CHAPTER II

BACKGROUND, METHODLOGY AND MODEL DESIGN

2.1 Industry Background and Data Selection

2.1.1 Industry Background

Paper is a material, a network, built up from individual fibers, fillers, additives and other components. The properties of the paper depend not only on the properties of these components but also very much on the interactions between them. Paper must, however, maintain its competitiveness through continuous product development in order to meet the ever-increasing demands on its performance. It must also be produced economically by environment-friendly processes with the minimum use of resources. The methods used for testing pulp and paper products are therefore regarded as essential parts of the papermaking science and technology series.

Basically, the properties that can be used to describe the character of paper quality include sufficient strength, suitable structure, correct optical properties, suitable surface properties and sufficient stiffness. Typical functional requirements for paper may relate to its running ability in printing and copying machines, its printability in different processes, and other aspects of end-use behaviors.

Paper making itself is a tremendously complicated process. The factors that will affect the final paper properties, especially affect the final paper curl indexes will be more complex. Paper curl is primarily due to strain variations in a paper structure. The origin of the curl can be different fiber orientation or fiber bonding on different sides of the paper. Fiber swelling and shrinkage due to moisture variations can also cause curl. In some cases, temperature variations are also a cause of curl. During sheet fed offset printing (one-sided printing) and ink-jet printing, water is applied to one side of the sheet. This also can cause paper curl. In practice, sheet fed offset printing occurs at constant temperature and relative humidity. Paper humidity after production must be set possibly the same as the humidity in printing situation to avoid severe curling problems in a printing operation. Therefore, so many factors are included in the whole process. There are no standard inputs choices for forecasting. However, selecting inputs for forecasting target of paper curl is a not easy work, it is the vital issue for the whole

research. In the real working environment, most parameters in the paper making process will more or less affect the final paper properties, so the relations among them are delicate. The methodology of multivariate process performance monitoring is demonstrated by application to the manufacture of gloss paper. The process is summarized in the Figure 2.1. The stock supply is made up of pulp, water and chemical additives. Water removal is carried out in three ways, by suction (the former), by applying pressure (the pressure rolls) and by heat (the drier rolls). The base paper obtained in this way is subsequently coated on both sides before drying. Ideally, there are six input parts that would affect the target. They are stock, paper machine, wet end (forming and press parts), size press, drying part and chemical. On the other hand, the paper properties basically can be separated into three categories: basic properties, optical properties and strength properties. Normally, the basic properties also will linearly affect other properties in faintly low level.

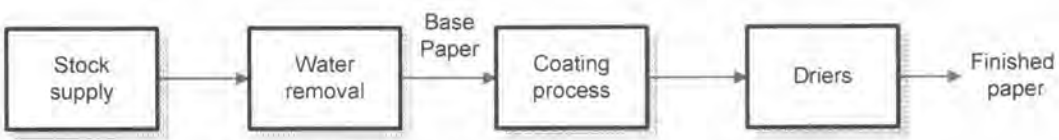

Figure 2.1: Stages involved in gloss paper production.

2.1.2 Data Selection

In paper making factory, the workflow can be controlled and monitored in real-time by production database system. Most parameters in paper machine can be checked and loaded from the system. Meanwhile, engineers and operators can easily observe the movements of each index parameters in ad hoc. Therefore, database supports to offer huge numbers of source data for our paper curl forecasting research. The sample data source interface is shown in the Figure 2.2.

Edwards et al [1] chose 10 parameters for reasons of availability and possibility through experimentation. Four of them were from the stock part or paper machine part. Other six actually come from the final paper properties, but most of them can be tested on line. In the research offered by Borolin [2], they selected 15 related inputs, four from final paper properties, the other from production process, most related to stock, calendar and paper machine part.

