Literature Review
CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, the literature is reviewed on forecasting using linear regression, multiple linear regression, Software Effort Estimation (SEE), ANN (Artificial Neural Network) and using SVR (Support Vector Regression).

2.2 FORECASTING USING LINEAR REGRESSION

Forest products such as paper, panels and sawn wood, appearing in a large number of end use products, affect daily lives. Furthermore, in several countries the forest sector contributes in a significant way to the general economic and social development. The increasing volume of trade and in particular the import of forest products into Europe, plays an important role also for global forest sector developments. A description of an econometric analysis of the forest sector in Europe and how the resulting models, together with assumptions regarding economic growth and price and cost developments, used to produce country specific projections of consumption, production and trade of wood products are studied (Jonsson, 2009). The effects of deflation and its impacts on net revenues and total costs in different productive lines of Mazandaran Wood and Paper Company are investigated (Majid Zadmirzaei and Limaei, 2013). Costs and net revenues data from financial balance sheet of 6 productive lines is collected over the period 2005 - 2010. Nominal data are converted to real data using consumer price index. t-test with significance level of 0.05 is used to determine the significant differences between nominal and real data using SPSS software. Regression analysis is used to predict the costs and net revenues models. The results have indicated that it is possible to predict costs and revenues with linear models.
Swanson and White (1997) took a model selection approach to the question of whether a class of adaptive prediction models like Artificial Neural Networks is useful for predicting future values of nine macroeconomic variables. A variety of out-of-sample forecast-based model selection criteria including forecast error measures and forecast direction accuracy is used. In order to compare the proposed predictions to professionally available survey predictions, a real-time forecasting procedure is implemented. One dimension of the approach is that the construction of a real-time economic data set which has the characteristic - which has the data available at time t do not contain any information which has been allowed to ‘leak’ in from future time periods, as often happens with fully revised macroeconomic data. It has also investigated the issue of appropriate window sizes for rolling-window-based prediction methods. Results have indicated that adaptive models often outperform a variety of non-adaptive models, as well as professional forecasters, when used to predict levels as well as the direction of change in various macroeconomic variables. Further, model selection based on an in-sample Schwarz Information Criterion (SIC) does not appear to be a reliable guide to out-of-sample performance, in the case of the variables considered in the proposed study. Thus, the in-sample SIC apparently fails to offer a convenient shortcut to true out-of-sample performance measures.

The study by Kayacan et al. (2012) is a primary econometric analysis to explore the factors explaining the changes in the industrial round wood demand in Turkey. The study has included demand forecasts based on the econometric models proposed herein. Two separate econometric models are constructed: one for national demand for domestically-produced sawlog and the other for national demand for domestically-produced non-sawlog industrial round wood. Models are originally designed in multiplicative form. The original models are then converted into the log-linear form so that the relevant coefficients of the regression equations would immediately reflect the elasticities. Estimation of the model parameters are
based on a panel data set of fifteen years (1995-2009) by twenty seven regional forest directorates in Turkey. In accordance with the maxim of less than the half of the 15 year period of data set, the demand forecasts are made for seven years beyond 2009. In view of the results, the explanatory power of the proposed models are arguably deemed satisfactory especially considering the lack of earlier studies of this scale and scope. This consequently increases the credibility of the demand projections. Notwithstanding signs of the estimated parameters of the models are for the most part congruent with those expected in light of the economic theory and practice, some intriguing results are obtained. Perhaps most notably, while the sign of the estimated price elasticity of sawlog demand occurred unexpectedly positive, the variation in sawlog demand is explained to a considerable extent by the variation in the price of imported sawlog. Also, notable is that the price of imported ‘fuel wood’ holds a positive relationship with the national demand for domestic non-sawlog industrial round wood, which is an expected cross elasticity since virtually all of the imported “fuel wood” is used as raw material for industry (e.g. chip and fibre board industry). Finally, both models suggest overall boost in demand; yet an upper bound of 4.5 million cubic meter for national demand for domestically-produced sawlog and of 15 million cubic meter for national demand for domestically-produced non- sawlog industrial round -wood can be expected by 2016.

Projections for US newsprint demand up to 2020 is studied. First, various specifications of the standard model used in forest product demand literature, called the classical model, are estimated using annual data from 1971–2000 (Hetemäki and Obersteiner, 2001). The results have indicated that structural change in the newsprint consumption pattern took place at the end of the 1980s. The classical model fails to explain and forecast the structural change. This finding has motivated the formulation of alternative models. Thus, a Bayesian model that allows industry experts prior knowledge about the future
demand for newsprint to be included in the projections is estimated. Also, an *ad hoc* model, in which newsprint demand is a function of changes in newspaper circulation, is used to compute projections. Finally, the forecasts of these models are evaluated along with some of the existing projections. Besides providing an outlook for US newsprint demand, the study contributes to the existing literature of long-term forest product demand by raising some methodological questions and by applying new models to compute projections.

A model of the supply side of the Ukrainian economy has been approached through estimation of aggregate production function in its simplest form, i.e. Cobb-Douglas (Kudina, 2000). Although drawbacks in data, brought by the transitional state of the economy and shortness of time series, made extrapolated and forecasted figures not as accurate has been aimed at the beginning, the final output seems to be rather encouraging and could be enhanced with the expansion of the historical period. Other way of model improvement is seen in further disaggregation. Despite ameliorating quality, a low reliability of indicators is going to persist since the problems of widespread non-monetary transactions and significant shadow activities cannot be resolved soon in Ukraine.

Average load forecasting errors for the holidays are much higher than those for weekdays. So far, many studies on the short-term load forecasting have been made to improve the prediction accuracy using various methods such as deterministic, stochastic, ANN and neural network-fuzzy methods. In order to reduce the load forecasting error of the 24 hourly loads for the holidays, the concept of fuzzy regression analysis is employed by Song et al. (2005) in the short-term load forecasting problem. According to the historical load data, the same type of holiday has showed a similar trend of load profile as in previous years. The fuzzy linear regression model is made from the load data of the previous three years and the coefficients of the model are found by solving the
mixed linear programming problem. The proposed algorithm shows good accuracy, and the average maximum percentage error is 3.57% in the load forecasting of the holidays for the years of 1996–1997.

