SOME DATA-DRIVEN METHODS IN PROCESS ANALYSIS AND CONTROL

Riku Mäkynen

PROCESS ENGINEERING
Bachelor's thesis
May 2018
SOME DATA-DRIVEN METHODS IN PROCESS ANALYSIS AND CONTROL

Riku Mäkynen

Advisor(s): Isokangas Ari, Paavola Marko

PROCESS ENGINEERING
Bachelor's thesis
May 2018
Abstract

Data-driven methods such as artificial neural networks have already been used in the past to solve many different problems such as medical diagnoses or self-driving cars and thus the material shown here can be of use in many different fields of science. A few studies that are related to data-driven methods in the field of process engineering will be explored in this thesis.

The most important finding related to neural network predictive controller was its better performance in the control of a heat exchanger when compared to several other controller types. The benefits of this approach were both energy savings and faster control. Another finding related to Evolutionary Neural Networks (EvoNNs) was the fact that it can be used to filter out the noise that is contained in the measurement data.
# TABLE OF CONTENTS

1 Introduction ........................................................................................................................................... 4

2 Methods .................................................................................................................................................. 6
   2.1 Artificial Neural networks .............................................................................................................. 6
      2.1.1 How to train a neural network ............................................................................................ 6
      2.1.2 Pareto Tradeoff ..................................................................................................................... 8
   2.2 Predictive control ............................................................................................................................ 9
      2.2.1 Introduction to Model Predictive Control ............................................................................. 9
      2.2.2 Neural Network Model Predictive Control (NNMPC) ......................................................... 10
   2.3 Noise in the data ............................................................................................................................ 11
   2.4 Predator–Prey algorithm ................................................................................................................. 11

3 Case studies ............................................................................................................................................ 13
   3.1 Case1: Heat exchanger .................................................................................................................. 13
   3.2 Case2: Iron Blast furnace .............................................................................................................. 20

4 Discussion ............................................................................................................................................. 25

5 Summary ................................................................................................................................................ 26

6 REFERENCES ......................................................................................................................................... 27
1 INTRODUCTION

Data-driven methods such as artificial neural networks have already been used in the past to solve many different problems such as medical diagnoses or self-driving cars and thus the material shown here can be of use in many different fields of science. Few studies that are related to data-driven methods in the field of process engineering will be explored in this thesis.

In this thesis the use of some data-driven methods in process engineering will be explored by studying two application examples. Comparisons between currently prevailing control & diagnosis schemes (such as General predictive control) and new schemes (such as evolutionary neural networks) will be shown when possible. This thesis is a study of existing literature on the field and tries to highlight the new possibilities for process diagnosis and control.

What are data-driven methods? Sometimes the scientific principles that the process consists of are either unknown or too trivial to put into any effective use. The data-driven process model tries to circumvent this problem. The idea behind a data-driven model is to capture the trends from the data itself. If an underlying physical trend in the process is significant for the process its presence should be seen in the data itself. Genetic programming and neural networks can be used to construct such a model. (Chakraborti 2014, pg. 89)

Why were process analysis and control chosen for focus of this thesis? The field of machine learning is quite massive but why choose these two? One of my main concerns when starting to write this thesis was: “What is the benefit of applying machine learning (data-driven methods) when the old methods seem to work just fine?“. This question that needs to be answered before there will be any motivation to either teach or apply data-driven methods in the industry. One of my goals was to give the reader a brief overview the possibilities that data-driven methods can offer in the diagnosis and control of process engineering systems. Special focus was given to finding articles that show side-by-side comparisons between data-driven methods and the more traditional
control strategies. Also articles that were explained well were favoured over those with plethora of mathematical equations and few explanations. Thus the papers that were chosen were easy to understand and/or could show clear benefits of using data-driven methods in the solution of process engineering problems. A few case studies that have these qualities will be focused on.

The most important finding related to neural network predictive controller was its better performance in the control of a heat exchanger when compared to several other controller types. The benefits of this approach were both energy savings and faster control. Another finding related to Evolutionary Neural Networks (EvoNNs) was the fact they it can be used to filter out the noise that is contained in the measurement data.

