Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 2 (2014) pp. 318-330

Comparison of Soft Computing Approaches for Prediction of Crude Oil Price

Lubna A.Gabralla[1] and Ajith Abraham[2, 3]

[1] Faculty of Computer Science & Information Technology,
Sudan University of Science and Technology,
Khartoum, Sudan
lubnagabralla@gmail.com

[2]Machine Intelligence Research Labs, WA, USA
[3]IT4Innovations, VSB - Technical University of
Ostrava, Czech Republic
ajith.abraham@ieee.org

***Abstract*: In this paper, we studied a vast array of machine learning approaches (ML). All are sound, robust techniques that are extremely applicable to practical prediction problems. To guarantee build successful machine learning model in predicting crude oil prices, we must apply practical steps for selecting best learning algorithm by running it over our data. We modeled the prediction process and analyzing the direct prediction models, which includes isotonic regression, SMOreg, Kstar, IBK ,ExtraTree, REPTree and several types of NNs includes FFN, RCN and RBF in previous articles . The purpose of this paper is to construct comparison among the previous direct models. Furthermore, the comparison of these algorithms is presented based on a root mean squared error (RMSE) and mean absolute error (MAE) to find out the best suitable approaches. We are confident that this study will be useful to researchers for the problem of predicting oil prices and similar problem.**

Keywords: machine learning, direct prediction models, predicting crude oil prices, Isotonic regression, SMOreg, Kstar, IBK ,ExtraTree, REPTree

I. Introduction

In 2014, oil prices are down sharply, losing over 50% of their value since June peak, when it was $115 a barrel and it is now below $51 [1]. This has raised a number of questions: Who is responsible for this decline? What are the factors and reasons that led to this change in prices? Will oil prices recover in 2015?? What is the effect of this decline on the countries that are highly dependent on oil based economy or importing countries? Falling oil prices have both positive and negative impacts. On one hand for many people, cheaper oil means lower gasoline prices and economies of importing countries may rebound a bit if the declining prices are exploited and on the other hand oil-exporting countries are very hardly hit. Similarly rise in prices have opposite effects in both directions. Crude oil price prediction is a challenging task due to its complex nonlinear and chaotic behavior. During the last couple of decades, both academicians and practitioners have devoted proactive knowledge to address this issue. A strand of them has focused on some key factors that may influence the crude oil price prediction accuracy [2-4], while others concentrated on designing models that will assist to predict crude oil with accurate results [5-7]. To support the global economy, companies and institutions to hedge against surprise changes to make sound decisions and building a healthy and successful economy. There is a vast and still growing literature that aims to explain and address the stochastic behavior of oil prices. Chen, et al. [8] concluded that the fluctuation of crude oil prices in the global market at present has caused a growing interest and efforts in examining current models and proposing new ones and identifying improved approaches in order to avoid the effects of crude oil price unpredictability. The aim of this paper is on the one hand to conduct comparative study between popular direct models for predicting crude oil prices this may lead to solve the problem of selecting or recommending a suitable subset of ML algorithms for a given task such as prediction crude oil price and on the other hand the experiments evidences can explain the role of selecting appropriate parameters for successful of each algorithm. The structure of this paper is as follows. Section 2 depicts the literature review followed by the research methodology in Section 3. Section 4 provides the details about dataset and data preprocessing, experimental results and discussions are described in Section 5. Comparisons between prediction models presented in Section 6 and Finally, the conclusions are given in Section 7.

II. Related Research

In the past decades, traditional statistical and econometric techniques, such as linear regression (LinR), co-integration analysis, Generalized Autoregressive Conditional Heteroskedastic (GARCH) models, naive random walk, vector auto-regression (VAR) and error correction models (ECM) have been widely applied to crude oil price prediction

[9]. Early attempts to model and forecast volatility Huntington [10] implemented a sophisticated econometric model to predict crude oil prices in the 1980s. Abramson and Finizza [11] suggested a probabilistic method for predictions of average annual oil prices. Gulen (1998) followed them and used co-integration analysis to predict the WTI price, using monthly data to cover the period of March 1983 to Oct 1995, and Barone-Adesi [12], [13] suggested a semi-parametric approach for oil price forecasting. Similarly Morana [14] proposed a semi-parametric approach based on the bootstrap approach, using daily oil prices for the period from 4 January 1982 to 21 January 1999 to predict the oil prices. The above models can provide good prediction results when the price series under study is linear or near linear. However, several experiments have proved that the prediction performance might be very poor if one continued using these traditional statistical and econometric models [15]. The major reason causing this phenomenon was that the traditional statistical and econometric models were built on linear assumptions and they cannot capture the nonlinear patterns hidden in the crude oil price series [9]. Due to the limitations of the traditional statistical and econometric models, some nonlinear and emerging Machine Learning (ML) models, such as artificial neural networks (ANN) are viewed as non-parametric, nonlinear, and assumption-free models [16]. Single ML techniques have been applied to predict crude oil prices using voluminous historical data to build prediction models [17]. Artificial neural networks (ANN) are designed to represent data by simulating the work of the human brain. ANN's emerged in different areas such as industrial, medical and business, and has achieved successful results. Therefore, many researchers also used ANN in the prediction of oil prices. Haidar, et al. [18] suggested a network to predict the oil prices using two groups of inputs, crude oil futures data, and Dollar index, S&P500, gold price and heating oil price. The authors measured performance by heat rate, root mean square error, correlation coefficient, mean squared error and mean absolute error. The authors concluded that heating oil spot price support forecast crude oil spot price in numerous steps prediction. Alizadeh and Mafinezhad [19] proposed General Regression Neural network (GRNN) using six factors monthly data to predict Brent crude oil price. Experiment results show that the model achieved high accuracy in normal ad crisis situations. Support Vector Machines (SVM) provide a class of competitive learning algorithms to improve generalization performance of neural networks and accomplish global optimum solutions simultaneously [20]. Khashman and Nwulu [21] designed an intelligent system based on SVM to predict the price of crude oil involving eight input factors (global demand; a random world event; among others). Empirical results show high prediction accuracy. The success of the application of support vector machine in solving several problems depend on the appropriate selection, and use of a kernel function[22], therefore Chiroma, et al. [22] compared the performances of five different kernel functions of the support vector machine to provide better understanding of the behavior of the kernel functions for support vector machine and improve the accuracy of crude oil prediction. Monthly data from 1987 to 2012 were utilized. The empirical results exhibited that the wave kernel function is significantly better than that of radial basis function, polynomial, exponential, and sigmoid kernel functions on crude oil prediction. Yu, et al. [7] proposed a new model based on rough-set-refined text mining (RSTM) for crude oil price predicting. The authors evaluated the model by comparing it with statistical models, time series models and neural network models, the empirical results display that RSTM outperforms other predicting models. Although, in the text mining model unlike other well-defined problem domains, expert opinions on the crude oil markets can vary wildly [23].

III. Research Methodology

A. Isotonic Regression

The isotonic regression [24] finds a non-decreasing approximation of a function while minimizing the mean squared error on the training data. The algorithm sweeps through the data and adjusts the estimate to the best possible fit with constraints. Sometimes it also needs to modify previous points to make sure the new estimate does not violate the constraints. The benefit of such a model is that it does not assume any form for the target function such as linearity [25].

B. Support vector regression

Support Vector Machines (SVM) are supervised learning models used for classification and regression analysis. It offers one of the most robust and accurate methods among all well-known algorithms [26]. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. Support Vector Regression (SVR) is an SVM algorithm to handle nonlinear prediction [27]. SMOreg is an iterative optimization algorithm proposed by Smola and Schölkopf [28] for using SVR regression. SMOreg uses constraints structural risk minimization as the model and has the good ability to model regression, prediction with non-linear data. SVM generalization performance depends on a good setting of their parameters. We used RBF as a kernel function and C =1, which indicates the complexity.

C. K Star

K Star (K*) is an instanced based classifier [29]. A new data instance is classified by comparing it to the stored examples in order to find the most similar ones. This approach is also called nearest neighbor classification and the main advantage of this approach is that arbitrary complex structures in the data can be captured and training and retraining this model is fast.