2.3 FORECASTING USING MULTIPLE LINEAR REGRESSION

The influence of economic and demographic variables on the annual electricity consumption in Italy has been investigated with the intention to develop a long-term consumption forecasting model. The time period considered for the historical data is from 1970 to 2007. Different regression models are developed, using historical electricity consumption, Gross Domestic Product (GDP), GDP per capita and population. A first part of the study by Bianco et al. (2009) has considered the estimation of GDP, price and GDP per capita elasticities of domestic and non-domestic electricity consumption. The domestic and non-domestic short run price elasticities are found to be both approximately equal to −0.06, while long run elasticities are equal to −0.24 and −0.09, respectively. On the contrary, the elasticities of GDP and GDP per capita present higher values. In the second part of the study, different regression models, based on co-integrated or stationary data, are presented. Different statistical tests are employed to check the validity of the proposed models. A comparison with national forecasts, based on complex econometric models, such as Markal-Time, has been performed and it has shown that the developed regressions are congruent with the official projections, with deviations of ±1% for the best case and ±11% for the worst. These deviations are to be considered acceptable in relation to the time span taken into account.

A formulation for predictive learning called multiple regression models and theoretical approach on construction of the regression model have been described. The key information of the article is the mathematical formulation for the forecast linear equation that estimates the multiple regression models. Calculation of the quantitative value of dependent variable forecast under influence of independent variables is done. The proposed study has presented the retail sales forecasting
with multiple model estimation. It is one of the most important decisions a retailer can make with information obtained from the multiple regression. Checking model on the goodness of fit and statistical significance is explored. Finally, the quantitative value of retail sales forecast based on multiple regression models is calculated. Amral et al. (2007) have investigated the short term (up 24 hours) load forecasting of the demand for the South Sulewesi’s (Sulewesi Island – Indonesia) Power System, using a Multiple Linear Regression (MLR) method. After a brief analytical discussion of the technique, the usage of polynomial terms and the steps to compose the MLR model is explained. Report on implementation of MLR algorithm using commercially available tool such as Microsoft EXCELTM is also discussed. As a case study, historical data consisting of hourly load demand and temperatures of South Sulawesi electrical system is used, to forecast the short term load. The results are presented and analyzed; the potential for improvement using alternative methods is also discussed.

Multilayer neural network has been successfully applied to the time series forecasting. Steepest descend, a popular learning algorithm for back propagation network, converges slowly and has the difficulty in determining the network parameters. Man-Chung et al. (2000) have introduced conjugate gradient learning algorithm with restart procedure to overcome these problems. Also, the commonly used random weight initialization does not guarantee to generate a set of initial connection weights close to the optimal weights leading to slow convergence. MLR provides a better alternative for weight initialization. The daily trade data of the listed companies from Shanghai Stock Exchange is collected for technical analysis with Neural Networks. Two learning algorithms and two weight initializations are compared. The results find that neural network can model the time series satisfactorily, when learning algorithm and weight initialization are adopted. However, the proposed conjugate gradient with MLR weight initialization requires a lower computation cost and learns better than steepest decent with
random initialization. Timm and Wiesner (2003) have reviewed the notion of predictive precision, model fit, model specification and mean squared error of prediction for multiple linear regression models with both fixed (non-stochastic) and random (stochastic) independent variables. Gold is a precious yellow commodity once used as money. The demand for this commodity is on the rise. Objective of the study by Ismail et al. (2009) has been to develop a forecasting model for predicting gold prices based on economic factors such as inflation, currency price movements and others. The most appropriate approach to the understanding of gold prices is the MLR model. MLR is a study on the relationship between a single dependent variable and one or more independent variables, as the case with gold price as the single dependent variable. The fitted model of MLR is used to predict the future gold prices. A naive model known as “forecast-1” is considered to be a benchmark model in order to evaluate the performance of the model. Variables such as Commodity Research Bureau future index (CRB); New York Stock Exchange (NYSE); Standard and Poor 500 (SPX); Treasury Bill (T-BILL) and US Dollar index (USDX) are considered to have influence on the prices. Parameter estimations for the MLR are carried out using SPSS with Mean Square Error (MSE) as the fitness function to determine the forecast accuracy. Two models are considered. The first model considered all possible independent variables. The model appeared to be useful for predicting the price of gold with 85.2% of sample variations in monthly gold prices explained by the model. The amount of variance explained is about 70% and the regression coefficients also provide a means of assessing the relative importance of individual variables in the overall prediction of gold price.

Medium-term forecasting is an important category of electric load forecasting that covers a time span of up to one year ahead. It suits outage and maintenance planning, as well as load switching operation. Abu-Shikhah et al. (2011) have proposed a new methodology that uses hourly daily loads to predict the next
year hourly loads and hence predict the peak loads expected to be reached in the coming year. The proposed technique is based on implementing multivariable regression on previous year’s hourly loads. Three regression models are investigated in the proposed research: the linear, the polynomial and the exponential power. The proposed models are applied to real loads of the Jordanian power system. Results obtained using the proposed methods showed that their performance is close and they outperform results obtained using the widely used exponential regression technique. Moreover, peak load prediction has about 90% accuracy using the proposed methodology. The methods are generic and simple and can be implemented to hourly loads of any power system. No extra information other than the hourly loads is required.

Efficient and accurate Short Term Load Forecasting (STLF) is vital for the economic operation of modern electrical power systems. Forecasting hourly electricity prices and loads have become one of the most essential tasks and bases for the decision making units. In the changing world conditions and new deregulated framework, producers and consumers are in need of short-term price and load forecasting to determine their bidding strategies. Accurate forecasting tools are necessary and crucial for both producers and consumers to maximize their profits and to maximize their utilities, respectively. The STLF implementation using Artificial Neural Network (ANN) and multiple linear regression computing technique, described approach has used the three-layered ANN paradigm with Levenberg-Marquardt back-propagation and gradient back-propagation algorithm. In the proposed project, the three-layered ANN paradigm has been used for forecasting the PJM region load values in America for the year of 1997. The data used are taken from Department of Electrics and Electronics at Middle East Technical University. The accuracy of this implementation is evaluated with the help of mean absolute error, mean absolute percentage error, mean-square-error and root-mean-square error by reporting the numerical results from a real-world case study.
Amongst the wealth of available machine learning algorithms for forecasting time series, linear regression has remained one of the most important and widely used methods, due to its simplicity and interpretability. A disadvantage, however, is that a linear regression model may often have higher error than models that are produced by more sophisticated techniques. Ristanoski et al. (2013) have investigated the use of a grouping based quadratic mean loss function for improving the performance of linear regression. In particular, segmenting the input time series into groups has been proposed and simultaneously optimizing both the average loss of each group and the variance of the loss between groups, over the entire series. The aim is to produce a linear model that has low overall error and is less sensitive to distribution changes in the time series and is more robust to outliers. The performance of the proposed method has experimentally been investigated and found that it can build models which are different from those produced by standard linear regression, whilst achieving significant reductions in prediction errors.