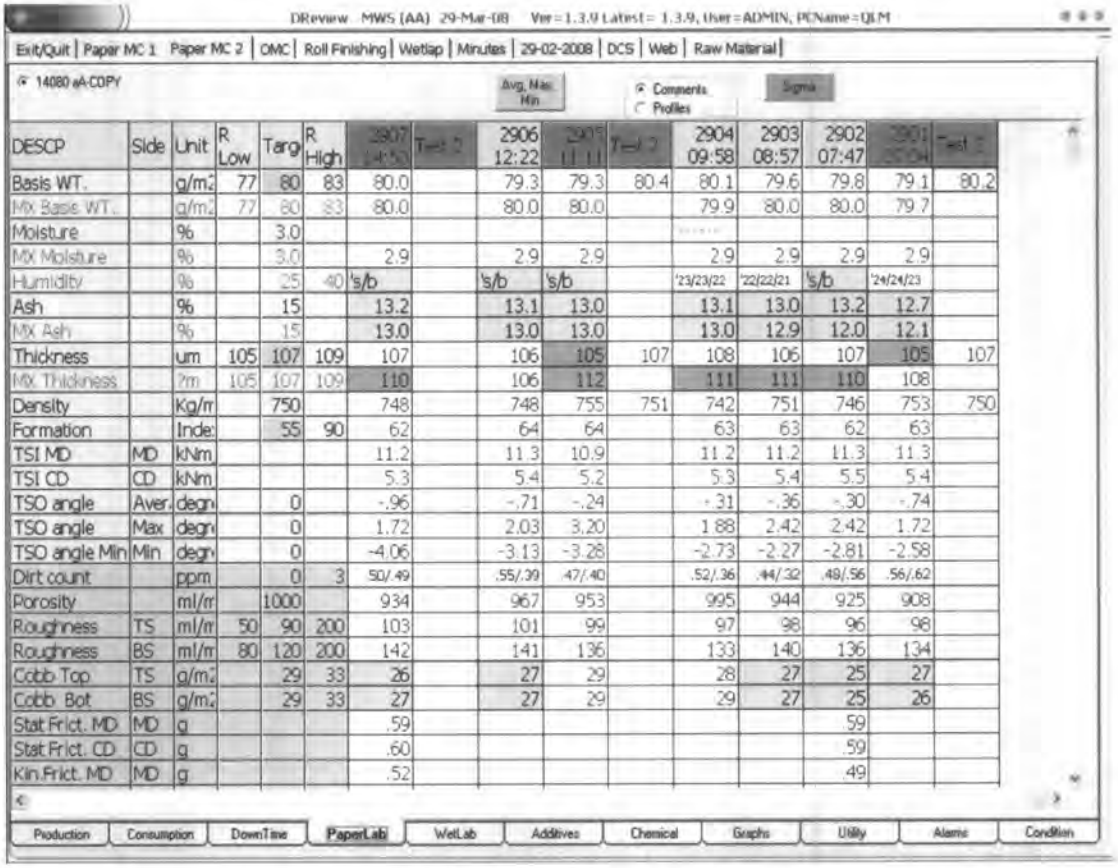

Figure 2.2: Interface of sample source data.

Approximately, one roll of jumbo reel will be produced in every one hour [2], and each jumbo reel should have its own curl indexes. Therefore, the interval among each record of paper properties is about one hour in the factory production database. The paper properties are assumed to be affected by the movements of vast indexes parameters in the last production hour.

According to the real-time working mechanism, we loaded 61 different kinds of index parameters covering all six parts of paper making process mentioned above, and the number of input data variables is 65 after adding 4 basic paper properties parameters (Basis WT, Ash, Moisture and Thickness). The details of the relation between the paper making factory database and the research source data set are shown in the following Figure 2.3.