The format of the thesis will be the following. In chapter 2 general concepts are explained. Chapter 3 in general is about case studies. In chapter 3.1 a neural network controller is being used for controlling a heat exchanger. Comparisons between the NN controller and with a more traditional PID controllers will be shown. In chapter 3.2 there will be an example of the use of evolutionary neural network (EvoNN) in the monitoring of Silicon content of an iron blast furnace. The related study also revealed that an EvoNN is capable of filtering out noise. Chapter 4 is a discussion section where some views will be given about the case studies. Chapter 5 is a summary where the main findings of each case study will be discussed shortly.
2 METHODS

Next the utilized methods will be described. First Information about neural networks and how they relate to the case studies. Also some basics about predictive control will be included. Lastly there will information regarding process measurement and the inherent noise in the system.

2.1 Artificial Neural networks

Artificial neural networks are a computational networks in which there is a large collection of neural units which loosely model the way the human brain works. These can be used in solving problems that are harder to solve with more traditional if-else type computer programs. Each individual neural unit usually has a summation function that combines all the values from each input. There can also be a threshold function that must be surpassed before the specific neuron can propagate a signal forward. ("Artificial neural network", n.d.)

2.1.1 How to train a neural network

There are several ways to train a neural network. In the first case study regarding the heat exchanger the network was trained based on collected data. In this method the data that has been collected from the running system is split into two parts. The first part will be used to train the neural network. The second part will be used to validate the trained network. This way we can be sure that the network learned the underlying model and not just the training data. The weight values in the hidden layers will be adjusted in the network until the output values match the “correct” values ("Training an Artificial Neural Network - Intro", n.d.).

The input information from the input layer moves to the hidden layer and then proceeds to the output layer, see Figure 1. In the hidden layer each input receives a nonzero weight as well as a weighted bias term. Often a nonlinear function such as a sigmoid function can be used when transferring information from one layer to the next one.
Once leaving the hidden layer again each output node gets a weight. Another weighted bias term was also added to the output of the hidden layer. This bias affects all the output layer nodes. In order to set the correct weight values one needs to train the network with known inputs and outputs (aka training data). During this process one should attempt to minimize the error that results from the neural network by adjusting the weights. Back propagation was not used here (Chakraborti 2014 pg. 91). Instead a multi-objective genetic algorithm is being used to train the network (Chakraborti 2014 pg. 91). Often this type of training can lead to the data being overfitted. In such a case the network is able to mimic the training data set accurately but fails to capture the essence of the process which can be seen as a failure to produce correct results with test data set. (Chakraborti 2014, pg. 91-92)

Figure 1 Neural network (Modified image from Chakraborti 2014, pg. 91)
2.1.2 Pareto Tradeoff

What is Pareto efficiency? “Pareto efficiency, or Pareto optimality, is a state of allocation of resources in which it is impossible to make any one individual better off without making at least one individual worse off” (“Pareto efficiency”, n.d.). An example related to economics can be seen in the picture below in figure 2 where the red line is called the Pareto frontier. A manufacturer that produces two items can max out its production by selecting a point on the red line. A decision maker can then select the best point from among points A, B, C, D, E, F, G and H depending on which one would benefit the company the most. Similar tradeoffs often have to be made in engineering applications.

Figure 2 Pareto frontier (Modified image from: “Pareto efficiency”, n.d.)

The evolutionary neural network (EvoNN) (Pettersson, Chakraborti & Saxén 2005 pg. 396) architecture and network topology are flexible. What this essentially means is that the amount of nodes in the network is not set in stone. The optimal solutions in the second case study are form a Pareto frontier. From these points a decision maker can then select a suitable solution. More about this subject will be discussed in chapter 3.2.
2.2 Predictive control

2.2.1 Introduction to Model Predictive Control

The usage of predictive control is a widely used strategy in process engineering. The purpose of predictive control is to predict the future behavior of the process based on the known model of the process. This is method is called the model predictive control (MPC) or model-based predictive control (MBPC). (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011). Model predictive controllers are relying on dynamic process models. These models are linear most of the time. Model predictive controller can predict future events in the system and takes control actions based on that information.

