CHAPTER XX

Applications of data analytics tools for data management[1]

By

Mo Jamshidi, Barney Tannahill[2], Maryam Ezell, Yunus Yetis and Halid Kaplan

ACE Laboratory, The University of Texas, San Antonio, TX USA
moj@wacong.org

ABSTRACT

Data, at a very large scale, has been accumulating in all aspects of our lives for a long time. Advances in sensor technology, the Internet, social networks, wireless communication, and inexpensive memory have all contributed to an explosion of "Big Data". Our interconnected world of today and the advent of cyber-physical or system of systems (SoS) are also a key source of data accumulation- be it numerical, image, text or texture, etc. SoS is basically defined as an integration of independently operating, non-homogeneous systems for certain duration to achieve a higher goal than the sum of the parts. Recent efforts have developed a promising approach, called "*Data Analytics",* which uses statistical and computational intelligence (CI) tools such as principal component analysis (PCA), clustering, fuzzy logic, neuro-computing, evolutionary computation, Bayesian networks, data mining, pattern recognition, etc. to reduce the size of "Big Data" to a manageable size and apply these tools to a) extract information, b) build a knowledge base using the derived data, c) optimize validation of clustered knowledge through evolutionary computing and eventually develop a non-parametric model for the "Big Data", and d) Test and verify the model. This chapter attempts to construct a bridge between SoS and Data Analytics to develop reliable models for such systems. Four applications of big data analytics will be presented, i.e. solar, wind, financial and biological data.

Keywords: Data analytics, Computational intelligence, Statistical techniques, Finances, Solar energy, Biology

xx.1. Introduction

System of Systems (SoS) are integrated, independently operating systems working in a cooperative mode to achieve a higher performance. A detailed literature survey on definitions of SoS and many applications can be found in texts by Jamshidi [1,2]. Application areas of SoS are vast. They span from software systems like the Internet to cloud computing, health care, and cyber-physical systems all the way to such hardware dominated cases like military missions, smart grid of electric energy, intelligent transportation, etc. Data

1. This work was supported, in part, by the Lutcher Brown Distinguished Chair, the University of Texas at San Antonio, USA
2. With Southwest Research Institute, San Antonio, TX, USA

analytics and its statistical and intelligent tools including clustering, fuzzy logic, neuro-computing, data mining, pattern recognition, principle component analysis (PCA), Bayesian networks, independent component analysis (ICA), regression analysis and post-processing such as evolutionary computation have their own applications in forecasting, marketing, politics, and all domains of SoS. SoS's are generating "Big Data" which makes modeling of such complex systems a big challenge.

A typical example of SoS is the future smart grid, destined to replace the conventional electric grid. The small-scale version of the smart grid is known as a micro-grid, designed to provide electric power to a home, an office complex or a small local community. A micro-grid is an aggregation of multiple distributed generators (DGs) such as renewable energy sources, conventional generators, and energy storage units which work together as a power supply networked in order to provide both electric power and thermal energy for small communities which may vary from one common building to a smart house or even a set of complicated loads consisting of a mixture of different structures such as buildings, factories, etc. [2]. Typically, a micro-grid operates synchronously in parallel with the main grid. However, there are cases in which a micro-grid operates in islanded mode, or in a disconnected state [3,4]. Accurate predictions of received solar power can reduce operating costs by influencing decisions regarding buying or selling power from the main grid or utilizing non-renewable energy generation sources.

In the new era of smart grid and distributed power generation big data is accumulated in a very large scale.. Another important example is the financial markets. Nowadays, artificial neural networks (ANNs), as a data analytics tool, have been applied in order to predict the stock exchange stock market index. ANNs are one of the data mining techniques that have the learning capability of the human brain. Due to stochastic nature of financial data several research efforts have been made to improve computational efficiency of share values [5, 6]. One of the first financial market prediction projects was by Kimoto, et.al. [7] have used an ANN for the prediction of Tokyo stock exchange index. Mizuno, et al. [8] applied an ANN to the Tokyo stock exchange to predict buying and selling signals with an overall prediction rate of 63%. Sexton, et.al. [9] have determined that use of momentum and start of training in neural networks may solve the problems that may occur in training process. Langdell [10] has utilized neural networks and decision trees to model behaviour of financial stock and currency exchange rates data

The object of this chapter is to use Big Data Analytics approaches to predict or forecast the behaviour of three important aspects of our times – Renewable energy availability, biological white cell behaviour, and stock market prediction. In each case, massive amounts of data are used to achieve these goals.

The remainder of this chapter is as follows: Section 2 briefly describes big data, big data analytics and five statistical and artificial intelligence-based tools. These are PCA, fuzzy logic, fuzzy C-Means, Artificial neural networks and genetic algorithms. Four applications of big data analytics – Solar energy, wind energy, Stock market index prediction and biological white blood cells behavior are presented. Section 4 provides some conclusions.

xx. 2. Big Data and Big Data Analytics

In this section a brief description of big data and data analytics are first given. Then five tools of statistics and AI will follow.

xx.2.1 Big Data

Big data is a popular term used to describe the exponential growth and availability of structured and/or unstructured data. Big data may be as important to business – and society – as the Internet has become. Big data is defined as a collection of data so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing techniques. These data sets have several attributes such as volume which is due to social media, advanced sensors, inexpensive memory, system of systems, (Cyber-physical systems), etc. [2,11]. Big data comes with different frequencies such as RFID tags, electric power smart meters, etc. It has different varieties such as structured and unstructured text, video, audio, stock ticker data, financial transactions, etc. Big data changes due to special events and hence has variability such as daily, seasonal, and event-triggered peaks such as world sports events, etc. [12,13] They can have multiple sources, relationships, and linkages, hence they possess complexity. For a more detailed coverage big data, refer to Chapter 1 in this volume.

xx.2.2 Big Data Analytics

Data analytics represents a set of statistical and artificial intelligence (AI) tools that can be applied to the data on hand to reduce its size, mine it, seek patterns and eventually deliver a non-parametric model. These tools are best to be used in a hybrid mode, where the big data is first pre-processed, mined for patterns, and a knowledge base is developed based on attained information, and eventually a model is constructed and evaluated. Fig.xx-1 shows the process just described [14].

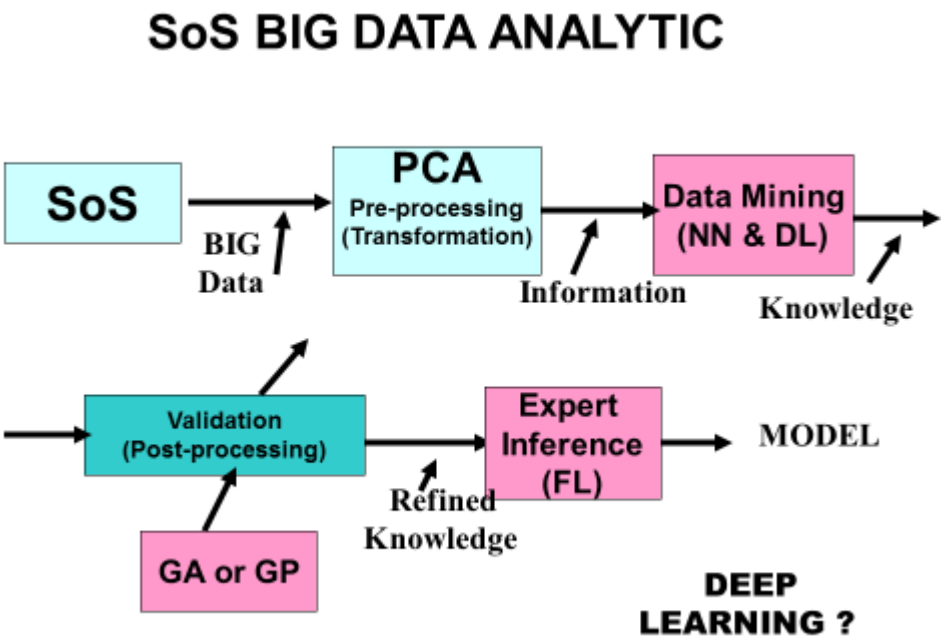

Fig. xx-1. A possible paradigm for Big Data Analytics

xx.2.2.1 Principal Component Analysis

Principal Components Analysis (PCA) is a statistical scheme which uses an orthogonal transformation to identify patterns in data sets so that its similarities and differences are highlighted as a set of values of linearly uncorrelated values called *Principal Components*. Since patterns in data can be hard to find in data of high dimensions, PCA can often help reduce the dimension of the data while bringing up the principal meaning of the information in the data. In other words, PCA can first find the pattern and then compress the data. PCA can work both with numerical as well as image data. Principal components will be independent if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables. [16]. The following steps cans summarize simple steps to perform PCA [16].