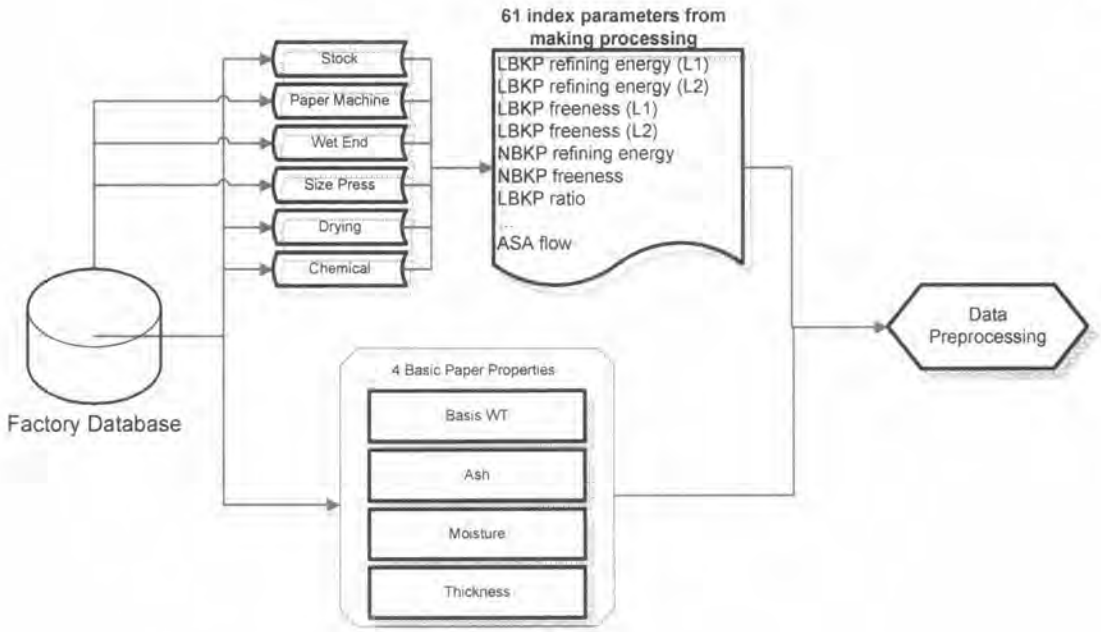

Figure 2.3: Source data selection flow.

The input details shows as following lists:

1) LBKP refining energy (L1): the consuming energy of refining the short fiber which be used for the assembly line L1.
2) LBKP refining energy (L2): the consuming energy of refining the short fiber which be used for the assembly line L2.
3) LBKP freeness (L1): after refining, the freeness of short fiber, for assembly line L1. The more refining energy, the less freeness will be.
4) LBKP freeness (L2): after refining, the freeness of short fiber, for assembly line L2. The more refining energy, the less freeness will be.
5) NBKP refining energy: the consuming energy of refining the long fiber.
6) NBKP freeness: after refining, the freeness of long fiber.
7) LBKP ratio: the ratio of short fiber in the fiber mixture.
8) NBKP ratio: the ratio of long fiber in the fiber mixture.
9) Stock to PM flow: the flow speed of stock to paper machine.
10) Fan pump 1 rpm: the rotating speed of No.1 fan pump.
11) Wire silo temperature: the temperature of wire silo.
12) Fan pump 2 rpm: the rotating speed of No.2 fan pump.

13) Attenuator pressure difference: be used to make the pressure and flow more stable.

14) Head box pressure: the pressure of head box.

15) Taper header diff. pressure: the pressure difference of TS (the front of the PM) and DS (the back of the PM).

16) Head box re-circle flow: be used to control the taper header pressure.

17) Jet wire ratio: The ratio of the speed of pulp flow and fabrics running speed.

18) Head box change length: the front-back width of the opening of head box slice.

19) Head box consistency: the pulp consistency in the head box.

20) Total retention: the ratio of fiber and material retention on the fabrics.

21) Filler retention: the ratio of filler retention on the fabrics.

22) Wire vacuum box vacuum: one of the five vacuums of the forming sections.

23) Top shoe unit vacuum: one of the five vacuums of the forming sections.

24) Middle suction box vacuum: one of the five vacuums of the forming sections.

25) Trans suction box vacuum 1: one of the five vacuums of the forming sections.

26) Trans suction box vacuum 2: one of the five vacuums of the forming sections.

27) Suction box vacuum: one of the five vacuums of the forming sections.

28) Couch roll low vacuum: vacuum of couch roll.

29) Couch roll high vacuum: vacuum of couch roll.

30) 1^{st} press nip: the nip pressure of the 1^{st} press stage.

31) Pick up press nip: the nip pressure of the 2^{nd} press stage.

32) 3^{rd} press nip: the nip pressure of the 3^{rd} press stage.

33) 4^{th} press nip: the nip pressure of the 4^{th} press stage.

34) LP steam pressure: LP low steam pressure.

35) The 4^{th} drying section pressure: the pressure of the 4^{th} drying section.

36) The 10^{th} drying section pressure: the pressure of the 10^{th} drying section.

37) Nip TS: nip pressure of TS size press part.

38) Nip DS: nip pressure of DS size press part.

39) Chamber pressure top: the pressure of top chamber.

40) Chamber pressure bottom: the pressure of bottom chamber.

41) Jet size to S/P top flow: the top flow for chemistry jet size.

42) Jet size to S/P bottom flow: the bottom flow for chemistry jet size.

43) Starch to S/P top flow: starch to the top surface sizing.

44) Starch to S/P bottom flow: starch to the bottom surface sizing.

45) PM speed: the speed of paper machine.

