Journal of Evolving Systems manuscript No.
(will be inserted by the editor)

Evolving Integrated Multi-Model Framework for On Line Multiple Time Series Prediction

Received: date / Accepted: date

Abstract Time series prediction has been extensively researched in both the statistical and computational intelligence literature with robust methods being developed that can be applied across any given application domain. A much less researched problem is multiple time series prediction where the objective is to simultaneously forecast the values of multiple variables which interact with each other in time varying amounts continuously over time. In this paper we describe the use of a novel Integrated Multi-Model Framework (IMMF) that combined models developed at three different levels of data granularity, namely the Global, Local and Transductive models to perform multiple time series prediction. The IMMF is implemented by training a neural network to assign relative weights to predictions from the models at the three different levels of data granularity. Our experimental results indicate that IMMF significantly outperforms well established methods of time series prediction when applied to the multiple time series prediction problem.

Keywords DENFIS, Local Trend Models, Dynamic Interaction Networks, Integrated Multi Model Framework, Transductive Modelling

1 Introduction

Previous research has shown that dynamic relationships exists over time between different variables relating to real world phenomena in the Biological and Economic domains. For example, it is well known that in a Gene Regulatory Network (GRN) the expression level of a gene is determined by its time varying interactions with other genes [11] [23] [25] [17]. Similarly, the movement of stock market index in a given country is influenced by the movement of other markets in the region that trade heavily with the country in question [5] [9] [10] [29].

The problem of forecasting multiple time series is fundamentally different from that of forecasting single time series. The multiple time series prediction problem cannot be simply decomposed into a set of single time series problems due to inter-dependencies between the variables. This means that traditional time series forecasting methods such as the Multi Layer Perceptron and Support Vector

Address(es) of author(s) should be given

Machines will be less effective. There has been some research into modeling the dynamics of interactions between multiple variables [8] [5] [10] [17] [33], but none of these studies have directly addressed the problem of forecasting multiple time series simultaneously.

This research seeks to fill this gap by exploring an integrated framework for multiple time series prediction. This framework, henceforth referred to as the Integrated Multi-Model Framework (IMMF) consists of three layers of models, namely Global, Local and Transductive models. Global models are built using all historical data and are useful for capturing long-term trends in data. However, they are less effective in tracking localised changes that take place at discrete points in time. Local models that work on subsets of data are better equipped to capture shorter term disturbances than global models. Clustering offers a natural solution to the problem of capturing discrete changes in data and we employ an evolving clustering method that has been inspired by Kasabov's Evolving Connectionist Systems [39] and more specifically by the DENFIS model [19] for building clusters incrementally. The third layer of the framework consists of the transductive model whose role is to learn from individual instances rather than from the entire dataset or subsets of it and perform prediction based on similarity with instances that have been presented in the past.

The key contributions in this research are formulation of an integrated framework for multiple time series prediction and the construction of methods that adapt to new data, evolve their structure and operate on different levels of data granularity. Our experimental results in section 5 on both real world and synthetic data show that each model has it own strengths. The main role of the global model is to explicitly identify complex inter-dependencies between the variables over time, while the local model tends to produce better accuracy than the Global model while being able to capture recurring trends. The transductive model when used on its own also performs similarly to the Local model, while sometimes outperforming both the Global and Local model in some scenarios. Our experimentation also showed that a high degree of synergy exists between the models as the integrated model consistently outperforms any of the three models on their own, thus reinforcing the need for an integrated mechanism.

The rest of the paper is organized as follows. Section 2 reviews prominent methods used in time series forecasting which have relevance to this research. In Section 3 we present the design principles behind the construction of each of the three different models developed at the global, local and instance levels. Section 4 formulates a weighting scheme that combines information from each of the models to form an overall prediction that is more robust than that produced by each model on it own. Our experimentation is presented in Section 5 where we show the accuracy of each model, the knowledge extracted and the role that integration plays. Our conclusions and directions for future research are outlined in Section 6.

2 Related Work

Univariate time-series prediction has been very well researched in both the Statistical and Machine Learning literature. Well developed and mature methods exist in the Statistical literature such as Box Jenkins Auto Regressive Moving Average (ARMA), and its generalised form, called the Auto Regressive Integrated Moving

Average (ARIMA) model. Such models represent the value of a time series at a given time step t as a recurrence relation consisting of a weighted product of its values and corresponding coefficients at previous time lags $t-1$, $t-2$, ..., $t-p$, together with the addition of white Gaussian noise. The coefficents of the recurrence relation are determined by formulating the problem as minimizing a function which is solved through the application of the ordinary least squares method.

Many methods for time series prediction have been proposed by the machine learning community and these include the well known Multi Layer Perceptron (MLP) and the Support Vector Machine, among others. Both types of methods are capable of modeling non linear trajectories and in general are much more accurate than traditional linear estimation methods such as Linear Regression. The main drawback of these methods in the univariate context is that they do not adapt to new data arriving in a stream. Thus in the context of a continuous stream of time series data, computationally expensive rebuilding of models would need to be performed on a regular basis in order to preserve a high level of accuracy. A range of Evolving Connectionist Systems or ECOS was proposed by Kasabov et al in a series of papers [15] [19] [18], [20] with DENFIS as one prominent method that is incremental in nature as it uses on line clustering. DENFIS is a powerful tool for neuro fuzzy inference that uses principles of evolving connectionist systems to incrementally build and maintain clusters. It creates Takagi-Sugeno type fuzzy rules for each cluster and thus can also be considered to belong to the class of evolving intelligent systems. It can be used for prediction purposes by fitting the data in each cluster with a regression function based on a Gaussian Kernel. It turns out, as our results in section 5 show, that DENFIS is better than methods such as MLP not just in terms of flexibility but also in terms of prediction accuracy. In view of the good performance of DENFIS [19] and its adaptive nature we used it as the core method to build our Local model.

We acknowledge that more recent methods such as Sequential Adaptive Fuzzy Inference System (SAFIS) [32], Flexible Fuzzy Inference Systems (FLEXFIS) [27], Evolving Fuzzy Model eFuMo [2], Evolving Takagi Sugeno model (eTS) [1] that we review below may outperform DENFIS in some situations, but our objective in this research is to demonstrate the effectiveness of the Integrated Framework for multiple time series prediction. In principle, any of the three aforementioned methods may be used to replace DENFIS with a careful estimation of parameter values, with a possible effect of increasing prediction accuracy further.

Angelov and Filev [1] proposed an Evolving Takagi Sugeno model (eTS) that recursively updates a TS model structure. The concept of information potential was defined as the spatial proximity of the new instance to all existing instances received so far to assess whether new data arriving should result in a new rule or whether existing rules should be modified. If the informative potential of the new data sample is higher than the average potential of the existing rules it is added to the rule base. If the new data, which is accepted as a focal point of a new rule is too close to a previously existing rule then an existing rule is replaced by a new one. The eTS scheme was compared with a number of other neuro-fuzzy systems such as ESOM and DENFIS. A simplified version of the eTS learning algorithm that simplifies the rule base, called the Simpl_eTS, was proposed by Angleov and Filev in a follow up paper [3].

The Sequential Adaptive Fuzzy Inference System (SAFIS) proposed by Rong et al uses the concept of rule influence of fuzzy rules to incrementally add or remove rules. SAFIS uses a five layered structure consisting of input variables, fuzzy mebership functions, fuzzy rules and output variables. An extended Kalman filter scheme was used to update fuzzy rules produced. SAFIS was compared to other fuzzy rule generators and was found to have similar accuracy to DENFIS, ESOM and eTS while producing more compact rule bases for benchmark time series datasets.

The FLEXFIS scheme uses vector quantization in combination with a vigilance parameter to update cluster centers with each new incoming data point and for the generation of a new cluster whenever required by new data. Vector quantization is used to incrementally update cluster centers when new data arrives. Unlike conventional use of vector quantization, when new data arrives it calculates the distance to the surface of existing clusters and thus prevents the generation of a new cluster very close to an existing cluster. New clusters are added depending on the value of the vigilance parameter which is set on the basis of a trial and error process. With proper setting of the vigilance parameter a new cluster center is never created that is far away from the centroid of a newly emerging mass of data, unlike conventional use of vector quantization methods. The FLEXFIS scheme was found to have better accuracy than DENFIS, SAFIS, eTS on a non linear dynamic prediction problem and the Mackay Glass dataset although its rule base was larger in the case of the Mackay Glass dataset.

Comprehensive surveys on Evolving Intelligent Systems and Evolving Neuo-Fuzzy Systems in particular can be found in [4] and [28] respectively.

In using DENFIS we note that it is limited to univariate time series prediction and the core algorithm needed to be modified in several ways to adapt it to the multivariate case as described in [43] and in section 3 of this paper.

In terms of multivariate methods two prominent techniques exist, one is the multivariate version of the Box Jenkins model, known as the VARMA and the other is the Kalman filter. Neither of these methods were designed to deal with continuous streaming data having non-stationary data distributions and hence require modification to deal with such scenarios. Such modifications are described in section 3.1 whereby the basic Kalman filter method was modified to incrementally update the trajectory of each time series variable without the need for recomputing the entire transition matrix that specifies the inter-relationships between the multiple time series variables.

3 Formulation of Global, Local and Transductive Models

In this section we describe the three basic schemes that we have used for forecasting. We start with the Global model which utilizes all available historical data to build a single model that is used for modeling as well as prediction. We then go on to describe the formulation of the Local model which is based on an evolving clustering architecture. Given a dataset, local modeling will give rise to a number of local models, each of which will define a partition or cluster of the dataset as whole. Forecasting is achieved by combining the predictions made by each of the local models. Lastly, we present the transductive scheme that makes predictions on the basis of each observation, rather than groups of observations.

3.1 Global Modeling with Evolving Dynamic Interaction Networks (eDin)

Given a set of observations $X = (x^1, x^2, ..., x^i)$ of a variable x at time points $(1, 2, ..., i)$, the estimated trajectory of x at time point $i+1$ is given by the Kalman filter recursive equation:

$$\hat{x}_e^{i+1} = A\hat{x}_u^i \tag{1}$$

where $\hat{x}_u^i$ denotes the updated value of x given the i^{th} observation and A is the transition matrix that governs the movement of x between consecutive time points. We initialize A to be the identity matrix at the start of the time series.