Algorithm xx.1 Standard PCA

Step 1: Get a data set
Step 2: Subtract the mean from each data value
Step 3: Calculate the covariance matrix
Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix.
Step 5: Choosing components and forming a feature vector

The principal component of the data will be near to the eigenvector of the covariance matrix with the largest eigenvalue. It is noted that this algorithm is not necessarily applicable for a truly "Big Data" scenarios. In such cases one may utilize deep belief networks [17]

xx.2.2.2 Fuzzy Logic

Fuzzy logic can be defined in two terms. On one hand it refers to multi-level logic based on fuzzy sets, which was first introduced by Zadeh in 1965 [18]. Fuzzy sets is the foundation of any logic, regardless of the number of truth levels it assumes. Fuzzy sets represent a continuum of logical values between 0 (completely false) and 1 (completely true). Hence, fuzzy logic treats many possibilities in reasoning in truth, i.e. human reasoning. Therefore, the theory of fuzzy logic deals with two problems 1) the fuzzy set theory, which deals with the vagueness found in semantics, and 2) the fuzzy measure theory, which deals with the ambiguous nature of judgments and evaluations [18].

On the other hand, fuzzy logic refers to a collection of techniques based on approximate reasoning called *fuzzy systems*, which include fuzzy control, fuzzy mathematics, fuzzy operations research, fuzzy clustering, etc. The primary motivation and "banner" of fuzzy logic is to exploit tolerance of costly exact precision., so if a problem does not require precision, one should not have to pay for it. The traditional calculus of fuzzy logic is based on fuzzy IF-THEN rules like: IF pressure is low and temperature is high then throttle is medium.

In this chapter fuzzy clustering, also called C-Means (next section), and fuzzy reasoning for building a non-parametric fuzzy expert system will be utilized for Data Analytics. However, its

application to "Big Data" is subject to future research and development in such areas as deep architectures and deep learning.

xx.2.2.3 Fuzzy C-Means Clustering

Cluster analysis, or clustering is a process of observation where the same clusters share some similar features. This is an unsupervised learning approach that has been used in various fields including machine learning, data mining, bio-informatics, and pattern recognition, as well as applications such as medical imaging and image segmentation.

Fuzzy C-means clustering algorithm is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 , 1 \le m < \infty$$

Where u_{ij} is the degree of membership of x_i being in cluster j, x_i is the *i*-th of the d-dimensional measured data, c_j is the d-dimension center of the cluster, and $\|*\|$ is any norm expressing the similarity between any measured data and the center. Fuzzy partition is carried out by iterative optimization of the objective function shown above, with an update in membership u_{ij} and cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \frac{\left\| x_i - c_j \right\|^{\frac{2}{m-1}}}{\left\| x_i - c_k \right\|}}$$

Where

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

This iteration will stop when $max_{ij}\{u_{ij}^{k+1} - u_{ij}^{k}\} < \varepsilon$ where ε a terminator criterion between 0 and 1 and k is the iteration step. This procedure converges to a local minimum or a saddle point of J_m. Application of fuzzy C-Means to biological white cells will be given in Section xx3.3.

xx.2.2.3 Traditional Artificial Neural Networks

Traditional artificial neural networks (ANNs) are information processing system that was first inspired by generalizations of mathematical model of human brain of human neuron (See Fig.xx-2).

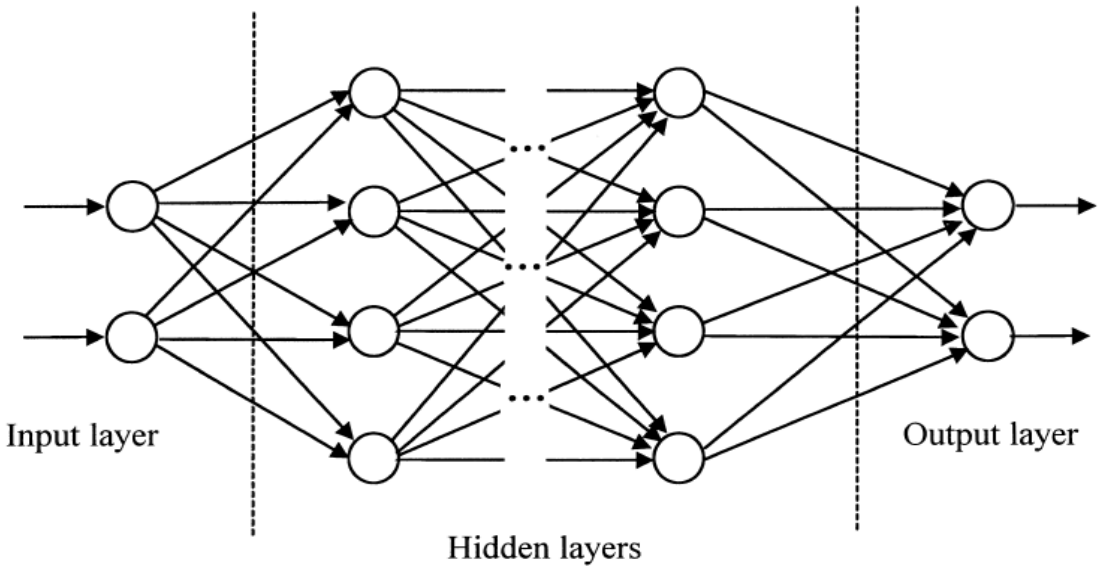

Fig. xx-2. Architecture of a feed forward Multi-Layer Perceptron

Each neuron receives signals from other neurons or from outside (input layer). The Multi-Layer Perceptron (MLP), shown in Figure 1 has three layers of neurons, where one input layer is present. Every neuron employs an activation function that fires when the total input is more than a given threshold. In this chapter, we focus on MLP networks that are layered feed-forward networks, typically trained with static *backpropagation*. These networks are used for application static pattern classification [19,20].

One of the learning methods in MLP neural networks selects an example of training, make a forward and a backward pass. The primary advantage of MLP networks is their ease of use and approximation of any input or output map. The primary disadvantage is that they train very slowly and require a lot of training data. It should be said that the learning speed will dramatically decrease according to the increase of the number of neurons and layers of the networks.

However, while traditional ANN have been around for over 30 years, they are not suitable for big data and deep architectures need to be considered [17]. However, these algorithm have to be used for real "big" data. But, the full efficiency of these algorithms have yet to be proven to the best of our knowledge.

xx.2.2.4 Traditional Genetic Algorithms

Genetic algorithm (GA) is a heuristic search approach that mimics the process of natural selection and survival of the fittest. It belongs to a larger class of computational techniques called *evolutionary computations* which also include *genetic programming*, where symbolic and text data is optimized. Heuristically, solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover will be obtained after several generations.