46) OBA to mixing chest flow: affects whiteness.

47) Retention aid to F/P 2 flow: affects operation process, increase retention.

48) BMA flow: increase retention.

49) Cationic starch to mixing chest flow: internal sizing, starch.

50) GCC flow: filler.

51) PCC flow: filler.

52) Dye flow: dying.

53) Perform flow: increase retention.

54) Edge flow TS: the valve open degree of TS (operation side).

55) Edge flow DS: the valve open degree of DS (driving side).

56) LP steam temperature: steam temperature.

57) Nip load: Nip pressure between calendar rolls.

58) Head box temperature: the temperature of pulp in head box.

59) 1st cationic starch flow: internal sizing, starch.

60) 2nd cationic starch flow: internal sizing, starch.

61) ASA flow: internal sizing, starch.

62) Basis WT: basic prosperity.

63) Ash: basic prosperity.

64) Moisture: basic prosperity.

65) Thickness: basic prosperity.

2.2 MLP Neural Networks

2.2.1 Introduction to Neural Networks

Neural network is not a new technology; it has been around the world since 1943. McCulloch and Pitts gave birth to the field of artificial neural networks. What is a neural network? First of all, when we are talking about a neural network, we should

more properly say "artificial neural network" (ANN), because that is what we mean most of the time. Biological neural networks are much more complicated than the mathematical models that we use for ANNs. But it is customary to be lazy and drop the "A" or the "artificial".

There is no universally accepted definition of an NN. But perhaps most people in the field would agree that an NN is a network of many simple processors ("units"), each possibly having a small amount of local memory. The units are connected by communication channels ("connections") which usually carry numeric (as opposed to symbolic) data, encoded by any of various means. The units operate only on their local data and on the inputs they receive via the connections. The restriction to local operation is often relaxed during training.

Some NNs are models of biological neural networks and some are not, but historically, much of the inspiration for the field of NNs came from the desire to produce artificial systems capable of sophisticated, perhaps "intelligent", computations similar to those that the human brain routinely performs, and thereby possibly to enhance our understanding of the human brain.

A neural network is first and foremost a graph, with patterns represented in terms of numerical values attached to the nodes of the graph, and transformations between patterns achieved via simple message-passing algorithms. Many neural network architectures, however, are also statistical processors, characterized by making particular probabilistic assumptions about data [6]. This conjunction of graphical algorithms and probability theory is not unique to neural networks, but characterizes a wider family of probabilistic systems in the form of chains, trees, and networks that are currently studied throughout AI.

Neural networks have found a wide range of applications, the majority of which are associated with problems in pattern recognition and control theory. In this context, it is best to view neural networks as a class of algorithms for statistical modeling and prediction. Based on a source of training data, the aim is to produce a statistical model of the process from which the data are generated, so as to allow the best predictions to be made for new data.

A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for usage [7]. It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process.

2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

Most neural networks involve combination [8], activation, error, and objective functions.

Combination functions: Each non-input unit in a neural network combines values that are fed into it via synaptic connections from other units, producing a single value called the "net input". For the function that combines values, it is called the "combination function". The combination function is a vector-to-scalar function. Most Neural networks use either a linear combination function (as in MLPs) or a Euclidean distance combination function (as in RBF networks).

Activation functions: Most units in neural networks transform their net input by using a scalar-to-scalar function called an "activation function", yielding a value called the unit's "activation". Except possibly for output units, the activation value is fed via synaptic connections to one or more other units. The activation function is sometimes called a "transfer", and activation functions with a bounded range are often called "squashing" functions, such as the commonly used tanh (hyperbolic tangent) and logistic (1/1+exp (-x)) functions. If a unit does not transform its net input, it is said to have an "identity" or "linear" activation functions.

Error functions: Most methods for training supervised networks require a measure of the discrepancy between the networks output value and the target value. The difference between the target and output values is called the "residual" or "error". This is NOT the "error function"! The residual can be either positive or negative, and negative residuals with large absolute values are typically considered just as bad as large positive residuals. Error functions, on the other hand, are defined so that the bigger is the worse.

Objective functions: The objective function is what you directly try to minimize during training. Neural network training is often performed by trying to minimize the total error or the average error for the training set. However, minimizing training error can lead to over-fitting and poor generalization if the number of training cases is small relative to the complexity of the network. A common approach to improving generalization error is regularization function. If no regularization function is used, the objective function is equal to the total or average error function.