The error co-variance of $\hat{x}_e^{i+1}$ is given from the error co-variance matrix, where P is defined by:

$$P_u^{i+1} = AP_u^i A^T + Q \tag{2}$$

where Q is an n by n noise co-variance matrix which is initialized to a very small value $1e^{-5}$. Likewise, P is also initialized to $1e^{-5}$. Now

$$\hat{x}_u^i = \hat{x}_e^i + K^i(\hat{x}^i - I\hat{x}_e^i) \tag{3}$$

where K is the Kalman gain given by:

$$K^{i+1} = P_e^{I+1} I^T (Q + IP_e^{i+1} I^T)^{-1} \tag{4}$$

The above equations $[1-4]$ are sufficient to model the progression of m variables $x_1, x_2, ..., x_m$ over time in off-line mode. However, to monitor the progression of these variables in on-line mode an efficient method is required of incrementally updating the transition matrix A instead of recalculating it at each new observation.

In order to achieve this we used an Expectation Maximization (EM) process where we iterate between Expectation and Maximization steps. In the Expectation step, we obtain the smoothed estimate of x at the previous time point i-1 as follows:
$J^{i-1} = P_u^{-1} A^T [P_e^i]^{-1}$, $\hat{x}_s^{i-1} = \hat{x}_u^{i-1} + J^{i-1}(\hat{x}_s^i - \hat{x}_e^i)$ where $\hat{x}_s^i = \hat{x}_u^i$

Note that the smoothing operation above needs to be applied to the previous time step only and not to the historical data as a whole. We can now perform the expectation step. Following [40] the sufficient statistics required in the expectation step are:

$E[x^i x^i|X] = P_s^i + \hat{x}_s^i \hat{x}_s^i$, $E[x^i x^{i-1}|X] = C_s^i + \hat{x}_s^i \hat{x}_s^{i-1}$ where C is the lag one co-variance smoother given by: $C_s^i = (I - K^i I)AP_u^{i-1}$

In the Maximization step, matrix A is updated to maximize the expectation of the joint probability density function over the posterior density:

$$A_{new} = [\sum_{k=1}^{i} E[x^k x^k|X]][\sum_{k=2}^{i} E[x^k x^{k-1}|X]]^{-1}. \tag{5}$$

Equation 5 above ensures that on-line predictions for a set of interacting variables x can be implemented efficiently without the need to recreate entire models at every time step.

In addition to the global prediction model given above we also construct a Dynamic Interaction Network (DIN) that captures the dynamics of the interactions

$$\begin{bmatrix} \mathbf{0.53} & \mathbf{-0.23} & \mathbf{0.36} \\ \mathbf{-0.01} & \mathbf{0.87} & \mathbf{0.43} \\ \mathbf{0.52} & \mathbf{-0.02} & \mathbf{0.31} \end{bmatrix} \text{(a)} \quad \begin{bmatrix} + & - & + \\ \mathbf{0} & + & + \\ + & \mathbf{0} & + \end{bmatrix} \text{(b)}$$

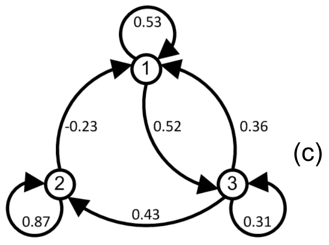


Fig. 1: Illustration of interaction network construction in DIN. (a) is the transition matrix; (b) is the corresponding influence matrix when a threshold of 0.1 is used; (c) is the interaction network.

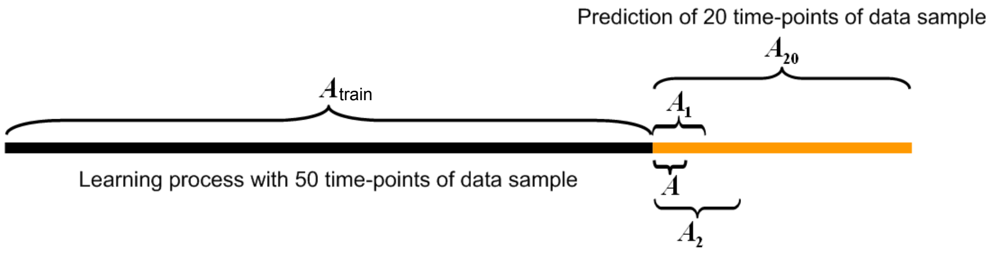

Prediction of 20 time-points of data sample
A_{20}
A_{train}
A_1
Learning process with 50 time-points of data sample
A
A_2

Fig. 2: Illustration of incremental learning in DIN. Transition matrix **A** is being re-estimated as new observations become available.

between the variables. The DIN modeling approach was first proposed by Kasabov et al in [17]. The construction of a DIN from a transition matrix is illustrated in Figure 1. Here, only the most significant time-series relationships defined by a threshold value, are elucidated from the transition matrix. All transitions $x_{i,j}$ with values in $\mathbf{A}$ that are greater than a user defined threshold (say 0.1) are flagged as positive, i.e. time-series x_i is influencing time-series x_j. By analogy, values below -0.1 are labeled as negative. The values and direction of influence are reflected in the directed network diagram.

The ability to capture time varying patterns of inter-relationships between multiple time-series is an important prerequisite to predicting the future values of the series. The DIN meets this requirement as it is able to evolve its structure and to learn incrementally with incoming data. As such a modeling and prediction process with on-line learning is implemented, whereby the transition matrix and the DIN model are updated at every time step. We will denote this model as eDIN in order to distinguish it from the original DIN model which did not posses an evolving capability.

As shown in Figure 2, a eDIN model is initially trained over a certain number of time steps, e.g. 50 time-points of data. Afterwards, the extracted eDIN model is used to predict the next time step's values. This eDIN model is then updated using equations [1−5] at the following time-point with new data arriving whereupon the updated eDIN model is used to predict values for next time step and so forth. This process is performed simply by using the recursive measurement update property of the Kalman filter and the EM algorithm as outlined in the algorithm above.

The incremental learning process then continues into the future, where prediction and training operations are interleaved with each other.

3.2 Local Modeling

We now turn our attention to local modeling. Traditionally time series prediction has been viewed as a global modeling activity whereby a single global model is induced from all available historical data [48] [47]. However, this may not be the optimal approach as global models often fail to track localized changes that take place at discrete points in time. This is due to the fact that trajectories produced by global models tend to smooth localized deviations by averaging the effects of such deviations over a long period of time [42].

In reality, localized deviations from the norm may be of great significance as they capture the conditions under which a time series behaves quite differently from its recent behavior. For example, financial markets react very favorably when interest rates are cut or when better than expected Economic fundamentals are announced by a Government under which they operate. Capturing such phenomena with a global model is difficult as it requires a discontinuity in the global trajectory function and this goes against the fundamental design philosophy behind the construction of any global model. This motivates a localized modeling approach. Several studies in time series prediction have shown that an ensemble of local models perform better than a single global model [7] [13] [50].

Local models are commonly created by forming clusters of instances that have many features in common. Clustering is ideally suited to building evolving models as new instances streaming in can be assigned to the cluster whose centroid most closely matches with the new instance, thus only requiring small localized (to the cluster in question) changes to be implemented without affecting the other local models comprising the ensemble. Thus clusters are naturally suited to data that changes over time. Furthermore, it is of interest to capture similar deviations from a global trajectory that take place repeatedly over time, in other words to capture recurring deviations from the norm that are similar in shape and magnitude. Such localized phenomena can only be captured accurately by localized models that are built only on data that defines the phenomenon under consideration and are not contaminated by data outside the underlying phenomenon. Our clustering approach is based on the Evolving Clustering Model (ECM) proposed by Kasabov and Song in [19]. We implement a two phased strategy in building local models. In stage 1, profiles of relationships between the time series variables are extracted and stored in a repository, while in stage 2 the detection and grouping of recurring trends of movement between time series is carried out when a particular profile emerges. It should be noted that the trends that we capture are not merely movements of a single time series but the pattern of movement of a group of time series with respect to each other. As an example consider the movement of 5 stock market indexes from New Zealand, Australia, Hong Kong, Japan and United States in the Pacific region. If one is able to learn that at a particular point in time that the New Zealand and Australia markets are moving together collectively, Hong Kong and Japan are progressing mutually, while the United States travels by itself, then it would be relevant to use only data of stock market indexes from the past which possesses the same profiles of relationships to predict future values of these stock market indexes, rather than to use the entire data set.

3.2.1 Extracting Profiles of Relationships from Multiple Time-Series

Most of the research in time-series clustering has concentrated on clustering at the instance level rather than on clustering at the level of variables [31]. However, one of the key tasks in our research is to group together series or variables, and not samples that are highly correlated or have similar shapes of movement, as it is our belief that multiple local models representing clusters of similar profiles will provide a better basis than a single global model for predicting future movements of the multiple time-series.

The first step in extracting profiles of relationships between multiple time-series is the computation of local cross-correlation coefficients between the observed time-series using Pearsons correlation analysis. Statistically significant correlations, which are determined through the use of the t-test with a confidence level of 95% is used. After the most significant correlations are identified, the Rooted Normalized One-Minus Correlation (RNOMC) coefficients [22] [31] (henceforth known as normalized correlation) is calculated to assess the local strength of the linear relationship between a pair of time series (a, b). The normalized correlation is given by:

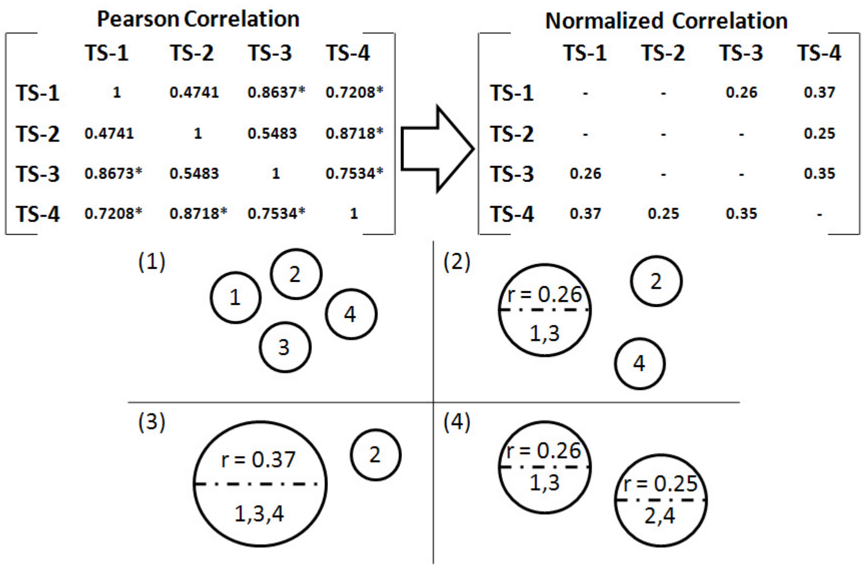
$$RNOMC(a,b) = \sqrt{\frac{(1 - corr(a,b))}{2}} \tag{6}$$

where $corr(a,b)$ is given by:

$$corr(a,b) = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}((a_i - \bar{a})^2 - (b_i - \bar{b}))^2}} \tag{7}$$

The normalized correlation coefficient defined by equation 6 ranges from 0 to 1,where 0 denotes high similarity and 1 signifies the opposite condition. Figure 3 illustrates the computation of the RNOMC measure from Pearson's correlation coefficient for a hypothetical example of 4 time series that are interacting with each other. The figure shows the statistically significant correlations at the 95% significance level from which the RNOMC measure is computed, which in turn drives the formation of two clusters, whereby series 1 and 3 fall into one cluster while series 2 and 4 end up in the other cluster.