In GA, an initial population is formed (see Fig. xx-3), possibly by chromosomes randomly represented by 0 and 1 bits. Then each pair of the chromosomes are crossed over or mutated (single

bit only) after selection. Once new extended population is obtained, through comparison of fitness function extra members are eliminated and the resulting new fitter population replaces the old one. Figure 3 shows the lifecycle of the GA algorithm [21]. GA has been used in Section xx.3.1 to enhance solar energy forecasting via ANN.

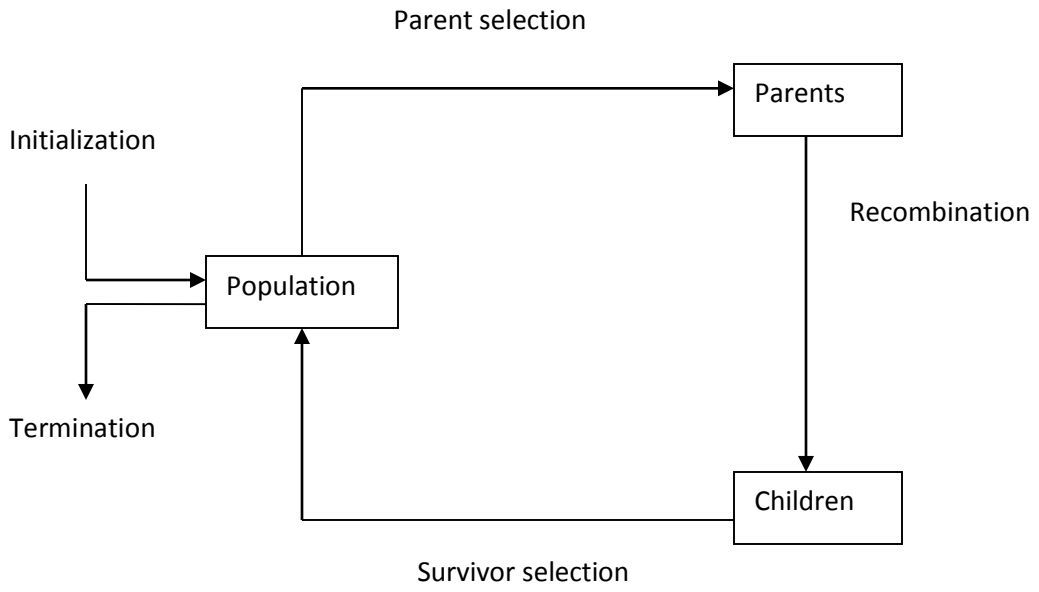

Fig. xx-3 GA Algorithm's Cycle [21]

xx.3 Applications of Data Analytics

In this section four different applications of physical, financial and biological data analytics will be presented.

xx. 3.1 Solar Energy Forecasting

This section provides an example showing how Data Analytics can be used to generate models to forecast produced photovoltaic (or solar) energy to assist in the optimization of a micro-grid SoS. Tools like fuzzy interference, neural networks, PCA, and genetic algorithms are used.

The object of this section and next is to use a massive amount of environmental data in order to derive an unconventional model capable of performing solar and wind energy predictions, which could be leveraged to control energy trading, power generation, and energy storage strategies to optimize operating costs. Recent studies primarily produced by Tannahill [22] have been performed that start looking into different aspects of this problem.

To ensure that the environmental input data for the different data analytics tools is comprehensive, data from different sources was combined to form the full dataset. This was possible because of the solar research projects occurring in Golden, CO, where the National Renewable Energy Laboratory (NREL) is conducting long term research and data recording to support the growing renewable energy industry. Data from the Solar Radiation Research Laboratory (SRRL), SOLPOS data made available by the Measurement and Instrumentation Data Center (MIDC), and data from the Iowa Environmental Mesonet (IEM) Automated Surface Observing System (ASOS) station near the Golden, CO site was also included to have current weather data in the set [22].

Data from the month of October 2012 was combined from the different sources of data. This final set includes one sample for each minute of the month and incorporates measured values for approximately 250 different variables at each data point. The data set was sanitized to only include data points containing valid sensor data prior to the analysis.

Once the viability of this approach was established, a year's worth of data from 2013 was retrieved from the online resources and was similarly sanitized in order to serve as a data set against which the effectiveness of the data analytics techniques could be evaluated for an entire year.

Since the micro-grid would benefit from predicted values of solar irradiance, it was decided that the output of the data analytics should be predicted values of three key irradiance parameters (Global Horizontal Irradiance (GHI), Direct Horizontal Irradiance (DHI), and Direct Normal Irradiance (DNI)). These values were shifted by 60 minutes so that they would serve as output datasets for the training of the fuzzy Inference System and Neural Network fitting tools that ultimately provided the means of non-parametric model generation in this exercise.

Once the objective of the data analytics was determined, relevant inputs to the data analytics tools needed to be identified. The full dataset contains approximately 250 different variables. Unfortunately, due to the curse of dimensionality, including all these variables in the data analytics was not practical due to memory and execution time constraints. If this exercise was to be conducted using distributed *cloud computing*, the number of variables to be considered might not need to be down-selected; however, since this effort took place on a single PC, the number of variables needed to be reduced. Ideally, a subject matter expert would be available to optimally identify the best variables to include in the evaluated dataset, or an adaptive training algorithm could be used to automatically perform the selection process. For the purposes of this section, several variables were selected based on intuition, including cloud levels, humidity, temperature, wind speed, and current irradiance levels.

Next, cleanup of the reduced dimension dataset was started by removing all data points containing invalid values from the data set. For instance, during night hours, many solar irradiance parameters contained negative values. Once these invalid data points were removed, the data set was further reduced by removing data points in which GHI, DHI, and DNI levels were very low. The primary reason for this second step was to reduce the amount of time and memory necessary for analysis. Fig. xx-4 is contains the measurements of GHI, DHI, and DNI over one day in the cleaned dataset.

After cleaning took place, the data could be fed into either of the two non-parametric model generating tools, the Fuzzy Inference System Generator and Back-Propagation Neural Network training tools included in the Matlab's fuzzy logic toolbox and the ANN Toolbox.

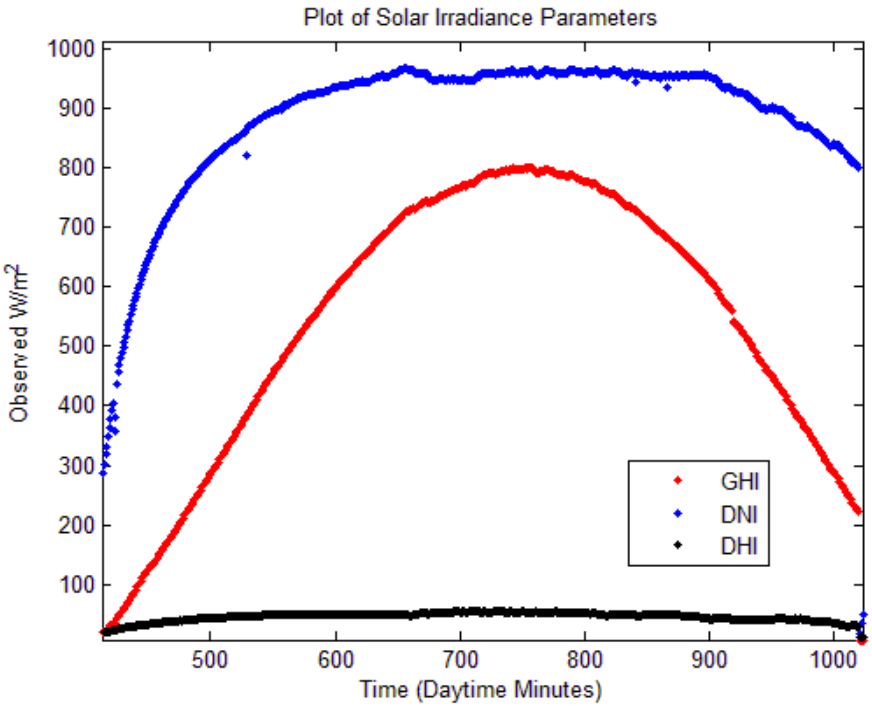

Fig. xx-4. Three Key Irradiance Parameter Plot for a Clear Day [22]

The Matlab Fuzzy Logic Toolbox function used in this step, *genfis3* uses Fuzzy C-Means clustering (Section xx.2.2.3) to cluster values for each variable which produces fuzzy membership functions for each of the variables in the input matrix and output matrix. It then determines the rules necessary to map each of the fuzzy inputs to the outputs to best match the training data set. These membership functions and rules can be viewed using the Matlab's FIS GUI tools such as *ruleview*. Fig. xx-5 shows the results of running *genfis3* on only four different variables in the dataset.