At time k which is the current state of the process the future states of both output (e.g. water flow rate) and control actions (e.g. the utilization percentage of a pump) are calculated from time k+1 to k+p. Here p refers to prediction horizon. See figure 3. These calculations of both predicted control and predicted output are based on the current measurements as well as the future predictions. (“Overall Objectives of Model Predictive Control”, n.d.). For example k+2 could be calculated by using the control actions and predicted outputs in time instants k and k+1. Instant k+2 will be calculated with the help of k, k+1 and k+2. The calculations are made so that they minimize objective function J (Overall Objectives of Model Predictive Control n.d.). Now the predicted control inputs and process outputs have been calculated one uses only the first control input (k+1) and adjusts the controller accordingly. Only the first step of the calculated predictions is implemented (k+1). Then the plant state is sampled again and all calculations above are being repeated for the new current state. ("Model_predictive_control", n.d.)
2.2.2 Neural Network Model Predictive Control (NNMPC)

Most of the MPC strategies in use are based on linear models of a process even though most of the processes contain nonlinearities (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011). In order to surpass the difficulties related to the nonlinearity of a process the model reduction technology is being used to help with the application of linear MPC in an effective way (Xie and Theodoropoulos 2010). Neural networks have recently become a tempting tool in the construction of various nonlinear systems (Zare and Aminian 2010; Patra, Jehadeesan, Rajeswari and Satyamurthy 2010). Heat exchanger play a major role in these systems. (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011)

By combining a neural network (NN) with a MPC approach this model can then be used to predict the future outputs of the process. (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011). It is expected that a neural network model predictive control (NNMPC or NNPC) is a superior alternative when compared to the widely used PID controlled systems (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011).

Model predictive control (MPC) consists of a range of different control methods that all have certain things in common (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011).
The model of the process is being used in order to predict the future state of the process up to a fixed number of steps (N) to the future. To arrive at this estimation information up to time k as well as the future control actions are being used. Therefore the future trajectory of the process is known. The future control trajectory can be calculated by solving an optimization problem consisting of a cost function and potentially from some additional constraints. (Vasičekaninová, Bakošová, Mészáros, Klemeš, 2011, pg.2)

2.3 Noise in the data

Often the data from simulations, experiments or from a real world processes is noisy. Sometimes this noise can be easily detected if it appears systematically through the whole data, but at times it can be random and comes from several different sources. If a process model should be built from this type of process then one should ensure that the correct physical trends are being retained in the model while at the same time minimizing the error caused by the noise. Sometimes this process doesn’t work very well and one might end up with a model that captures every fluctuation that appears in the data (overfitting). The reverse could also be true where your model doesn’t capture noise but it also doesn’t capture the major trends in your system either (underfitting). To balance these two extremes one should be aiming to build a model that will ignore the noise but will capture the major trends in the process. To solve the problems above one could use Evolutionary neural networks (EvoNN) as it has already been used in this way previously to do this job (Pettersson, Chakraborti & Saxén 2005). Such networks have been tested on several industrial and scientific problems. (Chakraborti 2014, pg. 90)

2.4 Predator–Prey algorithm

In PPGA the predators and prey are being placed in a computational grid that aims to mimic a forest that has one faction of hunters and another one being hunted. The prey is this case would be all the possible solutions for a problem and the job of weeding out the bad solutions is given to the predator group. Both predator and prey will be allowed to wander in the forest in addition to some extra restrictions. Prey can breed, unlike predators who do not breed. The usual genetic algorithm operations such as mutation
and crossover (Chakraborti 2014 pg. 92) are only being applied to the prey population. The predators do not die nor do they reproduce. A Fonseca scheme (Fonseca 1995 pg. 64) is being used to rank each population that survives a predator attack (Chakraborti 2014 pg. 92). The rank of an individual is computed like so:

\[ R_i = 1 + \Theta_i \]

Here \( \Theta_i \) is the amount of individuals who dominate individual i in a weakly dominant way (Chakraborti 2014 pg. 92). Strongly dominant solution would mean that it’s better in fulfilling all the objectives at hand (Deb 2001, pg. 32). The individuals that arrive from this procedure having rank value one are the approximation of the Pareto frontier. (Chakraborti 2014, pg. 92-93)
3 CASE STUDIES

Next two case studies will be explored that are related to the use of neural network in process engineering.