Algorithm 1 outlines the scheme for clustering together similar time series. Basically variables that are highly correlated with each other are assigned to the same cluster, as is the case in lines 11 and 15 below. Once a variable is assigned to a particular cluster (say X) it may be reassigned at a later point in time to another cluster (Y) should its correlation with one or more variables in cluster Y may be greater than its correlation with its variables in the former cluster X, as is the case in lines 17 and 26.



Pearson Correlation

	TS-1	TS-2	TS-3	TS-4
TS-1	1	0.4741	0.8637*	0.7208*
TS-2	0.4741	1	0.5483	0.8718*
TS-3	0.8673*	0.5483	1	0.7534*
TS-4	0.7208*	0.8718*	0.7534*	1

Normalized Correlation

	TS-1	TS-2	TS-3	TS-4
TS-1	-	-	0.26	0.37
TS-2	-	-	-	0.25
TS-3	0.26	-	-	0.35
TS-4	0.37	0.25	0.35	-

Fig. 3: Formation of clusters of time series variables.

3.2.2 Clustering Recurring Trends of a Time-Series

After the profiles have been extracted, the next step is to mine and cluster trends of movement from each profile. With n as the number of multiple time-series being analyzed, the worst case time-complexity of Algorithm 1 is $O(n^2)$, which means that it will be necessary to store and dynamically update relationships profiles in order to avoid expensive re-computation and extraction of profiles. Maintaining profiles of relationships between multiple time-series ensures that we can identify the time-series that most influences the movement of other time-series in a particular time locality. However, this type of knowledge on its own does not offer predictive power to estimate the future values of the time-series ensemble. To ensure prediction with local models, it will be necessary to capture the different shapes of movement of time series within a profile and to utilize these shapes in constructing an ensemble of local models that can be used for prediction. Shapes of movement can be modeled and captured in many different ways, including the use of polynomial regression functions. Widiputra et al in [12] reports on the use of polynomial functions of degree 2 and degree 3 in capturing non linear relationships between time series in a stock market environment. However, it is problematic to know the degree of the polynomial that should be used and so for these reasons we shall use kernel regression that can deal with any given data distribution.

The local models constructed through the process of kernel regression are derived from the clusters defined by Algorithm 1. However due to the dynamic nature of time series data, it will be necessary to adapt the clusters formed to reflect evolving trends in the underlying data. Our dynamic clustering scheme has been inspired by Evolving Clustering Method (ECM) [19], [35]. ECM is a flexible clustering scheme whereby changes in the data are easily accommodated by adding a

```
BEGIN {Algorithm 1}
1: Input: X, where $X_1, X_2, ..., X_n$ are observed time-series
2: Output: profiles of relationships between multiple time-series
3: calculate the normalised correlation coefficient of X
4: C is the set of clusters
5: $\rho$ is the RNOMC correlation coefficient threshold
6: $C_k$ is the cluster in which vector $X_k$ (time series variable) belongs to
7: $c_{ij}$ denotes the correlation between variables $X_i, X_j$
8: $c_{ik}^{max}$ denotes the maximum correlation between variable $X_i$ and all other variables present
   in a given cluster $C_k$
9: for each time-series $X_1, X_2, ..., X_n$ do
10:    if ($c_{ij} > \rho$) and ($X_i, X_j$ belong to different clusters) then
11:        allocate $X_i, X_j$ together in a new cluster
12:    end if
13:    pre-condition: $X_i$ belongs to a cluster; $X_j$ does not belong to any cluster
14:    if ($c_{ij} > \rho$), $c_{jk} > \rho$, $\forall X_k \in C_i$) then
15:        allocate $X_i, X_j$ together in a new cluster
16:    else if ($c_{ij} > c_{ik}^{max}$,$c_{jk} \leq \rho$,$\forall X_k \in C_k$) then
17:        remove $X_i$ from its cluster; allocate $X_i, X_j$ together in a new cluster
18:    end if
19:    pre-condition $X_i$ and $X_j$ belong to different clusters
20:    if ($c_{ij} > \rho$) then
21:        if ($c_{ik} > \rho$,$\forall X_k \in C_j$ $c_{jl} > \rho$, $X_l \in C_i$) then
22:            merge cluster of $X_i$ with cluster of $X_j$
23:        else if ($c_{ij} > c_{jk}^{max}$, $X_k \in C_j, c_{jl} > \rho$, $\forall X_l \in C_i$) then
24:            remove $X_j$ from its cluster; allocate $X_j$ to cluster of $X_i$
25:        else if ($c_{ij} > c_{ik}^{max}$,$X_k \in C_i$, $c_{ij} > c_{jl}^{max}$, $X_l \in C_j$) then
26:            remove $X_i, X_j$ from their clusters
27:            allocate $X_i, X_j$ together in a new cluster
28:        end if
29:    end if
30: end for
31: return clusters of multiple time-series
END {Algorithm 1}
```

new instance to an existing cluster when sufficient similarity exists or creating a new cluster when the instance is sufficiently different from all existing clusters.

We now describe in detail the algorithm used to cluster recurring trends. The algorithm proceeds by breaking up the dataset into chunks of data. We gather a sample of the dataset as a training sample and perform a bootstrapping process to obtain the chunk size. The bootstrap is implemented by performing an auto-correlation analysis on the training sample to determine the chunk size. The time lag that yields the maximum correlation determines the chunk size. Thereafter we scan new incoming chunks of data and use the Nadaraya Watson kernel weighted average function [6] to estimate the trajectories of each time series variable at each data chunk is described in Algorithm 2 below.

BEGIN {Algorithm 2}

1. Create the first cluster C_1 and populate it with trajectories $(X_1, X_2, ..., X_n)$ for each of the time series variables. The fitted value $\hat{X}_{ij}$ at the j^{th} observation

$= \frac{\sum_{j=1}^{m} w_{ij} X_{ij}}{\sum_{j=1}^{m} X_{ij}}$ where m is the size of a data chunk; w_{ij} is the kernel weight of the j^{th} observation for variable i. Each $\hat{X}_{ij}$ is fitted by the Gaussian membership function: $exp(-\frac{(z-X_{ij})^2}{2\alpha^2})$, where α is the kernel bandwidth and z takes values in the range $[0, 1, 2, ..., (\frac{m}{dX} + 1)]$; dX is the sampling rate for each observation X_{ij}.

The kernel weights w_{ij} are estimated using ordinary least square fitting so that $SSE = \sum_{j=1}^{m}(X_{ij} - \hat{X}_{ij})^2$ is minimized. The i^{th} term of the centroid of the cluster is then given by $c_i = \frac{\sum_{j=1}^{m} w_{ij}}{m}$.

2. If there are no more data chunks the algorithm terminates; else the next data chunk or snapshot is read. The new set of kernel weights are estimated as in step 1 above and distances between the trajectory X_i and all k existing cluster centers are calculated by: $d_{il} = RNOMC(w_i, c_i^l)$ where w_i, c_i represents the kernel weight vector of trajectory X_i and the cluster centroid of cluster l respectively. If, for an existing cluster l we have $d_{il} < R_l$ where R_l is the radius of cluster l, then we assign the current trajectory X_i to cluster C_l, else we perform step 3 below.

3. Find a cluster e that requires the minimum extension of radius to accommodate X_i. This cluster is given by $e = \underset{1 \leq l \leq k}{\text{argmin}}(d_{il})$.

4. If $S_{i,e} \leq 2Dthr$ (which is a threshold that determines the maximum size of a cluster radius), the current trajectory X_i is assigned to cluster e. Cluster e is updated by moving its center, c_e, and increasing the value of its radius R_e. The updated radius R_e^{new} is set to $S_{i,e}/2$ where

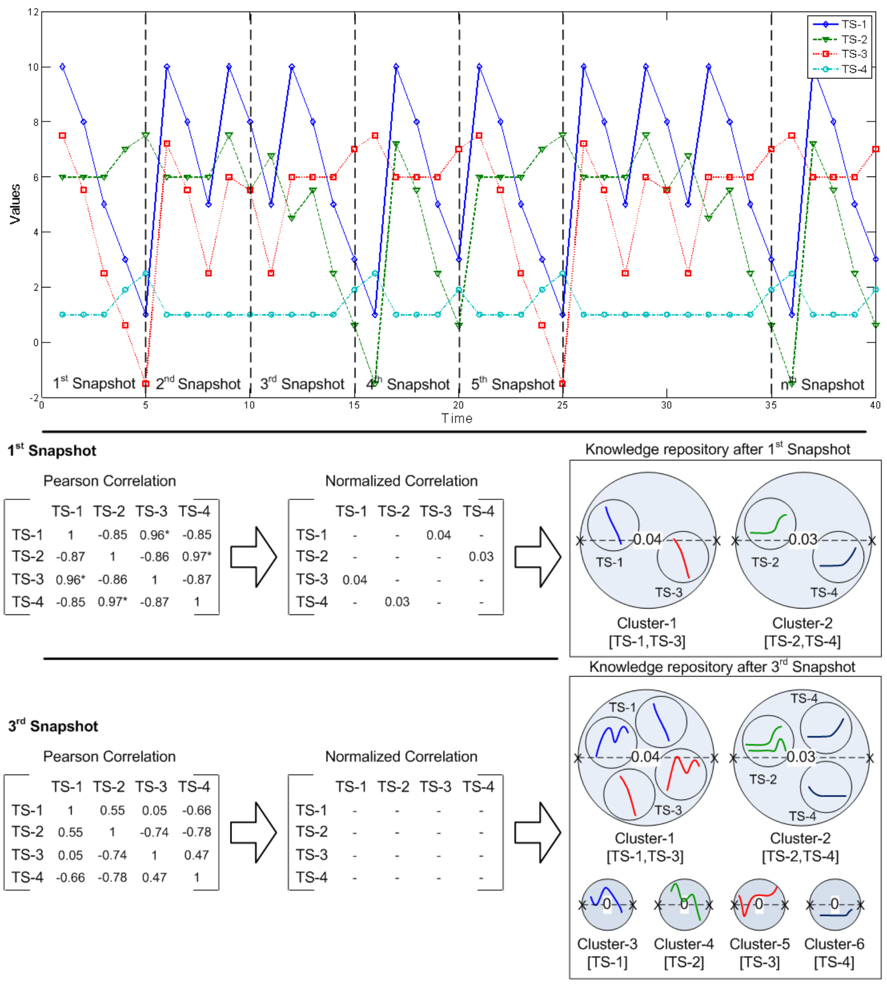
$$S_{i,e} = D_{i,e} + R_e \tag{8}$$

and the new center c_a^{new} is now the mean value of all trends of movement that belong to cluster e. Distance from the new center c_a^{new} to current trend w_i, is equal to R_{new}. The algorithm then returns to Step 2.

