A MultiPopulation Particle Swarm Optimization-based Time Series Predictive Technique

Cry Kuranga[a1*], Tendai S. Muwani[b2], Njodzi Ranganai[c2]

[a*]kurangacry@gmail.com, [b]tendaimuwani@gmail.com, [c]ranganai1981@gmail.com

[1]Department of Computer Science, University of Pretoria, Lynnwood Road, Hillcrest, Pretoria 0002, South Africa (+27652427298)

[2]Department of Computer Science and Information Systems, Manicaland State University of Applied Sciences, Stair Guthrie Rd., Fernhill, P. Bag 7001, Mutare, Zimbabwe

Abstract

A time series may exhibit concept drifting data patterns due to different underlying data generating processes. However, presenting more than one pattern to a predictor may reduce prediction precision. Several machine learning predictive techniques use a fixed observation window that is heuristically selected. The selected window may have concept drifting data patterns. This work proposes a multi-population particle swarm optimization-based nonlinear time series predictive technique. The proposed technique decomposes a predictive task into three sub-tasks: observation window optimization, predictive model induction task, and forecasting horizon prediction. Each sub-task is optimized by a particle swarm optimization sub-swarm in which the sub-swarms are executed in parallel. The proposed technique was experimentally evaluated on fifteen Electric load time series using root mean square error, mean absolute percentage error, and computation time as performance measures. The results obtained show that the proposed technique was effective to induce a forecasting model of improved predictive and computation performance to outperform the benchmark techniques on all datasets. Also, the proposed algorithm was competitive with state-of-the-art techniques. The future direction of this work will consider an empirical analysis of the search and solution spaces of the proposed technique and perform a fitness landscape analysis.

Keywords: multi-population, adaptive window, particle swarm optimization, nonlinear autoregressive model, prediction, nonstationary time series

Introduction

Time series are commonly used to infer and analyze the change of a phenomenon over time (Lütkepohl, 2005). As such, the order of observations in a time series is significant (Box et al., 2016). Thus, it is hypothesized that successive observations in the series are dependent, in which the dependence is based on the position of the observation, i.e., observations at position, t are more similar to those at position, $t-1$ (Madsen, 2007). However, time series are usually stochastic in nature whereby the past partially determines the future. Therefore, it is almost impossible to have error-free forecasts (Ciaburro et al., 2021).

The technique of using prior time steps to predict the next time step is called a sliding window. A sliding observation window provides the basis to convert a time series into a supervised machine learning problem (Qiao et al., 2018). There is a growing use of machine learning algorithms combined with sliding window techniques in forecasting time series (Makridakis et al., 2018a).

Time series usually consists of data patterns that are generated by generating processes that change over time (Kang et al., 2020). As such, a time series may exhibit concept drifting data patterns. Usually, supervised machine learning uses a fixed observation window that is heuristically selected since there is no consensus on the selection of an optimal window. The selected window may have concept drifting patterns (Gomes et al., 2019). An optimal sliding observation window saves computing power and improves the predictions, especially if the predictive models are generated periodically (Surakhi et al., 2021). However, presenting more than one pattern to a predictor may reduce prediction precision.

One of the challenging issues in machine learning is dealing with data whose data-generating processes change over time. To learn from such data, strategies for at least three tasks are required: sensing when change occurs; determining which instances to keep and which to discard (or, more broadly, maintaining adequate statistics); and adjusting the existing model(s) when substantial change is detected. A machine learning hybrid technique, dynamic particle swarm optimization-based nonlinear autoregressive time series prediction technique (NARX-QPSO) was proposed by Kuranga et al., (2022). The NARX-QPSO is based on a fixed observation window and it dynamically adapts the time series predictive model if a concept drifting data pattern is detected, thereby, enhancing the performance of the induced model. However, NARX-QPSO was incapable to capture inherent nonlinearity in the observation window.

A predictive task for nonstationary data can be considered a multi-dimensional problem that consists of a predictive model induction, an optimal observation window (adaptive window), and forecasting horizon prediction. This work proposes a multi-population particle swarm optimization-based nonlinear time series predictive technique. The proposed technique decomposes a predictive task into three sub-tasks: observation window optimization, predictive model induction task, and forecasting horizon prediction. Each sub-task is optimized by a particle swarm optimization sub-swarm in which the sub-swarms are executed in parallel. The proposed technique dynamically adapts the predictive model if a concept drifting data pattern is detected in the sliding observation window. The main contributions of this work are to:

a) Design a multi-population particle swarm optimization algorithm in hybrid with a nonlinear autoregressive model for time series predictive tasks,
b) Design an *adaptive* sliding observation window and a forecasting horizon prediction, and;
c) Perform a comparative study of the proposed technique to the baseline techniques and state-of-the-art techniques.

The work is organized as follows: Section 2 provides a background, and Section 3 discusses the multi-population particle swarm optimization predictive technique. Section 4 discusses experimental evaluations, and Section 5 presents the results and discussion. Section 6 concludes this work and provides future directions.

2. Background

A time series is an observation made over time in a specific order (Arunkumar et al., 2021). Time series data is characterized by high volumes and dimensionality and need to be updated regularly. Time series modeling is the process of gathering and analyzing historical data to fit a framework that explains the internal structure of a time series. The model that is created is then used to better predict the values of the series (Fentis et al., 2019). Therefore, any time series prediction model should be able to manage nonlinearity, voluminous data, fluctuating patterns, and consider the movement of time (Ma et al., 2020).

Time series forecasting models are classified into three types: statistical, machine learning, and hybrids (Nti et al., 2020). Statistical models are used when there is a lack of insight into the data generation process and the factors that influence it (Fentis et al., 2019). A statistical model is made up of measured variables (X_k) and a target variable (Y_k), which is based on the assumption that there is a relation between X_k and Y_k. Thus, the statistical model attempts to find a mathematical way to describe a relationship between X_k and Y_k and is typically distinguished by a plethora of computational possibilities that imply a high level of complexity, particularly for data patterns that depict nonlinearity (Fentis et al., 2019).

An autoregressive model (AR) and an autoregressive integrated moving average (ARIMA) are among the most used statistical models (Fentis et al., 2019). The nonlinear autoregressive model with exogenous inputs (NARX) is a variant of AR tailored for nonlinear time series predictive tasks. The NARX model links the latest value of output to both previous values of the same output as well as current and previous values of extrinsically influencing inputs (Fentis et al., 2019). Despite being one of the most computational complex statistical models and the least durable, the NARX model usually provides accurate predictions (Munawar et al., 2007).