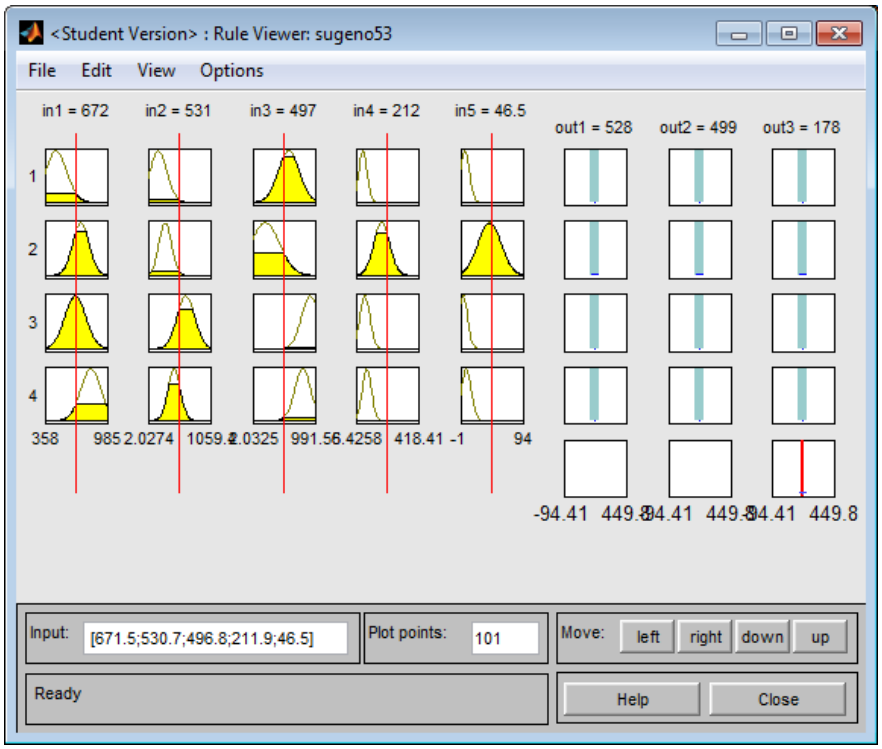

Fig. xx-5. Rule View of Generated FIS Membership Functions [22]

When comparing the predicted values to the actual values of GHI, DHI, and DNI, differences in the observed and predicted data points could corresponds to the presence of clouds or other anomalies that could not be predicted an hour in advance using the variables input to the function. In addition to unpredictable weather phenomena, such anomalies could include sensor accuracy error, data acquisition noise or malfunctions, missing data, and other issues associated with the acquisition of the environmental data.

The second model generating method used was the Matlab's ANN training toolbox. By default, this tool uses the Levenberg-Marquardt back propagation method to train the network to minimize its mean squared error performance. Fig. xx-6 shows a representation of the feed-forward neural network generated when training using 13 inputs variables and one hidden layer comprised of 10 neurons.

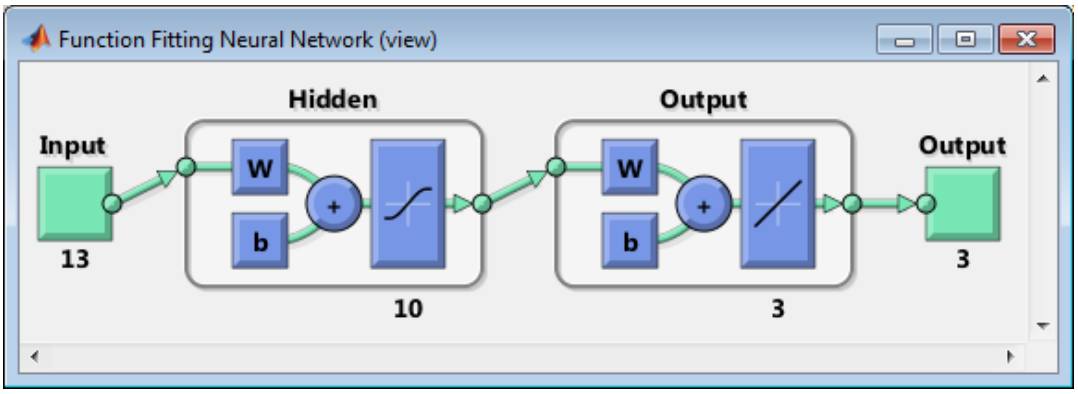

Fig. xx-6. Trained Feed-Forward Neural Network Representation

In order to improve the performance of the model generation methods, the following additional data derived points were included in the training sets: $x(t)^2$, sin(x(t)), cos(x(t)), slope(x(t-1):x(t)), slope(x(t-60):x(t)), mean(x(t-60):x(t)), and stdev(x(t-60):x(t)). These functions in some cases made the data more useable and added memory to the training algorithms, which greatly improved performance. Where necessary, PCA was used to reduce the dimension of the data while minimizing the amount of useful data lost.

The results of this analysis showed that the best results were achieved with the Neural Network training tool. The results (mean Square Errors) are discussed in Table xx-1 below. NN10 refers to a standard feed forward neural network consisting of 10 neurons in one hidden layer.

Table xx-1. Performance Comparison of *GENFIS3* and *NFTOOL* Prediction Models

Model Type	**Input Params.**	**PCA?**	**Final Dim**	**MSE GHI**	**MSE DNI**	**MSE DHI**	**R**
FIS	244	Y	50	2.96E+03	2.08E+04	1.05E+03	0.967
NN10	244	Y	150	9.97E+02	2.95E+03	4.57E+02	0.994

These models performed significantly better than a simple predictor based on the average solar parameter values observed at different points during the day. The *nftool* model training time was noted to be linear with respect to the number of variables in the training data set, but *genfis3*'s training time was observed to be much longer with higher dimensions. Its training time appears to be a function of the training set dimension squared. The training time with both functions was linearly related to the number of variable instances in the training dataset, but the slope of this line was an order of magnitude smaller when using the *nftool*.

Next, Genetic Algorithms (Section xx.2.2.4) were used to improve the accuracy of the best performing generated model while minimizing the number of inputs necessary to implement the system. Each population element was a vector of binary variables indicating which of the available

variables in the training set would be used for training the model. This is useful because it would reduce the amount of sensors necessary to implement a system.

Over time, the genetic algorithm solver within the Matlab's Global Optimization Toolbox reduced the training data set from 244 variables to 74 variables. The final solution had a MSE GHI of 7.42E+02, a MSE DNI of 2.60E+03, and a MSE GHI of 1.78E+02.

Finally, the methods discussed above were used to evaluate the models against a larger data set (the entire year of 2013). After some experimentation, it was apparent that generating a single model to make predictions about the entire year was not an ideal approach. Instead, a prediction engine was designed to generate a model for each desired prediction based on recent data. The engine performs the following:

1. Searches the pre-processed data set in order to find the prediction time index of interest
2. Fetches the appropriate training data set (ten day window starting an hour before the prediction time)
3. Performs any additional pre-processing to prepare the training data set for model generation training.
4. Trains two solar prediction models using the training data.
5. Selects the models with the highest R values and uses them to make the requested prediction

The prediction engine was used to make a prediction every 6 hours throughout the year. The resulting error statistics are shown in Table xx-2 below.