5. If $S_{i,a} > 2Dthr$, where Dthr, then X_i does not belong to any of the existing clusters. A new cluster is then created in the same way as described in Step 1, and the algorithm returns to Step 2;

6.The prediction of the next value of a given time series variable is made by considering its distance from all the cluster centers. Clusters that are closer contribute to a greater extent. The contribution of cluster j to variable X_i is given by: $w_{ij} = \frac{(max(D)-(D_{ij}-min(D))}{max(D)}$ where D_{ij} represent the distance between variable i and cluster j, while $min(D), max(D)$ represent the distance to the nearest and furthest clusters respectively.

END {Algorithm 2}



1st Snapshot

Pearson Correlation

	TS-1	TS-2	TS-3	TS-4
TS-1	1	-0.85	0.96*	-0.85
TS-2	-0.87	1	-0.86	0.97*
TS-3	0.96*	-0.86	1	-0.87
TS-4	-0.85	0.97*	-0.87	1

Normalized Correlation

	TS-1	TS-2	TS-3	TS-4
TS-1	-	-	0.04	-
TS-2	-	-	-	0.03
TS-3	0.04	-	-	-
TS-4	-	0.03	-	-

3rd Snapshot

Pearson Correlation

	TS-1	TS-2	TS-3	TS-4
TS-1	1	0.55	0.05	-0.66
TS-2	0.55	1	-0.74	-0.78
TS-3	0.05	-0.74	1	0.47
TS-4	-0.66	-0.78	0.47	1

Normalized Correlation

	TS-1	TS-2	TS-3	TS-4
TS-1	-	-	-	-
TS-2	-	-	-	-
TS-3	-	-	-	-
TS-4	-	-	-	-

Fig. 4: Creation of knowledge repository ltm-knowledge repository.

The predicted value is then given by: $X_i^{next} = \dfrac{\sum_{j=1}^{m} w_{ij} c_j}{\sum_{j=1}^{m} w_{ij}}$

Figure 4 illustrates how a repository containing trajectories are built and maintained over a period of time for a hypothetical dataset. After the arrival of the first data chunk, the first cluster contains time-series #1 (TS1) and time-series #3 (TS3) as they are correlated and moving together, while the second cluster captures the fact that time-series #2 (TS2) and time-series #4 (TS4) are progressing in a similar fashion. As the second data chunk becomes available, the same trends for each of the four time series remains the same, so the four trajectories in the two clusters are retained. However, the second chunk also contains two additional trends for TS1 and TS3, thus adding two new entries to cluster 1, while an additional trend for TS4 is created as the trend for TS2 remains the same. In the third data chunk all four time series are uncorrelated with each other, thus causing four additional clusters to be created, each containing one of the time series. The knowledge repository created acts as a knowledge base and is the key data resource to perform prediction.

3.3 Transductive Inference and Prediction

In contrast to global and local prediction systems where models are constructed from sets of instances, transductive prediction is made on a set of instances that most closely match with a given instance. In its most basic form, the learning is made with a k nearest neighbor algorithm and the prediction for a given instance is simply the average of the values of the nearest neighbors. In an early version of this research the K-NN and WK-NN algorithms were used to perform prediction of stock market indexes in the Asia Pacific region [45]. A WW-KNN method [16], that suggests using weighted importance of the variables in the local neighbourhood when weighing the distance between the instances, was applied on bioinformatics problems [16]. The results showed that transductive learning is capable of capturing recurring patterns of movement from the past.

In general, transductive reasoning has the advantage of using only the data that is most relevant to the instance given. It also has the potential to reduce the effect of outliers, since only a relevant subset of the instances in the dataset is used to derive the solution. Transductive learning has been used in wide range of applications such as data stream mining [34], text classification [14], bioinformatics [18] and surveillance [24], amongst others.

In this research we use a modified version of the Neuro Fuzzy Inference (NFI) system proposed by Song and Kasabov [36]. NFI is a dynamic neuro-fuzzy inference system which uses either Zadeh-Mamdani [49], or Takagi-Sugeno [37] types of fuzzy inference. In the Zadeh-Mamdani type of NFI model, Gaussian fuzzy membership functions are applied to each fuzzy rule on both antecedent and consequent parts, while for the Takagi-Sugeno variation, the consequent part is represented by a linear or non-linear function. A back propagation learning algorithm is used for optimizing the parameters of the fuzzy membership functions. We make two modifications to the basic NFI algorithm. We use the first order rate of change of value instead of the absolute value for each time series variable. The second change involves the use of Correlation Coefficient instead of Euclidean distance measure to quantify similarity level between instances when locating nearest neighbors. Figure 5 shows how prediction is performed with our modified version of NFI which we refer to as mNFI.

4 An Integrated Framework for Multiple Time Series Prediction

All three approaches discussed so far with respect to prediction have their own strengths and limitations. Global modeling has the advantage of using a larger quantity of data but a possible drawback is that global models may fail to track localized changes that take place at discrete points in time. Local models on the

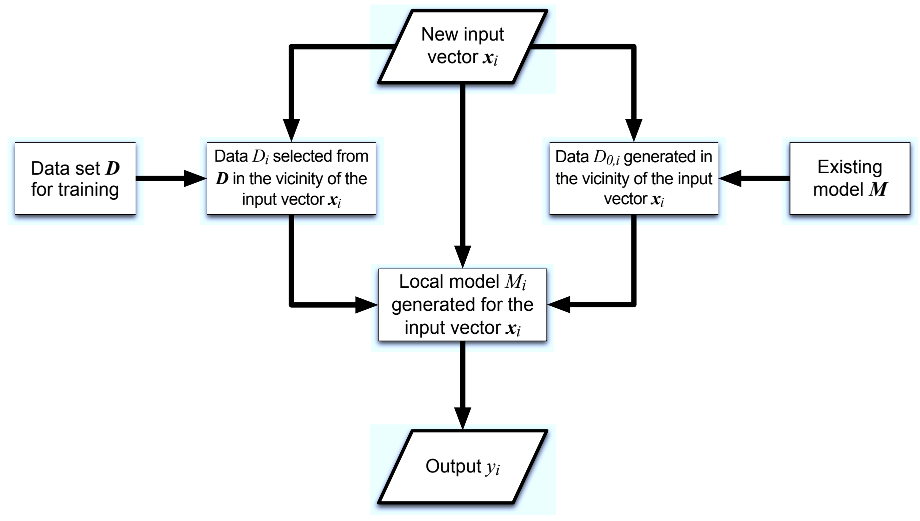

New input vector $\boldsymbol{x}_i$
Data set $\boldsymbol{D}$ for training
Data D_i selected from $\boldsymbol{D}$ in the vicinity of the input vector $\boldsymbol{x}_i$
Data $D_{0,i}$ generated in the vicinity of the input vector $\boldsymbol{x}_i$
Existing model $\boldsymbol{M}$
Local model M_i generated for the input vector $\boldsymbol{x}_i$
Output y_i

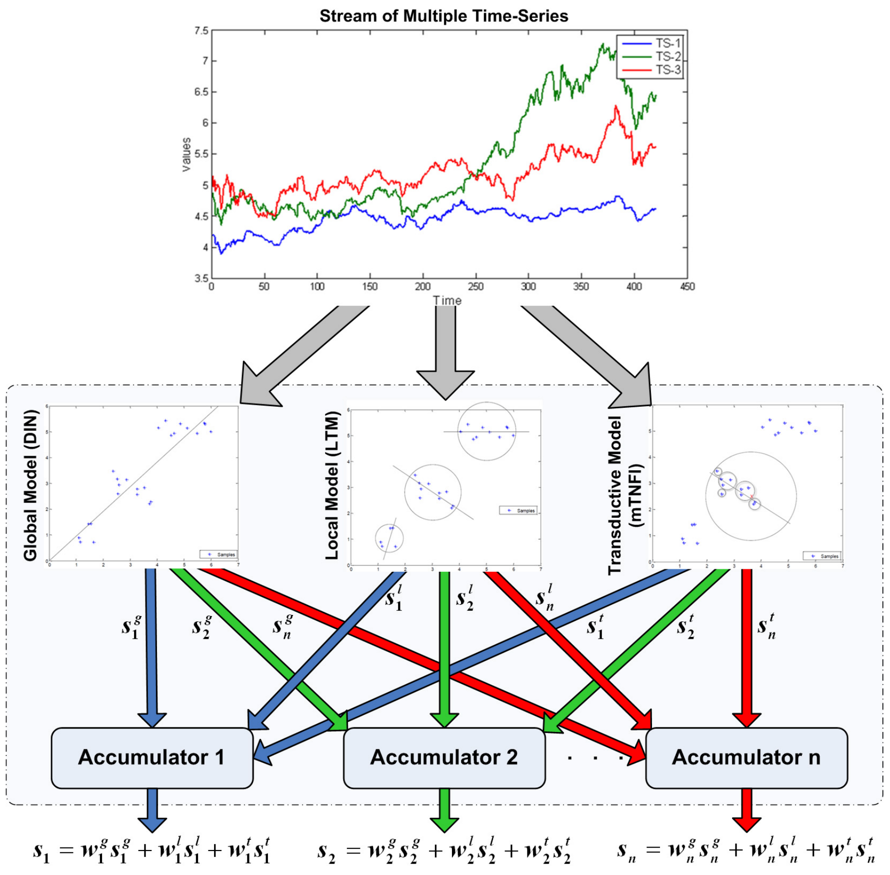
Fig. 5: Transductive prediction with NFI

other hand have the advantage of capturing changes that take place in different time localities and are also more flexible as prediction can be performed either on the basis of a single model that covers a subset of the space or from an ensemble of local models that collectively cover the entire space. Transductive models take the local model concept a step further by personalizing prediction to instances rather than subsets of data. Given the advantages of each approach, it is of interest to investigate how they would perform together in one integrated framework. Such an integrated framework should be capable of weighing the contributions of each approach and form a composite prediction based on a weighted sum of the predicted values from each of the three different models.