The use of machine learning in time series modeling is gaining traction (Hannan et al., 2021; Struye et al., 2020). Machine learning techniques that use artificial intelligence algorithms to model time series data typically outperform statistical models, despite several drawbacks such as overfitting and being stuck in local minima (Preece et al., 2018; Gunning et al., 2019; Cuaresma et al., 2004; Kandananond, 2012). The following artificial intelligence algorithms are commonly used in time series modeling: particle swarm optimization (PSO), genetic algorithms, support vector regression (SVR), and artificial neural networks (Makridakis et al., 2018b).

Usually, the effectiveness and accuracy of the predictive model are improved by combining several techniques. The outstanding performance of hybrid methods indicates that there are several possible ways to develop forecasting models (Smyl, 2020; Montero-Manso et al., 2020). However, hybrid models are typically computationally intensive (Ristanoski et al., 2013).

A sliding window can be described as a transient approximation over temporal data, and it captures the most recent time points (Iqbal et al., 2019; Osman et al., 2021). Most sliding window-based algorithms are based on the fixed-size window, which deletes the earliest known batch of data when a new batch of data arrives (Baek et al., 2021). An example of a sliding window technique is illustrated in Figure 1. The numbers, $1 - 10$, represent the data points under observation, the window size is 8 units, and the window slides by 1 unit. The units can be hours, days, months, or even years.

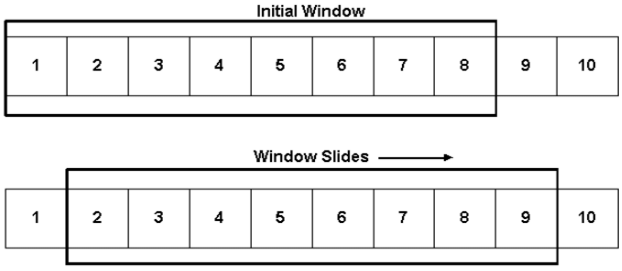


Figure 1. Sliding Window Technique

Prior information is required to determine the window size. A much wider window may include unnecessary patterns which tend to reduce the effectiveness of the predictive model. Conversely, a narrow window may result in concept drift which also affects predictive accuracy (Li et al., 2017). In a dynamic sliding window, when the concept drift is detected, the window dynamically resizes (Li et al., 2017). Thus, the dynamic window tends to expand if the data generating process is fixed and contracts if the data generating process fluctuates (Maurya et al., 2015).

2.1 Related Work

A dynamic particle swarm optimization-based nonlinear autoregressive time series prediction technique (NARX_QPSO) hybridizes a quantum-inspired PSO algorithm with a NARX model (Kuranga et al., 2022). In NARX_QPSO, a NARX model determines the model's initial parameters, and quantum-inspired PSO (QPSO) is tasked to optimize both the parameters and the model's structure to produce a structurally optimal nonlinear predictive model (Blackwell et al., 2002). Also, QPSO is used to detect the presence of concept drifting patterns in the time series. A significant deterioration in the performance of the prevailing predictive model suggests a change in the underlying data generating process that triggers a predictive model adaptation technique. The predictive model is adapted using the current observation window. A NARX_QPSO model is based on a fixed sliding window. NARX_QPSO was experimentally evaluated on time series datasets using root mean square error (RMSE) and mean absolute percentage error (MAPE). The NARX_QPSO outperformed all baselines techniques and several state-of-the-art techniques, however, it was outperformed by the empirical mode decomposition-based ensemble technique. The reduced performance of NARX_QPSO to empirical mode decomposition-based ensemble technique may suggest the presence of concept drifting in the observation window that reduced the prediction accuracy of the induced model.

An adaptive sliding window-based predictive model technique for resource utilization at data centers was proposed by Iqbal et al., 2020. The proposed model used the historical data and the selected sliding windows to induce a predictive model. The proposed technique was experimentally evaluated on three public data center datasets: Alibaba, Bitbrains, and Google data using RMSE and mean absolute error (MAE). Several baseline techniques which include SVR and K-nearest neighbors were used to benchmark the proposed technique. The proposed adaptive sliding window-based predictive model technique outperformed all baseline techniques in terms of predictive accuracy. The results also indicated that an optimal window size of 60 minutes was ideal to build a predictive model for resource utilization at data centers.

Also, an adaptive sliding window-based predictive technique was proposed by Alberg et al., 2020. The proposed technique combined three techniques: incremental learning, seasonal, and non-seasonal ARIMA models. The proposed model builds the predictive model regularly using the sliding window and consists of four components: a learning module – to compute the parameters for the ARIMA models using training data passed through the sliding window. The repository component stores the ARIMA model derived from the most recent test data, the predictive component evaluates each model in the repository, and the meta-learning component determines the size of the sliding window to pass on the next slide. A new predictor was induced whenever a concept drift was detected in the data. Four variants of the proposed models were experimentally evaluated using six electric load datasets (not publicly available). The adaptive sliding window technique outperformed the benchmark models and baselines techniques in terms of predictive accuracy.

This work proposes an adaptive sliding window-based predictive technique which is built on a machine learning hybrid technique, NARX_QPSO.

3. Multi-population Particle Swarm Optimization Predictive Technique

A multi-population quantum-inspired particle swarm optimization predictive technique (multiPQPSO) is proposed that considers a predictive task as a multi-dimensional problem: sliding

observation window optimization, predictive model induction, and prediction of the forecasting horizon, in which each dimension is tackled independently.

The proposed multiPQPSO uses a QPSO swarm that is divided into three sub-swarms. Each sub-swarm is tasked to optimize one dimension of the problem. As such, the optimization process is executed in parallel. In the first dimension, sliding observation window optimization, the sub-swarm consists of sliding windows of varying sizes. The objective is to find the optimal sliding window (window with the least irrelevant data points). The second dimension adopts the NARX_QPSO model induction technique (Kuranga et al., 2022). In NARX_QPSO, the NARX estimates model parameters, and then the QPSO determines the optimal model structure and model parameters. The third dimension, forecasting horizon prediction, uses the induced predictive model from the second dimension to predict the forecasting horizon - the number of data points that yields the best predictive model accuracy. As in the first dimension, the sub-swarm consists of forecasting horizons of varying sizes. The objective is to find the optimal forecasting horizon that yields the best forecasting accuracy.

The sliding observation window consists of training and testing partitions. The initial size of observation windows in the first dimension and forecasting horizons in the third dimension are randomly generated within the lower and upper bounds of the training partition of the sliding observation window. Figure 2 illustrates two windows, w_1 and w_2, that are randomly generated within the lower and upper bounds of the training partition of the sliding window. The final value of each window or horizon is the last value of the training partition, i.e., the range for $w_1: 8-12$ and $w_2: 4-12$. Therefore, a sub-swarm consists of different-sized sliding windows or forecasting horizons.