Table xx-2. Solar Prediction Engine Error Statistics

	RMSE GHI	**RMSE DHI**	**RMSE DNI**	**Solar Model R Value**
Mean	4.250	1.491	-0.9052	0.981739
Min	-651.4	-707.8	-234.9	0.941555
Max	778.6	809.0	198.3	0.998421
STDEV	90.55	111.3	40.88	0.009399
RMSE	90.62	111.3	40.88	N/A

Next, the prediction model was used to generate a high time resolution prediction data set over a short period (one day) in order to illustrate the effectiveness of the engine with a more easily displayed data set (see Figs xx-7- xx-9).

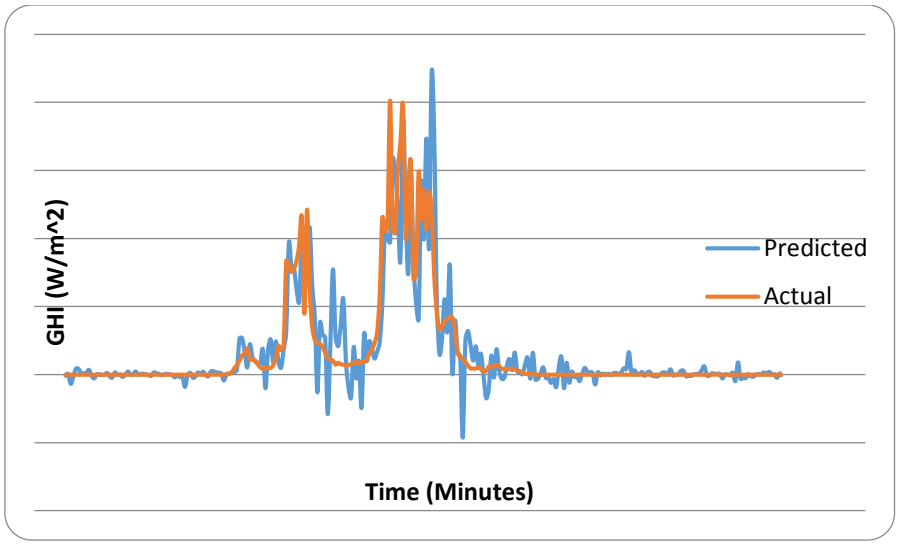

Fig. xx-7. Prediction Engine GHI Results (5 Minute Resolution)

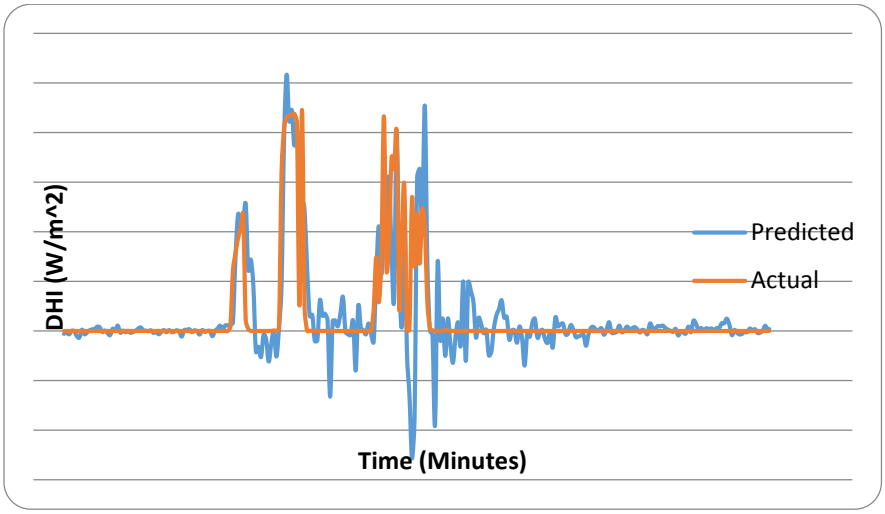

Fig. xx-8. Prediction Engine DHI Results (5 Minute Resolution)

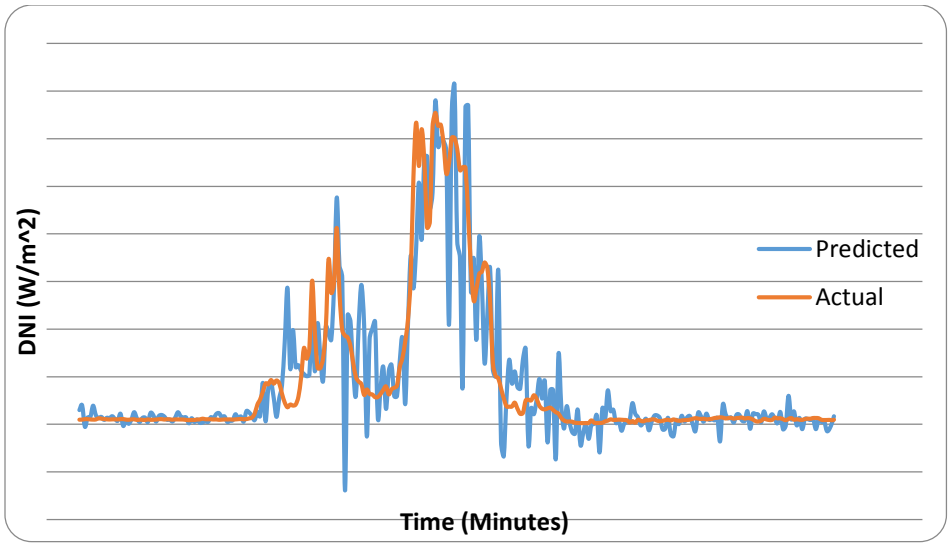

Fig. xx-9. Prediction Engine DNI Results (5 Minute Resolution)

xx.3.2 Wind Energy Forecasting

Many of the data analytics techniques discussed in Section xx.3.1 were also evaluated in their ability to generate models capable of predicting available wind power capacity. Similar to the solar predictions, knowing the available wind power ahead of time could be useful in energy trading or control algorithms.

It was decided that the output of the data analytics for this exercise should be predicted values of wind speed at three different altitudes (19 ft, 22 ft, and 42 ft in altitude). These values were shifted by 60 minutes so that they would serve as output datasets for the training of the neural networks investigated in the following neural network types:

- Standard Feed-Forward Neural Network
- Time Delay Network
- Nonlinear Autoregressive Network with Exogenous Inputs (NARXNET) Neural Network
- Layer Recurrent Neural Network

Two different input data sets were used for this investigation. The first was merely reduced to a dimension of 21 using PCA [16]. The second was first expanded to include derived, preprocessed values including slope and average values from the past parameter values. Then, this second dataset was also reduced to a dimension of 21 using PCA.

A variety of configurations were tested with each network type, but surprisingly, the best performing neural networks were those using a pre-expanded (via nonlinear expansion) data set fed into a conventional feed forward neural network with ten neurons in the hidden layer. The figures below show the results and error generated using this network to predict wind speed an hour in advance.

With this information in mind, year-long predictions were made using the model prediction engine discussed in section xx.2.2. For the purposes of this work, predicted wind power availability

was calculated assuming the wind speed across the entire wind turbine blade area was the same. It was also assumed that the air density was 1.23 kg/m3 throughout the year, and that the power coefficient Cp was 0.4. These assumptions were used in conjunction with the wind turbine equation found [23] to calculate power density availability (Watts/m2). This quantity can be multiplied by the cumulative sweep area of a wind turbine farm to calculate total available wind power; however, this step was left out of this exercise to keep the results more generalized.