3.1 Case1: Heat exchanger

A heat exchanger is a device which is used to transfer heat between one or more fluids. The fluids may be separated by a solid wall or the fluids may also be in direct contact. ("Heat exchanger", n.d.). Heat exchanger control is a complex process mainly due to its nonlinearity and complexity that can be caused by several different factors such as contact resistance, friction, leakage, temperature-dependent flow properties, unknown fluid properties, etc. (Vasiččkaninová, Bakošová, Mészáros, Klemeš, 2011). A draft of the heat exchanger can be seen in figure4.

![Figure 4 Tubular heat exchanger (Modified image from: Vasiččkaninová, Bakošová, Mészáros, Klemeš, 2011, pg.2)](image-url)
The controlled variable is the temperature of the heated stream at the end of the heat exchanger. The energy consumptions of NNPC control are then compared with a normal PID controller. The goal is to have smaller energy consumption when compared to PID control. (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011, pg.1)

In this study a neural network was used to estimate the future state of a tubular heat exchange system. The model of the neural network includes one hidden layer with 6 neurons and the network is trained off line. The predictive NNMPC model is used to calculate the proper inputs for the heat exchanger so that the correct temperature (at the end of the heat exchanger) for the heated stream is assured. The performance is evaluated based on energy consumption. In this study the energy consumption can be noticed in the amount of heating water consumed. The NNPC strategy is then compared with a normal PID controlled system. (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011, pg.2)

The system that is being controlled is a co-current tubular heat exchanger. In the system water is being used to heat petroleum through a copper tube. A draft of the heat exchanger can be seen in figure 4.

The controlled variable is the petroleum temperature at the end of the heat exchanger. The variable that controls the petroleum temperature is the water volume flow rate $q_{v3}(t)$, the rest of the inlet variables are being kept constant. The model assumes that the all fluids are moving in a plug velocity profile. Water temperatures $q_{1}(z,t)$, $q_{2}(z,t)$, $q_{1}(z,t)$ are a function of time $t$ and the axial $z$ coordinate. The flow rate of petroleum is constant while water flow is being changed in time. Water flow rate remains the same across the pipe but can vary as a function of time. The densities and heat capacities of petroleum, tube materials and water is expected to remain constant.

(Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011, pg.1-2)
The neural network model consists of two layers. A sigmoid transfer function is found in the hidden layer and a linear transfer function can be found in the output layer. To train the neural network a prediction error between model output and actual plant output was used. Figure 5 shows the structure of the neural network, $u(k)$ represents the input of the system, $y(k)$ is the output of the plant, $\hat{y}(k+1)$ is the output of the plant model. The previous values of the input signals are stored in the TDL (tapped delay lines). $IW^{i,j}$ represents the weight matrix of input j to layer i, $LW^{i,j}$ represents the weight matrix from input j to layer i. (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011, pg.3)

The neural network can be trained in batch mode offline by using data collected from previous plant operation. The Levenberg–Marquardt method was used in the training of the neural network. The LM algorithm can be used for solving non-linear least squares problems (“Levenberg–Marquardt algorithm”, n.d.).

![Figure 5 Neural network Structure, (Modified image from: Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011, pg.3)](image)

The following neural network parameters were used: 3 delayed inputs from the plant, 6 neurons in hidden layer and 2 delayed outputs from the plant. The previously mentioned Levenberg–Marquardt algorithm was used for training the network. The training data was obtained of the process that was controlled with a sampling interval of 2 seconds. The amount of training samples was six hundred. The training of the network was done
offline. Training results for the validation data can be seen in figure 6. (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011, pg.4)

Figure 6 shows the performance of the neural network against the validation data. By comparing the plant output graph with the neural network output it can be noticed that they match quite closely with each other. Because of this it is safe to say that the NN is not an overfitted one. Thus it can be said that the formulated neural network represents the process quite well and the neural network training was successful.
Figure 6 Neural network performance against the validation data. (Modified image from: Vasičckanová, Bakošová, Mészáros, Klemeš, 2011, pg.5)
The neural network controller was compared with two different PID controllers. The set point was changed at time 300s from 40°C to 39°C and later at time 600s to 40.5°C. The NNPC control has the least overshoot and is also settles the fastest. More precise results can be seen in the table 1 below where IAE (integrated absolute error) and ISE (integrated squared error) have been listed for all controllers. The NNPC method has the smallest errors in both categories IAE and ISE. (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011, pg.5)

The result can be seen in figure 7.