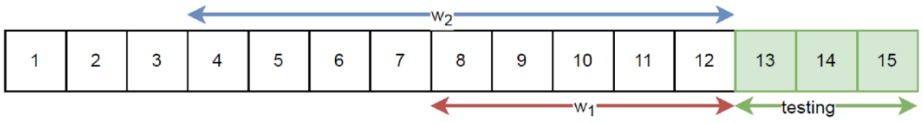


Figure 2 Sliding window

A NARX model can be expressed as:

$$\hat{y}_i = \sum_{\tau=0,\sum_{d=1}^{D}\delta_d=\tau}^{p,q} \left(\alpha_{(\delta_{\alpha_1},\ldots,\delta_{\alpha_D})} \prod_{k=1}^{D} y_{i,k}^{\delta_{\alpha_k}}, \beta_{(\delta_{\beta_1},\ldots,\delta_{\beta_D})} \prod_{l=1}^{D} x_{i,l}^{\delta_{\beta_k}} \right) \quad (1)$$

where $\{p, q\}$ are the maximum polynomial orders for y and x respectively, D is dimensionality, $\{\alpha_{(\delta_{\alpha_1},\ldots,\delta_{\alpha_D})}, \beta_{(\delta_{\beta_1},\ldots,\delta_{\beta_D})}\}$ are a real-valued coefficients, and $\{\delta_\alpha, \delta_\beta\}$ are the natural-valued orders. The coefficients are computed using the least-squares approximation technique.

The second-dimension, predictive model induction, sub-swarm consists of particles in the form of Eq. (1) which comprises of unique term-coefficient mapping from Eq. (2):

$$\theta = \left\{ \{(t_0 \to \alpha_0), \ldots, (t_{n_k} \to \alpha_{n_k})\}, \{(t_0 \to \beta_0), \ldots, (t_{n_l} \to \beta_{n_l})\} \right\} \quad (2)$$

where n is the number of the terms, and the coefficients: $\{\alpha_\vartheta, \vartheta \epsilon \{0, \ldots, n_k\}; \beta_\vartheta, \vartheta \epsilon \{0, \ldots, n_l\}\}$. If input variable(s) is m and the natural-valued order, φ_k, τ_l, variable-order mappings (T) is defined as:

$$T_{\varepsilon,\sigma} = \left\{ \left\{ (y_{\varepsilon,1} \to \delta_{\alpha_\varepsilon,1}), \ldots, (y_{\varepsilon,n_y} \to \delta_{\alpha_\varepsilon,n_y}) \right\}, \{(x_{\sigma,1} \to \delta_{\beta_\sigma,1}), \ldots, (x_{\sigma,n_x} \to \delta_{\beta_\sigma,n_x})\} \right\} \quad (3)$$

where n is the number of variables $\{y_{\varepsilon,k}, k\epsilon\{0,\dots,n_y\}, \{x_{\sigma,l}, l\epsilon\{0,\dots,n_x\}\}$ are input variable-integer representations. If a concept drifting data pattern is detected, the term coefficient mappings of each QPSO particle in the second dimension are updated. Algorithm 1 summarizes the multiPQPSO.

Algorithm 1 Multi-population Quantum-inspired Particle Swarm Optimization

```
Initialize a sliding window (s_window)
Set subPop_window, subPop_model, and subPop_horizon to be initial sub_swarms
DO
    Slide the s_window
    IF a concept drifting data pattern is detected
      While i < max_iteration:
      #Cycle through particles in each sub_swarm and evaluate fitness
         For j in range (num_particles): #if sub_swarms are of same size
            subPop_window[j].evaluate
            subPop_model[j].evaluate
            subPop_horizon[j].evaluate

            # Update local best solution(pBest)
             If subPop_window[j].err < w_err_pBest:  # Window_dimension
                w_pBest = subPop_window[j].position
                w_err_pBest = subPop_window[j].err
             If subPop_model[j].err < m_err_pBest:   # Model_dimension
                m_pBest = subPop_model[j].position
                m_err_pBest = subPop_model[j].err
             If subPop_horizon[j].err < h_err_pBest: # Horizon_dimension
                h_pBest = subPop_horizon[j].position
                h_err_pBest = subPop_horizon[j].err

         # Update global best solution(gSolution)
         If gSolution.err[0] > w_err_pBest:
            gSolution.position[0] = w_pBest
            gSolution.err[0] = w_err_pBest
         If gSolution.err[1] > m_err_pBest:
            gSolution.position[1] = m_pBest
            gSolution.err[1] = m_err_pBest
         If gSolution.err[2] > h_err_pBest:
            gSolution.position[2] = h_pBest
            gSolution.err[2] > h_err_pBest
         gSolution_err = gSolution.evaluate

         # Update velocities and positions using gSolution
         For particles in sub_swarms:
            Update velocity using gSolution
            Update position using gSolution
         i+=1
   Else
     gSolution_err = gSolution.evaluate # Evaluate current predictive model
Until no further data to analyze
```

The predictive solution (gSolution) consists of the global best of each sub-swarm: window size (*w_pBest*), predictive model (*m_pBest*), and forecasting horizon (*h_pBest*). Therefore, a gSolution is evaluated to get the predictive performance of the multiPQPSO. Each particle in the sub-swarm is evaluated by substituting its dimension in the gSolution, i.e., a particle in the first dimension replaces the first dimension of the gSolution. Therefore, the adapted gSolution is evaluated to obtain the fitness of that particle (Georgieva et al., 2014). If the sub-swarm local best fitness is

better than the gSolution fitness, then the sub-swarm local best becomes the global best for that dimension. A sliding window can grow, if there is an improvement in accuracy, beyond the initial size of the sliding window up to the *max_window* size. The *max_window* is a parameter.

The multiPQPSO detects a concept drifting data pattern in the sliding observation window if there is a significant fitness deterioration, *fit_det*, of gSolution (Kuranga, et al. 2022). The *fit_det* is a parameter. As illustrated in Figure 2, the sliding window consists of two parts: training $(1-12)$ and testing $(13-15)$. The window slides by the *n* data points (the size of the test partition), i.e., $n=3$. Thus, the testing partition is considered as new incoming data. The training partition is only used if a concept drifting pattern is detected to update the gSolution (update using the recent historical data).