D. Instance Based Learning

Instance-based learning (IBL) algorithms are derived from the nearest neighbor machine learning philosophy. IBK is the number of nearest neighbors (k) can be set manually or determined automatically. Each unseen instance is always compared with existing ones using a distance metric. Instance-based algorithms have numerous advantages one benefit of this approach is its simplicity[30]. We selected a default value of K=1 based on cross-validation.

E. Extra-Tree

The Extra-Trees algorithm constructs an ensemble of unpruned decision or regression trees. At each node number of attributes were selected randomly and splitting a node with minimum sample size. It is generated numerous times with the original learning sample to produce an ensemble model. The predictions of the trees are combined to get the final prediction, by majority vote in classification problems and average in regression problems. The basic differences with other tree based ensemble methods are it uses the whole learning sample and splits nodes by choosing cut- points fully randomly [31].

F. REP-Tree

Reduced Error Pruning Tree (REPtree) is a fast decision tree learner. It builds a decision tree based on information gain or reducing the variance and prunes it using reduced-error pruning (REP) with back over fitting [32].

G. Neural Networks (NNs)

NNs are computer models constructed to mimic the functions of the human brain through parallel computation of several input vectors. NNs are composed of neurons distributed in the input, hidden, and output layers [33]. We used several types of supervised networks such as Feed forward neural network (FFN), Radial basis function (RBF), and dynamic Recurrent network (RCN). FFN networks are most frequently used for prediction and pattern recognition. RBF provides an alternative, fast method for designing nonlinear feed-forward networks. Dynamic networks use memory and recurrent feedback connections to recognize spatial and temporal patterns in data. They are commonly used for time-series prediction, nonlinear dynamic system modeling, and control systems applications (Demuth et al., 2008, Demuth, Beale, 2000).

1) Feed forward Neural Networks (FFN)

Backpropagation [34] method is a supervised learning scheme and the most popular technique in multilayer networks when a set of input produces its own actual output and then compare it with the target value by calculating the error, after that error is fed back through the network. The weights of each connection are adjusted to reduce the error in several ways, such as gradient descent etc. until sufficient performance is achieved. To improve the generalization, there are several learning methods such as Levenberg – Marquardt (LM), Bayesian regularization (BR) and BFGS quasi-Newton (BFG-QN) back propagation algorithm [35]. In addition, each neuron in a particular layer is connected with all neurons in the next layer. The connection between the i^{th} and j^{th} neuron (in a different layer) is characterized by the weight coefficient ωij and the i^{th} neuron itself is characterized by the threshold coefficient υi (Figure 1).The weight coefficient reflects the degree of importance of the given connection in the neural network. The output value of the i^{th} neuron xi is determined by Eqs. (1) and (2)):

$$x_i = f(\xi_i) \tag{1}$$

$$\xi_i = \upsilon_i + \sum_{j=1}^{N} \omega_{ij} x_j \tag{2}$$

Where N is the neurons' number, ξi is the potential of the ith neuron and function f (ξi) is the so-called transfer function. The supervised adaptation process varies the threshold coefficients υ_i and weight coefficients ωij to minimize the sum of the squared differences between the computed and required output values. This is accomplished by minimization of the objective function E, given in equation (3):

$$E = \sum_{O} \frac{1}{2}(x_O - x_d)^2 \tag{3}$$

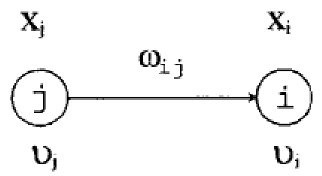

Figure 1. Connection between two neurons i and j

Where xO, and xd, are vectors composed of the computed and desired output neurons and summation runs over all output neurons O.

2) Recurrent Neural Network (RCN)

RCN is the state of the art in nonlinear time series prediction, system identification, and temporal pattern classification. As the output of the network at time t is used along with a new input to compute the output of the network at time t +1, the response of the network is dynamic [36] .

$$x_i(t) = \upsilon_i(t) + \sum_{j=1}^{N} \omega_{ij} x_j(t-1) \tag{4}$$

3) Radial Basis Function (RBF)

Radial Basis Function (RBF) [36] is an artificial neural network that uses radial basis functions as activation functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters. RBF is successful in numerous fields especially for system control, time series and prediction.

IV. Data set and Experimental Environment

A. Dataset Description

The dataset for experiments are obtained cooperative by Faculty of Management and Economic Sciences, Sousse University, Tunisia. It consists of 3337 records as instances and 14 variables as attributes to predict the West Taxes Intermediate (WTI) as output. The data set was taken from different sources such as [37, 38]. Attributes are listed as below:

- **Date (DT):** The daily data from 4 January 1999 to 10 October 2012. Dates are converted to numeric form when the input file is read.
- **West Texas Intermediate (WTI):** It is the most famous benchmark [9], and plays an important role as a reference

point to determine the price, and it constitutes a crucial factor in the configuration of prices of all other commodities [39].

- **Federal Fund rate (FFR):** One of the most influential interest rates in the U.S. economy, because it effects on monetary and financial conditions, which in turn have an impact on fundamental aspects of the broad economy including employment, growth and inflation [40].
- **Volatility Implied Equity Index (VIX):** Measures the contribution of the instability of the market.
- **The regional Standard and Poor's equity index (SPX):** Represent the market performance.
- **New York Harbor conventional gasoline spot prices (GPNY):** As example to assesses oil products.
- **US Gulf Coast conventional gasoline spot prices(GPUS):** As example to assesses oil products.
- **New York Harbor No. 2 heating oil spot price (HP):** As indication of seasonality in the energy market.
- **Future contracts 1 (FC1):** For WTI to maturity traded on NYMEX
- **Future contracts 2 (FC2):** For WTI to maturity traded on NYMEX
- **Future contracts 3 (FC3):** For WTI to maturity traded on NYMEX
- **Future contracts 4 (FC4):** For WTI to maturity traded on NYMEX.
- **Exchange rate (ER):** The price of oil and exchange rates of other currencies against the U.S. Dollar price.
- **Gold prices (GP):** Gold is that less volatile than crude oil and could reflect the real trend in the commodity market rather than the noise and gold used as the results of investors hedge against inflation caused by the oil price shock [41].

B. Data preprocessing

Before constructing a model we selected several aspects of initial preparation of data. Feature selection, normalization and data partition are used for preparation the inputs. It is worth mentioning that these steps are often used when designing any model in this research. We implemented first feature selection methods which is defined as a process of selecting a subset of features, d, out of the larger set of D features, which maximize the classification or prediction performance of a given procedure over all possible subset data. The second method is normalization, which shifts the instance values in specific and obviously means to represent information contained within the data and the data set [42]. Finally divided the dataset to groups according to deferent percentages of training and testing. The detail for each approach in the Sections below.

1) Feature selection methods

We formulated 10 different sub datasets, which were derived from the original dataset after implementing the several attribute selection algorithms. For instance SBDS1 and SBDS2 are as a result of Correlation based Feature Selection (CFS) algorithm by evaluating the value of a group of attributes by concerning the individual predictive ability of each feature as well with the possibility of redundancy among the features with several search methods such as best-first, which keeps a list of all attribute subsets evaluated so far, sorted in order of the performance measure. We used Forward selection, where we start with no attributes and add them one at a time and Backward, where we start with all the attributes and delete each one at a time, stops when the addition/deletion of any residual attributes results in a decrease in evaluation. In a case of one, begin with all the attributes or with none of them and this called bidirectional search method [43]. In SBDS3 and SBDS4, we utilized Genetic algorithm, which is based on search processes on the principle of natural selection [43]. $SBDS_5$ is formulated after a Random search in the space of attribute subsets. Random search starts from a random point and reports the best subset found. If a start set is supplied, Random searches randomly for subsets, which is useful or better than the start point with the same or fewer attributes. [44].We performed forward selections with a limited number of k attributes, based on the ranking using training data to decide, which attribute is added in each iteration of forward selection, and the test data is only used to evaluate the "best", P best subsets of a particular size. To determine the "optimal" subset size, we average the P scores on the test data for each subset size, and choose the size with the highest average. Then, a final forward selection is performed on the complete data set to find a subset of that optimal size and SBDS6 is created. When we used Classifier subset evaluator algorithm, we get SBDS7 by evaluating attribute subsets on training data or a separate hold out testing set using Support vector regression to estimate the 'merit' of a set of attributes with genetic search method. We get SBDS8 using Relief attribute evaluation algorithm, which evaluates the quality of attributes according to the value of the given attribute for the near instance to each other and different predicted (class) value [45]. We used ranker as search method, which Ranked the list of attributes based on individual evaluation of each attribute [46]. SBDS9and SBDS10 used wrapper algorithm, which evaluate attribute sets by using SMOreg algorithm. It is called wrapper because the learning algorithm is wrapped into a selection task [43]. We implemented the best-first search method in two directions: forward and backward respectively. Table 1 illustrates the categories and attributes for each algorithm.