Neural networks offer a computational approach that is quite different from conventional digital computation. Digital computers operate sequentially and can do arithmetic computation extremely fast. Biological neurons in the human brain are extremely slow devices and are capable of performing a tremendous amount of computation tasks necessary to do everyday complex tasks, commonsense reasoning, and dealing with fuzzy situations. The underlining reason is that, unlike a conventional computer, the brain contains a huge number of neurons, information processing elements of the biological nervous system, acting in parallel. Neural networks are thus a parallel, distributed information processing structure consisting of processing elements interconnected via unidirectional signal channels called connection weights.

2.2.2 MLP Neural Network architecture

In the architecture of neural networks, typically, the network consists of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. These neural networks are commonly referred to as multilayer perceptrons (MLPs). Many applications based on MLP neural network have successfully solved the problems in related fields.

A multiplayer perceptron has three distinctive characteristics:

1. The model of each neuron in the network includes a nonlinear activation function. The important point to emphasize here is that the nonlinearity is smooth (i.e., differentiable everywhere), as opposed to the hard-limiting used in

Rosenblatt's perceptron [9]. A commonly used form of nonlinearity that satisfies this requirement is a sigmoid nonlinearity defined by the logistic function:

$$y_j = \frac{1}{1+\exp(-v_j)}$$

where v_j is the induced local field (i.e., the weighted sum of all synaptic inputs plus the bias) of neuron j, and y_j is the output of the neuron. The presence of nonlinearities is important because otherwise the input-output relation of the network could be reduced to that of a single-layer perceptron. Moreover, the used of the logistic function is biologically motivated, since it attempts to account for the refractory phase of real neurons.

2. The network contains one or more layers of hidden neurons that are not part of the input or output of the network. These hidden neurons enable the network to learn complex tasks by extracting progressively more meaningful features from the input patterns (vectors).

3. The network exhibits high degrees of connectivity, determined by the synapses of the network. A change in the connectivity of the network requires a change in the population of synaptic connections or their weights.

Figure 2.4 displays the network diagram that represents the corresponding statistical model of a MLP multiple layer perceptron architecture consisting of one input layer with three inputs, neutrons or units (X_1, X_2, X_3) with three input weights (W_1, W_2, W_3) going into a single hidden layer with one hidden unit and an activation function that is connected to a single output unit.

MLPs are general-purpose, flexible, nonlinear models that, given enough hidden neurons and enough data, can approximate virtually any function to any desired degree of accuracy [10]. In other words, MLPs are universal approximators. MLPs can be used when you have little knowledge about the form of the relationship between the independent and dependent variables.

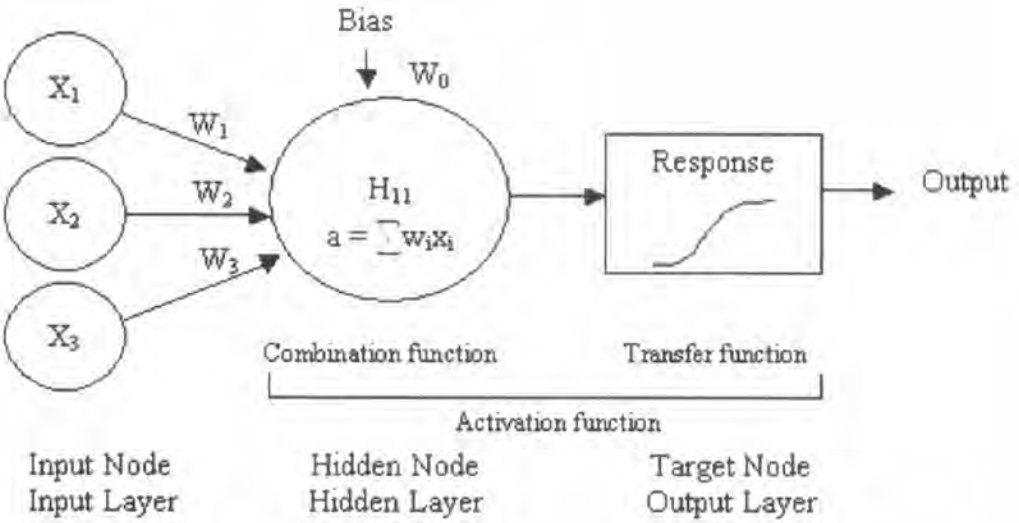

Figure 2.4: A neural network MLP architecture.