The fitness of each particle is calculated using the adapted adjusted coefficient of determination:

$$R_a^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \times \frac{m_i}{m_D} \times \frac{w_i}{w_D}, \quad (4)$$

where $\bar{y}$ is the average value, $\hat{y}_i$ is the predicted value, y_i is the actual value, m_D is the number of coefficients in the initial model and m_i is the number of coefficients in the current model, w_D is the initial window size and w_i is the current window size. Thus, the fitness function tends to penalize a complex model, a model induced using a wider window and a larger number of coefficients.

4. Experimental Evaluation

4.1. Dataset

The performance of the proposed technique is evaluated using fifteen Electric load time series datasets from AEMO (AEMO, 2022). Two CPU resource utilization characteristics datasets for Bitbrains (BitB, 2022) and Alibaba (Alib, 2022) are used to benchmark the proposed algorithm to state-of-the-art techniques. The discussions on the datasets are provided in the literature (Kuranga et al., 2022; Iqbal et al., 2020).

The AEMO Electric load datasets are from New South Wales, Tasmania, Queensland, South Australia, and Victoria. The data were recorded daily at a 30-minute interval for the period 2013-2015. As such, each year had 17, 520 data points. In this work, each dataset is linearly scaled to [0, 1]. Daily minimum and maximum temperatures for the following places are considered as exogenous variables: New South Wales (NSW) - Canberra; Tasmania (TAS) - Launceston airport and Hobart; Queensland (QLD) - Brisbane; South Australia (SA) - Adelaide; and Victoria (VIC) - Melbourne (AusMet, 2022).

Alibaba dataset is extracted from cluster workloads that serve both batch and interactive processing. The data consists of cluster logs for 12 hours of 1,313 machines. The following metrics were measured for all the machines sampled at the 5-minute interval: disk, memory, and CPU.

The Bitbrains datasets are extracted from the cloud services data center. The data consists of 1,750 virtual machines (VMs) performance traces for 30 days. The following metrics were measured for all the VMs sampled at the 5-minute interval: disk, memory, network, and CPU. The experiments are conducted using a CPU workload of 20 randomly selected VMs with average utilization of greater than 30%.

4.2 Performance Measure

In each experiment, the following metrics are measured: computation, and predictive accuracy (RMSE and MAPE). The computation time is measured as the time taken between the start and end of the predictive model induction. The RMSE and MAPE are computed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i' - y_i)^2}{n}} \tag{5}$$

$$MAPE = \sum_{i=1}^{n}\left|\frac{y_i' - y_i)}{y_i}\right| \times 100, \tag{6}$$

where n is the testing data patterns, y_i' is the predicted value, and y_i is the actual value.

4.3 Experimental Details

The AEMO datasets are used to perform a comprehensive evaluation of the multiPQPSO technique. A comparative study of the multiPQPSO to the baseline and adaptive window-based predictive state-of-the-art techniques is performed using Alibaba and Bitbrains datasets.

The following techniques are used to benchmark the performance of the proposed technique: NARX_SVR, and NARX_QPSO (Kuranga, et al., 2022). The multiPQPSO, NARX_SVR, and NARX_QPSO induce a nonlinear predictive model to predict x_t for a two-day horizon (96 steps) or one-week horizon (240 steps) using input features: month, day, time, load value, and min and max temperatures. Each dataset is split: training (75%) and testing (25%), and the order of the data points is maintained since *t* had an inherent meaning to the data pattern. A training set is used to induce a predictive model which is evaluated using an out-of-sample testing dataset.

Two experiments are conducted to evaluate the proposed technique. In Exp 1, the initial sliding window is set to 384 data points (week) and the testing partition is set to 96 data points, the size of the forecasting horizon. The benchmark techniques, NARX_SVR and NARX_QPSO use a fixed sliding window of 384 data points (training partition) to update the predictive model if a concept drift is detected in the data while multiPQPSO is expected to adapt the sliding window.

In Exp 2, the initial sliding window is set to 1440 data points (one month) and the testing partition is set to 240 data points, the size of the forecasting horizon. As in Exp 1, the benchmark techniques, NARX_SVR and NARX_QPSO use a fixed sliding window of size 1440 data points (training partition) to update the predictive model if a concept drift is detected in the data.

The experiments are conducted in a Python environment on an Intel Core *i*7 processor (3.1 GHz) and 16 GB RAM desktop running on Windows 10. Hyper-parameter tuning is performed using GridSearchCV for the parameter ranges from the literature (Pedregosa et al., 2011). Table 1 presents the parameters that are tuned and the range of values used. The SVR parameters: *c*, *epsilon*, and *gamma* are optimized by QPSO during each iteration, and an "rbf" kernel is used.

The statistical analysis is performed using the non-parametric Friedman test at $\alpha = 0.05$ significance level. The null hypothesis, H_0 states that there exists no statistically significant difference between the central tendency of the populations. If there exists a statistically significant difference then a post-hoc Nemenyi test is performed (Herbold et al., 2020).

Table 1: Hyper-Parameter Tuning

Parameter	Range	Optimal Value
c_1	[1, 2]	1.4961
c_2	[1, 2]	1.4961
ω	[0, 1]	0.7298
r_Q	[1, 5]	2
max_window	[384, 1440]	1024
subPop_window	[5, 30]	10
subPop_horizon	[5, 30]	10
subPop_model	[5, 30]	25
iterations	[50, 200]	100
fit_det	[5, 25]	10%

5. Results and Discussion

Table 2 presents the results obtained from performing Exp 1 using the multiPQPSO and the benchmark techniques: NARX_SVR and NARX_QPSO on AEMO datasets. The performance of NARX_QPSO and multiPQPSO in terms of RMSE and MAPE were in the same error range exhibiting outstanding performance for all 15 datasets. However, NARX_SVR exhibited the worst performance on all the datasets.

Figure 3 illustrates the performance profiles: RMSE and MAPE for the benchmark techniques and the proposed algorithm, multiPQPSO, for Exp 1. As shown in Figure 3, the predictive performance of NARX_SVR was the worst for NSW and TAS datasets, however, there was a significant predictive performance gain on QLD and VIC datasets. The outstanding predictive performance exhibited by NARX_QPSO shows that the sliding observation window of 384 data points was optimal to induce a model to forecast 96 data points.