2) Normalization

Most models work well with normalized data sets the data were normalized using Eq. (5) by scaling the instance to the range between -1 and 1 to improve prediction accuracy and CPU processing time [47] .

$$n_o = \frac{k_i - x_{min}}{p_{max} - x_{min}} \tag{5}$$

Where n_0 = normalized dataset k_i = raw dataset, x_{min} = minimum value of the dataset and p_{max} = maximum value of the dataset.

3) Data Partition

There are various alternatives to recognize the training and testing split process such as cross-validation, bootstrap and holdout [43]. According to holdout method, we divided dataset randomly into two parts, one half of training and the other half for testing. It is common to hold out one-third of the data for testing and use the remaining two-thirds for training [43]. However, several researchers achieved good results with other divisions, for example Lai, et al. [48] created their model using 60% for training and 40% for testing while Yu, et al. [49] utilized 80% for training and 20% for testing. We investigated the effect of training and testing data by randomly splitting them as shown in Table 2. We used several percentages to increase the opportunities for achieving better results. In the literature, there are also some studies conducted by using such divisions for training and test data [50] .

Table 1. Attribute selection methods and their features

Sub Dataset	Attributes evaluator	Search method	Attributes
$SBDS_1$	Correlation based Feature Selection subset evaluator	Best-first-Forward	WTI,SPX,FG1
$SBDS_2$	Correlation based Feature Selection subset evaluator	Best-first- Backward	DT,VIX,WTI,SPX,GPNY GPUS,HP,ER,FC1,FC2,FC3,FC4
$SBDS_3$	Correlation based Feature Selection subset evaluator	Genetic	VIX,WTI,GPNY, ER, FC1
$SBDS_4$	Correlation based Feature Selection subset evaluator	Genetic	WTI,GPNY,FC1
$SBDS_5$	Correlation based Feature Selection subset evaluator	Random	WTI,SPX,ER, FC1
$SBDS_6$	Correlation based Feature Selection subset evaluator	Subset Size Forward Selection	VIX,WTI,GPNY, FC1
$SBDS_7$	Classifier subset evaluator	Genetic- SMOreg	VIX,WTI,SPX, GPNY, ER, FC1, FC2
$SBDS_8$	Relief attribute evaluation	Ranker	WTI,FC1,FC2, FC3, FC4, VIX, GPUS, HP,GP,FFR,SPX,DT,ER
$SBDS_9$	Wrapper subset evaluator (SMOreg)	Best-first- Forward	WTI,GPUS
$SBDS_{10}$	Wrapper subset evaluator (SMOreg)	Best-first- Backward	WTI,FC1

Table 2. Training and testing percentages

Training	Testing	Label
90%	10%	(A)
80%	20%	(B)
70%	30%	(C)
60%	40%	(D)

V. Experimental Results

The purpose of this Section is to measure the performance of direct prediction models. On the one hand, we used ten sub datasets (SBSD1, SBDS2, SBDS3, SBDS4, SBDS5, SBDS6, SBDS7, SBDS8, SBDS9 and SBDS10) which is derived from the original dataset by using several attribute selection algorithms mentioned in Table 1 and on the other hand we used four groups (A-B-C-D), which contain different training and testing percentages as displayed in Table 2. It is worth mentioning that we repeated the training and testing experiments ten times with different random sample for each sub dataset to guarantee that the full dataset represented in the training and testing sets in the correct way and the error rates on the different iterations are averaged to yield an overall error rate. To judge the prediction performances and evaluate the accuracy of prediction, there are two basic criteria: the Mean Absolute Error (MAE) and Root Mean Square error (RMSE). The smaller the value of the evaluation indexes, the higher the performance of the algorithm. Willmott and Matsuura [51] indicated that MAE is a more natural measure of average error, it is unambiguous and comparisons of average model performance error should be based on MAE. e.g. [52], [53] and [54].

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \tilde{y}_i| \quad (6)$$

RMSE is a widely used measure to calculate differences between the values predicted by a model or a predictor and the values actually observed e.g. [52],[49], and [55]. RMSE is defined by the formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2} \quad (7)$$

To simplify the extensive list of experiments, we classified the experiments of direct prediction models in two groups as follows:

A. *First Phase Experiments and Results*

In this Section, we implemented six direct algorithms, namely Isotonic Regression, SMOreg, Kstar, IBK, ExtraTree and REPTree [27] . Figure 2 shows the performance of six algorithms in order to determine best approaches. Table 3 reports the empirical results illustrating MAE and Table 4 presents the RMSE for the six algorithms.

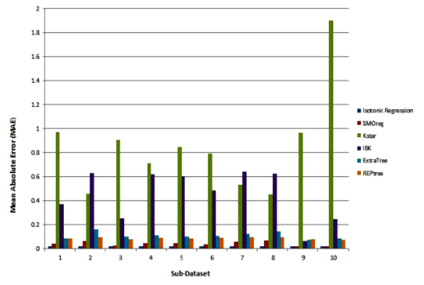

Figure 2. MAE for six prediction models with 10 sub-data set

Table 3. MAE for first phase experiment

Prediction Model	Data	SBDS$_1$	SBDS$_2$	SBDS$_3$	SBDS$_4$	SBDS$_5$	SBDS$_6$	SBDS$_7$	SBDS$_8$	SBDS$_9$	SBDS$_{10}$
Isotonic Regressin	A	**2.22E-02**	**2.22E-02**	**2.20E-02**	**2.22E-02**	**2.22E-02**	**2.22E-02**	**2.22E-02**	**2.22E-02**	**2.22E-02**	**2.22E-02**
	B	2.42E-02	2.42E-02	2.40E-02	2.42E-02	2.42E-02	2.42E-02	2.42E-02	2.42E-02	2.42E-02	2.42E-02
	C	2.78E-02	2.78E-02	2.80E-02	2.78E-02	2.78E-02	2.78E-02	2.78E-02	2.78E-02	2.78E-02	2.78E-02
	D	3.25E-02	3.25E-02	3.30E-02	3.25E-02	3.25E-02	3.25E-02	3.25E-02	3.25E-02	3.25E-02	3.25E-02
SMOreg	A	4.40E-02	6.60E-02	2.86E-02	4.94E-02	4.85E-02	3.77E-02	5.73E-02	6.79E-02	2.23E-02	2.32E-02
	B	3.93E-02	7.01E-02	3.50E-02	4.77E-02	4.77E-02	3.84E-02	5.98E-02	7.06E-02	2.63E-02	2.44E-02
	C	4.07E-02	6.99E-02	3.99E-02	5.20E-02	4.64E-02	3.68E-02	6.31E-02	7.32E-02	2.86E-02	2.65E-02
	D	4.44E-02	6.99E-02	3.90E-02	5.16E-02	5.58E-02	4.09E-02	6.31E-02	7.52E-02	3.10E-02	**2.21E-02**
Kstar	A	9.72E-01	4.56E-01	9.08E-01	7.10E-01	8.44E-01	7.94E-01	5.35E-01	**4.55E-01**	9.65E-01	1.90E+00
	B	9.76E-01	4.69E-01	9.01E-01	7.11E-01	8.53E-01	7.92E-01	5.47E-01	4.74E-01	9.60E-01	1.90E+00
	C	9.88E-01	4.87E-01	9.00E-01	7.23E-01	8.61E-01	7.98E-01	5.62E-01	4.91E-01	9.73E-01	1.91E+00
	D	1.00E+00	5.16E-01	9.13E-01	7.39E-01	8.70E-01	8.11E-01	5.87E-01	5.20E-01	9.83E-01	1.93E+00
IBk	A	3.74E-01	6.32E-01	2.52E-01	6.23E-01	6.04E-01	4.83E-01	6.42E-01	6.27E-01	**6.66E-02**	2.50E-01
	B	3.96E-01	6.62E-01	2.67E-01	6.48E-01	6.29E-01	5.00E-01	6.72E-01	6.56E-01	6.91E-02	2.65E-01
	C	4.18E-01	6.91E-01	2.82E-01	6.77E-01	6.55E-01	5.24E-01	7.02E-01	6.87E-01	7.35E-02	2.81E-01
	D	4.47E-01	7.31E-01	3.05E-01	7.19E-01	6.91E-01	5.57E-01	7.45E-01	7.32E-01	8.11E-02	3.07E-01
ExtraTree	A	8.32E-02	1.60E-01	1.01E-01	1.13E-01	1.00E-01	1.07E-01	1.23E-01	1.46E-01	**7.73E-02**	8.51E-02
	B	8.67E-02	1.76E-01	1.09E-01	1.18E-01	1.10E-01	1.23E-01	1.32E-01	1.62E-01	8.29E-02	1.00E-01
	C	1.04E-01	1.76E-01	1.22E-01	1.19E-01	1.09E-01	1.18E-01	1.54E-01	1.68E-01	9.40E-02	1.09E-01
	D	1.19E-01	1.95E-01	1.32E-01	1.49E-01	1.31E-01	1.45E-01	1.73E-01	1.90E-01	1.05E-01	1.31E-01
REPree	A	8.32E-02	9.70E-02	8.30E-02	9.35E-02	8.42E-02	9.24E-02	9.57E-02	9.45E-02	8.14E-02	**7.54E-02**
	B	9.67E-02	1.14E-01	9.70E-02	1.04E-01	9.76E-02	1.03E-01	1.09E-01	1.11E-01	9.39E-02	8.61E-02
	C	1.11E-01	1.30E-01	1.12E-01	1.19E-01	1.12E-01	1.18E-01	1.24E-01	1.30E-01	1.08E-01	1.01E-01
	D	1.27E-01	1.50E-01	1.26E-01	1.39E-01	1.28E-01	1.38E-01	1.43E-01	1.46E-01	1.23E-01	1.14E-01