2.3 Learning Algorithms

Considered about the size of the inputs data in this research, two training algorithms from Newton-Based methods suitable for middle-sized data are chosen to optimize the learning process. The Newton-based methods are derived from using the first-order or second-order Taylor series expansion. The Hessian matrix contains the second derivatives of the error function with respect to the weight estimates. Newton methods is one of the most powerful and well-known convergence methods for solving root-finding problems i.e. f(x) = 0. It is regarded as one of the fastest and most reliable convergence algorithms for unconstrained optimization of polynomial error functions given the fact that the initial parameter estimates are a good approximation.

2.3.1 Quasi-Newton Algorithm

The Quasi-Newton method is a variant that approximates the Hessian matrix with a positive definite matrix using only first derivatives [7]. The basic equation using in Quasi-Newton is as follows:

$$\delta^{(n)} = -\eta^{(n)} \cdot (H^{(n)})^{-1} \cdot g^{(n)}$$

where $g^{(n)}$ is a vector of first derivatives of the error function with respect to the weight estimates at the n^{th} step or iteration. $H^{(n)}$ is the Hessian matrix of second derivatives of the error function with respect to the weight estimates at the n^{th} step. That is, the Hessian matrix plays the same role as the design matrix in ordinary least-squares

regression modeling. $(H^{(n)})^{-1}$ is an approximation to the Hessian matrix in the iteration. The algorithm firstly initializes $x^{(0)}$ and any real positive definite Hessian matrix H_0. If $g^{(k)} = 0$ then the algorithm will stop, otherwise, it will compute:

$$d^{(k)} = -(H_k^{-1}) \cdot g^{(k)}$$

$$\alpha_k = \arg\min f(x^{(k)} + \alpha_k d^{(k)}) \text{ for } \alpha > 0$$

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$$

$$\Delta x^{(k)} = \alpha_k d^{(k)}$$

$$g^{(k+1)} = Q \cdot x^{(k+1)}$$

$$\Delta g^{(k)} = g^{(k+1)} - g^{(k)}$$

$$H_{k+1} = H_k + \frac{(\Delta x^{(k)} - H_k \Delta g^{(k)})(\Delta x^{(k)} - H_k \Delta g^{(k)})'}{\Delta g^{(k)'}(\Delta x^{(k)} - H_k \Delta g^{(k)})}$$

After computation, the steps should be iterated until $g^{(k)} = 0$. Quasi-Newton technique avoids the computation of the Hessian matrix in iterations and calculates an approximation to the matrix of second derivatives based on the updates gradient [11]. The approximation is updated at each iteration similar to the Hessian matrix re-evaluated based on the Newton's method. Therefore, the Quasi-Newton method builds up the approximation in each step.

2.3.2 Double Dogleg Algorithm

The Double Dogleg method combines both the Quasi-Newton and the trust region methods. For the part of Quasi-Newton, we've already mentioned it. For the part of trust region, it is a small hyper-elliptic region or radius around the current search point. The trust region method takes a different approach in determining the next iteration step with comparison to the other minimization techniques. The step length is similar to the Newton direction when the value of identity matrix of scaling coefficients equals to 0. When the value gets large, the step length is similar to steepest descent direction.

The equation of double dogleg method is shown as follows:

$$\delta^{(n)} = \alpha_1 \cdot s_1^{(n)} + \alpha_2 \cdot s_2^{(n)}$$

The method searches along the dogleg trajectory at each iteration between the steepest descent direction $s_1^{(n)}$ and the Quasi-Newton direction $s_2^{(n)}$. Momentum constants (α_1, α_2) are applied to each direction to accelerate convergence.

Because of its calculation in small but efficient steps in the iterative process with slow convergence, Double Dogleg algorithm is a suitable method matched with early stopping regularization, which is meaningful to avoid bad local minimums.

2.4 Dimension Reducing Methods

In predictive modeling, there are two reasons for eliminating variables from the analysis: redundancy and irrelevancy. In other words, most of the modeling selection routines are designed to minimize input redundancy and maximize input relevancy. At times, the statistical model can potentially consist of an enormous number of input variables in the model to predict the target variable. Therefore, irrelevancy in some of the input variables might not provide a sufficient amount of information in describing or predicting the target variable. Redundancy in the input variables suggests that a particular input variable does not provide any added information in explaining the variability in the target variable that has not already been explained by some other input variables already in the model.