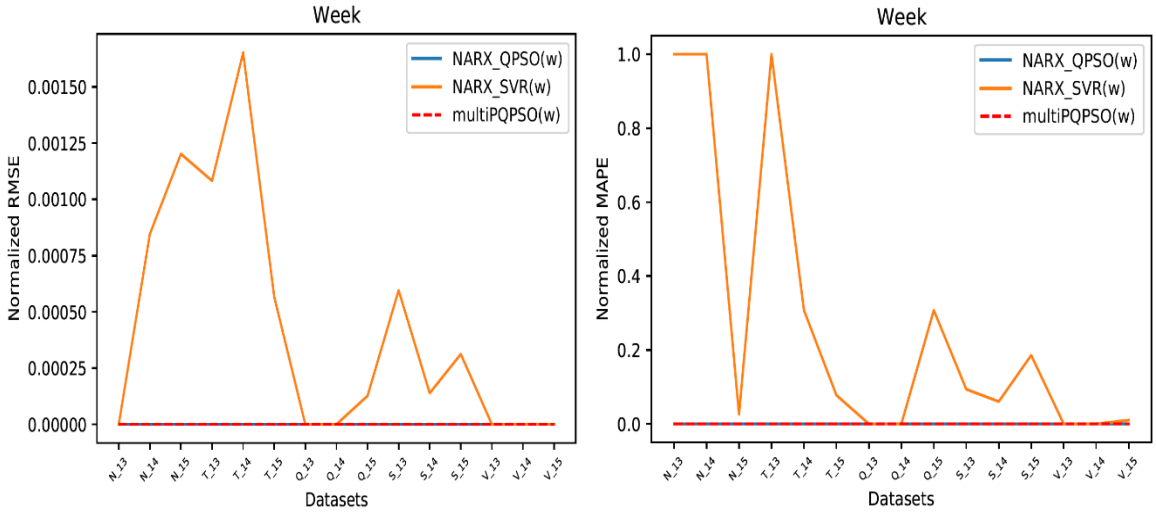


Figure 3 Performance profiles for Exp 1

Table 3 presents the results obtained for Exp 2. The predictive performance of NARX_QPSO and NARX_SVR deteriorated in Exp 2. However, NARX_SVR significantly outperformed NARX_QPSO. Two possibilities may result in performance deterioration for NARX_QPSO and NARX_SVR; either the observation window had concept drifting patterns or the user-defined forecasting horizon was not optimal for the induced predictive model.

Table 2: Results for Exp 1

Dataset	Measure	NARX_SVR	NARX_QPSO	EMD_NARX_QPSO	multiPQPSO
NSW_2013	RMSE	7.19e-08	6.39e-23	5.70e-12	**3.64e-23**
	MAPE	2.83e-02	7.42e-16	**5.47e-16**	6.24e-16
	Runtime	194.24	62.71	238.29	**55.10**
NSW_2014	RMSE	2.50e-01	**2.29e-23**	2.38e-23	5.12e-23
	MAPE	8.31e-03	**7.34e-23**	8.28e-23	5.35e-16
	Runtime	179.66	66.52	221.87	**15.21**
NSW_2015	RMSE	6.24e-01	4.81e-23	**3.53e-23**	1.28e-22
	MAPE	3.93e-05	5.78e-16	**5.33e-16**	8.13e-16
	Runtime	178.05	48.02	258.31	**43.35**
TAS_2013	RMSE	4.10e-01	2.37e-23	8.80e-25	**5.66e-25**
	MAPE	1.45e-02	2.08e-15	2.70e-13	**3.32e-16**
	Runtime	184.91	47.81	302.59	**32.46**
TAS_2014	RMSE	3.36e-01	6.45e-25	**6.13e-25**	8.17e-25
	MAPE	6.18e-04	5.32e-16	**4.27e-16**	4.28e-16
	Runtime	170.92	47.52	248.94	**25.99**
TAS_2015	RMSE	8.81e-02	7.06e-25	6.47e-25	**2.04e-26**
	MAPE	3.42e-04	4.12e-16	4.47e-16	**6.09e-17**
	Runtime	131.37	49.31	198.62	**33.97**
QLD_2013	RMSE	7.07e-08	3.64e-24	**2.42e-24**	6.43e-23
	MAPE	3.78e-08	1.92e-16	**1.72e-16**	6.47e-16
	Runtime	172.38	**32.27**	264.27	36.19
QLD_2014	RMSE	7.69e-08	1.60e-24	**1.19e-24**	6.93e-24
	MAPE	3.76e-08	1.26e-16	**4.34e-17**	2.96e-16
	Runtime	213.41	**33.09**	402.76	33.88
QLD_2015	RMSE	2.68e-01	2.92e-23	**4.42e-24**	3.07e-23
	MAPE	1.63e-03	6.10e-16	**6.04e-16**	6.14e-16
	Runtime	168.61	**28.26**	290.47	64.30
SA_2013	RMSE	8.47e-01	3.33e-24	**2.35e-24**	3.53e-24
	MAPE	1.93e-03	9.08e-16	**1.23e-16**	9.21e-16
	Runtime	139.48	**24.70**	215.60	22.99
SA_2014	RMSE	2.03e-08	4.96e-24	2.59e-12	**3.79e-24**
	MAPE	1.10e-08	1.16e-15	1.48e-15	**9.69e-16**
	Runtime	143.52	31.01	216.23	**29.30**
SA_2015	RMSE	5.49e-01	2.72e-24	2.16e-24	**1.76e-24**
	MAPE	3.51e-03	7.80e-16	6.89e-16	**5.99e-16**
	Runtime	165.43	**22.83**	231.54	23.35
VIC_2013	RMSE	6.68e-08	4.38e-23	3.77e-23	**1.34e-23**
	MAPE	3.95e-08	7.68e-16	7.13e-16	**4.13e-16**
	Runtime	168.93	32.51	326.41	**22.52**
VIC_2014	RMSE	6.82e-08	2.41e-23	**2.27e-23**	2.92e-23
	MAPE	4.34e-08	6.37e-16	**5.78e-16**	7.54e-16
	Runtime	177.48	**23.85**	353.68	56.58
VIC_2015	RMSE	4.00e-04	4.32e-23	3.36e-12	**8.54e-24**
	MAPE	9.58e-05	1.08e-15	3.51e-15	**4.06e-16**
	Runtime	186.47	**21.07**	309.63	27.73

If there exist concept drifting patterns in the observation window, then the performance of the predictive model is expected to deteriorate since the model is built on the past historic pattern which may not be reflective of the current data-generating process. The performance gain by NARX_SVR shows that a wider sliding window was favorable for the SVR technique. The proposed model, multiPQPSO, exhibited an outstanding performance in Exp 2 to outperform NARX_QPSO and NARX_SVR on all datasets. The NARX_QPSO exhibited the worst predictive performance for both RMSE and MAPE.