The statistics of the resulting data from the year-long wind power prediction are included in Table below.

Table xx-3 Wind Prediction Engine Error Statistics

	RMSE DNI	**Wind Model R Value**
Mean	-0.01645	0.9113
Min	-10.15	0.6670
Max	9.617	0.9700
STDEV	1.614	0.03700
RMSE	1.614	N/A

Fig. xx-10 shows a graph of the regression analysis's index values for both the solar and wind prediction model performance over the year-long data set.

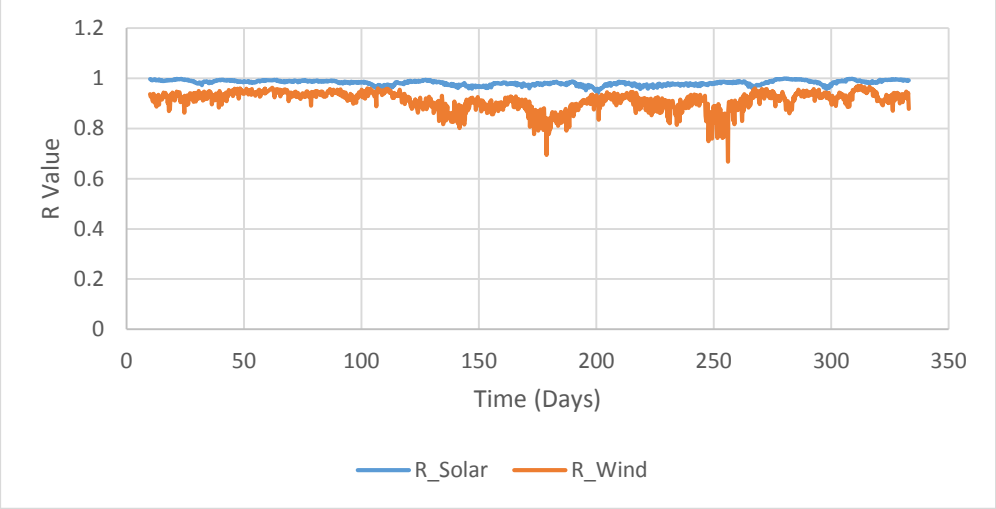

Fig. xx-10. Prediction Engine Model R Values (6 Hour Resolution)

A higher resolution prediction loop based is shown in Fig. xx-11.

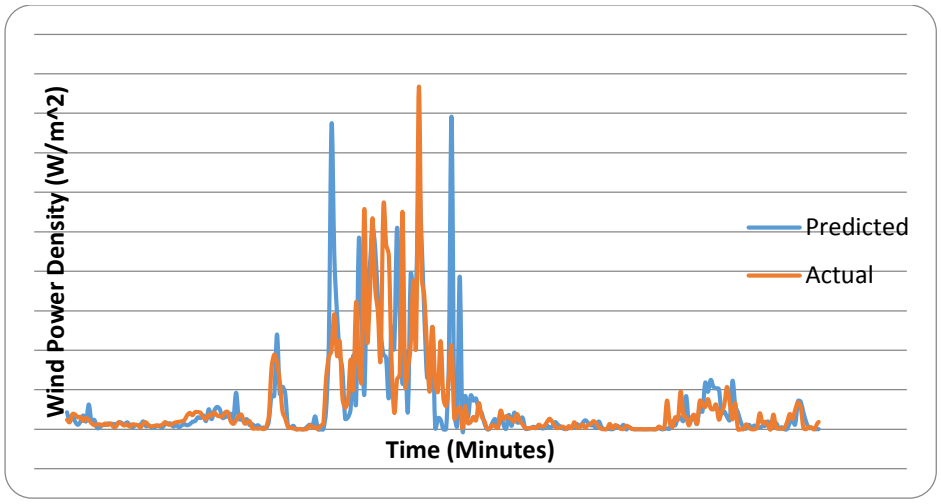

Fig. xx-11. Prediction Engine Average Wind Power Density Results (5 Minute Resolution)

xx.3.3 Financial Data Analytics

In this section artificial neural networks (ANN) have been used to predict stock market index. ANN has long been used as a data mining tool. This section presents a forecasting scheme of NASDAQ's stock values using ANN. For that purpose, actual values for the exchange rate value of the NASDAQ Stock Market index were used. The generalized feedforward network, used here, was trained with stock market prices data between 2012 and 2013. Prediction of stock market price is an important issues in national economies. Many researchers have proposed forecasting market price [24, 25].

xx.3.3.1 Training Process

Training process of ANN is through a gradient-based optimization approach (similar to conjugate gradient approach) through adjusting of inter-layer links weights between neurons. This algorithm, whose most celebrated approach is called "backpropagation," is also called training or learning phase of a multi-layer perceptron (MLP).

The initial values of weights are determined through a random generation. The network is adjusted based on a comparison of the output and the target during the training (Fig. xx-12).

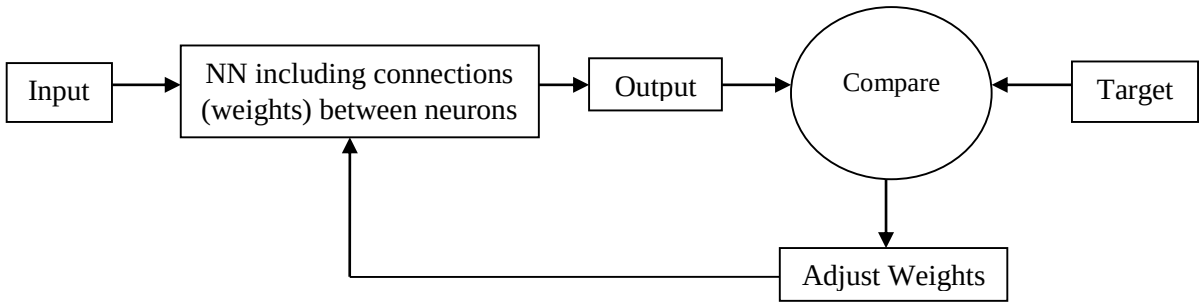

Fig. xx-12 A network based on a desired target and output comparison [26]

The training process requires a set of examples of proper network behaviour and target outputs. During training, the network of the weights and biases are repeated to minimize the network performance function. Mean square error (MSE) performance index during training of feed-forward neural network. MSE is the average squared error between the network outputs and the target outputs [27].

Training is the process of propagating errors back though the system from the output layer towards. Backpropagation is a common paradigm for the training process, utilizing errors between layers of the network. Output is the only layer which has a target value. Training occurs until the errors in the weights reduces to a pre-set minimum value. Training has been kept on until its iterative process reaches such that the errors in weights are below a threshold. MLP is the most common feed-forward networks.

With these settings, the input and target vectors will be randomly divided into three sets as follows: 70% will be used for training and 15% will be used to validate the network in generalizing and to stopping training before overfitting. The last 15% will be used as a completely independent test of network generalization.

For simulation purpose, the NASDAQ dataset of daily stock prices has been used [28, 29]. We used five input variables for ANN such as opening price, the highest price, the lowest price, volume of stock, and adjusted daily closing price of the day. Moreover, the architecture and pre-set parameters for ANN were: 10 hidden neurons, 0.4 as learning rate, 0.75 as momentum constant and 1000 was chosen as maximum epochs. Mean squared error (MSE) is the average squared of error between outputs and targets. If the test curve had increased significantly before the validation curve increased, it means it is possible that some over fitting might have occurred. The result of the simulation were acceptable and reasonable.

Error histogram provided additional verification of network performance. It can be clearly seen that errors are between -120 and +100 (Fig. xx-13). Data set represented hundreds of thousands, so these errors were found negligible considering that the error was smaller than about 0.02 % of targets.

