all the features are following a gaussian distribution i.e, normal distribution.

# Implementation Methods

## Feature Selection using Fuzzy Set

The proposed selection method based on Fuzzy Information Measure consists of three steps:

- Fuzzy Information Measure
- Feature Subset Selection
- Feature Selection Classification

The first step defines the corresponding membership function of each fuzzy set of each feature. In this Fuzzy information measure, the numeric feature can be discretized into finite fuzzy sets. The number of fuzzy sets is affecting the result of classification. The discretization of a numeric feature is an essential process before feature selection. Fuzzy C Means clustering algorithms are used to generate cluster centers and construct membership function to fuzzify all features. We calculated the fuzzy entropy for features in the dataset by using the membership function in MATLAB. The second step selects feature subsets based on the proposed fuzzy entropy measure focusing on boundary samples. In third step, the output for second step is classified by using Weka tool for accuracy calculation. The threshold value Tc is used in this proposed algorithm for constructing the membership functions of the fuzzy sets of a numeric feature and the threshold value Tr is used in the proposed algorithm for feature subset selection.

*Fuzzy C-Means clustering algorithm:*

This algorithm divides the numeric feature to k clusters. Membership function can be produced by using the centers of these clusters. For this purpose, the centers of these clusters are used as centers of fuzzy subsets. Increasing the number of clusters causes overfitting. To solve this problem a threshold value(Tc ) is used.

Step 1: Use Fuzzy C-means cluster to generate k cluster based on the values of a feature f, where k≥2.

Step 2: Calculate a new cluster center mi for each cluster until each cluster is not changed.

Step 3: Construct the membership functions of the fuzzy sets based on the k cluster centers for the feature f.

Assign cluster centers to the ith cluster center mi, where $mL$ denotes the left cluster center of $mi$ , $mR$ denotes the right cluster center of $mi$ , Umin represents the minimum value of a feature, and Umax denotes the maximum value of a feature.

$$mL = \begin{cases} Umin - (mi - Umin), \text{ if } i = 1 \\ mi-1 \text{ , Otherwise} \end{cases}$$

$$mR = \begin{cases} Umax + (Umax - mi), \text{ if } i = k \\ mi+1 \text{ , Otherwise} \end{cases}$$

Construct a membership function μvi of the fuzzy set vi based on the ith cluster center mi

$$\mu vi\,(x) = \begin{cases} max\,\{1 - (mi-x \,/\, mi-mL)\text{ , }0\} \text{ if } x \le mi \\ max\,\{1 - (x-mi \,/\, mR-mi)\text{ , }0\} \text{ if } x > mi \end{cases}$$

Step 4: Calculate the fuzzy entropy FE of a fuzzy set Â.

The FCM algorithm computes the membership of each pattern in all clusters (centroids vectors mj ) and then normalizes the membership of each specific pattern xk in all clusters. If this process is to be applied along each feature rather than each pattern, then the membership of each feature in all clusters is mapped to sum up to one.

Step 5: Calculate the entropy of the class H( C ) $H(C) = \sum pi \log 2\ pi$ , i varies from 1 to n

*Feature Subset Selection:*

Create Fuzzy Membership Grade Matrix for each feature f;
Calculate fuzzy Information of each feature;
While *true*
{
Select feature f with Maximum value of fuzzy information;
Add f into previous selected subset and update combined Fuzzy Membership Grade Matrix ; Calculate fuzzy Information of new selected subset according to Tr;
If (new Fuzzy information value > previous fuzzy Information value) or (fuzzy Information =zero) or (there is no additional feature for selection)
Break;

} While feature exists in dataset D
}

Among the 9 features provided in the dataset, our feature selection method got 6 features as important features which are then fed to ANN to get required results. The accuracy using these 6 features is 91.6 per cent. Feature selection was performed across around 11000 data sets for each needle type.

## Predictive Model using Artificial Neural Network

Keras framework is used. Keras has high-level neural networks written in python and which focuses on deep learning. Programming language is chosen as Python because of its flexibility and greater relevance in this field.