Kasabov [16] has proposed such an integrated framework in the Bioinformatics domain. However, no specification as to how such a framework can be actually put into practice was given. One of the key contributions of this research is a detailed scheme for implementing the integrated framework. Such an implementation fundamentally involves solving a linear optimization problem of finding weights w_G, w_L and w_T that represents the importance of global, local and transductive models respectively. Figure 6 shows from a high level perspective how the global, local and transductive models can be combined into one single framework.

The requirement that the weights be adapted as the data distribution of each series changes continuously in time suggests the use of a neural network architecture for solving this problem. However, since the optimization required is linear in nature a simple architecture based on a single hidden layer with a single neuron will be sufficient to determine an accurate solution. Employing such a simple architecture has the important advantage of efficiency as a relatively straightforward back propagation process can be used to estimate the optimal weights.

Figure 7 presents the Adaline neural network [46] as a solution to the weight optimization problem in model integration. The Adaline neural network has a feed forward structure and in the learning phase the weights are adjusted according to



Stream of Multiple Time-Series
TS-1
TS-2
TS-3
Values
Time
Global Model (DIN)
Local Model (LTM)
Transductive Model (mTNFI)
Samples
s_1^g
s_2^g
s_n^g
s_1^l
s_2^l
s_n^l
s_1^t
s_2^t
s_n^t
Accumulator 1
Accumulator 2
Accumulator n
$s_1 = w_1^g s_1^g + w_1^l s_1^l + w_1^t s_1^t$
$s_2 = w_2^g s_2^g + w_2^l s_2^l + w_2^t s_2^t$
$s_n = w_n^g s_n^g + w_n^l s_n^l + w_n^t s_n^t$

Fig. 6: Adalaine network architecture for weight adaptation and optimization .

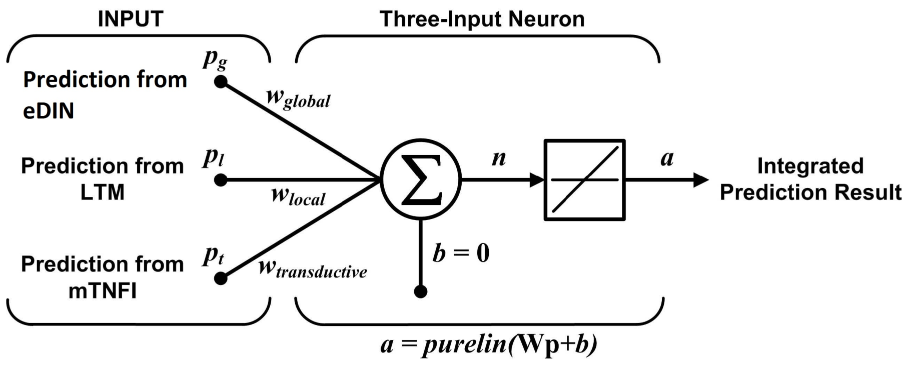


Fig. 7: Adalaine network architecture for weight adaptation and optimization .

the weighted sum of the inputs, instead of passing inputs through an activation (transfer) function in the standard perceptron model. The *purelin(wp+b)* represents a linear function of the weight vector and the inputs which are the predictions made by each of the global, local and transductive models. A bias b is introduced in general in an Adaline network to model any external interference in the system. In our case we consider the optimization to represent a closed system with no external interference, so b is set to 0. The objective function to be minimized is given by:

$$s_i = w_i^g p_i^g + w_i^l p_i^l + w_i^t p_i^t, i = 1, 2, ..., n \tag{9}$$

where n is the number of time series and the p vectors correspond to the prediction from each of global, local and transductive models. In effect the network presented in Figure 7 is a simplified version as an input is required for each combination of time series variable and model type.

The learning algorithm employed in the optimization process is outlined as follows:

BEGIN {Algorithm 3}
Step 1: Calculate the mean absolute error of the prediction for each combination of time series variable and model type. The weight vectors are initialized by mapping the absolute error of predictions at time step 0 to a Gaussian membership function where the model with the smallest absolute error is assigned to the center of the membership function.

Step 2: Weight vectors at time step t are updated based on the difference between the prediction value of the objective function equation 9 and the actual value at time step $t - 1$.

Step 3: The optimization process terminates when the number of epochs reaches a given threshold value or the predictions obtained by using the weight vectors on the integrated model is less than some user defined tolerance value.
END {Algorithm 3}

As our experimentation shows, the integrated model improves the accuracy of prediction. The key to this improvement is the adaptation of the weights to the changing data distribution. Thus for example, when conditions for the local model are not optimal for it to perform best it assigns more weight to the predictions made by the global and transductive models, thus compensating for shortcomings in each of the models at different points in time.

5 Experimental Study

Our experimentation took two different forms. We first experimented with synthetic data in order to conduct a performance study of each of the models under controlled conditions. Secondly, we conducted two real life studies from two different application domains, namely stock market index prediction and prediction of air pressure levels across geographically dispersed sites in a selected country. We start by describing our experimentation with synthetic data.

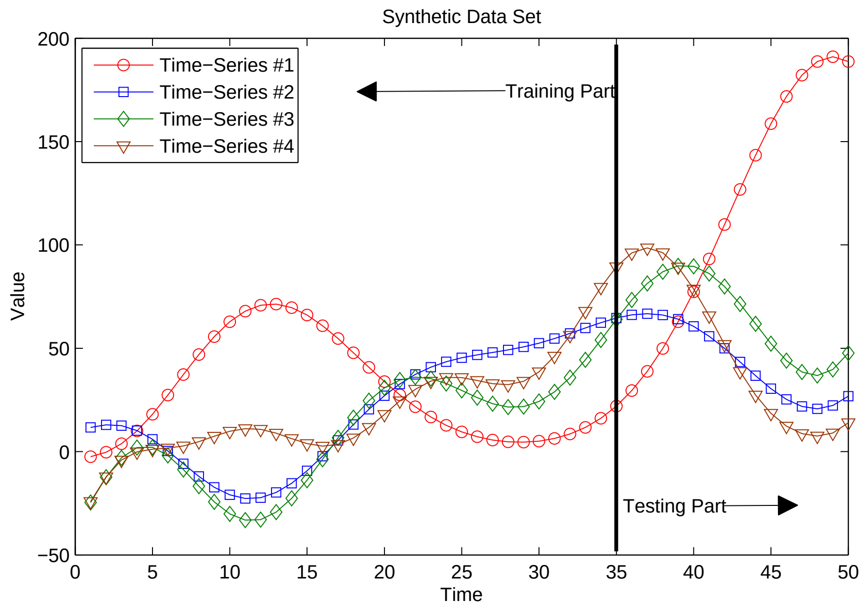


Fig. 8: Trajectories of synthetic time series.

5.1 Experimentation on Synthetic Data

We generated 4 different synthetic time series exhibiting different characteristics in order to test the performance of the different prediction models. We modeled the time series using 3 different polynomial functions of degree 6, thus producing non linear trajectories. The functions are given by:

$$f_1(x) = \frac{1}{30}(x^6 - 22.5x^5 + 189.5x^4 - 735.75x^3 + 1296.56x^2 - 822.66x + 180) \quad (10)$$

$$f_2(x) = 0.08x^6 - 1.8x^5 + 15.16x^4 - 58.86x^3 + 103.72x^2 - 65.81x + 8 \quad (11)$$

$$f_3(x) = -0.1x^6 - 2.25x^5 + 18.95x^4 + 73.5x^3 - 1296x^2 + 85x - 7.5 \quad (12)$$

In order to inject inter-dependency, another 4 functions were derived from expressions [10−12] as follows: $S_1 = f_1(x) \times f_2(x)$ $S_2 = f_1(x) \times f_3(x)$ $S_3 = f_2(x) \times f_3(x)$ $S_4 = \frac{(f_3(x) \times S_3)}{10}$
Figure 8 shows that all of the 4 time series have non linear trajectories and exhibit dynamically changing dependencies. Thus for example, in the initial period (from around time step 2 through to about time step 30) , time series 2 and 4 are strongly interrelated, whereas from around time step 30 to time step 35 in the training phase, it is series 3 and 4 that have strong dependencies on each other.

Before examining the performances of the 4 different models on the time series, an understanding of the shifting dynamics of the 3 basic models (i.e the global, local and transductive) over time is needed. This in turn will help to understand the role that the Integrated model plays. Figure 9 shows that the relative weights of the models change over time for time series 1. At the start and at the end, the

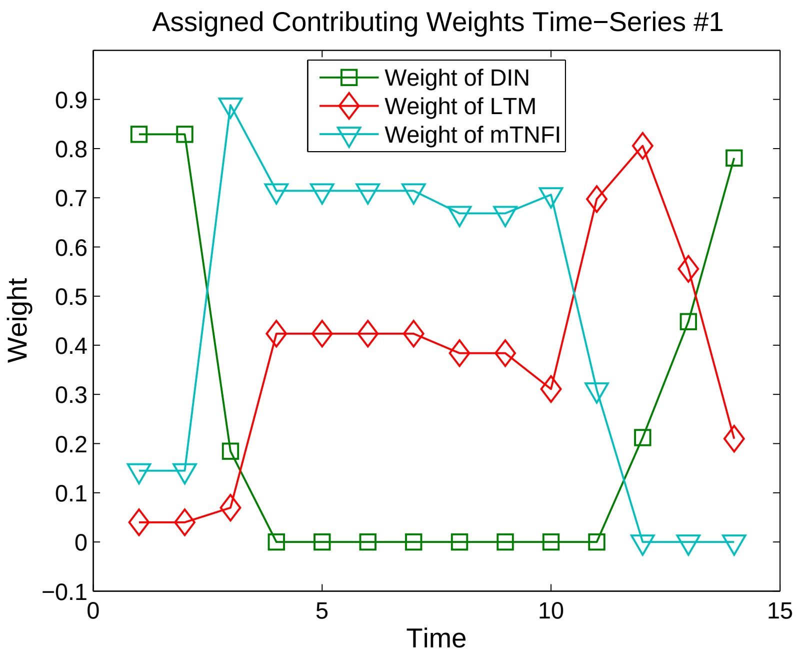


Fig. 9: Weight dynamics of the basic models for time series 1

global (eDIN) model tends to dominate over the two models but at other points either the Local (LTM) or Transductive (mTNFI) models have the highest weight, thus showing that all 3 models play an important part in the prediction process. This also demonstrates that the Integrated model does indeed have a positive role to play as it will shift emphasis from one model to the other as time goes on, and thus can be expected to result in an improved prediction accuracy than any of the 3 basic models when taken over a given period of time.