Table 4. RMSE for first phase experiment

Prediction Model	Data	SBDS$_1$	SBDS$_2$	SBDS$_3$	SBDS$_4$	SBDS$_5$	SBDS$_6$	SBDS$_7$	SBDS$_8$	SBDS$_9$	SBDS$_{10}$
Isotonic Regression	A	**5.27E-02**	**5.27E-02**	**5.27E-02**	**5.27E-02**	**5.27E-02**	**5.27E-02**	**5.27E-02**	**5.27E-02**	**5.27E-02**	**5.27E-02**
	B	5.34E-02	5.34E-02	5.34E-02	5.34E-02	5.34E-02	5.34E-02	5.34E-02	5.34E-02	5.34E-02	5.34E-02
	C	6.16E-02	6.16E-02	6.16E-02	6.16E-02	6.16E-02	6.16E-02	6.16E-02	6.16E-02	6.16E-02	6.16E-02
	D	7.12E-02	7.12E-02	7.12E-02	7.12E-02	7.12E-02	7.12E-02	7.12E-02	7.12E-02	7.12E-02	7.12E-02
SMOreg	A	6.71E-02	9.19E-02	5.32E-02	7.00E-02	7.06E-02	6.10E-02	8.06E-02	9.27E-02	4.49E-02	3.10E-02
	B	6.99E-02	1.00E-01	6.62E-02	7.55E-02	7.54E-02	6.99E-02	8.93E-02	1.01E-01	5.63E-02	3.18E-02
	C	7.13E-02	1.00E-01	6.99E-02	7.95E-02	7.43E-02	6.91E-02	9.10E-02	1.04E-01	5.85E-02	3.53E-02
	D	7.97E-02	1.05E-01	7.47E-02	8.44E-02	8.75E-02	7.86E-02	9.48E-02	1.10E-01	6.63E-02	**3.03E-02**
Kstar	A	1.85E+00	6.74E-01	1.64E+00	1.04E+00	1.31E+00	1.21E+00	7.88E-01	**6.71E-01**	2.20E+00	3.35E+00
	B	1.89E+00	7.07E-01	1.66E+00	1.04E+00	1.35E+00	1.22E+00	8.05E-01	7.16E-01	2.25E+00	3.41E+00
	C	1.95E+00	7.45E-01	1.69E+00	1.06E+00	1.39E+00	1.24E+00	8.32E-01	7.52E-01	2.31E+00	3.47E+00
	D	1.96E+00	8.16E-01	1.71E+00	1.11E+00	1.39E+00	1.29E+00	9.02E-01	8.24E-01	2.31E+00	3.50E+00
IBk	A	5.51E-01	8.76E-01	3.82E-01	8.76E-01	8.17E-01	7.20E-01	8.90E-01	8.74E-01	**1.28E-01**	4.11E-01
	B	5.90E-01	9.50E-01	4.40E-01	9.33E-01	8.64E-01	7.65E-01	9.56E-01	9.37E-01	1.40E-01	4.38E-01
	C	6.14E-01	9.93E-01	4.56E-01	9.77E-01	9.00E-01	8.09E-01	1.01E+00	9.85E-01	1.43E-01	4.56E-01
	D	6.58E-01	1.07E+00	5.04E-01	1.05E+00	9.46E-01	8.79E-01	1.08E+00	1.07E+00	1.66E-01	4.93E-01
ExtraTree	A	2.15E-01	3.68E-01	2.75E-01	2.63E-01	2.41E-01	2.70E-01	2.75E-01	3.47E-01	**1.94E-01**	2.34E-01
	B	2.15E-01	3.68E-01	2.75E-01	2.63E-01	2.41E-01	2.70E-01	2.75E-01	3.47E-01	1.94E-01	2.34E-01
	C	2.68E-01	4.17E-01	3.05E-01	2.98E-01	2.80E-01	2.88E-01	4.11E-01	3.81E-01	2.19E-01	3.01E-01
	D	2.79E-01	4.66E-01	3.23E-01	3.77E-01	3.27E-01	3.90E-01	4.37E-01	4.49E-01	2.41E-01	3.67E-01
REPree	A	1.82E-01	2.29E-01	1.78E-01	2.38E-01	1.83E-01	2.38E-01	2.47E-01	2.24E-01	1.76E-01	**1.66E-01**
	B	2.36E-01	2.93E-01	2.36E-01	2.75E-01	2.37E-01	2.74E-01	2.82E-01	2.78E-01	2.32E-01	2.03E-01
	C	2.47E-01	3.06E-01	2.51E-01	2.81E-01	2.49E-01	2.79E-01	2.93E-01	3.19E-01	2.40E-01	2.35E-01
	D	3.01E-01	3.67E-01	2.96E-01	3.56E-01	3.03E-01	3.53E-01	3.61E-01	3.63E-01	2.93E-01	2.65E-01

Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 2 (2014) pp. 318-330

Prediction Model	Data	MAE	RMSE	Sub dataset	Time
Isotonic	A	2.220E-02	5.270E-02	All-SBDS	**00:00:04**
SMOreg	D	**2.210E-02**	**3.030E-02**	$SBDS_{10}$	00:05:14
Kstar	A	4.546E-01	6.706E-01	$SBDS_8$	00:25:19
IBk	A	6.660E-02	1.284E-01	$SBDS_9$	00:00.11
ExtraTree	A	7.730E-02	1.936E-01	$SBDS_9$	00:00.05
REPree	A	7.540E-02	1.661E-01	$SBDS_{10}$	00:00.08

Table 6. Summary of the results for direct prediction models

As illustrated in Figure 2, the K Star algorithm did not perform well for all the training and testing and for the different attributes. Time needed by the system to learn is another important criteria that may be considered in model selection [56], therefore in this Section we hold comparisons between prediction models based on time. According to Table 5 Isotonic regression and Extra Tree consumed less time with, while Kstar fails to achieve suitable time comparing with other models and SMOreg succeed to get less error but consumed long time so this consider drawback for it . The recorded time in this Table represents the time required to by each algorithm for all 10 sub data sets