A random seed of 7 is chosen for reproducibility. We load the train and test datasets using numpy.loadtxt into our program.We have experimentally deduced the architecture of our network. We initially start off with 6 nodes, that is, the input layer has 6 nodes corresponding to the number of features. Following the rules that the next layer will contain thrice the number of nodes (including the bias), we have 21 nodes in the first hidden layer and 63 nodes in the second hidden layer. Finally we have 1 node in output layer which predicts the fabric wastage. Layers are added using add.Dense() where the dense signifies that the weight matrix should be dense.
Once the architecture is set we compile the model using compile(). For this problem, we use ‘rmsprop’ as our optimizer and the error analysis is done using ‘Mean Squared Error’. Rmsprop is a variant of the Gradient Descent method where we try to minimize error function by moving to its local minima. The learning rate is taken as 0.001 and decay is set to zero.

In rmsprop, we keep a moving average of squared gradient for each weight,
MeanSquare(w, t) = 0.9 * MeanSquare(w, t−1) + 0.1 * ( (∂E/∂w)(t) ) ^2

We fit our dataset using the model we create using fit(). The batch size for all operations is taken as 32 and we train our model for 1000 epochs. The training data is subdivided into train and validation using a 3:1 ratio.We finally send the test data to the model to obtain the calculated output using predict() function.

We compare the output obtained with the actual output and determine the efficiency of the network. We get sufficiently accurate results as shown in the results section. We can further apply optimization techniques on this model to improve its accuracy.

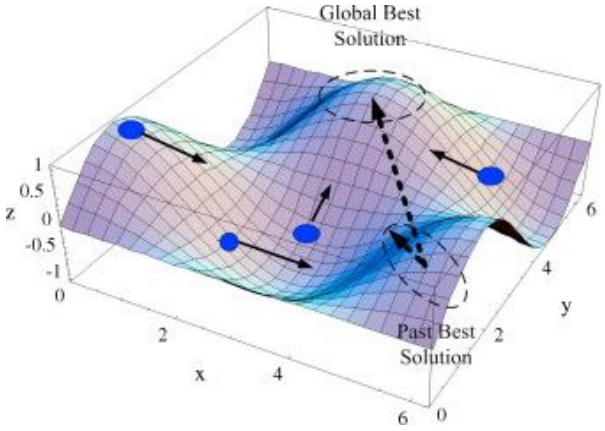
# Optimization Using PSO

The ANN model from the previous step has been trained using PSO algorithm.The most popular method of finding weights and biases is to use back propagation technique, but here PSO is used to find optimal weights and biases of the network[4].
Algorithm:
PSO is initialized with a group of random particles (solutions) which then searches for optima by updating generations. In every iteration, each particle is updated by following two best values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called ‘pbest’. Another best value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called ‘gbest’. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called ‘lbest’.

After finding the two best values, the particle updates its velocity and positions with following equations:

v[i] = v[i] + c1 * rand() * (pbest[i] - present[i]) + c2 * rand() * (gbest[i] - present[i])
present[i] = present[i] + v[i]

Where: v[i] is the particle velocity, persent[i] is the current particle (solution). pbest[i] and gbest[i] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors.




Python’s Pyswarm Package has been used to implement the algorithm with the below mentioned parameters.

**Fitness Function**: Mean squared error function has been used to find out fitness value or loss to update Velocity and Position to find optimised weights and biases.This gives the loss value between desired output and ANN output.

**Parameters**:
Cognitive Parameter(c1) ,Social Parameter (c2) are the learning factors to optimise the values based on particles ability to explore(global search) and exploit(local search) the space.Inertia(w) favors exploitation with lower value as velocity gets updated easily and exploration with higher velocity.

We have set these three values to a particular value based on our experiment results and precious research results, which have shown the following values to be optimized ones[7].
c1=2,c2=2,w=0.25
Minimum number of swarm particles model gives better result. We used this value based on our experiment results.
Number of particles in the space = 5

## Predictive Model using Bayesian

The following explains the organization and functionality of the code modules.