Short-term (0 – 36 h ahead) wind power forecast is a central issue for the correct management of a grid connected wind farm. A combination of physical and statistical treatments to post-process Numerical Weather Predictions (NWP) outputs is needed for successful short-term wind power forecasts. One of the most promising and effective approaches for statistical treatment is the Model Output Statistics (MOS) technique. An MOS based on multiple linear regressions has been proposed by Ranaboldo (2011): the model screens the most relevant NWP forecast variables and selects best predictors in order to fit a regression equation that minimizes the forecast errors, utilizing wind farm power output measurements as input. The performance of the method is evaluated in two wind farms, located in different topographical areas and with different NWP grid spacing. Due to the high seasonal variability of NWP forecasts, it has been considered appropriate to implement monthly stratified MOS. In both wind farms, first predictors are always wind speeds (at different heights) or friction velocity. When friction velocity is the
first predictor, proposed MOS forecasts have resulted to be highly dependent on the friction velocity – wind speed correlation. Negligible improvements are encountered when including more than 2 predictors in the regression equation. Proposed MOS has performed well in both wind farms and its forecasts compare positively with actual operative model in use at Risø DTU and other MOS types, showing minimum bias and improving NWP power forecast of around 15% in terms of root mean square error. Further improvements could be obtained by the implementation of a more refined MOS stratification, e.g. fitting specific equations in different synoptic situations.

2.4 SOFTWARE EFFORT ESTIMATION

Stensrud, E., Foss et al. (2003) have investigated conclusion instability problem based on the evaluation criteria and the subset of the data used in the evaluation. With the use of non-parametric methods (the Mann-Whitney U test), it shows how to avoid conclusion instability. Menzies et al. (2006) have proposed a study that ranked 158 effort estimation methods via three different evaluation criteria and hundreds of different randomly selected subsets. The same four methods are ranked higher than the other 154 methods regardless of which evaluation criteria or data subset is applied. The recommended non-parametric evaluation is to evaluate and prune effort estimation methods. More specifically, while learning about the effort estimators from COCOMO-style data, it is found that the manual stratification defeats many complex algorithmic methods. To the best of their knowledge, this report is the first to offer stable conclusions regarding effort estimation across such a wide range of methods.

Often effort estimation technique requires a generalizing from a small number of historical projects. Generalization from such limited experience is an inherently under constrained problem. Hence, the learned effort models may possibly exhibit large deviations that prevent standard statistical methods (e.g., t-tests) from distinguishing the performance of alternative effort-estimation methods.
The COSEEKMO effort-modeling workbench applies a set of heuristic rejection rules to comparatively assess results from alternative models. Using these rules, and despite the presence of large deviations, COSEEKMO could rank alternative methods for generating effort models. Based on the experiments with COSEEKMO, Menzies et al. (2006) have proposed a new view on supposed “best practices” in model-based effort estimation: 1) Each such practice should be viewed as a candidate technique which may or may not be useful in a particular domain, and 2) tools like COSEEKMO are used to help analysts explore and select the best method for a particular domain.

Software estimation process helps in the prediction of the effort and cost which is required to develop software. Bisoi et al. (2012) provide a general overview of software estimation models and techniques. Model has been categorized as a Size-Based, Function-Based, Learning-Based and Expertise-Based. Both Size-based and Function-based models are termed as Parametric as they use a function or formula of fixed form for software cost/effort estimation. Each has its own strengths and weaknesses. A key factor in selecting an estimation model is the accuracy of its estimates. Unfortunately, it is true that no single technique is best in all the situations and there a careful comparison of the results with several approaches is most likely to produce realistic estimates.

The effort invested in a software project is one of the most important and most analyzed variables in recent years in case of the process of project management. Computing techniques was a consortium of methodologies centering in fuzzy logic, Artificial Neural Networks and evolutionary computation. These methodologies are complementary and synergistic, rather than competitive and provided in one form or another flexible information processing capability for handling real life ambiguous situations. These methodologies are currently used for reliable and accurate estimate of software development effort. Ramesh et al. (2013) proposed to analyze algorithmic models and non-algorithmic models to provide in
depth review of software and project estimation techniques in industry and literature based on the different test datasets along with their advantages and disadvantages.

Software Cost Estimation with resounding reliability, productivity and development effort is a challenging task. This has incited the software community to give much needed thrust and delve into extensive research in software effort estimation for evolving sophisticated methods. Estimation by analogy is one of the expedient techniques in software effort estimation field. However, the methodology utilized for the estimation of software effort by analogy is unable to handle the categorical data in an explicit and precise manner. Malathi and Sridhar (2011) have proposed a new approach that has been developed to estimate software effort for projects represented by categorical or numerical data using reasoning by analogy and fuzzy approach. The existing historical datasets, analyzed with fuzzy logic, produce an accurate result in comparison to the dataset analyzed with the earlier methodologies. Layman et al. (2008) have proposed a software development process, in which scheduling and predictability are considered important components to delivering a product on time and within budget. Effort estimation artifact offers a rich data set for improving scheduling accuracy and understanding the development process. Effort estimation data for 55 features in the latest release of Visual Studio Team System (VSTS) are collected and analyzed for trends, patterns and differences. Statistical analysis shows that actual estimation error is positively correlated with feature size and that in-process metrics of estimation error are also correlated with the final estimation error. These findings suggest that smaller features could be estimated more accurately, and that in-process estimation error metrics could be provided a quantitative supplement to developer intuition regarding high-risk features during the development process.

Jorgensen et al. (2004) have proposed an approach to improve analyses of errors that occur in software effort estimation, within one software development company and collected information about estimation errors through: i) interviews
with employees in different roles who are responsible for estimation, ii) estimation experience reports from 68 completed projects and iii) statistical analysis of relations between characteristics of the 68 completed projects and estimation error. The role of the respondents, the data collection approach and the type of analysis have an important impact on the reasons given for estimation error and found, for example, a strong tendency to perceive factors outside the respondents’ own control as important reasons for inaccurate estimates. Reasons given for accurate estimates are typically cited factors that are within the respondents’ own control and are determined by the estimator’s skill or experience.