2.4.1 Principal Components Analysis

The purpose of principal components analysis is both data reduction and interpretation of a linear combination of the input variables in the data that best explains the covariance or correlation structure. The analysis is designed to reduce the dimensionality of the data while at the same time preserving the structure of the data. The advantage is that a smaller number of linear independent variables, or principal components, without losing too much variability in the original data source, where each principal component is a linear combination of the input variables in the model [12].

Principal components analysis is based on constructing an independent linear combination of input variables in which the coefficients (eigenvectors) capture the maximum amount of variability in the data. Typically, the analysis creates as many principal components as there are input variables in the data set in order to explain all the variability in the data where each principal component is uncorrelated to each other. This solves one of two problems in the statistical model. Firstly, the reduction in the number of input variables solves the dimensionality problem. Secondly, this will solve co-linearity among the input variables since the components are uncorrelated to each

other. The goal of the analysis is first finding the best linear combination of input variables with the largest variance in the data, called the first principle component. The basic idea is to determine the smallest number of the principal components to account for the component consists of a line that is perpendicular to each data point by minimizing the total squared distance from each point that is perpendicular to the line. This is analogous to linear regression modeling, which determines a line that minimizes the sum-of-squares vertical distance from the data points that is always perpendicular to the axis of the target variable [13]. Principal components analysis is designed so that the first principal component is perpendicular, orthogonal, and uncorrelated to the second principal component, with the second principal component following the first principal component in explaining the most variability in the data. The number of principal components to select is an arbitrary decision. This can be achieved by observing the magnitude of the eigenvectors within each principal component.

The principal components are comprised of both eigenvalues and eigenvectors that are computed from the variance-covariance matrix. The eigenvalues are the diagonal entries in the variance-covariance matrix. The principal components are the linear combination of the input variables with coefficients equal to the eigenvectors of the corrected variance-covariance matrix.

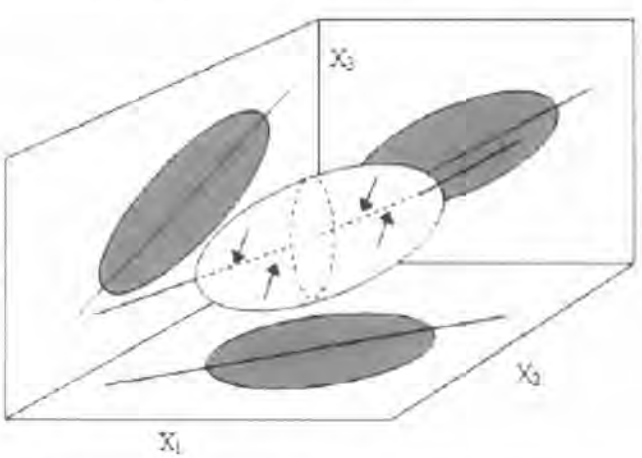

Figure 2.5: Geometric interpretation of the first principal component.

Principal components analysis is basically designed to determine the minimum distance between all the data points in the model. This is illustrated in the figure 2.5. In the figure 2.5, the line accounts for the largest majority of variability in the

data where the principal components line is not parallel to any of the three variable axes. The elliptical-shape sphere represents the distribution of the data points in a three-dimensional plane and the shadows display the two-dimensional scatter plots, where there is a positive correlation between all pairs of variables in the data.

2.4.2 Step-wise Regression

The R-square statistic is based on the linear relationship between each input variable in the predictive model and the target variable to predict. The R-square statistic is calculated in determining how well the input variables in the predictive model explain the variability in the target variable. Stepwise method is used in selecting the best linear combination of input variables to the model with the modeling selection procedure terminating when the improvement in the R-square is less than 0.0005.

R^2 is a statistic that will give some information about the goodness of fit of a model. In regression, the R^2 coefficient of determination is a statistical measure of how well the regression line approximates the real data points. An R^2 of 1.0 indicates that the regression line perfectly fits the data. The explanation of the value of R^2 is shown as follows.

A data set has values y_i each of which has an associated modeled value f_i. Here, the values y_i are called the observed values and the modeled values f_i are sometimes called the predicted values. The "variability" of the data set is measured through different sums of squares:

$SS_{tot} = \sum_i (y_i - \bar{y})^2$, the total sum of squares (proportional to the sample variance);

$SS_{reg} = \sum_i (f_i - \bar{f})^2$, the regression sum of squares, also called the explained sum of squares;

$SS_{err} = \sum_i (y_i - f_i)^2$, the sum of squared errors, also called the residual sum of squares.