1.Data Load and Split

The data is loaded into the program as Comma Separated Values (.csv). The loaded dataset is then cleaned up by removing the header line. The data is then converted to the integer form by multiplying the output values with the highest power of 10 needed to remove the decimal point. The data is then split based on the input splitRatio into training and testing datasets. This split is done by randomly choosing an index value from 0 to the length of remaining dataset - 1. This randomly chosen index is then used to load the train data values. This is done until all the required data values are loaded. The dataset remaining is the test dataset.

2. Summarization of Classes

The summarization of classes is firstly done by segregating the dataset vectors to clusters such that all input values that produce a particular output class are grouped into one dictionary entry as a key value pair. These separated classes are then summarized in terms of the mean and standard deviation of each attribute in the input set for each output class.

3. Predictions

The prediction phase checks each target class for the probability of input belonging to that class by means of the mean and standard deviation calculated in the previous phase. The class with the highest probability is then displayed are the required target class.

# Experiment and Result Analysis

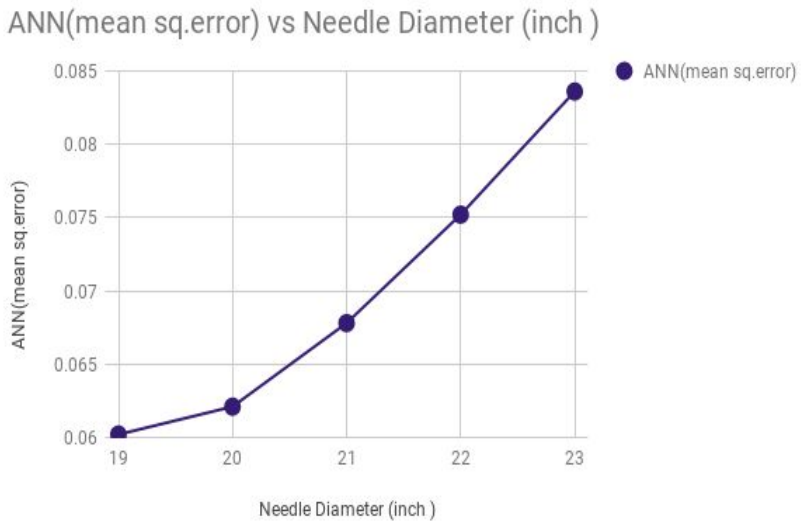
**DATASET**
Our dataset has 5 components, each component having a different value of needle diameter.

We have computed and compared the error for each needle diameter, and have compared the errors obtained for Normal ANN, ANN Optimized with PSO and Bayesian

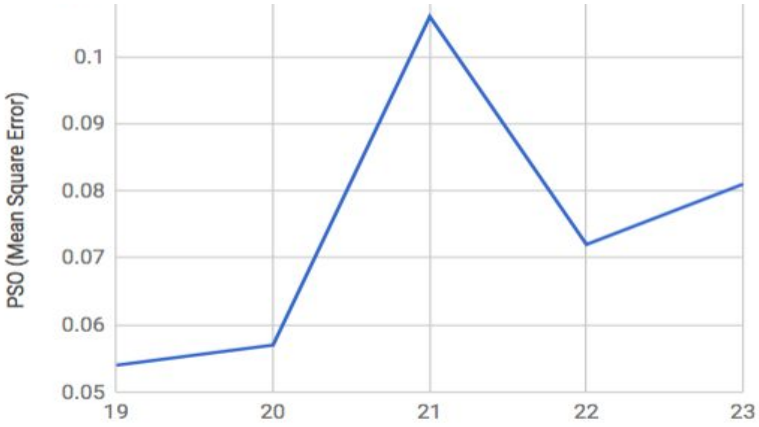
Result Comparison
No. of train samples=7470
No. of test samples=3680

| Needle Diameter (inch ) | ANN(mean sq.error) | PSO(mean sq.error) | Bayesian(mean sq.error) |
|---|---|---|---|
| 19 | 0.0602 | 0.054 | 0.044 |
| 20 | 0.0621 | 0.057 | 0.058 |
| 21 | 0.0678 | 0.106 | 0.062 |
| 22 | 0.0752 | 0.072 | 0.068 |
| 23 | 0.0836 | 0.081 | 0.0421 |