Figure 7 Outlet stream temperature when using different types of controllers. -V- Cohen-coon PID controller, -o- Strejc PID controller, -◊- NNPC. (Modified image from: Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011, pg.5)
Table 1 IAE and ISE values of different controllers. (Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011, pg.6)

<table>
<thead>
<tr>
<th>Method</th>
<th>IAE</th>
<th>ISE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNPC</td>
<td>159,9</td>
<td>279,4</td>
</tr>
<tr>
<td>PID Cohen-Coon</td>
<td>212,2</td>
<td>304,7</td>
</tr>
<tr>
<td>PID Strejc</td>
<td>233,0</td>
<td>297,4</td>
</tr>
</tbody>
</table>

Finally in figure 8 one can see the total hot water consumption of each control strategy. The NNPC method has the smallest hot water consumption out of the three.

Figure 8 Hot water consumption with each of the three controllers: -∇- Cohen-coon PID controller, -○- Strejc Pid controller, -◊- NNPC. (Modified image from: Vasičckaninová, Bakošová, Mészáros, Klemeš, 2011, pg.6)

The results of the simulation show that the NNPC control strategy was successful in the heat exchanger control. The benefits of NNPC versus a classical PID controller system...
was faster settling time with less oscillation and more energy savings. A notable advantage was also the fact that input constraints could be directly included in the model. When comparing the NNPC with classical PID control both tracking as well as energy savings were better with the NNPC model. Similar NNPC model can be applied to a real world process as well by using training data from a real process and training the neural network with that data. (Vasičkaninová, Bakošová, Mészáros, Klemeš, 2011, pg.6)

3.2 Case2: Iron Blast furnace

A blast furnace is a metallurgical furnace that is used for smelting in the production of industrial metals. In an iron blast furnace flux, ores and fuel are supplied from the top of the furnace Pre-heated air is being blown to the lower section of the furnace. (“Blast Furnace”, n.d.). Hot air escapes from the top while slag and iron are extracted from the bottom at regular intervals (Ricketts 2016). The Silicon content measurement data is noisy and the goal is to use an EvoNN to remove noise and thus reveal the true Silicon content.

In order to create a Pareto front as seen in figure 9 the neural network had to be trained several times. Once the training was complete with one network its network complexity and training error of the network were written down. Then this process had to be repeated several times each time making adjustments to the network complexity.
Figure 9 Pareto frontier (Modified image from: Chakraborti 2014 pg. 98)

EvoNN based on certain output and input values (Chakraborti 2014 pg. 95) formed the Pareto frontier in figure 9. Each of the points represent a tradeoff between network complexity and training error. From these points a decision maker can then select the best choice. Different squares on the graph give different levels of error for the underlying EvoNN as well as different model complexities. (Chakraborti 2014, pg. 95-96) Generally speaking a simpler model has a lower chance of becoming overfitted. More complex models have a higher chance to become overfitted (Pettersson, Chakraborti & Saxén 2005 pg. 389). (Chakraborti 2014, pg. 95)
From the Pareto frontier in figure 9 a decision maker can now pick the best fitting solution. But that raises a question: which of the points should be picked? For this task a statistical information criteria has been effectively used in some earlier papers such as Bhattacharya et al. 2009 (Chakraborti 2014 pg. 96). The potential candidate indicators for this could be Bayesian information criteria (BIC) Akaike’s information criteria (AIC) or the corrected AIC (AICc) (Chakraborti 2014 pg. 97).