$\bar{y}$ and $\bar{f}$ are the means of the observed data and modeled (predicted) values respectively. And the most general definition of the coefficient of determination is $R^2 = 1 - \frac{SS_{err}}{SS_{tot}}$.

Stepwise selection begins like forward selection with no centers in the network. At each step, a center is added or removed. If there are any centers in the network, the one that contributes least to reducing the objective function is subjected to a statistical test (usually based on the F statistic) to see if it is worth retaining in the network; if the center fails the test, it is removed. If no centers are removed, then the centers that are not currently in the network are examined; the one that would contribute most to reducing the objective function is subjected to a statistical test to see if it is worth adding to the network; if the center passes the test, it is added. When all centers in the network pass the test for staying in the network, and all other centers fails the test for being added to the network, the stepwise method terminates.

Process will be performed in the stepwise regression method from the R^2 variable selection routine where the input variables are added or removed from the predictive model as follows:

Firstly, correlation analysis will be performed between each input variable and the target variable. All input variables are retained to the model with a squared correlation r-square statistic greater than the value of 0.005.

Secondly, forward stepwise R-square regression will be performed to the input variables that are not rejected from the previous step. The input variables that have the largest squared correlation coefficient value to the target variable are first entered into the regression model. Input variables with an improvement in the R^2 statistic less than the threshold criterion value 0.0005 will be removed from the model.

2.5 Forecasting Process Flow Design

Usually, the data in every special field are intricate with much noise and it is almost impossible to get a successful neural network modeling performance with initially setting parameters.

In this thesis study, we propose a compared MLP neural modeling based on the idea of choosing different data preprocessing and different training algorithms

and then, selecting the best solution to go on the forecasting process from the modeling evaluation. In data preprocessing part, we will separately use PCA (Principal Components Analysis) and Step-wise regression statistic to reduce the dimension of the inputs data. For neural network model, the algorithms of Quasi-Newton and Double Dogleg will be implemented concerned by the real training parameter counts.

In this study, a workflow is designed for the whole forecasting process. Firstly, the input data loaded from the factory production database in every ten minutes interval will be transposed into six columns for every hour to match the related paper properties and target paper curl. Then, this data set will be tagged as the first source data set. The second source data set which used for forecasting is just from the transaction of averaging every of the six transposed columns. After physical cleansing, the technique of PCA and the standard variable selection technique based on Step-wise regression will be used respectively. Two different MLP neural networks models will be trained individually with the two kinds of preprocessed data from the above steps. Finally, an accuracy evaluation would be made to help to choose the best model with the highest performance. The forecasting process flow design is given in the following Figure 2.6.

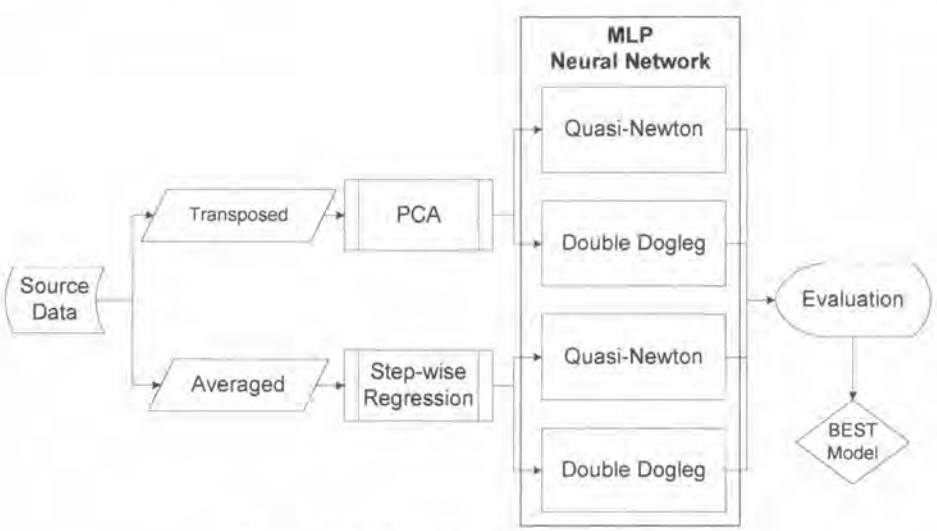

Figure 2.6: Forecasting process flow.