Most work has focused based on algorithmic cost models such as COCOMO and Function Points. These could suffer from the disadvantage such as the need to calibrate the model to each individual measurement environment coupled with very variable accuracy levels even after calibration. An alternative approach has been to use analogy for estimation. Shepperd et al. (1996) demonstrate that this method has considerable promise and is shown to outperform traditional algorithmic methods for six different datasets. A disadvantage of estimation by analogy is that it requires a considerable amount of computation. An automated environment that supports the collection, storage and identification of the most analogous projects in order to estimate the effort for a new project is known as ANGEL is proposed. ANGEL is based upon the minimization of Euclidean distance in dimensional space. The software is flexible and could deal with differing datasets both in terms of the number of observations (projects) and in the variables collected. The analogy approach is evaluated with six distinct datasets drawn from a range of different environments and is found to outperform other methods. It is widely accepted that effective software effort estimation demands used more than one technique and has shown that estimation by analogy is a candidate technique and that with the aid of an automated environment is an eminently practical technique.
Software is the most expensive element for all computer based systems. For complex custom systems, a large effort estimation error makes the difference between profit and loss. Cost (Effort) Overruns may possibly be disastrous for the developer. The basic input for the effort estimation is the size of project. A number of models have been proposed to construct a relation between software size and Effort; however problems for effort estimation are still there because of uncertainty existing in the input information. Accurate software effort estimation is a challenge in the industry. Prasad Reddy et al. (2010) have proposed three software effort estimation models by using soft computing techniques. The performance of the developed models is tested by NASA software project dataset. The developed models are able to provide good estimation capabilities.

Sadiq et al. (2013) have used organic software projects because in each case the projects size lies between 2-50 KLOC and have applied the linear regression model i.e. Effort = -1.5 + 0.1804 FP to predict the software project effort and function point. After obtaining the software effort, project manager can arrange the project progress, control the cost and ensures the quality more accurately.

Software cost estimation is one of the prerequisite managerial activities carried out at the software development initiation stages and also repeated throughout the whole software life-cycle so that amendments to the total cost are made. In software cost estimation, typically, a selection of project attributes is employed to produce effort estimations of the expected human resources to deliver a software product. However, choosing the appropriate project cost drivers in each case requires a lot of experience and knowledge on behalf of the project manager which can only be obtained through years of software engineering practice. A number of studies indicate that popular methods applied in the literature for software cost estimation, such as linear regression, are not robust enough and do not yield accurate predictions. Recently the dual variables Ridge Regression (RR) technique has been used for effort estimation yielding promising results.
Papatheocharous et al. (2010) show that results may be further improved if an AI method is used to automatically select appropriate project cost drivers (inputs) for the technique and proposed a hybrid approach combining RR with a Genetic Algorithm, the latter evolving the subset of attributes for approximating effort more accurately. The proposed hybrid cost model has been applied on a widely known high-dimensional dataset of software project samples and the results obtained shows that accuracy may be increased if redundant attributes are eliminated.

Sehra et al. (2011) proposed effort invested in a software project which has been probably one of the most important and most analyzed variables in recent years in the process of project management. The limitation of algorithmic effort prediction models is their inability to cope with uncertainties and imprecision surrounding software projects at the early development stage. More recently attention has turned to a variety of machine learning methods and soft computing in particular to predict software development effort. Soft computing was a consortium of methodologies centering in fuzzy logic, artificial neural networks and evolutionary computation. It is important, to mention here, that these methodologies are complementary and synergistic, rather than competitive. They provide in one form or another flexible information processing capability for handling real life ambiguous situations. These methodologies are currently used for reliable and accurate estimate of software development effort, which has always been a challenge for both the software industry and academia. The aim of this study is to analyze soft computing techniques in the existing models and to provide in depth review of software and project estimation techniques existing in industry and literature based on the different test datasets along with their strength and weaknesses.

Even though there are a number of software size and effort measurement methods proposed in literature, they are not widely adopted in the practice. According to literature, only 30% of software companies use measurement, mostly as a method for additional validation. In order to determine whether the objective
metric approach could give results of the same quality or better than the estimates relying on work breakdown and expert judgment and has validated several standard functional measurement and analysis methods, on the selected set of small and medium size real-world web based projects at Capability Maturity Model Integration (CMMI) level 2. Evaluation performed Popović and Bojić (2012) provides objective justification and guidance for the use of a measurement-based estimation in these kinds of projects.

Software effort prediction is an important task in the software development life cycle. Many models including regression models, machine learning models, algorithmic models, expert judgment and estimation by analogy have been widely used to estimate software effort and cost. Nassif et al. (2012) have proposed a treeboost (Stochastic Gradient Boosting) model that was put forward to predict software effort based on the Use Case Point method. The inputs of the model include software size in use case points, productivity and complexity. A multiple linear regression model is created and the Treeboost model was evaluated against the multiple linear regression model, as well as the use case point model by using four performance criteria: MMRE (Mean Magnitude of Relative Error), PRED (Prediction), MdMRE (Median Magnitude of Relative Error) and MSE (Mean Square Error). Experiments show that the Treeboost model could be used with promising results to estimate software effort.

Neural Networks are often selected as tools for software effort prediction because of their capability to approximate any continuous function with arbitrary accuracy. A major drawback of neural networks is the complex mapping between inputs and output, which is not easily understood by a user. Setiono et al. (2010) have used a rule extraction technique that derives a set of comprehensible IF-THEN rules from a trained neural network are applied to the domain of software effort prediction. The suitability of this technique is tested on the ISBSG R11 (International Software Benchmarking Standards) data set by a comparison
with linear regression, radial basis function networks and CART (Classification And Regression Tree). It has been found that the most accurate results are obtained by CART, though the large number of rule limits comprehensibility. Considering comprehensible models only, the concise set of extracted rules outperform the pruned CART tree, making neural network rule extraction the most suitable technique for software effort prediction when comprehensibility is important.

Nagpal et al. (2012) have proposed software estimation techniques, an inclusive set of directives for software project developers, project managers and the management in order to produce more accurate estimates or predictions for future developments. The estimates also facilitate allocation of resources for Software development. Estimations also smooth the process of re-planning, prioritizing, classification and reuse of the projects. Various estimation models are widely being used in the Industry as well for research purposes. Several comparative studies have been executed on them, but choosing the best technique was quite intricate. Estimation by Analogy (EA) is the method of making estimations based on the outcome from k most analogous projects. The projects close in distance is potentially similar to the reference project from the repository of projects. This method has widely been accepted and is quite popular as it impersonates human beings inherent judgment skill by estimating with analogous projects. Grey Relational Analysis (GRA) is used as the method for feature selection and also for locating the closest analogous projects to the reference project from the set of projects. The closest k projects are then used to build regression models. Regression techniques like Multiple Linear Regression, Stepwise Regression and Robust regression techniques are used to find the effort from the closest projects.