Table 3: Results for Exp 2

Dataset	Measure	NARX_SVR	NARX_QPSO	multiPQPSO
NSW_2013	RMSE	2.54	253.12	**1.01e-18**
	MAPE	8.11e-06	0.0011	**3.58e-14**
	Runtime	1719.76	130.63	**51.79**
NSW_2014	RMSE	1.98	295.47	**3.46e-23**
	MAPE	2.84e-06	0.0013	**5.65e-16**
	Runtime	1647.13	129.53	**31.67**
NSW_2015	RMSE	2.97	519.13	**3.28e-23**
	MAPE	4.13e-06	0.0015	**5.38e-16**
	Runtime	1641.61	117.37	**31.67**
TAS_2013	RMSE	37.61	378.64	**2.88e-25**
	MAPE	2.34e-03	0.0126	**4.54e-16**
	Runtime	1455.89	31.49	**71.4**
TAS_2014	RMSE	20.33	16.28	**1.31e-24**
	MAPE	2.01e-03	0.0019	**6.87e-16**
	Runtime	1442.03	132.69	**23.89**
TAS_2015	RMSE	31.94	21.44	**5.53e-24**
	MAPE	4.41e-03	0.0020	**7.60e-16**
	Runtime	1150.10	102.96	**31.00**
QLD_2013	RMSE	253.64	1412.09	**1.66e-23**
	MAPE	1.52e-02	0.0049	**4.81e-16**
	Runtime	2196.11	72.83	**22.36**
QLD_2014	RMSE	128.58	1889.21	**2.28e-23**
	MAPE	1.54e-04	0.0046	**5.23e-16**
	Runtime	2045.94	71.89	**17.04**
QLD_2015	RMSE	108.06	2119.47	**4.78e-23**
	MAPE	1.36e-04	0.0053	**7.14e-16**
	Runtime	2373.08	74.53	**17.81**
SA_2013	RMSE	332.24	1421.42	**4.81e-23**
	MAPE	2.06e-02	0.0180	**5.84e-15**
	Runtime	1924.27	85.97	**126.54**
SA_2014	RMSE	108.06	1462.10	**2.88e-24**
	MAPE	1.17e-03	0.0182	**1.10e-15**
	Runtime	1163.83	80.25	**17.46**
SA_2015	RMSE	286.37	1758.99	**3.31e-24**
	MAPE	1.35e-3	0.0189	**8.19e-16**
	Runtime	1205.78	62.69	**128.23**
VIC_2013	RMSE	6.35e-08	5435.14	**7.94e-22**
	MAPE	3.93e-08	0.0104	**8.58e-15**
	Runtime	2907.15	70.75	**27.90**
VIC_2014	RMSE	6.03e-08	4815.00	**5.47e-23**
	MAPE	4.02e-08	0.0104	**8.48e-16**
	Runtime	2032.71	94.61	**35.48**
VIC_2015	RMSE	5.89e-08	4095.72	**3.37e-23**
	MAPE	3.93e-08	0.0094	**8.05e-16**
	Runtime	2585.21	80.78	**16.15**

Figure 4 illustrates the performance profiles: RMSE and MAPE for the benchmark techniques and the proposed algorithm for Exp 2. The NARX_SVR yielded an improved predictive performance, except for TAS and SA_2013 datasets. However, multiPQPSO exhibited an outstanding performance to outperform the other techniques on all datasets. The outstanding performance exhibited by multiPQPSO indicates that the predictive model was built either using data patterns that were reflective of the current data-generating process or the predicted forecasting horizon was optimal for the induced predictive model, or both cases were satisfied.

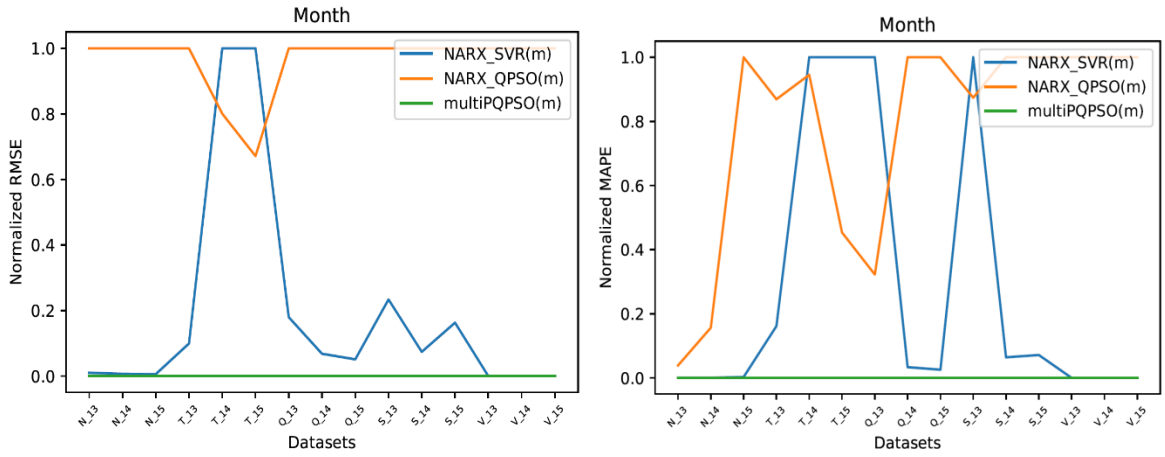


Figure 4 Performance profiles for Exp 2

Figure 5 is an illustration of computation time for Exp 1 and Exp 2. In Figure 5, the week represents Exp 1 and the month Exp 2. In Exp 1, the computation time for NARX_QPSO and multiPQPSO were in the same error range for all the datasets except for fewer cases. Conversely, NARX_SVR exhibited the highest computation time in Exp 1.