Table 5. Time schedule for direct prediction models

Prediction Model	Data	Time
Isotonic Regression	**A**	**00:00:04**
	B	**00:00:04**
	C	**00:00:04**
	D	**00:00:04**
SMOreg	**A**	00:10:15
	B	00:09:24
	C	00:07:58
	D	00:05:14
Kstar	**A**	00:25:19
	B	00:43:27
	C	00:54:58
	D	01:03:02
IBK	**A**	00:00.11
	B	00:00.19
	C	00:00.22
	D	00::00:25
ExtraTree	**A**	00:00:05
	B	**00:00:04**
	C	**00:00:04**
	D	**00:00:04**
REPtree	**A**	00:00:08
	B	00:00:11
	C	00:00:14
	D	00:00:15

Summarizes the important results as follows: SMOreg, Isotonic Regression, REPtree and ExtraTree achieved less MAE 2.21E-02,2.22E-02,7.54E-02 and 7.73E-02 respectively and IBK accomplished good results 6.66E-02 with $SBDS_9$ only. Training and testing (A) which represent 90% training & 10 testing achieved best results with most algorithms also the best results focused in $SBDS_9$ & $SBDS_{10}$ and poor results were posted in ($SBDS_5$, $SBDS_7$ & $SBDS_8$) thus were removed from further experiments. SMOreg surpassed other algorithms in RMSE (3.03E-02). Table 6 display this results.We conclude from above discussions that Kstar was not appropriate to solve our problem.

B. *Second Phase Experiments and Results*

Numerous important characteristics of neural networks make them proper and valuable for data mining and machine learning so the objective of this Section is to provide a variety of the training and testing percentages with a set of different inputs using several kinds of neural networks to get high accuracy for the model. Neural network experiments are accomplished in MATLAB [57]. Neural networks with one and sometimes two hidden layers are widely used, for the large majority of problems and have performed very well (Panchal et al., 2011). Increasing the number of hidden layers are extremely hard to train, increases computation time and may lead to over-fitting which leads to poor out-of-sample predicting performance for this reason we used one hidden layer in this work. One of the most important characteristics of a network is the number of neurons in the hidden layer (s). If an insufficient number of neurons are used, the network will be unable to model complex data, and the resulting fit will be poor. Despite its importance, there is no formula for selecting the optimum number of hidden neurons. Therefore, scholars depend on experimentation. We used 40-45-50-55-60 neurons in the hidden layer based on trial and error approach. In most cases, the literature suggests the use of a trial-and-error approach to configuring network parameters, where the performance goal is set by the user. For instance, (Rene et al., 2013) used trial-and-error approach to determine network parameters. Selection of the training algorithm, which is suitable for our problem, depends on many factors such as the complexity of the problem and the number of inputs and others (Demuth et al., 2008). We used the Levenberg-Marquardt (LM), Bayesian regularization (BR) and BFGS Quasi-Newton (BFG-QN) algorithms because they are commonly used for regression problems and is easy to compare with other algorithms. The transfer function we applied is tan-sigmoidal for the hidden layer and pure linear function in the output layer, the maximum number of epochs is set to 1000 and the training goal is set to 0.

1) *Feed Forward Neural Network (FFN)*

According to Table 7, FFN utilized the 7 sub-datasets, which were selected as the best sub-dataset based on previous experiments. The best results were obtained when using the Bayesian regulation (BR) back-propagation method with 80% training and 20% testing , and sub-dataset1 ($SBDS_1$) achieved a MAE= 3.843E-05 with 90% training and 10% testing using 45 neurons. Figure 3 shows a comparison between the training algorithms for 7 sub-datasets and 4 groups of training and testing illustrating the superiority of BR. We measured the performance using MAE and RMSE.

2) *Recurrent Neural Network (RCN)*

We implemented RCN using one hidden layer with 10 neurons and used three training algorithms: Levenberg –Marquardt (LM), Bayesian regularization (BR) and BFGS Quasi-Newton (BFG-QN). Bayesian regularization method outperformed other algorithms by 51.85%. It is noted from

Table 8 for all the sub-datasets (80% training and 20% testing) is the best (shaded area), except in sub-dataset ($SBDS_6$) 90% training and 10% testing is the best. On the other hand, the lowest value of

Table 7. Performance of FFN

Sub-datasets	Data	Mean Absolute Error			Hidden layer neurons
		LM	BR	BFG-QN	
$SBDS_1$	**A**	1.48352E-03	**3.84294E-05**	8.00000E-04	45
	B	4.81400E-03	**1.35000E-04**	4.00000E-04	45
	C	5.77900E-03	**3.55000E-04**	1.50000E-03	45
	D	8.15200E-03	**4.38000E-04**	6.30000E-03	40
$SBDS_2$	A	9.17000E-04	2.83700E-03	**9.00000E-04**	40
	B	4.48000E-04	2.18100E-03	**4.00000E-04**	40
	C	2.74700E-03	**1.02200E-03**	1.90000E-03	45
	D	3.15700E-03	**7.69000E-04**	1.00000E-03	60
$SBDS_3$	A	1.83600E-03	**1.26594E-04**	1.10000E-03	50
	B	3.00400E-03	**1.24897E-04**	9.00000E-04	50
	C	1.21500E-02	5.31100E-03	**3.10000E-03**	45
	D	1.69940E-02	4.06200E-03	**1.80000E-03**	45
$SBDS_4$	**A**	5.58850E-02	**5.74155E-05**	7.80000E-03	50
	B	2.86700E-03	**6.45000E-05**	1.60000E-03	50
	C	2.48740E-02	**3.04484E-04**	7.70000E-03	50
	D	1.67079E-02	**1.94000E-04**	2.40000E-03	50
$SBDS_6$	A	3.50300E-03	**9.65000E-04**	2.00000E-03	55
	B	1.40900E-03	**6.80000E-05**	8.00000E-04	60
	C	2.19900E-02	**3.73327E-04**	2.80000E-03	40
	D	4.28700E-02	**2.10900E-03**	2.60000E-03	40
$SBDS_9$	A	1.44560E-02	**6.23000E-05**	5.70000E-03	40
	B	1.94854E-02	**6.04640E-05**	3.10000E-03	60
	C	3.23723E-01	**3.49000E-04**	4.95000E-02	55
	D	1.07220E-01	**1.62000E-04**	2.25000E-02	55
$SBDS_{10}$	A	5.69780E-02	**1.90200E-03**	1.92300E-01	40
	B	1.39500E-02	**1.97000E-04**	9.70000E-03	55
	C	9.65800E-02	**2.25700E-03**	3.45000E-02	40
	D	2.83030E-01	**4.50000E-04**	3.32000E-02	55