Accurate and credible software effort estimation is a challenge for academic research and software industry. From many software effort estimation models in existence, Estimation by Analogy (EA) is still one of the preferred
techniques by software engineering practitioners because it mimics the human problem solving approach. Accuracy of such a model depends on the characteristics of the dataset, which is subject to considerable uncertainty. The inherent uncertainty in software attribute measurement has significant impact on estimation accuracy because these attributes are measured based on human judgment and are often vague and imprecise. To overcome this challenge Azzeh et al. (2010) have proposed a new formal EA model based on the integration of Fuzzy set theory with GRA. Fuzzy set theory is employed to reduce uncertainty in distance measure between two tuples at the kth continuous feature. GRA is a problem solving method that is used to assess the similarity between two tuples with M features. Since some of these features are not necessary to be continuous and may have nominal and ordinal scale type, aggregating different forms of similarity measures will increase uncertainty in the similarity degree. Thus the GRA is mainly used to reduce uncertainty in the distance measure between two software projects for both continuous and categorical features. Both techniques are suitable when relationship between effort and other effort drivers is complex. Experimental results show that using integration of GRA with FL produced credible estimates when compared with the results obtained using Case-Based Reasoning, MLR and ANN methods.

Knowing the estimated cost of a software project early in the development cycle is a valuable asset for management. Shan et al. have proposed an evolutionary computation method, Grammar Guided Genetic Programming (GGGP), to fit models, with the aim of improving the prediction of software development costs. Valuable results are obtained, significantly better than those obtained by simple linear regression. In this research GGGP, because of its flexibility and the ability of incorporating background knowledge, also shows great potential in being applied in other software engineering modeling problems. Many research projects on software estimation use software size as a major
explanatory variable. However, practitioners sometimes use the ratio of effort for early phase activities such as planning and requirement analysis, to the effort for the whole development phase of the software in order to estimate effort. Tsunoda et al. (2013) have focused on effort estimation based on the effort for early phase activities. The goal of the research is to examine the relationship of early phase effort and software size with software development effort. To achieve the goal, and build effort estimation models using early phase effort as an explanatory variable and compared the estimation accuracies of these models to the effort estimation models based on software size. In addition, the estimation models using both early phase effort and software size. The ISBSG (International Software Benchmarking Standards Group) dataset, which is collected from software development companies and regarded planning phase effort and requirement analysis effort as early phase effort. The result of the experiment has shown that when both software size and sum of planning and requirement analysis phase effort are used as explanatory variables, the estimation accuracy is most improved (Average Balanced Relative Error is improved to 75.4% from 148.4%). Based on the result, both early phase effort and software size be used as explanatory variables, because that combination shows the high accuracy and does not have multicollinearity issues.

Accurate software development effort estimation is critical to the success of software projects. Although many techniques and algorithmic models have been developed and implemented by practitioners, accurate software development effort prediction is still a challenging endeavor in the field of software engineering, especially in handling uncertain and imprecise inputs and collinear characteristics. Du et al. (2013) have proposed a hybrid intelligent model combining a neural network model integrated with fuzzy model (neuro-fuzzy model) to improve the accuracy of estimating software cost. The performance of the proposed model is assessed by designing and conducting evaluation with published project and
industrial data. Results have shown that the proposed model demonstrates the ability of improving the estimation accuracy by 18% based on the Mean Magnitude of Relative Error (MMRE) criterion.

Jorgensen (2004) has proposed models whose purpose is to explain the accuracy and bias variation of an organization’s estimates of software development effort through regression analysis. The collected information about variables are believed to affect the accuracy or bias of estimates of the performance of tasks completed by the organization. In total, information about 49 software development tasks is collected and found that the following conditions have led to inaccuracies in estimates: (i) Estimates are provided by a person in the role of ‘‘software developer’’ instead of ‘‘project leader’’, (ii) The project has as its highest priority time-to-delivery instead of quality or cost and (iii) The estimator does not participate in the completion of the task. The following conditions have led to an increased bias towards underestimation: (i) Estimates are provided by a person with the role of ‘‘project leader’’ instead of ‘‘software developer’’. (ii) The estimator has assessed the accuracy of own estimates of similar, previously completed tasks to be low (more than 20% error). In addition, there are several important threats to the validity of the coefficients suggested by the models. An analysis of the estimators’ own descriptions of the reasons for achieved estimation accuracy on each task suggests that it will be difficult to include all important estimation accuracy and bias factors in regression-based models. For this reason, not realistic to expect such models to replace human judgment in estimation uncertainty assessments and as input to plans for the improvement of estimates. It is, nevertheless, possible that the type of formal analysis and regression-based models are presented.

2.5 FORECASTING USING ARTIFICIAL NEURAL NETWORK (ANN)

ANNs have powerful pattern classification and pattern recognition capabilities. One major application area of ANNs is forecasting. Several
distinguishing features of ANNs make them valuable and attractive for a forecasting task. ANNs are data-driven self-adaptive methods. They can generalize i.e. after learning data are presented (a sample), ANNs can often correctly infer the unseen part. Also, ANNs are universal functional approximators and are nonlinear.

2.5.1 Software Effort Estimation using ANN

Estimating software development effort is an important task in the management of large software projects. The task is challenging and it has been receiving the attentions of researchers ever since software was developed for commercial purpose. A number of estimation models exist for effort prediction. However, there has been a need for novel model to obtain more accurate estimations. Kaur et al. (2010), have proposed a precise method of estimation by selecting the most popular models in order to improve accuracy and explore the use of Soft Computing Techniques to build a suitable model structure to utilize improved estimation of software effort for NASA software projects. A comparison between ANN Based Model is carried out. The evaluation criteria are based upon MRE (Magnitude of Relative Error) and MMRE (Mean Magnitude of Relative Error). Consequently, the final results are very precise and reliable when they are applied to a real dataset in a software project. The results show that ANNs are effective in effort estimation. Software development effort estimation is one of the most major activities in software project management. A number of models have been proposed to construct a relationship between software size and effort; however still have problems for effort estimation. This is because project data, available in the initial stages of project are often incomplete, inconsistent, uncertain and unclear. The need for accurate effort estimation in software industry is still a challenge. Artificial Neural Network models are more suitable in such situations. Reddy et al. (2009) have proposed to develop software effort estimation models based on artificial neural networks. The models are designed to improve the performance of the network that suits to the COCOMO Model. Artificial Neural
Network models are created using Radial Basis and Generalized Regression. A case study based on the COCOMO81 database compares the proposed neural network models with the Intermediate COCOMO. The results are analyzed using five different criterions MMRE, MARE, VARE, Mean BRE and Prediction. It is observed that the Radial Basis Neural Network provides better results.