Each input variable of ANN was preprocessed. Mean value, average of the training set was small as compared to its standard deviation. Index rage was between -1 and +1 [30]. We were able to use simple formula which is $Index(x) = (Index(x) - Min\ (Index)) / (Max\ (Index) - Min\ (Index))$ [30]. It can be clearly seen the regression plot of the training set (See Figs. xx-14- xx-15). Each of the figures corresponds to the target from the output array. Regression values (correlation coefficients) are very close to 1. It indicates that the correlation between the forecasted figures and the target is very high.

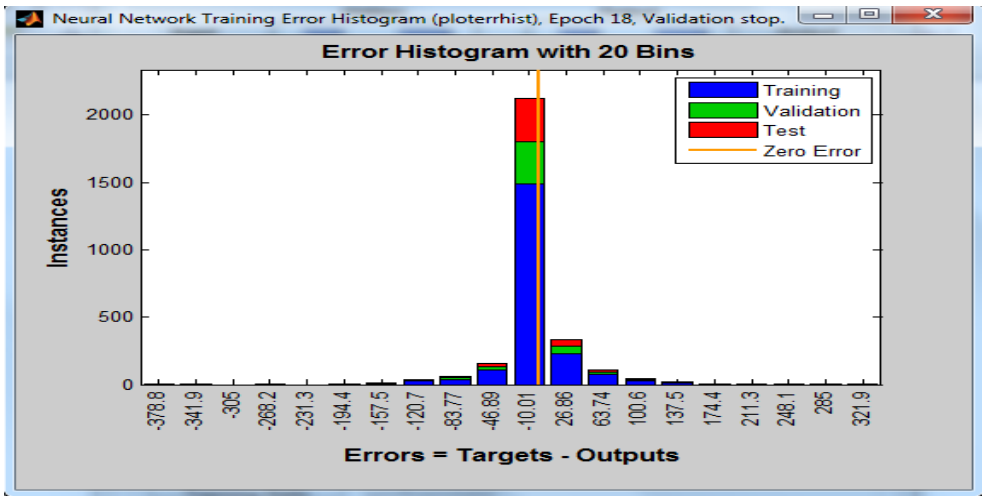

Fig. xx-13. Error Histogram

Regression was used to validate the network performance. For a perfect fit, the data fell along a 45^{o} degree line, where the network outputs are equal to the targets. For this problem, the fit is reasonably good for all data sets, with R values in each case of 0.99 or above (Figures x-13-x-14).

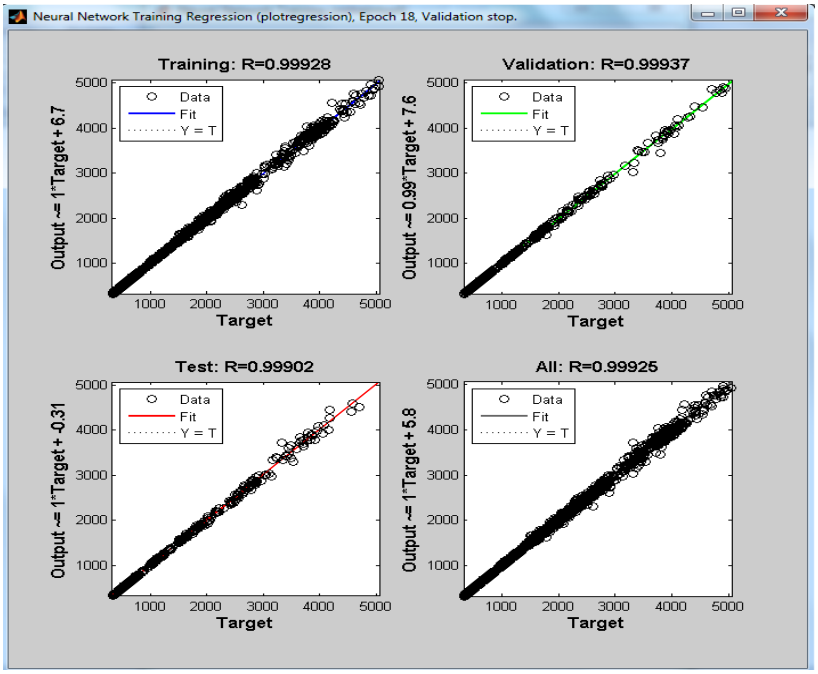

Fig. xx-14. Regression plot for training

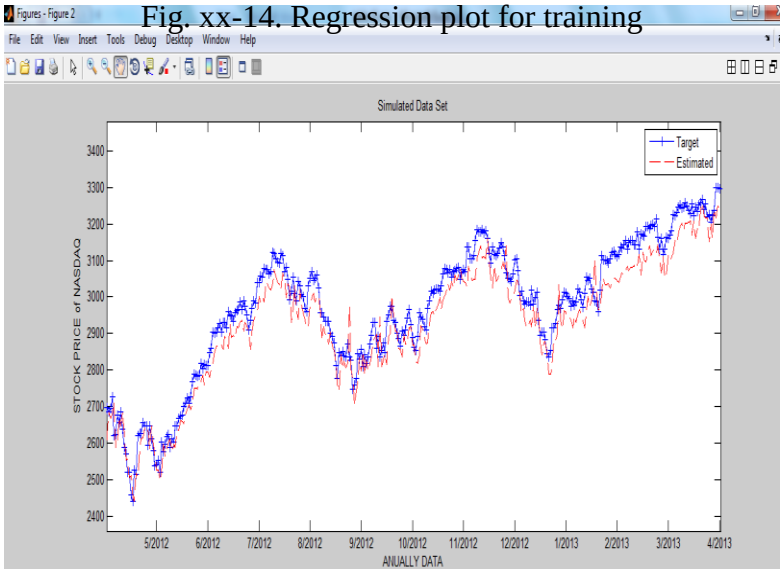

Fig. xx-15. Target and Estimated data of annually NASDAQ data

xx.3.3.2 Biological Data Analytics

This brief section describes the creation of a server on the University of Texas, San Antonio (UTSA) Open Cloud Institute's private cloud and how it was used to run code on some data that is typically too large to be handled by a personal computer.

A virtual machine on the UTSA private cloud running Ubuntu 12.04 as its operating system was created. The code and images were securely copied onto the server. The code was opened in the operating system's *vi editor* and was edited to be used on a batch of images before execution began.

The Fuzzy C-Means algorithm (Section xx.2.2.3) was utilized and the code was modified to be used on our dataset. The data set was a batch of grey-scale images and Fuzzy C-Means algorithm has been used to form clusters based on the pixel values. The code was originally used for clustering one image but it was embedded in a loop to read 312 images, one at a time, find and print the centroids and membership values and then show the clustered images.

Each image is the image of a white blood cell on a blood film and we have chosen to divide the image data (the pixel values) into 4 clusters which are represented by colors white, black, light gray and dark gray. Each image was first converted into the form of a vector of pixel values.

It took the same amount of time to run the code for one image on the virtual machine of the personal laptop as it took on the cloud, but it kept running the code on the cloud for one image after another without crashing the server, whereas running it on the laptop would heat up the CPU and crash it. Fig. xx-16 shows the results, while the centroids of R G B for all 4 clusters are summarized in Table x-4.

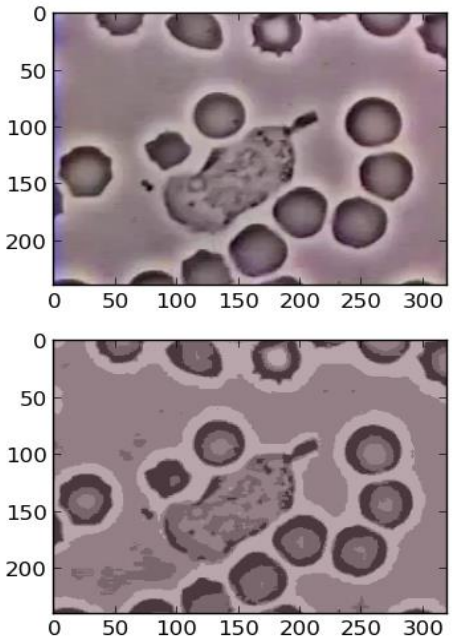

Fig. xx-16. The results of a fuzzy C-mean clustering for a white blood cell.