Whilst the details of the weight dynamics for the other 3 time series are different, the same basic trend of shifting emphasis of the 3 models over time is present and thus we have omitted presentation of these figures due to space constraints. We are now in a position to examine the performance of the models on the 4 time series. Table 1 gives the Root Mean Square (RMS) error of the predictions for each of the time series on the testing segments of the 4 datasets. Consistent with Figure 9 the table shows that no one of the 3 basic models performed consistently the best for all time series, although the LTM and mTNFI performed very well. The integrated model was consistently the best performer, in line with expectations given that the distribution of weights between the models varied widely over time. While it would be tempting to say that the eDIN (global) model was redundant given that it consistently gave the highest RMS value, it plays an important part in the overall prediction process as it contributes strongly to the predictions at certain points in time as indicated by Figure 9. Moreover the eDIN model is able to describe in a graphical manner the interactions between the time series, as demonstrated in Figure 1 and is thus a valuable tool in time series prediction.

Table 1: Relative Error Rates of eDIN, LTM, MTNFI and IMMF Models

Variable	eDIN	LTM	mTNFI	IMMF
Series 1	11.97	7.87	7.24	4.78
Series 2	5.51	4.19	6.38	2.69
Series 3	7.47	4.86	3.09	2.35
Series 4	7.69	5.48	1.97	1.58

Table 2: Relative Error Rates of IMMF Models versus Traditional Time Series Prediction Methods

Variable	IMMF	MLR	MLP	DENFIS
Series 1	4.78	62.06	15.14	14.32
Series 2	2.69	43.73	7.57	4.65
Series 3	2.35	37.06	7.66	6.10
Series 4	1.58	10.19	8.77	8.86

We were also interested in assessing performance of established prediction methods such as Multiple Linear Regression (MLR), Multi Layer Perceptron and DENFIS agianst the IMMF model. These methods, although widely used in single time series prediction were not designed to capture interactions between multiple time series methods and so it would be interesting to see how they cope against methods such as eDIN, LTM and mTNFI that were designed to operate in a multiple time series environment. Table 2 shows clearly that the multiple time series methods outperform their single time series counterparts. As expected, MLR was the worst performer. This is for two reasons, one is the fixed linear trajectory that MLR uses. Secondly, the trajectory was estimated on the training segment where the data behaves quite differently from the testing segment, and since MLR has no adaptive capability the prediction accuracy is severely affected. MLP, on the other hand is able to cope better on account of its non linear capability but it too suffers from the fact that its model (trajectory) is determined from the training segment and since it has no adaptive capability its accuracy is also significantly lower than that of the multiple time series models. DENFIS is interesting due to the fact that it has adaptive capability. Thus it uses the training segment to bootstrap a model which it incrementally updates during the testing segment. The evolving nature of DENFIS helps it to perform best amongst the single time series methods. Despite this it still cannot perform as well as the multiple time series methods as it does not take into account the interactions between variables, thus confirming the importance of capturing inter-dependencies between variables.

Having established the superiority of multiple time series methods on different synthetic time series we next examine their performance on two real world case studies, starting with the Stock Market Case Study.

5.1.1 Modeling of Stock Market Indexes in the Asia Pacific Region

The globalised security markets of today are characterized by interdependencies, and often demonstrate contagious behavior in periods of crisis. A study by Phylaktis and Ravazzolo [30] found that the relationship between markets is stronger

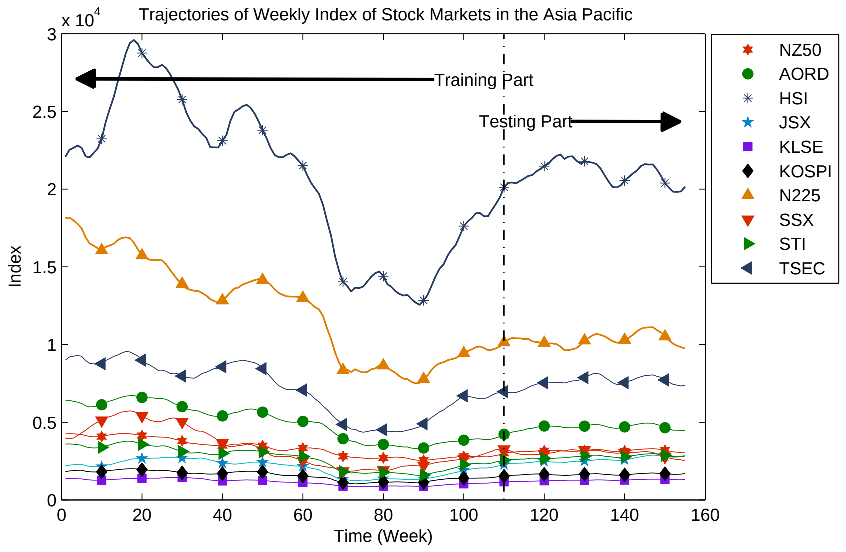


Fig. 10. Trajectories of selected markets in Asia Pacific region

than in the past, due to the relaxation of foreign ownership restrictions. Other studies also found that movement in local stock markets are generally influenced by major players in Europe, North America and Australasia. The existence of known interdependencies between markets thus makes the stock market environment an ideal one to study interactions between multiple time series variables. The 10 markets that we have selected are: Australia AORD, Hong Kong HSI, Indonesia JSX, Malaysia KLSE, South Korea KOSPI, Japan Nikkei 225, New Zealand NZ50, Shanghai China SSX, Singapore STI and Taiwan TSEC, all within the Asia Pacific region. Figure 10 depicts the trajectories of the 10 stock markets during the period of study. As Figure 10 shows, markets such as Japan's Nikkei and Hong Kong's HSI showed a greater degree of volatility than the others in the study period.

Figure 11 shows the prediction accuracies of the models on the most volatile market, Japan's Nikkei 225. The plot shows that IMMF is by far is the most accurate model, consistent with our results from synthetic data. The same trends hold true for the other markets as Table 3 shows. Also consistent with the synthetic time series the mTNFI model on its own performed very well and outperformed the eDIN and LTM models on all markets except for the Hong Kong HSI. Even though the eDIN did not emerge the winner in any of the markets, weight analysis shows that it was a significant contributor to the overall prediction for all markets at various different time points in the study period.

Table 4 shows that IMMF significantly outperformed all of the single time series prediction methods, consistent with the experimentation on synthetic data.

An analysis of the network generated by the eDIN model at a selected time point shows some interesting interactions between the markets. Figure 12 shows that no market moves in isolation, every single market influences or is being influenced by other markets. This result supports findings from previous studies that

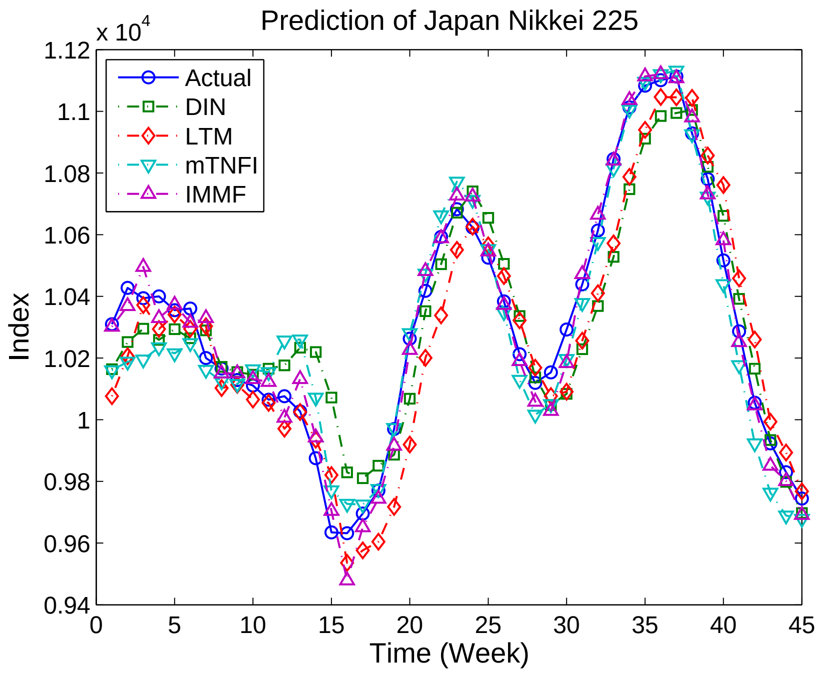


Fig. 11: Prediction for Nikkei 225

Table 3: Relative Error Rates of eDIN, LTM and mTNFI and IMMF Models for Selected Stock Markets

Stock Market	eDIN	LTM	mTNFI	IMMF
Australia AORD	68.86	60.59	36.72	25.13
Hong Kong HSI	330.44	254.44	258.99	156.33
Indonesia JSX	47.61	29.31	21.11	19.85
Malaysia KLSE	12.40	9.40	9.02	5.29
South Korea KOSPI	28.03	20.23	15.49	13.56
Japan Nikkei 225	156.06	149.46	105.96	61.81
New Zealand NZ50	38.11	27.16	17.8	8.90
Shanghai China SSX	56.56	48.14	40.43	30.78
Singapore STI	50.81	27.25	24.81	16.10
Taiwan TSEC	112.34	94.77	89.02	57.36

analyzed the existence of interdependencies in global stock markets [9] [26] [44]. Figure 12 clearly shows how the two leading markets in the Asia Pacific, the Hong Kong HSI and Japan Nikkei 225, affect the other markets significantly. Interestingly, the network also shows that movement of the Shanghai's SSX is affected by most of the other markets in the region (as indicated by the inward arrows pointing into SSX). Again, this finding is consistent with a previous study conducted by Widiputra et al [44] which spanned a different time period from the one used in this study. Thus the structure of the network that is derived from the eDIN model has the potential to reveal insights into dependencies between markets which are

Table 4: Relative Error Rates of IMMF Models versus Traditional Time Series Prediction Methods

Stock Market	IMMF	MLR	MLP
Australia AORD	25.13	171.80	88.52
Hong Kong HSI	156.34	460.38	428.70
Indonesia JSX	19.85	77.36	41.72
Malaysia KLSE	5.29	36.45	26.65
South Korea KOSPI	13.56	47.62	28.60
Japan Nikkei 225	61.81	311.88	293.48
New Zealand NZ50	8.90 126.17 48.99		
Shanghai China SSX	30.78	175.56	105.78
Singapore STI	16.10	75.30 60.75	
Taiwan TSEC	57.36	198.68	141.32