Tronto et al. (2006), have proposed a machine learning techniques such as neural networks, rule induction, genetic algorithm and case-based reasoning for application in a wide variety of fields such as computer vision, econometrics and medicine, where human abilities have proven to be superior to those of computers. Such techniques hold the promise of being able to make sense of a variety of inputs of different types in producing an output. Software effort modeling has always appeared to be a rather hit-or-miss business where statistical methods frequently result in low accuracy of prediction. Some experiments using an ANN have been conducted, highlighting some of the problems that arise when machine learning techniques are applied to software effort modeling. These experiments show that, compared with conventional regression analysis, improved accuracy of prediction is possible. A software effort estimation model based on artificial neural networks is constructed. The model is designed accordingly to improve the performance of the network that suits to the COCOMO model. A multi-layer feed forward neural network is used to accommodate the model and its parameters to estimate software development effort. The network is trained with back propagation learning algorithm by iteratively processing a set of training samples and comparing the network's prediction with the actual effort. COCOMO dataset was used to train and to test the network and it is observed that proposed neural network model improves the estimation accuracy of the model. The test results from the trained neural network are compared with that of the COCOMO model. The preliminary results obtained suggest that the proposed architecture could be replicated for accurately forecasting the software development effort.
To enhance the estimation accuracy of COCOMO model, the estimated effort is more close to the actual effort.

The software effort estimation model is constructed based on Artificial Neural Networks. The model is designed accordingly to improve the performance of the network that suits to the COCOMO Model. A single layer feed forward neural network is used to accommodate the model and its parameters to estimate software development effort. The network is trained with back propagation learning algorithm and Resilient Back propagation algorithm (RPROP) by iteratively processing a set of training samples and comparing the network’s prediction with the actual effort. COCOMO dataset is used to train and to test the network and it is observed that proposed neural network model improves the estimation accuracy of the model. The test results from the trained neural network are compared with those of the COCOMO model. By comparing the results of these two models, it is proved that both models (SLANN with BP and SLANN with RPROP) work better than COCOMO and SLANN with RPROP is an optimal neural network model for software effort estimation. SLANN with BP works well only for projects with small size, where as SLANN with RPROP works well for all kinds of projects as the convergence rate of RPROP algorithm is very high. The preliminary results obtained suggest that the proposed architecture could be replicated for accurately forecasting the software development effort.

Dimitri P Solomatine et al. (2004) have proposed Flexible and Optimal M5 Model Trees with Applications to Flow Predictions. Numerical prediction is otherwise called as regression. M5-modelled decision tree is one of the classification methods and the tree is created by portioning the large size input space using the entropy measures and gives unique labels in the leaf nodes. The actual values of input parameters are applied from the root node down to the leaf node and the leaf node gave the classified label. For regression, leaf nodes are assigned to the average of the output variable. If the input domain has complex regression
relationships among them, the leaf nodes are designed to give the relationship functions rather than average value of the output variable. M5 tree is created progressively by dividing the input domain into many subsets. The division is stopped when the regression error is lesser than some acceptable error level from the training data.

A Modified algorithm based on Support Vector Machines for classification of hyper spectral images in a similarity space is proposed by Hosseini et al. (2012). SVM-based classifier is a more accurate classification method especially when the size of the sample space is very small. The most important aspect of SVM classifier is the kernel function selection. The main purpose of kernel function was to minimize the distance between instances within the same cluster (intra cluster) and maximize the distance between instances of different clusters (inter clusters). The design of kernel functions is done at the time of training the training set and then is tested with test data. Finding the best kernel suitable for the given data set is a time consuming process. Trafalis and Ince (2000) have compared the SVM developed by Vapnik with other techniques such as Back propagation and Radial Basis Function (RBF) Networks for financial forecasting applications. The theory of the SVM algorithm is based on statistical learning theory. Training of SVMs leads to a QP problem. Preliminary computational results for stock price prediction are carried out. The support vector machines for regression, a robust technique for function approximation is used. Preliminary computational results in the MATLAB environment seem to be quite promising.

2.5.2 Forecasting using ANN in other domains

Prediction of stock market returns is an important issue in finance. Artificial Neural Networks have been used in stock market prediction during the last decade. Studies are performed for the prediction of stock index values as well as daily direction of change in the index. In some applications, it has been specified that artificial neural networks have limitations for learning the data.
patterns or that they may perform inconsistently and unpredictable because of the complex financial data used. In Turkey, artificial neural networks are mostly used in predicting financial failures. There had been no specific research for prediction of Turkish stock market values. The aim is to use artificial neural networks to predict Istanbul Stock Exchange (ISE) market index value. Preliminary research performed on Turkish stock market has suggested that the inputs to the system might be taken as: previous day’s index value, previous day’s TL/USD exchange rate, previous day’s overnight interest rate and 5 dummy variables each representing the working days of the week. After the inputs have been determined, the data have been gathered for the period of July 1, 2001 through February 28, 2003 from the Central Bank of Republic of Turkey. Training set is determined to include about 90% of the data set and the rest 10% will be used for testing purposes. Network architecture has been determined to be Multi-Layer Perceptron (MLP) and Generalized Feed Forward networks. Training and testing have been performed using these two network architectures. However, subsystems are considered, which have different number of hidden layers (1, 2 and 4) for a mean-squared error value of 0.003. The results are then compared with the results of moving averages for 5 and 10-day periods, which show that artificial neural networks have better performances than those of moving averages.