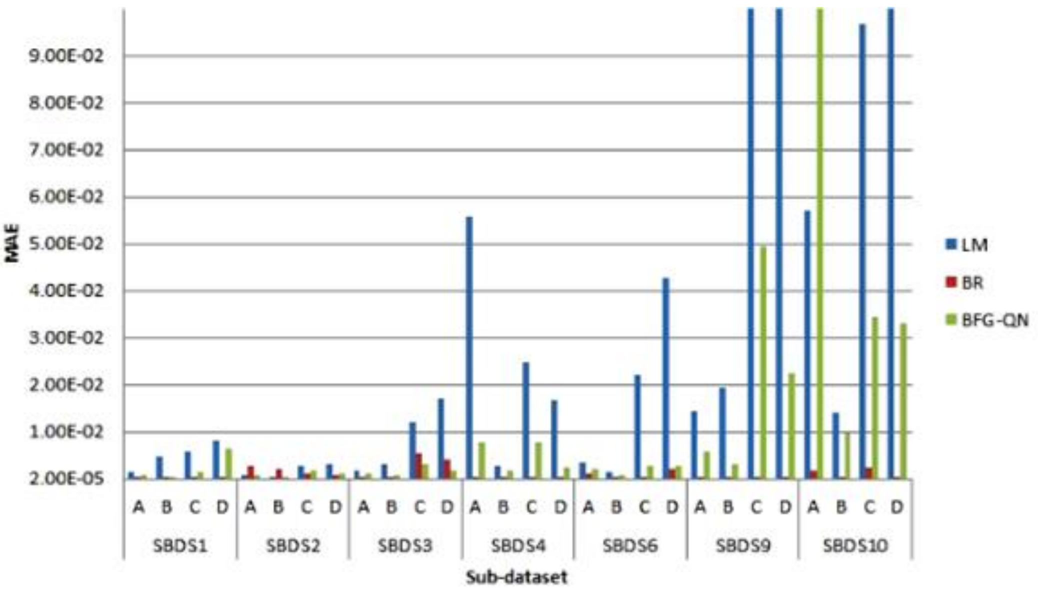

Figure 3. Comparison between training algorithms

Table 8. Performance of RCN

Sub-datasets	Data	Mean Absolute Error		
		LM	BR	BFG-QN
SBDS1	**A**	**1.14102E-03**	1.27500E-03	2.52400E-03
	B	1.17400E-03	2.48800E-03	**5.79000E-04**
	C	1.03650E-02	**6.98300E-03**	1.25780E-02
	D	1.29940E-02	**7.69800E-03**	1.29150E-02
SBDS2	A	4.60000E-04	**3.77000E-04**	2.18328E-02
	B	2.22000E-04	**1.74000E-04**	1.18390E-02
	C	**1.36700E-03**	1.44800E-03	7.03600E-02
	D	8.61000E-04	**3.32000E-04**	3.45920E-02
SBDS3	A	**5.22000E-04**	4.43900E-03	2.55500E-03
	B	3.57762E-04	**1.05000E-04**	6.40100E-03
	C	4.21100E-03	**2.82000E-04**	4.10660E-02
	D	6.82000E-04	**1.68000E-04**	1.67200E-02
SBDS4	**A**	7.68000E-04	**4.80285E-05**	1.58640E-02
	B	5.20102E-05	**3.94799E-05**	7.16800E-03
	C	4.73000E-04	2.17658E-04	**2.00530E-02**
	D	2.08400E-03	**1.81824E-04**	6.17200E-03
SBDS6	A	**3.94114E-05**	9.52860E-05	4.38700E-03
	B	1.16000E-04	**1.12000E-04**	4.13600E-03
	C	**4.41680E-04**	3.77708E-01	1.42909E-01
	D	4.95000E-04	**3.62000E-04**	9.70200E-03
SBDS9	A	**1.83283E-04**	1.51900E-03	3.33800E-03
	B	**1.72000E-04**	1.89800E-03	1.58400E-03
	C	**1.82716E-03**	6.89000E-03	6.86900E-03
	D	**9.69000E-04**	5.59500E-03	4.03800E-03
SBDS10	A	5.80000E-04	**2.48000E-04**	1.01690E-02
	B	**1.47000E-04**	2.19000E-04	2.20500E-03
	C	1.30400E-03	**1.01200E-03**	4.03600E-03
	D	**4.30824E-04**	6.10500E-03	1.03750E-02

the error is 3.941 E-05 when using 90% training and 10% testing with sub-dataset ($SBDS_6$).

3) Radial Basis Function Network (RBF)

We constructed the network until it reached a maximum number of neurons or the sum-squared error falls beneath an error goal. Table 9 shows the results obtained using the seven sub-datasets and different number of neurons: 40, 45, 50, 55 and 60. The shaded area indicates the best results when using 60 neurons in few sub-datasets then followed by 55 neurons. According to the percentage of training and testing sub-dataset ($SBDS_1$– $SBDS_3$-$SBDS_4$- $SBSD_9$ & $SBDS_{10}$) achieved the best results with 80% training and 20% testing. The best results over all sub-datasets is an MAE of 2.206 E-05 in SBDS9 with 80% training & 20 % testing using 45 neurons.

Measure performance by RMSE is depicted in Table 10 for best results for each NN algorithms. In order to investigate the effect of the algorithms we calculated the average of the training period for FFN, RCN, and RB when using BR as a training algorithm. Based on the results in Table 11, the training time depends on the sub dataset size for example SBDS2 include 13 features consumes more time than other sub-datasets as well as training rate and its impact, decrease the training ratio leads to increased speed, therefore always D is better than A. We observed from the Table 11 that RBF is characterized by it speed followed by RCN and finally FFN.

VI. A Comparison Analysis of Direct Prediction Models

From the results presented, we can draw the following comparisons: Based on experiments in Section 5.1.1 SMOreg outperformed other algorithms but it suffers from the consumed time, Kstar gets high error. We compared the results of three different types of neural networks as shown in Figure 4 using RMSE and observed that the RBF network outperformed other methods in obtaining the lowest error (MAE= 2.206 E-05 & RMSE= 1.291E-03). Also, the data set using training 80% and testing 20% accomplished the best results for RCN and RBF neural network methods. In the FFN and RBF networks, the best results were obtained when using 45 neurons and RBF proved its superiority. RBF networks outperformed again in the time factor, as it was faster than feed-forward and recurrent neural networks. Table 12 shows this comparison.

Table 9. Performance of RBF

Sub-datasets	Data	Mean Absolute Error based on the number of neurons				
		40	45	50	55	60
SBDS$_1$	A	1.03146E-04	9.34926E-05	1.00340E-04	5.70070E-05	5.08729E-05
	B	1.53000E-04	1.48000E-04	1.09000E-04	2.58000E-04	**2.40010E-05**
	C	4.84160E-05	5.01626E-05	5.02828E-05	5.02436E-05	4.98570E-05
	D	**3.88153E-05**	**3.81595E-05**	**3.81548E-05**	**3.81543E-05**	3.81548E-05
SBDS$_2$	A	5.05680E-03	**2.70885E-03**	**1.53678E-03**	**1.33255E-03**	**1.36009E-03**
	B	**4.09600E-03**	3.23000E-03	3.02800E-03	1.85800E-03	1.68000E-03
	C	6.13000E-03	6.04800E-03	5.16800E-03	4.47700E-03	2.85700E-03
	D	8.42200E-03	8.58200E-03	7.28400E-03	5.07200E-03	4.30200E-03
SBDS$_3$	A	**2.97000E-04**	**2.93000E-04**	**2.91000E-04**	**2.86000E-04**	2.75000E-04
	B	7.31000E-04	7.51000E-04	7.21000E-04	3.81000E-04	**2.24000E-04**
	C	1.50700E-03	6.52000E-04	5.11000E-04	9.74410E-04	6.96864E-04
	D	1.60300E-03	1.56300E-03	1.51400E-03	1.52400E-03	1.54100E-03
SBDS$_4$	A	2.11138E-04	2.09055E-04	2.09022E-04	2.09002E-04	2.08946E-04
	B	**6.08910E-05**	**6.23224E-05**	**6.23330E-05**	**6.23224E-05**	**6.35603E-05**
	C	1.25481E-04	1.10437E-04	1.13763E-04	1.16843E-04	1.16840E-04
	D	4.06000E-04	3.84000E-04	3.81000E-04	3.80000E-04	3.80000E-04
SBDS$_6$	A	1.00800E-03	8.17000E-04	3.54000E-04	1.60000E-04	4.34000E-04
	B	**1.53000E-04**	**1.03000E-04**	**1.26000E-04**	**1.04000E-04**	1.21483E-04
	C	4.14000E-04	4.01671E-04	4.01344E-04	4.28000E-04	2.34880E-04
	D	2.54000E-04	2.16000E-04	1.97000E-04	1.85000E-04	**8.74530E-05**
SBDS$_9$	A	9.04430E-02	9.04440E-02	9.04440E-02	9.04440E-02	9.04440E-02
	B	**2.21914E-05**	**2.20646E-05**	**2.21172E-05**	**2.21087E-05**	**2.21087E-05**
	C	4.51944E-05	4.51615E-05	4.49053E-05	4.48180E-05	4.47060E-05
	D	3.51792E-05	3.41134E-05	3.40980E-05	3.35330E-05	3.35330E-05
SBDS$_{10}$	A	2.34000E-04	2.85000E-04	2.76000E-04	2.72000E-04	2.76000E-04
	B	**4.03840E-05**	**4.03361E-05**	**3.98230E-05**	**3.98158E-05**	**3.98230E-05**
	C	5.37473E-05	5.34101E-05	5.34009E-05	5.36361E-05	5.36107E-05
	D	1.21000E-04	1.20000E-04	1.20000E-04	1.16000E-04	1.16000E-04