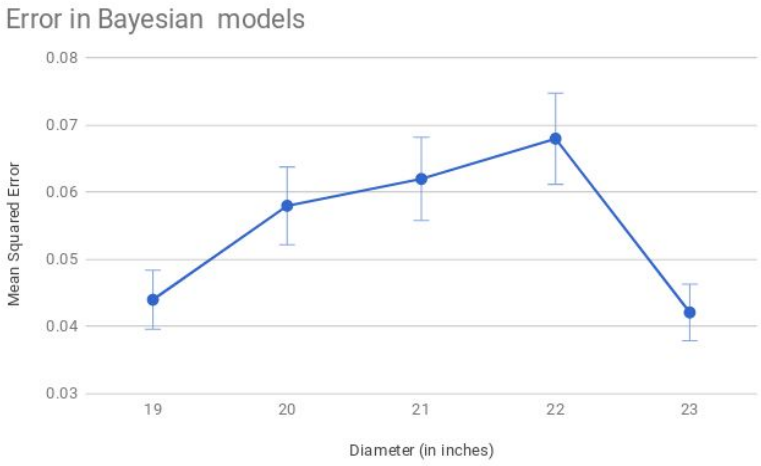
**RESULTS IN TERMS OF GRAPHS**



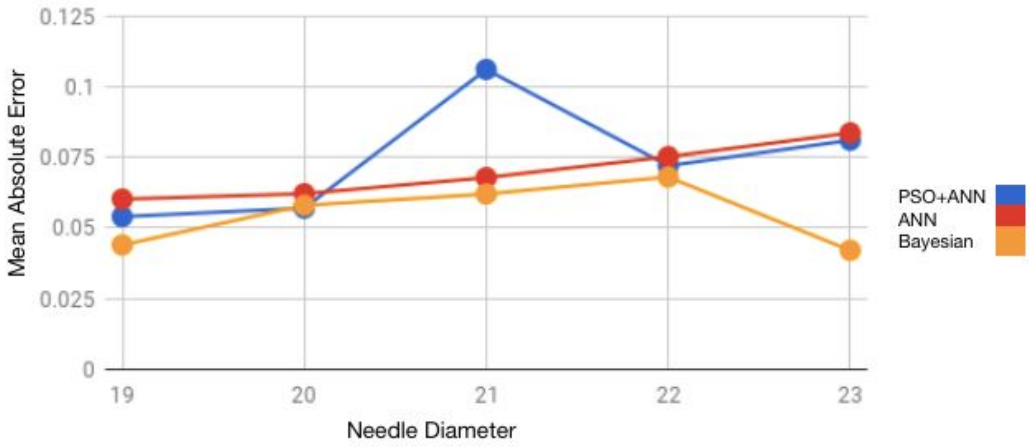

MEAN SQUARE ERROR FOR ANN




MEAN SQUARE ERROR FOR ANN WITH PSO




MEAN SQUARE ERROR FOR BAYESIAN MODEL




COMBINED MEAN SQUARE ERROR GRAPH

The combined graph is as shown above. We can see that all methods give approximately same values. The simple ANN model shows a gradual increase in Mean Squared Error as the needle diameter increases. The Particle Swarm Optimization performed on it decreases its error for all diameters but one. The Bayesian curve shows the least error whatsoever be the needle diameter, since the problem statement is particularly favourable to be modelled using this method.

# Conclusion

The textile manufacturing process is characterised by the high consumption of resources like water, fuel and a variety of chemicals in a long process sequence that generates a significant amount of waste. Our project predicts the amount of textile wastage beforehand which leads to a cleaner production. Hence it is an attractive approach to tackle environmental problems associated with industrial production and poor material efficiency. It leads to significant financial saving and environmental improvements. It also improves the company's public image by highlighting the initiative it has taken to protect the environment. Here the concept of Neural Networks and Machine Learning is applied to the Textile Sector to solve an Environmental problem. Attempts can be made to further optimize the created models.