Load forecasting is the technique for prediction of electrical load. In a deregulated market, it is much needed for a generating company to know about the market load demand for generating near to accurate power. If the generation is not sufficient to fulfill the demand, there will be problem of irregular supply and in case of excess generation the generating company will have to bear the loss. Neural network techniques has been recently suggested for short-term load forecasting by a large number of researchers. Firstly, a literature survey is conducted and most of the reported models are based on the MLP network. There are numerous model suggestions, but the large variation and lack of comparisons
make it difficult to directly apply proposed methods. It is concluded that a comparative study of different model types seems necessary. Several models are developed and tested on the real load data of a Finnish electric utility. Most of them use an MLP network to identify the assumed relation. Sheikh and Unde (2012) have carried out short term load forecasting for P.D.V.V.P.COE, Ahmednagar college campus using ANN technique. ANN is implemented on MATLAB-10. MLP (Multi-layer Perceptron) is made with input as days and hourly load. Hourly load means the hourly power consumption in college. Error is calculated as MAPE (Mean Absolute Percentage Error) and with error of about 0.956% is successfully carried out. It could be implemented by any intensive power consuming company/college for predicting the future load and will prove to be very useful tool while sanctioning the load. Park et al. (1991) have proposed an ANN electric load forecasting, the ANN is used to learn the relationship among past, current and future temperatures and loads. In order to provide the forecasted load, the ANN interpolates among the load and temperature data in a training data set. The average absolute errors of the one-hour and 24-hour ahead forecasts in the test on actual utility data show 1.40% and 2.06%, respectively. This is compared with an average error of 4.22% for 24-hour ahead forecasts with a currently used forecasting technique applied to the same data.

Lee et al. (1992) have proposed ANN method to forecast the short-term load for a large power system. The load has two distinct patterns: weekday and weekend-day patterns. The weekend-day pattern includes Saturday, Sunday and Monday loads. A nonlinear load model is proposed and several structures of ANN for short-term load forecasting are tested. Inputs to the ANN are past loads and the output of the ANN is the load forecast for a given day. The networks with one or two hidden layers are tested with various combination of neurons and results are compared in terms of forecasting error. The neural network, when grouped into different load patterns, gives good load forecast. Majumder and Hussain (2010)
presented a computational approach for predicting the S&P CNX Nifty 50 Index. A neural network based model has been used in predicting the direction of the movement of the closing value of the index. It could be used to predict price index value of the stock market. After studying various features of the network model, an optimal model is proposed for the purpose of forecasting. The model has used the pre-processed data set of closing value of S&P CNX Nifty 50 Index. The data set encompasses the trading days from 1st January, 2000 to 31st December, 2009. The model has been validated across 4 years of the trading days. Accuracy of the performance of the neural network is compared using various out of sample performance measures. The highest performance of the network in terms of accuracy in predicting the direction of the closing value of the index is reported at 89.65% and with an average accuracy of 69.72% over a period of 4 years.

Interest in using ANNs for forecasting has led to a tremendous surge in research activities in the past decade. While ANNs provide a great deal of promise, they also embody much uncertainty. Researchers to date are still not certain about the effect of key factors on forecasting performance of ANNs. Zhang et al. (1998) present a state-of-the-art survey of ANN applications in forecasting. It has provided (i) a synthesis of published research in this area,(ii) insights on ANN modeling issues and (iii) the future research directions. Hung et al. (2009) have presented a new approach using an Artificial Neural Network technique to improve rainfall forecast performance. A real world case study is set up in Bangkok; 4 years of hourly data from 75 rain gauge stations in the area are used to develop the ANN model. The developed ANN model is being applied for real time rainfall forecasting and flood management in Bangkok, Thailand. Aimed at providing forecasts in a near real time schedule, different network types are tested with different kinds of input information. Preliminary tests show that a generalized feed forward ANN model using hyperbolic tangent transfer function has achieved the best generalization of rainfall. Especially, the use of a combination of
meteorological parameters (relative humidity, air pressure, wet bulb temperature and cloudiness), the rainfall at the point of forecasting and rainfall at the surrounding stations, as an input data, advanced ANN model to apply with continuous data containing rainy and non-rainy period, have allowed model to issue forecast at any moment. Additionally, forecasts by ANN model are compared to the convenient approach namely simple persistent method. Results show that ANN forecasts have superiority over the ones obtained by the persistent model. Rainfall forecasts for Bangkok from 1 to 3 h ahead are highly satisfactory. Sensitivity analysis has indicated that the most important input parameter besides rainfall is the wet bulb temperature in forecasting rainfall. Rui and El-Keib (1995) have proposed an extensive survey of ANN-based load forecasting models. Six factors which are believed to have a considerable effect on the accuracy, reliability and robustness of the models are emphasized. The surveyed publications and the authors own experience lead to the conclusion that the ANN structure, input variables, number of hidden neurons and BP algorithm parameters are mainly system dependent. The development of a more general ANN model is to handle the STLF problem.

2.6 FORECASTING USING SUPPORT VECTOR REGRESSION (SVR)

Support Vector Regression (SVR), has been successfully used to solve nonlinear regression. The SVR model applies the structural risk minimization principle to minimize the upper bound of the generalization error, instead of minimizing the training error, employed by most conventional neural network models.

2.6.1 Software Effort Estimation using SVR

Basha and Ponnurangam (2010) have proposed a reliable effort estimation which remains an ongoing challenge to software engineers. Accurate effort estimation is the state of art of software engineering, effort estimation of software was the preliminary phase between the client and the business enterprise. The relationship between the client and the business enterprise begins with the
estimation of the software. The credibility of the client to the business enterprise increases with the accurate estimation. Effort estimation often requires generalizing from a small number of historical projects. Generalization from such limited experience is an inherently under constrained problem. Accurate estimation is a complex process because it could be visualized as software effort prediction, as the term indicates prediction never becomes an actual. This work follows the basics of the empirical software effort estimation models. The goal of this is to study the empirical software effort estimation. The primary conclusion is that no single technique is the best for all situations and that a careful comparison of the results of several approaches is most likely to produce realistic estimates. Pahariya et al. (2009), have proposed a new computational intelligence sequential hybrid architectures involving Genetic Programming (GP) and Group Method of Data Handling (GMDH) viz. GP-GMDH, GMDH-GP and recurrent architecture for GP for software cost estimation. Three linear ensembles based on (i) arithmetic mean (ii) geometric mean and (iii) harmonic mean are also developed and performed GP based feature selection. The efficacy of MLR, Polynomial Regression, SVR, Classification And Regression Tree (CART), Multivariate Adaptive Regression Splines (MARS), Multilayer FeedForward Neural Network (MLFF), Radial Basis Function Neural Network (RBF), Counter Propagation Neural Network (CPNN), Dynamic Evolving Neuro–Fuzzy Inference System (DENFIS), TreeNet, Group Method of Data Handling and Genetic Programming was tested on the International Software Benchmarking Standards Group (ISBSG) release 10 dataset. Ten-fold cross validation is performed throughout the study. The results obtained from the experiments indicate that the GP-GMDH and GMDH-GP have outperformed all the other techniques and also have performed t-test to see if the performances of the hybrids developed are statistically significant.