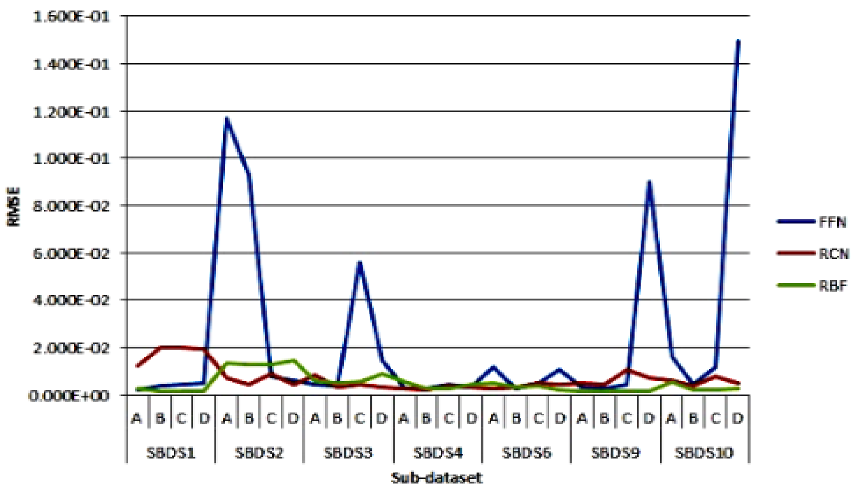

Figure 4. Comparison among 3 type of NNs

Table 10. RMSE for FFN, RCN & RBF

Sub-datasets	**Data**	**RMSE**		
		FFN	**RCN**	**RBF**
SBDS$_1$	**A**	**2.280E-03**	**1.237E-02**	2.613E-03
	B	3.606E-03	1.994E-02	1.521E-03
	C	4.494E-03	1.994E-02	1.677E-03
	D	4.669E-03	1.956E-02	**1.377E-03**
SBDS$_2$	A	1.170E-01	7.114E-03	1.337E-02
	B	9.274E-02	4.125E-03	**1.273E-02**
	C	7.629E-03	8.823E-03	1.275E-02
	D	**6.181E-03**	**4.065E-03**	1.463E-02
SBDS$_3$	A	4.123E-03	8.368E-03	6.071E-03
	B	3.464E-03	3.187E-03	**4.650E-03**
	C	5.568E-02	4.009E-03	5.396E-03
	D	**1.421E-02**	**2.889E-03**	8.677E-03
SBDS$_4$	**A**	2.775E-03	2.539E-03	5.295E-03
	B	**2.490E-03**	**1.951E-03**	**2.423E-03**
	C	4.164E-03	3.521E-03	2.508E-03
	D	3.114E-03	3.007E-03	4.344E-03
SBDS$_6$	A	1.138E-02	**2.300E-03**	4.632E-03
	B	**2.569E-03**	3.285E-03	3.166E-03
	C	4.615E-03	5.015E-03	3.657E-03
	D	1.024E-02	4.243E-03	**2.085E-03**
SBDS$_9$	A	2.898E-03	4.959E-03	1.563E-03
	B	**2.408E-03**	**4.078E-03**	**1.291E-03**
	C	4.461E-03	1.020E-02	1.596E-03
	D	9.000E-02	6.941E-03	1.459E-03
SBDS$_{10}$	A	1.597E-02	5.769E-03	5.598E-03
	B	**4.359E-03**	**3.764E-03**	1.960E-03
	C	1.134E-02	7.592E-03	**1.744E-03**
	D	1.493E-01	4.628E-03	2.398E-03

Table 11. Comparison between NNs based on time

Sub-datasets	**Data**	**TIME**		
		FFN	**RCN**	**RBF**
SBDS$_1$	**A**	00:03:20	00.04.54	00:00:20
	B	00:02:53	00.04.04	00:00:17
	C	00:01:50	00.03.23	00:00:14
	D	**00:01:24**	**00.03.21**	**00:00:13**
SBDS$_2$	A	00:13:33	00.06.57	00.00.34
	B	00:13:29	00.06.42	00.00.19
	C	00:12:52	00.05.27	00.00.17
	D	**00:11:40**	**00.04:35**	**00:00:16**
SBDS$_3$	A	00:03:53	00:05:00	00:00:28
	B	00:03:45	00:04:30	00:00:25
	C	00:04:15	00:03:53	00:00:21
	D	**00:02:48**	**00:03:15**	**00:00:20**
SBDS$_4$	**A**	00:03:32	00:04:19	00:00:27
	B	00:02:31	00:03:49	00:00:24
	C	00:02:18	00:03:30	00:00:20
	D	**00:02:14**	**00:03:11**	**00:00:19**
SBDS$_6$	A	00:03:39	00:04:28	00:00:28
	B	00:03:30	00:04:05	00:00:24
	C	00:03:14	00:03:06	00:00:21
	D	**00:02:39**	**00:03:04**	**00:00:19**
SBDS$_9$	A	00:01:58	00:04:38	00:00:28
	B	00:01:29	00:04:05	00:00:24
	C	00:00:47	00:03:43	00:00:21
	D	**00:00:22**	**00:03:10**	**00:00:19**
SBDS$_{10}$	A	00:02:01	00:04:49	00:00:26
	B	00:01:15	00:04:22	00:00:24
	C	00:01:38	00:03:09	00:00:21
	D	**00:00:32**	**00:02:46**	**00:00:19**

Table 12. Summary for the results which explain the comparison between NNs

Prediction Model	Data	MAE	RMSE	Sub-dataset	Time
FFN	A	3.84294E-05	2.280E-03	SBDS1	00:03:20
RCN	B	3.94114E-05	2.300E-03	SBDS6	00:04:05
RBF	B	**2.20646E-05**	**1.291E-03**	**SBDS9**	00:00:24

VII. Conclusions

In this research, simple machine-learning approaches were applied to predict the daily WTI price for every barrel of crude oil in USD. List of features used as the input factors were divided into ten sub-datasets resulting in numerous attribute selection algorithms and four data sets with different percentages of training and testing. Experiments start with six direct prediction models namely isotonic regression, SMOreg, Kstar, IBK, ExtraTree, REPTree followed by several types of NNs including FFN, RCN and RBF. This paper provides successful comparisons of simple prediction models, sub- datasets, and different group of training and testing data sets

References

[1] E.L. (2014). *Why the oil price is falling*. Available: http://www.economist.com/blogs/economist-explains/2014/12/economist-explains-4 [Accessed 23/2/2015]

[2] L. Kilian and D. P. Murphy, "The role of inventories and speculative trading in the global market for crude oil," *Journal of Applied Econometrics*, vol. 29, pp. 454-478, 2014.

[3] R. A. Lizardo and A. V. Mollick, "Oil price fluctuations and US dollar exchange rates," *Energy Economics*, vol. 32, pp. 399-408, 2010.

[4] C. Morana, "Oil price dynamics, macro-finance interactions and the role of financial speculation," *Journal of banking & finance*, vol. 37, pp. 206-226, 2013.

[5] X. Zhang, Q. Wu, and J. Zhang, "Crude oil price forecasting using fuzzy time series," in *Knowledge Acquisition and Modeling (KAM), 2010 3rd International Symposium on*, 2010, pp. 213-216.

[6] M. Sompui and W. Wongsinlatam, "Prediction Model for Crude Oil Price Using Artificial Neural Networks," *Applied Mathematical Sciences*, vol. 8, pp. 3953-3965, 2014.

[7] L. Yu, S. Wang, and K. Lai, "A rough-set-refined text mining approach for crude oil market tendency forecasting," *International Journal of Knowledge and Systems Sciences*, vol. 2, pp. 33-46, 2005.

[8] A.-S. Chen, M. T. Leung, and L.-H. Wang, "Application of polynomial projection ensembles to hedging crude oil commodity risk," *Expert Systems with Applications*, vol. 39, pp. 7864-7873, 2012.

[9] L. Yu, S. Wang, and K. K. Lai, "Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm," *Energy Economics*, vol. 30, pp. 2623-2635, 2008.

[10] H. G. Huntington, "Oil price forecasting in the 1980s: what went wrong?," *The Energy Journal*, pp. 1-22, 1994.