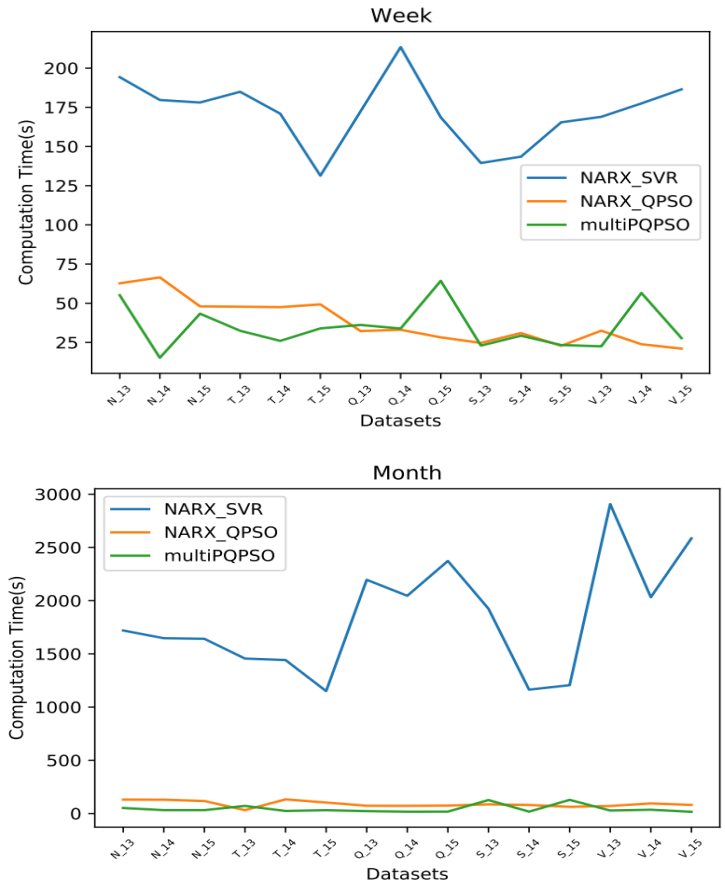
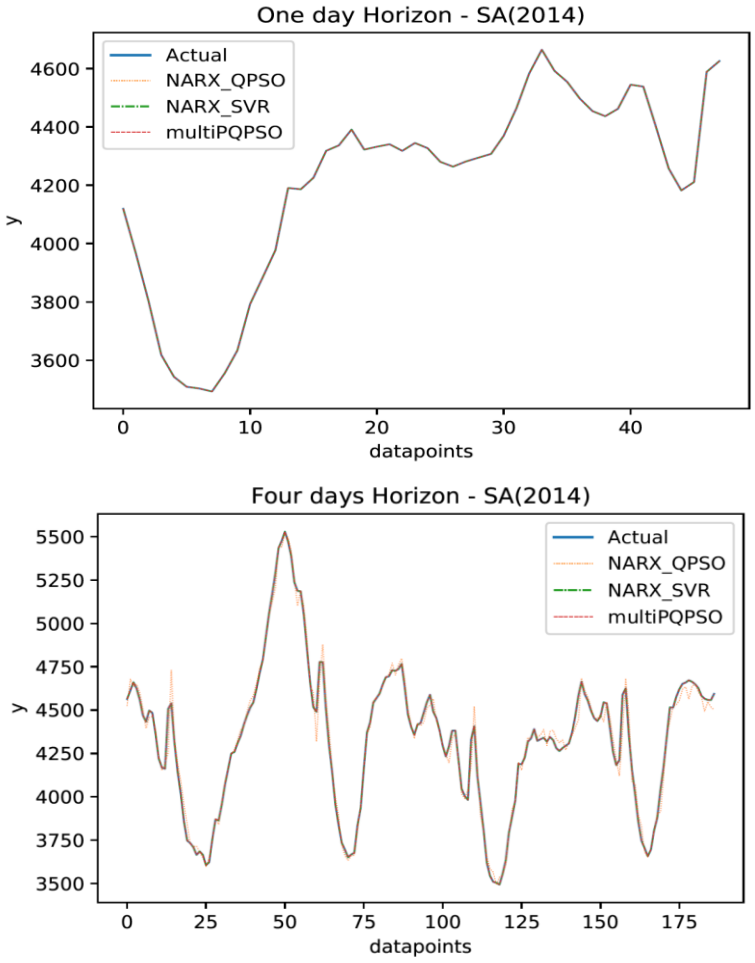


Figure 5 Computation time for Exp 1 and Exp 2

In Exp 2 (month), the computational performance of NARX_QPSO deteriorated slightly as the observation window and forecasting horizon were increased. The NARX_SVR exhibited the highest computation time on all the datasets. However, the predictive performance of the NARX_SVR greatly improved for both RMSE and MAPE compared to Exp 1 as indicated in Table 2. The performance exhibited by the NARX_SVR technique indicated that the SVR model was a computationally intensive technique.

The computational deterioration of multiPQPSO in Exp 2 for SA_2013 and SA_2015 suggests the presence of several concept drifting data patterns that resulted in several executions of the multiPQPSO to adapt the predictive model. However, the predictive performance of multiPQPSO did not deteriorate as indicated by RMSE and MAPE in Table 3. Also, the computation performance of multiPQPSO was outstanding which suggests the effectiveness of multiPQPSO parallel execution, the overall performance of the algorithm improved.

Figure 6 illustrates actual versus predicted data points for Exp 1 and Exp 2 for the SA_2014 dataset. As illustrated in Figure 6, all techniques induced predictive models that resemble the actual data in Exp 1. In Exp 2, there was a noticeable performance deterioration of NARX_QPSO.