The equations for AIC and BIC each are shown below:

\[
AIC = 2k + n \cdot \ln\left(\frac{RSS}{n}\right)
\]

\[
BIC = k \cdot \ln(k) - n \cdot \ln\left(\frac{RSS}{n}\right)
\]

Here k is the amount of parameters that are being used in the model. For a neural network this would mean the number of all connections and biases are included as well. RSS refers to residual sum of squares for the current model that is being created and n refers to the amount of observations. The goal is to use as low value for these as possible. AIC model has a habit of creating models that are slightly over parameterized while BIC criteria instead punishes any increase in parameter count quite a lot. (Chakraborti 2014 pg. 96-98)

The AICc version is a modified AIC version which basically penalizes the increase in parameter count (“Akaike information criterion”, n.d.). The AICc model:

\[
AICc = AIC + \left(\frac{2k(k + 1)}{n - k - 1}\right)
\]

The Si% content is usually quite complicated to monitor in an iron blast furnace and large fluctuations are also very likely. The trained network produced a Pareto frontier as shown in figure 10. The darkened square represents the square that was chosen with the AICc criteria. (Chakraborti 2014 pg. 98-99)
The efficiency of this network can be seen in figure 11 where one can see a comparison with the real data and the estimate from the network. Based on the graph it would seem like the EvoNN captured the major fluctuations of the Si% content very well while ignoring the large fluctuations caused by noise. Therefore this EvoNN has the capacity to weed out noise (figure 11). It should be noted that another method for dealing with noise would be a Kalman filter that has been shown effective in earlier papers (Chakraborti 2014, pg. 97-99)

Figure 10 (Modified image from: Chakraborti 2014 pg. 99) Pareto frontier for the hot metal Si % content. Each square represents a trained network. Based on the AICc criteria the network located in the darkened square was selected.
Figure 11 (Modified image from: Chakraborti 2014, pg. 99) Noisy operation data from blast furnace vs AICc based neural EvoNN model prediction
4 DISCUSSION

The advantage of using data-driven methods is the fact that you don’t have to know everything that is going on in the process as long as all the relevant information can be found in the training data. Another upside for using data-driven methods was the ability to tweak the network complexity during training phase. If the model that was being built was about to become overfitted then one could simply reduce the network’s complexity (reducing the amount of nodes in the network) to reduce this problem. By reducing the network complexity one also reduces the effect of random noise. The problem with reducing network complexity is a risk of the network becoming underfitted. In chapter 3.2 AICc (Akaike Information Criterion corrected) was used to find the best tradeoff between network complexity and accuracy. Some other possible ways to handle overfitting are validation, regularization (“Overfitting”, n.d.) or early stopping (Wang, Venkatesh, Judd 1995).

As discussed earlier in chapter 3.2, neural networks have the ability to filter out noise. It should be noted that a similar effect might also be achieved by using either a Kalman filter, extended Kalman filter or some other filtering technique. It would have been interesting to see a side-by-side comparison between the Kalman filter and an EvoNN but unfortunately such a study was not found.
5 SUMMARY

In this thesis a focus was given to highlight the potential or benefits that data-driven methods can give when applied to process engineering problems. In the first case a neural network was being used in an attempt to control a heat exchanger. A faster overall control of the heat exchanger system was achieved. The neural network controller also saved energy which could be seen as smaller consumption of heating water. In this case the neural network controller performed better when compared to several different PID-controllers.

On the second case the possibilities of an evolutionary neural network (EvoNN) were explored in the diagnosis of Si% content in an iron blast furnace. In this case the EvoNN was able to pick the major shifts in the Si% content even with a very noisy dataset.

In summary one can conclude that neural network based control strategies offer indeed a valid alternative when compared to the more traditional control strategies such as PID-controllers.
6 REFERENCES


Pettersson Frank, Biswas Arijit, Sen Pranay Kumar, Chakraborti Nirupam 2009. Analyzing Leaching Data for Low-Grade Manganese Ore Using Neural Nets and


10.1016/S1570-7946(10)28069-0. Available: https://www.researchgate.net/publication/228398843_An_off-line_Model_Reduction-based_Technique_for_On-line_Linear_MPC_Applications_for_Nonlinear_Large-Scale_Distributed_Systems