[11] B. Abramson and A. Finizza, "Probabilistic forecasts from probabilistic models: a case study in the oil market," *International Journal of forecasting*, vol. 11, pp. 63-72, 1995.

[12] G. Barone-Adesi, Bourgoin, F., Giannopoulos, K.: , "Don't look back. Risk " *Energy policy*, vol. 14, 1995.

[13] G. Barone-Adesi and F. Bourgoin, "K. Giannopoulos,(1998)." Don't look back"," *Risk*, vol. 11, pp. 100-103, 1998.

[14] C. Morana, "A semiparametric approach to short-term oil price forecasting," *Energy Economics*, vol. 23, pp. 325-338, 2001.

[15] A. S. Weigend, "Time series prediction: forecasting the future and understanding the past," *Santa Fe Institute Studies in the Sciences of Complexity*, 1994.

[16] E. M. Azoff, *Neural network time series forecasting of financial markets*, 1st ed.: John Wiley & Sons, Inc., 1994.

[17] H. Chiroma, S. Abdulkareem, A. ABUBAKAR, and J. U. MOHAMMED, "Computational Intelligence Techniques with Application to Crude Oil Price Projection: A literature Survey from 2001-2012," *Neural Network World*, vol. 23, pp. 523-551, 2013.

[18] I. Haidar, S. Kulkarni, and H. Pan, "Forecasting model for crude oil prices based on artificial neural networks," in *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008. International Conference on*, 2008, pp. 103-108.

[19] A. Alizadeh and K. Mafinezhad, "Monthly Brent oil price forecasting using artificial neural networks and a crisis index," in *Electronics and Information Engineering (ICEIE), 2010 International Conference On*, 2010, pp. V2-465-V2-468.

[20] J. Wang, W. Xu, X. Zhang, Y. Bao, Y. Pang, and S. Wang, "Data Mining Methods for Crude Oil Market Analysis and Forecast," *Data Mining in Public and Private Sectors: Organizational and Government Applications*, vol. 184, 2010.

[21] A. Khashman and N. I. Nwulu, "Intelligent prediction of crude oil price using Support Vector Machines," in *Applied Machine Intelligence and Informatics (SAMI), 2011 IEEE 9th International Symposium on*, 2011, pp. 165-169.

[22] H. Chiroma, S. Abdulkareem, A. I. Abubakar, and T. Herawan, "Kernel Functions for the Support Vector Machine: Comparing Performances on Crude Oil Price Data," in *Recent Advances on Soft Computing and Data Mining*, ed: Springer, 2014, pp. 273-281.

[23] H. Pan, I. Haidar, and S. Kulkarni, "Daily prediction of short-term trends of crude oil prices using neural networks exploiting multimarket dynamics," *Frontiers of Computer Science in China*, vol. 3, pp. 177-191, 2009.

[24] W. B. Wu, M. Woodroofe, and G. Mentz, "Isotonic regression: Another look at the changepoint problem," *Biometrika*, vol. 88, pp. 793-804, 2001.

[25] F. Pedregosa. (2013). *isotonic regression* Available: http://fa.bianp.net/blog/2013/isotonic-regression/ [Accessed 2/3/2014]

[26] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, *et al.*, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, pp. 1-37, 2008.

[27] L. A. Gabralla, R. Jammazi, and A. Abraham, "Oil price prediction using ensemble machine learning," in *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*, 2013, pp. 674-679.

[28] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, pp. 199-222, 2004.

[29] J. G. Cleary and L. E. Trigg, "K^*: An Instance-based Learner Using an Entropic Distance Measure," in *ICML*, 1995, pp. 108-114.

[30] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine learning*, vol. 6, pp. 37-66, 1991.

[31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, pp. 3-42, 2006.

[32] W. Mohamed, M. N. M. Salleh, and A. H. Omar, "A comparative study of Reduced Error Pruning method in decision tree algorithms," in *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on*, 2012, pp. 392-397.

[33] H. Chiroma, S. Abdulkareem, and A. Y. u. Gital, "An Intelligent Model Framework for Handling Effects of Uncertainty Events for Crude Oil Price Projection: Conceptual Paper," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2014.

[34] Y. Chauvin and D. E. Rumelhart, *Backpropagation: theory, architectures, and applications*: Psychology Press, 1995.

[35] H. Demuth, M. Beale, and M. Hagan, "Neural network toolbox™ 6," *User's guide*, 2008.

[36] A. Abraham, "Artificial neural networks," *Handbook of Measuring System Design*, vol. 0-470-02143-8.

, pp. 901-908, 2005.

[37] M. M. Inc. (2012). *Growing gold and silver producer in the Americas*. Available: http://www.mcewenmining.com/ [Accessed 15/5/2013]

[38] EIA. (2012). *PETROLEUM & OTHER LIQUIDS*. Available: http://www.eia.gov/petroleum/data.cfm [Accessed 23/4/2013]

[39] A. Alexandridis and E. Livanis, "Forecasting crude oil prices using wavelet neural networks," *Published in the proc. of 5th FSDET, Athens, Greece* vol. 8, 2008.

[40] investopedia. (2012). *Federal Funds Rate*. Available: http://www.investopedia.com/terms/f/federalfundsrate.asp [Accessed 7/8/2014]

[41] I. El-Sharif, D. Brown, B. Burton, B. Nixon, and A. Russell, "Evidence on the nature and extent of the relationship between oil prices and equity values in the UK," *Energy Economics*, vol. 27, pp. 819-830, 2005.

[42] S. Zhang, C. Zhang, and Q. Yang, "Data preparation for data mining," *Applied Artificial Intelligence*, vol. 17, pp. 375-381, 2003.

[43] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann, 2005.

[44] H. Liu and R. Setiono, "A probabilistic approach to feature selection-a filter solution," in *ICML*, 1996, pp. 319-327.

[45] M. Robnik-Šikonja and I. Kononenko, "An adaptation of Relief for attribute estimation in regression," in *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, 1997, pp. 296-304.

[46] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.

[47] O. Kaynar, I. Yilmaz, and F. Demirkoparan, "Forecasting of natural gas consumption with neural network and neuro fuzzy system," *Energy Education Science and Technology Part A–Energy Science and Research*, vol. 26, pp. 221-238, 2011.

[48] K. K. Lai, K. He, and J. Yen, "Modeling VaR in crude oil market: a multi scale nonlinear ensemble approach incorporating wavelet analysis and ANN," in *Computational Science–ICCS 2007*, ed: Springer, 2007, pp. 554-561.

[49] L. Yu, Y. Zhao, and L. Tang, "A compressed sensing based AI learning paradigm for crude oil price forecasting," *Energy Economics*, vol. 46, pp. 236-245, 2014.

[50] X. Zeng, "Machine Learning Approach for Crude Oil Price Prediction," Doctor of Philosophy Doctoral level ETD - final, Faculty of Engineering and Physical Sciences, School of Computer Science, The University of Manchester Manchester, UK, 2014.

[51] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, p. 79, 2005.

[52] A. Shabri and R. Samsudin, "Crude Oil Price Forecasting Based on Hybridizing Wavelet Multiple Linear Regression Model, Particle Swarm Optimization Techniques, and Principal Component Analysis," *The Scientific World Journal*, vol. 2014, 2014.

[53] H. Chiroma, S. Abdul-Kareem, A. Abubakar, A. M. Zeki, and M. J. Usman, "Orthogonal Wavelet Support Vector Machine for Predicting Crude Oil Prices," in *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, 2014, pp. 193-201.

[54] R. A. Ahmed and A. B. Shabri, "A Hybrid of EMD-SVM Based on Extreme Learning Machine for Crude Oil Price Forecasting," *Australian Journal of Basic and Applied Sciences*, vol. 8(15), pp. 341-351, 2014.

[55] D. Xu, Y. Zhang, C. Cheng, W. Xu, and L. Zhang, "A Neural Network-Based Ensemble Prediction Using PMRS and ECM," in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, 2014, pp. 1335-1343.

[56] C. Giraud-Carrier, "Beyond predictive accuracy: what?," in *Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation*, 1998, pp. 78-85.

[57] L. A. Gabralla, H. Mahersia, and A. Abraham, "Ensemble Neurocomputing Based Oil Price Prediction," in *Afro-European Conference for Industrial Advancement*, 2015, pp. 293-302.