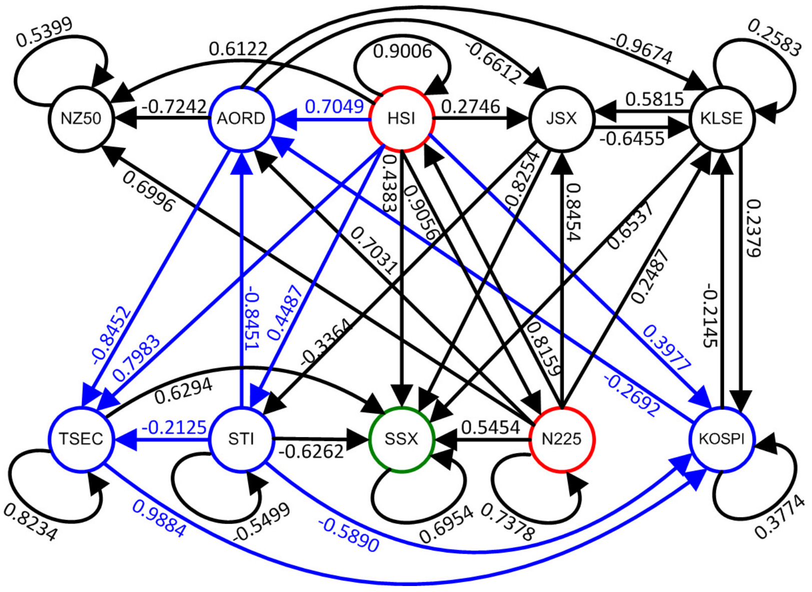


Fig. 12: Interaction between the Stock Markets in the Asia Pacific region

ultimately reflect economic policies set by countries and their trading patterns. With respect to the inward flows into China's SSX, the following explanations can be offered. Firstly, China has become one of the largest investors in many Asian countries. Secondly, by proposing to negotiate a free trade agreement with the ASEAN (Association of South East Asia Nations) countries, China has offered to share the benefits of its economic growth with countries in the region while establishing its influence over the smaller economies in Asia. All these contribute to the unique position of China, and to the number of vertexes that it is involved with in the interaction network.

Furthermore, the network also identifies the existence of interactions between Australia's AORD, Hong Kong's HSI, South Korea's KOSPI, Singapore's STI, and Taiwan's TSEC, which is in agreement with previous findings by Masih and Masih

[29] in their research on the dynamics of stock market interdependency in 1998 which found that these 5 stock markets are interdependent on each other.

It is also important to note that the interdependencies mentioned above are sustained throughout the study period although the exact degree of dependencies between markets varies with time. The LTM and mTNFI models also capture inter-dependencies between markets in the form of clusters and fuzzy rules respectively, both of which evolve continuously in time. Details of these are in [43] and [41]. In conclusion, we note that each of the globalized (eDIN), localized (LTM) and transductive (mTNFI) approaches play an important part in analyzing and accurately predicting the movement of stock markets, while the integrated model provides an even greater degree of accuracy by weighing the relative contributions from each of these models over time.

5.1.2 Modeling Weather Patterns across selected Weather stations in New Zealand

Weather prediction is a complex activity as a number of factors such as wind direction and intensity, temperature, air pressure, moisture content, among others are needed to accurately forecast the weather pattern at any given point in space and time. In this Case Study we concentrate exclusively on one factor, namely the prediction of air pressure across selected weather stations in New Zealand which are geographically dispersed from each other. Findings from a previous study on global weather systems revealed that small changes to one part of the system led to a complete change in the weather system as a whole [38]; as such this was the key motivation behind experimenting with such data.

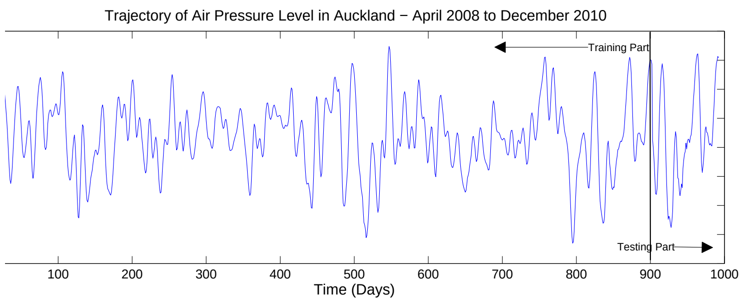


Fig. 13: Air Pressure Trajectory in the Auckland weather station during study period

Air pressure readings are recorded daily from various locations in New Zealand by the National Institute of Weather and Atmosphere, New Zealand. The dataset used in this study covers a period of almost three years, ranging from the beginning of April 2008 to the end of December 2010. Spatial coordinates were used to define the multiple variables, with the air pressure at four different locations in New Zealand (Auckland, Hamilton, Paeroa and Reefton) comprising the multiple time series variables. These weather stations were carefully chosen so as to reflect a range of proximity, with Auckland and Hamilton being relatively close being around 125 kilometers apart, whereas Auckland and Reefton are approximately 988 kilometers apart. The trajectories of observed air pressure data from two of

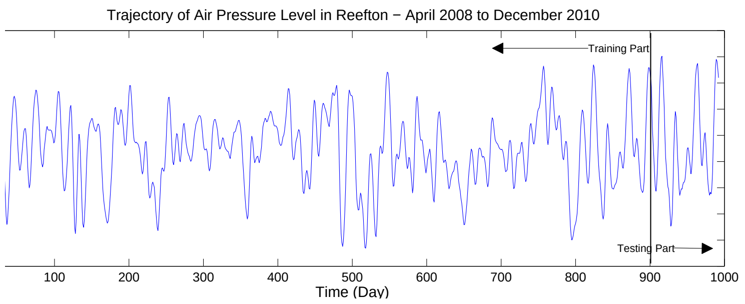


Fig. 14: Air Pressure Trajectory in the Reefton weather station during study period

Table 5: Relative Error Rates of eDIN, LTM, mTNFI and IMMF on Weather data

Station	eDIN	LTM	mTNFI	IMMF
Auckland	1.99	1.10	1.01	0.64
Paeroa	1.97	1.10	0.98	0.60
Hamilton	1.96	1.11	1.02	0.65
Reefton	2.46	1.31	1.05	0.65

Table 6: Relative Error Rates of IMMF Models versus Tradtional Time Series Prediction Methods on Weather data

Station	IMMF	MLR	MLP
Auckland	0.64	3.52	3.04
Paeroa	0.60	3.43	3.16
Hamilton	0.65	3.73	3.50
Reefton	0.65	4.17	3.91

these stations, namely Auckland and Reefton are illustrated in Figures 13 and 14. The plots show that the air pressure profiles between the two stations are similar during the majority of the study period, however Reefton experienced more severe low pressure points than Auckland at several different points in the study period. The air pressure profiles for the other two stations had a greater degree of similarity with that of Auckland and were thus omitted.

Table 5 shows that IMMF outperforms the eDIN, LTM and mTNFI methods on their own, consistent with our experimentation on synthetic data and the stock market dataset.

We also tested IMMF against the single time series methods such as MLR and MLP to verify the significance of including interactions in the prediction process for this dataset. Table 6 shows that IMMF is vastly superior to both MLR and MLP, once again reinforcing the need for incorporating interdependencies into the prediction process.

Figure 15 shows that interactions are present between all weather stations but the interactions between Auckland, Paeroa and Hamilton are the greatest. Paeroa and Hamilton affect Auckland pressure whereas the opposite is not true. This can be explained in terms of the geography of the locations: Auckland is located on a narrow spit of land which is around 25 kilometers at its widest from West to East

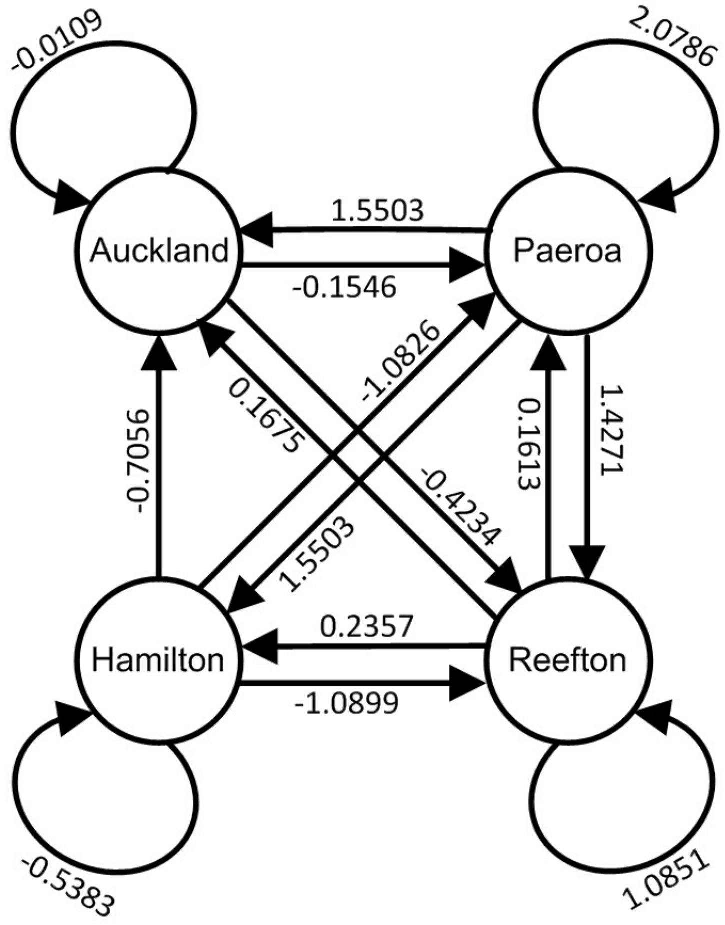


Fig. 15: Interactions between stations during study period

coast and is therefore vulnerable to weather systems developing from Paeroa to its East and Hamilton to its South.

Profiles of relationships extracted from the Local model (LTM) shows that in the period of study there is a significant number of occasions (437) when the trajectories of Auckland, Hamilton and Paeroa are moving together while Reefton moves independently the vast majority of the time (516). This is in agreement with the findings of the Global (eDIN) model discussed above. In conclusion, the results of this study are in close agreement with that of the Stock Market study and the synthetic data on the performance of the multiple time series models.

6 Conclusion and Future Work

This research shows that in terms of prediction accuracy, special purpose models built to capture interactions between multiple time series perform much better than established methods for single time series prediction. Furthermore, the integrated model that was proposed in this research performed better than any of the multiple time series methods on their own, suggesting that in multiple time series data with varying dynamics, the predictions from each of the methods need to be corrected or adjusted by predictions from the other two methods. This process was accomplished by feeding the predictions into an Adaline neural network that dynamically adjusted the relative contributions from each of the models over time.

The research also showed that valuable information could be extracted from the models developed. The Global method produced an interaction network that changed its structure over time. The vertexes of the network represented interactions between nodes (e.g. country specific stock markets) which could be interpreted in terms of domain specific events such as economic policies set by countries leading to strong trading patters with other countries. In the case of the weather case study, the air pressure at certain weather stations was affected by the pressure at other stations in their proximity on account of their geographical position.