Table xx-4. Clustering centroids results for biological data

	R	G	B
Cluster 1	0.72771727	0.65201056	0.67813281
Cluster 2	0.47189973	0.40160155	0.42703744
Cluster 3	0.58333669	0.50191924	0.53275565
Cluster 4	0.29300364	0.22106555	0.24503933

xx.4 Conclusions

This chapter describes the use of a few tools of data analytics paradigms to assess future estimates of physical, economical and biological variables. Physical phenomena like the intermittency of renewable energy sources, the volatility of stock markets or the inconclusiveness of health related issues all cry out for accurate estimation. Solar and wind energy data were used to forecast availability of solar and wind energy up to 6 hours and can be extended to 24 hours.

ANN was used to predict future behavior of stock market indices such as NASDAQ index in the United States. The stock market index prediction using artificial neural networks has produced a very efficient results for over a year. Regression index R was within 2% of actual value. The final application of data analytics was the clustering of biological white blood cells utilizing UTSA's Open Cloud Institute's private cloud for extensive computing.

Future work will include testing much larger data for analytics work and utilization of new tools like "Deep Learning" [31, 32]. Deep learning can also be combined with tools like Bayesian networks, PCA, GA, GP (genetic programming g) and fuzzy logic to enhance the final forecast.

References

[1] Jamshidi, M. (ed.), *Systems of Systems Engineering – Principles and Applications*, CRC – Taylor & Francis Publishers, London, UK, 2008.

[2] Jamshidi, M. (ed., 2009), System of Systems Engineering – Innovations for the 21st Century, John Wiley & Sons, Publishers, New York, NY.

[3] Manjili, Y. S., Rajaee, A., Jamshidi, and Kelley, B. (2012) "Fuzzy Control of Electricity Storage Unit for Energy Management of Micro-Grids", Proc. World Automation Congress, Puerto Vallarta Mexico, 2012, pp.1-6.

[4] Tannahill, B. K., Maute, C., Yetis, Y., Ezell, M. N., Jaimes, A., Rosas, R. Motaghi, A., Kaplan, H. and Jamshidi, M. (2013) "Modeling of System of Systems via Data Analytics – Case for "Big Data" in SoS," Proc. 8th IEEE SoSE, Maui, HI, June 2-4, 2013, EDAS # 1569744777

[5] Sergiu, C. (2011) "Financial Predictor via Neural Network". Retrieved from http://www.codeproject.com

[6] M. Jamshidi, M. JAMSHIDI, B. Tannahill, Y. Yetis and H. Kaplan, "Big Data Analytics via Soft Computing Paradigms," Chapter 12, Frontiers of Higher Order Fuzzy Sets, Springer-Verlag, Germany, pp. 229-258, 2015

[7] T. Kimoto, K. Asawaka, M. Yoda, and M. Takeoka, "Stock Market Prediction System with Modular Neural Network", Processing of International Joint Conference on Neural Networks, 1990, pp. 1-6.

[8] H. Mizono, M. Kosaka, H. Yajma, and N. Komoda, "Application of Neural Network to Technical Analysis of Stock Market Prediction", Studies in Informatics and Control, 1998, Vol.7, No.3, pp.111-120.

[9] S. R. Sexton, R. E. Dorsey and J. D. Johnson, "Toward Global Optimization of Neural Network: A Comparison of the Genetic Algorithm and Backpropagation", 1998, Decision Support Systems vol.22, pp.117-185.

[10] S. Langdell, "Examples of the use of data mining in financial applications," http://www.nag.com/IndustryArticles/DMinFinancialApps.pdf, Access July 15, 2015

[11] M. Jamshidi and B. K. Tannahill, "Big Data Analytics Paradigms - From PCA to Deep Learning, Proc. AAAI Workshop on Big Data, Stanford, CA, March, 2014.

[12] http://www.sas.com, February 20, 2014.

[13] http://en.wikipedia.org/wiki/Big_Data, July 25, 2015

[14] http://en.wikipedia.org/wiki/Data_analytics , July 25, 2015

[15] J. Shlens. (2009, Apr. 22). A Tutorial on Principal Component Analysis [Online]. Available: http://www.snl.salk.edu/~shlens/pca.pdf

[16] L. I. Smith. (2002, Feb. 26). A Tutorial on Principal Components Analysis [Online].
[17] G. Hinton, S. Osindero and Y.-W. The, "A fast learning algorithm for deep belief nets," Neural Computation, Vol 18, No. 7, pp. 1527-1554.
[18] A. Zilouchian and M. Jamshidi, (eds.), Intelligent Control Systems with Soft Computing Methodologies, CRC Publishers, Boca Raton, FL, 2001, Chapters, 8-10.
[19] D. E. Rumelhart, G. E. Hinton, R. J. Wiliams, "Learning Internal Representation by Error Propagation: Parallel Distributed Processing Explorations in the Microstructures of Cognition", McClelland J. L. (eds.), 1:318-362, MIT Press, Cambridge, 1986.
[20] A. Zilouchian and M. Jamshidi, (eds.), Intelligent Control Systems with Soft Computing Methodologies, CRC Publishers, Boca Raton, FL, 2001, Chapter 2.
[21] www.cse.hcmut.edu.vn/~dtanh/download/Genetic_Algorithm.doc, Accessed July 15, 2015
[22] B. Tannahill, "Big Data Analytics Techniques: Predicting Renewable Energy Capacity To Facilitate The Optimization Of Power Plant Energy Trading And Control Algorithms," M.S. Thesis, ACE Laboratory, ECE Department, The University of Texas at San Antonio, December 2013.
[23]The Royal Academy of Engineering. (2014). Wind Turbine Power Calculations [Online]. Available: http://www.raeng.org.uk/education/diploma/maths/pdf/exemplars_advanced/
[24] M.T. Philip, K. Poul, S.O. Choy, K. Reggie, S.C. Ng, J. Mark, T. Jonathan, K. Kai, and W. Tak-Lam, "Design and Implementation of NN5 for Hong Stock Price Forecasting", Journal of Engineering Application of Artificial Intelligence, vol. 20, 2007, pp.453-461.
[25] X. Wu, M. Fund and A. Flitman, "Forecasting Stock Performance using Intelligent Hybrid Systems", Springerlink, 2001, pp. 447-456.
[26] R. Ball and P. Tissot, "Demonstration of Artificial Neural Network in Matlab", Division of Near shore Research, Texas A&M University, 2006
[27] S. Neenwi, P. O. Asagba, L. G. Kabari, "Predicting the Nigerian Stock Market Using Artificial Neural Network", European Journal of Computer Science and Information, Vol.1, No.1, pp.30-39, June 2013
[28] Yahoo Historical Prices of NASDAQ, Retrieved May 5, 2013 "http://finance.yahoo.com/q/hp?s=%5EIXIC&a=00&b=2&c=2013&d=03&e=15&f=2013&g=d"
[29] Mathworks Website, November 2013 , http://www.mathworks.com/ help/nnet/ug/analyze-neural-network-performance-after-training.html
[30] Ciumac Sergiu (25 May 2011) "Financial Predictor via Neural Network". Retrieved from http://www.codeproject.com
[31] http://www.deeplearning.net/tutorial/ July 20, 2014.
[32] A. Moussavi and M. Jamshidi, "Cascaded and Partially Cascaded Stacked Autoencoders Applied to Traffic Flow Prediction," Neural Computing & Applications Journal, to appear, 2015