Xing and Guo (2005) have proposed SVR to build Software Reliability Growth Model (SRGM). SRGM is an important aspect in software reliability
Software reliability is the probability that a given software will be functioning without failure during a specified period of time in a specified environment. In order to obtain the better performance of SRGM, practical selection of parameter C for SVR is discussed in the experiments. Experimental results with the classical Sys1 and Sys3 SRGM data set show that the performance of the proposed SVR-based SRGM is better than conventional SRGMs and relative good prediction and generalization ability are achieved. Software cost estimation is the process of predicting the effort required to develop a software system. The basic input for the software cost estimation is coding size and set of cost drivers, the output was Effort in terms of Person-Months (PM’s). Here, the use of SVR has been proposed for the estimation of software project effort and has used the COCOMO dataset and the results are compared to Intermediate COCOMO as well as to MOPSO model results for this dataset. It has been observed from the simulation that SVR outperforms other estimating techniques. Kumari and Pushkar (2013) provide a comparative study on SVR, Intermediate COCOMO and Multiple Objective Particle Swarm Optimization (MOPSO) model for estimation of software project effort. To forecast the demand of wood pulp, SVM regression analysis is used with kernel functions. SVM is a machine learning method used to project the given data into a higher dimensional space. It has maximized the margins between two different classes and minimized the regression error margin. Training data are collected for the demand from the paper mills over years and support vector regression model is created. After training any data irrespective of training data could be given for testing. Training and testing using this method have given improved accuracy.

Accurate cost estimation helps to complete project within time and budget. Many estimation models have been proposed over the last 30 years. Kumari Pushkar (2013) provides a detailed overview of existing software cost estimation models and techniques. Cost estimation models are basically of two
types: algorithmic and non-algorithmic. It also includes the recent developed techniques for software cost estimation field. The strength and weakness of various software cost estimation methods. It also focuses on some of the relevant reasons that cause inaccurate estimation. A comparative analysis among existing popular models are performed and the performance is analyzed and compared in terms MMRE (Mean Magnitude of Relative Error) and PRED (Prediction).

2.6.2 Forecasting using SVR in other domain

Travel time is a fundamental measure in transportation. Accurate travel-time prediction also is crucial to the development of intelligent transportation systems and advanced traveler information systems. Chun-Hsin Wu et al. (2004) have applied SVR for travel-time prediction and have compared its results to other baseline travel-time prediction methods using real highway traffic data. Since support vector machines have greater generalization ability and guarantee global minima for given training data, it is believed that SVR will perform well for time series analysis. Compared to other baseline predictors, the results show that the SVR predictor can significantly reduce both relative mean errors and root-mean-squared errors of predicted travel times and then demonstrate the feasibility of applying SVR in travel-time prediction and proves that SVR is applicable and performs well for traffic data analysis. Bo-Juen Chen et al. (2004) have proposed SVM model successfully applied to load forecasting. Important conclusions from the results are that temperature might not be useful in such a midterm load forecasting problem and that the introduction of time series concept may improve the forecasting.

As financial time series are inherently noisy and non-stationary, it is regarded as one of the most challenging applications of time series forecasting. Due to the advantages of generalization capability in obtaining a unique solution, SVR has also been successfully applied in financial time series forecasting. In the modeling of financial time series using SVR, one of the key problems is the inherent high noise. Thus, detecting and removing the noise are important but
difficult tasks when building an SVR forecasting model. To alleviate the influence of noise, a two-stage modeling approach used for independent component analysis (ICA) and support vector regression is proposed in financial time series forecasting. ICA was a novel statistical signal processing technique that is originally proposed to find the latent source signals from observed mixture signals without any prior knowledge of the mixing mechanism. Lu, C.J. et al. (2009) have proposed to use ICA to the forecasting variables for generating the Independent Components (ICs). After identifying and removing the ICs containing the noise, the rest of the ICs are then used to reconstruct the forecasting variables which contain less noise and served as the input variables of the SVR forecasting model. In order to evaluate the performance of the proposed approach, the Nikkei 225 opening index and TAIEX closing index are used as illustrative examples. Experimental results show that the proposed model outperforms the SVR model with non-filtered forecasting variables and a random walk model.

Traditional time series forecasting models, like ARIMA and regression models, can hardly capture nonlinear patterns. SVR, a novel neural network technique, has been successfully used to solve nonlinear regression and time series problems. The SVR model applies the structural risk minimization principle to minimize the upper bound of the generalization error, instead of minimizing the training error, employed by most conventional neural network models. Thus, parameter determination for an SVR model is appropriate to achieve high forecasting accuracy. Several evolutionary algorithms, such as genetic algorithms and simulated annealing algorithms have been used in parameter selection, but these algorithms often suffer from the possibility of being trapped in local optimum. This is used in an improved ant colony optimization algorithm in an SVR model, called SVRCACO, for selecting suitable parameters, with encouraging local search in areas where forecasting accuracy improvement continues to be made, then, autocatalytically converge to promising regions.
Simutis R. et al. (2008) have proposed two different methods used to forecast the daily cash demand for Automatic Teller Machines (ATM). The first method is based on flexible ANN. The generalization properties of this ANN are improved using special adaptive regularization term. The second forecasting method employs the Support Vector Regression (SVR) algorithm. Performed simulation studies and experimental tests show tolerable forecasting capacities using the both proposed methods. The forecasting schema based on flexible ANN is in the implementing phase for intelligent cash management in ATM network. Trafalis and Ince (2000) have compared SVM developed by Vapnik with other techniques such as Back propagation and Radial Basis Function (RBF) Networks for financial forecasting applications. The theory of the SVM algorithm is based on statistical learning theory. Training of SVMs leads to a Quadratic Programming (QP) problem. Preliminary computational results for stock price prediction is also validated. The comparison is done for support vector machines for regression with Back propagation and RBF networks. The support vector machines for regression are a robust technique for function approximation. Preliminary computational results in the MATLAB environment seem quite promising.

Wang, L. and Ji Zhu (2010) have proposed a two-step kernel learning method based on the SVR for financial time series forecasting. In a number of candidate kernels, they learn a sparse linear combination of these kernels so that the resulting kernel could be used to predict well on future data. The L1-norm regularization approach is used to achieve kernel learning. Since the regularization parameter might be carefully selected, to facilitate parameter tuning, they develop an efficient solution path algorithm that solves the optimal solutions for all possible values of the regularization parameter. The kernel learning method has been applied to forecast the S&P500 and the NASDAQ market indices and it shows promising results.