In terms of future work there are a number of ways that this work can be extended. First of all, the Global method (eDIN) currently only captures linear relationships between variables. Non linear relationships can be captured by simply incorporating the Extended Kalman Filter but then the challenge will be to adapt the prediction process in an on line manner as we have done in the case of the basic version of the filter. Along the same lines a correlation ratio can be used instead of Pearson's correlation to represent non linear relationships with the Local method. Another challenge would be to automate the process of parameter selection. Each of the methods have a number of parameters which are currently set by hand based on performance with the training dataset. The logical next step would be the optimization of these parameters using evolutionary computation methods such as quantum inspired Particle Swarm Optimization [21].

Finally we note that this research has not conducted a performance study to determine the run time overhead introduced by the methods used such as the Global, Local and Transductive methods used in the prediction process. However, we observe that all of these methods make only one pass through the data and build their models incrementally thus making them suitable for on line prediction. However a study that quantifies the computational costs of each of the models is desirable in order to further optimize the performance of these methods.

Acknowledgement The work was funded by the Knowledge Engineering and Discovery Research Institute ($KEDRI, www.kedri.info$) and the School of Computing and Mathematical Sciences of the Auckland University of Technology, New Zealand.

References

1. Angelov, P. A., Dimitar P. Filev.: An Approach to Online Identification of Takagi-Sugeno Fuzzy Models IEEE Transactions on Systems, Man and Cybernetics - Part B **34:1**. 484–498 (2004)
2. Dovzan, D.; Logar, V.; Skrjanc, I.: Solving the sales prediction problem with fuzzy evolving methods IEEE Congress on Evolutionary Computation (2012) 1–8 (2012)
3. Angelov, P. A., Filev, D..: Simpl_eTS: a simplified method for learning evolving TakagiSugeno fuzzy models In: Proceedings of the The 14th IEEE Internat. Conf. on Fuzzy Systems (2005) 1068–1073

4. Angelov, P.A., Filev, D., Kasabov N.: Evolving Intelligent Systems: Methodology and Applications, John Wiley & Sons 2010.
5. Antoniou, A., Pescetto, G., Violaris, A.: Modelling international price relationships and interdependencies between the stock index and stock index futures markets of three eu countries: A multivariate analysis. Journal of Business Finance & Accounting, **30**, 645–677 (2003)
6. Bierens, H.: The Nadaraya-Watson kernel regression function estimator. Tech. rep., NATO ASI Series (1994)
7. Cevikalp, H., Polikar, R.: Local classifier weighting by quadratic programming. IEEE Transactions on Neural Networks **19**, 18321838 (2008)
8. Chiang, T., Doong, S.: Empirical analysis of stock returns and volatility: Evidence from seven asian stock markets based on TARGARCH model. Review of Quantitative Finance and Accounting **17:3**, 301–318 (2001)
9. Chowdhury, A.: Stock market interdependencies: Evidence from the asian NIEs. Journal of Macroeconomics **16:4**, 629–651 (1994)
10. Collins, D., Biekpe, N.: Contagion and interdependence in african stock markets. South African Journal of Economics **71:1**, 181–194 (2003)
11. Davidson, E.H.: The regulatory genome : gene regulatory networks in development and evolution. Academic, Burlington, MA (2006). URL `http://www.loc.gov/catdir/enhancements/fy0668/2006445256-d.html`
12. H Widiputra, H., Kho, H., Lukas, Pears, R., Kasabov, N.: A Novel Evolving Clustering Algorithm with Polynomial Regression for Chaotic Time-Series Prediction. In: Proceedings of the 16th International Conference on Neural Information Processing: Part II series, ICONIP (2009).
13. Islam, M., Yao, X., Nirjon, S., Islam, M., Murase, K.: Bagging and Boosting Negatively Correlated Neural Networks. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSPART B: CYBERNETICS **38:3**, 771–784 (2008)
14. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the sixteenth international conference on machine learning, ICML (1999)
15. Kasabov, N.: Evolving fuzzy neural networks for supervised/unsupervised on-line knowledge-based learning. IEEE Transactions on Systems, Man and Cybernetics Part B**31**, 902–918 (2001)
16. Kasabov, N.: Global, local and personalised modeling and pattern discovery in bioinformatics: An integrated approach. Pattern Recognition Letters **28:6**, 673–685 (2007)
17. Kasabov, N., Chan, Z., Jain, V., Sidorov, I., Dimitrov, D.: Gene regulatory network discovery from time-series gene expression data: a computational intelligence approach. Lecture Notes in Computer Science **3316**, 13331353 (2004)
18. Kasabov, N., Pang, S.: Transductive support vector machines and applications in bioinformatics for promoter recognition. In: Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, (2003)
19. Kasabov, N., Song, Q.: DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Transactions on Fuzzy Systems **10:2**, 144–154 (2002)
20. Kasabov, N.: (Evolving Connectionist Systems: The Knowledge Engineering Approach), second edition, Springer Verlag, London, 2007.
21. Kasbov, N., Hamed, H.: Quantum-inspired Particle Swarm Optimisation for Integrated Feature and Parameter Optimisation of Evolving Spiking Neural Networks. International Journal of Artificial Intelligence **7:11**, 114–124 (2011)
22. Kim, R., Ji, H., Wong, W.: An improved distance measure between the expression profiles linking co-expression and co-regulation in mouse. BMC Bioinformatics **7**, 1–8 (2006)
23. Levine, M., Davidson, E.H.: Gene regulatory networks for development. Gene regulatory networks for development Proceedings of the National Academy of Sciences of the United States of America **102**(14), 4936–4942 (2005). DOI 10.1073/pnas.0408031102. URL `http://dx.doi.org/10.1073/pnas.0408031102`
24. Li, F., Wechsle, H.: Watch List Face Surveillance Using Transductive Inference. In: ICBA'04 (2004)
25. Li, X., Chen, H., Li, J., Zhang, Z.: Gene function prediction with gene interaction networks: a context graph kernel approach. Trans. Info. Tech. Biomed. **14:1**, 119–128 (2010)
26. Lucey, B., Muckley, C.: Global stock market interdependencies and long-term portfolio diversification. Tech. rep., SSRN eLibrary (2010)

27. Lughofer, E.:FLEXFIS: A Robust Incremental Learning Approach for Evolving TakagiSugeno Fuzzy Models IEEE Transactions on Fuzzy Systems. **16:6**, 1393–1410 (2008)
28. Lughofer, E.:Evolving Fuzzy Systems - Methodologies, Advanced Concepts and Applications, Springer 2011
29. Masin, R., Masih, A.: Dynamic modeling of stock market interdependencies: An empirical investigation of australia and the asian NICs. Review of Pacific Basin Financial Markets and Policies **4:2**, 235264 (2001)
30. Phylaktis, K., Ravazzolo, F.: Stock market linkages in emerging markets: implications for international portfolio diversification. Journal of International Financial Markets **15:2**, 91–106 (2005)
31. Rodrigues, P., Gama, J., Pedroso, J.: Hierarchical clustering of time-series data streams. IEEE Trans. on Knowl. and Data Eng. **20**, 615–627 (2008)
32. Hai-Jun, R., Sundararajan, N., Guang-Bin, H., Saratchandran, P.:Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinearsystem identification and prediction, Journal of Fuzzy Sets and Systems **157:9**, 1260–1275 (2006)
33. S Ozawa, S.P., Kasabov, N.: Online Feature Extraction for Evolving Intelligent Systems. IEEE Press (2010)
34. Shaker, A., Hullermeier, E.: Instance-Based Classification and Regression on Data Streams. In: Sayed-Mouchaweh, M., Lughofer, E.,(eds), Learning in Non-Stationary Environments, Springer Verlag, 185–201, (2012).
35. Song, Q., Kasabov, N.: ECM - a novel on-line, evolving clustering method and its applications. In: Posner, M.I.,(ed.), Foundations of cognitive science (2001)
36. Song, Q., Kasabov, N.: NFI: a neuro-fuzzy inference method for transductive reasoning. IEEE Transactions on Fuzzy Systems **13:6**, 799–808 (2005)
37. Takagi, T., Sugano, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics **15:1**, 116–132 (1985)
38. Vitousek, P.: Global environmental change: An introduction. Review of Ecology and Systematics **23**, 1–14 (1992)
39. Watts, M.: A decade of Kasabov's evolving connectionist systems: a review. IEEE Transactions on Systems, Man, and Cybernetics, Part C **39:3**, 253–269 (2008)
40. Welling, M.: The Kalman Filter. Tech. rep., California Institute of Technology (2001)
41. Widiputra, H.: Integrated Multi-Model Framework for Adaptive Multiple Time-Series Analysis and Modelling. Ph.D. thesis, Knowledge Engineering and Discovery Research Institute, Auckland University of Technology (2011)
42. Widiputra, H., Pears, R., Kasabov, N.: Dynamic Interaction Networks versus Local Trend Models for Multiple Time Series Prediction. Cybernetics and Systems **42:2**, 100–123 (2011)
43. Widiputra, H., Pears, R., Kasabov, N.: Multiple time-series prediction through multiple time-series relationships profiling and clustered recurring trends. In: Proceedings of the pacific asia cconference on knowledge discovery and data mining 2011, PAKDD 11 (2011)
44. Widiputra, H., Pears, R., Serguieva, A., Kasabov, N.: Dynamic interaction networks in modelling and predicting the behaviour of multiple interactive stock markets. Intelligent Systems in Accounting, Finance and Management **16**, 189–205 (2009)
45. Widiputra, H., Pears, R.,Kasabov, N.: Personalised modelling for multiple time-series data prediction: a preliminary investigation in asia pacific stock market indexes movement. In: Proceedings of the 15th International conference on advances in neuro-information processing, 2008, ICONIP '08, 1237–1244 (2008)
46. Widrow, B., Lehr, M.: Adaptive neural networks and their applications. International Journal of Intelligent Systems **8**, 453–507 (1993)
47. Wu, D., Bennett, K., Cristianini, N., Shawe-Taylor, J.: Large margin trees for induction and transduction. In: Proceedings of the sixteenth international conference on machine learning, ICML 99 (1999)
48. Wu, D., Bennett, K.P., Cristianini, N., Shawe-taylor, J., Holloway, R.: Functional multilayer perceptron: a non-linear tool for functional data analysis. In: Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999, (1999)
49. Zadeh, L.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man and Cybernetics Part C **3:1**, 28–44 (1973)
50. Zhou, Z., Jiang, Y.: Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble. IEEE Transactions on Information Technology in Biomedicine **7**, 37–42 (2003)