# References

## Feature Selection papers

[1]A. Chaudhary, S. Kolhe, Rajkamal ,(2013)" Performance Evaluation of feature selection method for Mobile devices" Int. Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 3, Issue 6, pp.587-594
[2] A.M. Kozae1 , A.A. Abo Khadra, T. Medhat (2007) , Reduction Of Multi-Valued Information Systems, International Journal of Pure and Applied Mathematics Volume 39 No. 1 , 91-99
[3] Anil Rajput, Ramesh Prasad Aharwal (2011) "J48 and JRIP Rules for E-Governance Data "International Journal of Computer Science and Security (IJCSS), Volume (5): Issue (2).
[4] Anshul Goyal and Rajni Mehta,(2012),"Performance Comparison of Naïve Bayes and J48 Classification Algorithms" , International Journal of Applied Engineering Research, ISSN 0973-4562 Vol.7 No.11.
[5] Antonio Arauzo-Azofra,, José Luis Aznarte, Jose M. Benitez, (2011)," Empirical study of feature selection methods based on individual feature evaluation for classification problems", Expert Systems with Applications 38 ,8170–8177.
[6] Anuj Sharma, Shubhamoy Dey (2012)," Performance Investigation of Feature Selection Methods and Sentiment Lexicons for Sentiment Analysis" Special Issue of International Journal of Computer Applications (0975 – 8887) on Advanced Computing and Communication Technologies for HPC Applications – ACCTHPCA.

[7] Aparna Choudhary, Jai Kumar Saraswat (2014),"Survey on Hybrid Approach for Feature Selection", International Journal of Science and Research (IJSR),ISSN: 2319-7064 , Volume 3 Issue 4, 2014.
[8] Artur J. Ferreira, Mário A.T. Figueiredo, (2012) "Efficient feature selection Filters for high dimensional data" ,Pattern Recognition Letters 33, 1794–1804.
[9] Been-Chian Chien , Chih-Hung Hu and Steen-J Hsu (2004)," Generating Hierarchical Fuzzy Concepts from Large Databases" 0-7803-8566-7/04, IEEE.
[10] Benoit Frenay, Gauthier Doquire, Michel Verleysen,(2013)," Theoretical and empirical study on the potential inadequacy of mutual information for feature selection in classification" ,Neurocomputing 112 , 64–78.
[11] Changyou, G. U. O (2011). "An Efficient Feature Selection Algorithm Based on Information Entropy in Decision Table." Journal of Information & Computational Science 8: 14 2941–2947.
[12] D. L. Gupta,A. K. Malviya ,Satyendra Singh (2012)," Performance Analysis of Classification Tree Learning Algorithms ",International Journal of Computer Applications (0975 – 8887) Volume 55– No.6.
[13] DildarKhan T. Pathan, Pushkar D. Joshi, S. U. Balvir ,(2014) ," Prediction of soil quality for agriculture", IRJSSE / Volume :2 / Issue: 3 , ISSN: 2347-6176

# ANN Papers

1. Zhang, Guoqiang, B. Eddy Patuwo, and Michael Y. Hu. "Forecasting with artificial neural networks:: The state of the art." *International journal of forecasting* 14.1 (1998): 35-62.
2. Specht, Donald F. "A general regression neural network." *IEEE transactions on neural networks* 2.6 (1991): 568-576.
3. Mitchell, Tom M. "Artificial neural networks." *Machine learning* 45 (1997): 81-127.
4. Han Lock Siew, Md Jan Nordin, "Regression Techniques for the Prediction of Stock Price Trend", Statistics in Science, Business, and Engineering (ICSSBE), 2012 International Conference.
5. Hongran Li and Shigeru Yamamoto, "A Model-Free Predictive Control Method Based on Polynomial Regression", Control Systems (ISCS), 2016 SICE International Symposium.
6. N. J. Park, George and N. Park, "A Multiple Regression Model for Trend Change Prediction", Financial Theory and Engineering (ICFTE), 2010 International Conference.
7. Pelin Yıldırım, Derya Birant, Tuba Alpyıldız, "Improving prediction performance using ensemble neural networks in textile sector" Computer Science and Engineering (UBMK), 2017 International Conference.
8. Nishchol Mishra, Dr.Sanjay Silakari, "Predictive Analytics: A Survey, Trends, Applications, Opportunities & Challenges", International Journal of Computer Science and Information Technologies, Vol. 3 (3) , 2012, 4434- 4438
9. Pelin Yildirim, Derya Birant, Tuba Alpyildiz, "Data mining and machine learning in textile industry", Wiley Online Library".
10. Tung Le, Ming Luo, Junhong Zhou, Hian L. Chan, "Predictive maintenance decision using statistical linear regression and kernel methods", Emerging Technology and Factory Automation (ETFA), 2014 IEEE.

# PSO Papers

1. J. Kennedy and R. Eberhart, "Particle Swarm Optimization", in: *Proc. IEEE Intl. Conf. on Neural Networks (Perth, Australia)*, IEEE Service Center,Piscataway, NJ, IV:1942-1948, 1995.

2. Venu G. Gudise and Ganesh K. Venayagamoorthy. “Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks”
3. C. Zhang, H. Shao. "An ANN’s Evolved by a New Evolutionary System and Its Application" in *In Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 4, pp. 3562-3563, 2000.
4. Beatriz A. Garro and Roberto A. Vázquez, “Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms,” Computational Intelligence and Neuroscience, vol. 2015, Article ID 369298, 20 pages, 2015. doi:10.1155/2015/369298
5. Marcio Carvalho, Teresa B. Ludermir. “Particle Swarm Optimization of Neural Network Architectures and Weights”, Seventh International Conference on Hybrid Intelligent Systems
6. M. Carvalho, T.B. Ludermir, "Particle Swarm Optimization of Feed-Forward Neural Networks with Weight Decay," his, p. 5, in *Sixth International Conference on Hybrid Intelligent Systems* (HIS’06), 2006.
7. RANE, Vishal A.. Particle Swarm Optimization (PSO) Algorithm: Parameters Effect And Analysis.International Journal of Innovative Research and Development, [S.l.], jul. 2013. ISSN 2278-0211.
8. Beatriz A. Garro, Humberto Sossa and Roberto A. Vázquez: Back-Propagation vs Particle Swarm Optimization Algorithm: which Algorithm is better to adjust the Synaptic Weights of a Feed-Forward ANN?**,** International Journal of Artificial Intelligence, ISSN 0974-0635; Int. J. Artif. Intell. Autumn (October) 2011, Volume 7, Number A11
9. Marco Dorigo et al. (2008) Particle swarm optimization. Scholarpedia, 3(11):1486.

# Bayesian Papers

1. Jen-Tzung Chien, Member, IEEE, and Guo-Hong Liao, “Transformation-Based Bayesian Predictive Classification Using Online Prior Evolution”
2. Helge Langseth, Thomas D. Nielsen, “Classification using Hierarchical Naive Bayes models”
3. Juho Piironen, Aki Vehtar - “Comparison of Bayesian predictive methods for model selection”
4. Tiago M. Fragoso and Francisco Louzada Neto - “Bayesian model averaging: A systematic review and conceptual classification”, 29 Sep 2015
5. Tomohiro Ando - “Predictive Bayesian Model Selection”
6. Roxana A. Ion, Hans Kalishoek, Dimitrios M. Karydas, Eindhoven University of Technology, Iulian Nitescu, Philips TASS BV, Embedded Systems Institute- “Bayesian model for early reliability prediction”
7. John Geweke and Gianni Amisano- “Comparing and evaluating bayesian predictive distributions of asset returns”
8. Dalene K. Stangl - Prediction and decision making using Bayesian hierarchical model