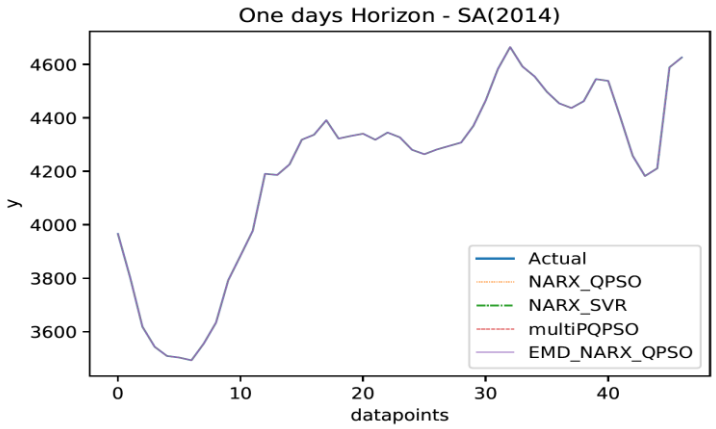
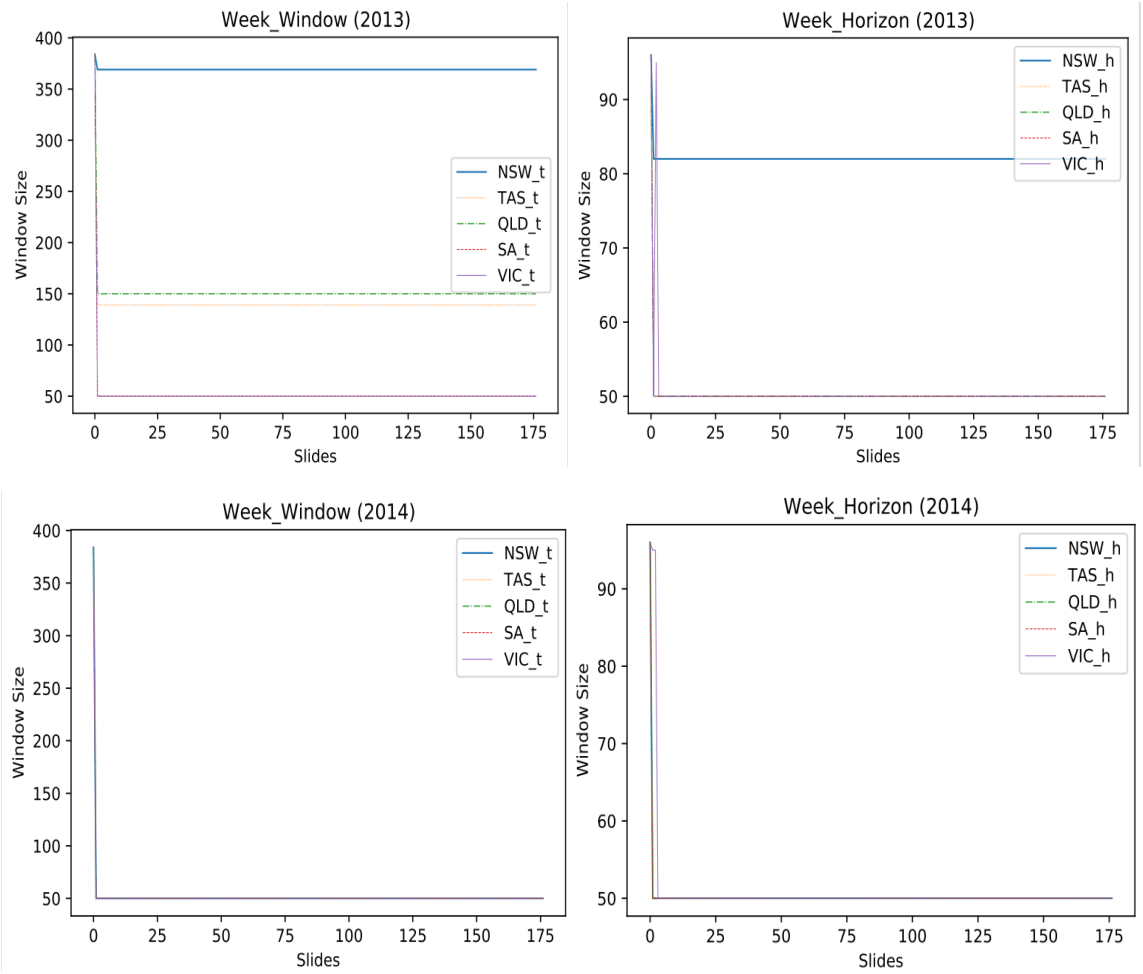


Figure 6 Actual vs. Predicted Data points for SA_2014 Dataset

Figure 7 illustrates the multiPQPSO's optimized observation window and the predicted forecasting horizon for Exp 1.



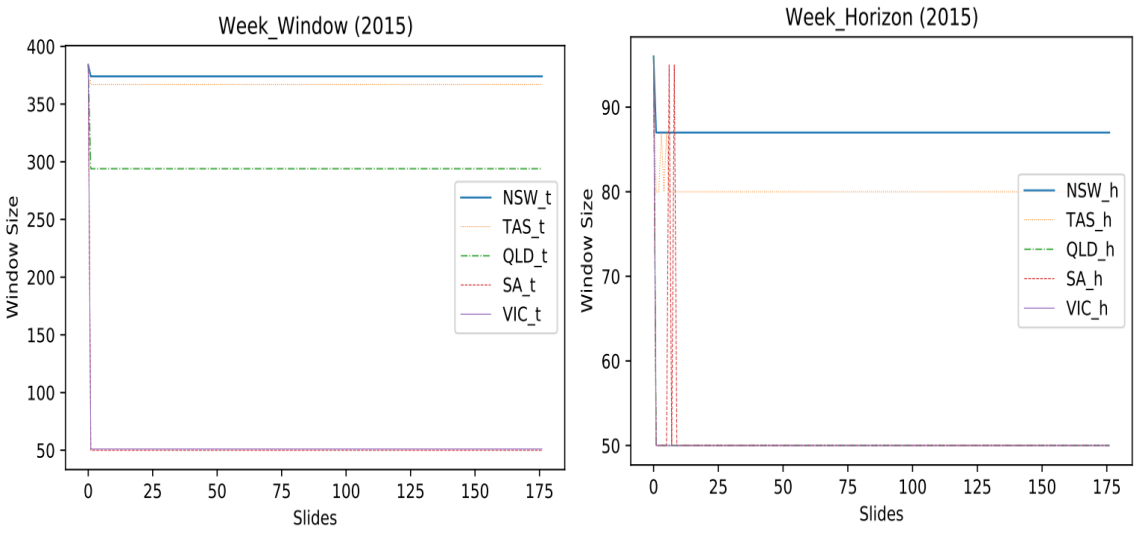


Figure 7 Analysis of Observation Window and Forecasting Horizon for Exp 1

In 2013, multiPQPSO dropped more than 40%, except for the NSW dataset, of the initial observation window, and predicted a forecasting horizon that was less than the user-defined. The dropping of data points in the sliding window suggests the presence of concept drifting data patterns in the observation window. The dropping of data points in both the observation window and forecasting horizon resulted in performance gain shown in Table 2. In 2014, multiPQPSO predicted an optimal forecasting horizon that was almost half of the user-defined forecasting horizon and dropped more than 80% of observation window data points in the first 10 slides to enumerate an optimal window of 50 data points that was maintained throughout the subsequent slides.

The multiPQPSO dropped a few data points for NSW_2015 and TAS_2015 and more than half of the observation window size for SA_2015 and VIC_2015. As observed in the year 2013, the same characteristics were observable in 2015 forecasting horizons in which an optimal horizon of 50 data points was enumerated in the first 10 slides.

Figure 8 is an illustration of the multiPQPSO's optimized observation window and the predicted forecasting horizon for Exp 2. As illustrated in Figure 8, for VIC_2013, multiPQPSO enumerated different observation windows for the first 10 slides and then picked an optimal observation window of 1000 data points thereafter.

Also, the predicted forecasting horizon for VIC_2013 was oscillating for the first 10 slides. A user-defined forecasting horizon of 240 data points and a wider sliding window led to great performance deterioration for NARX_QPSO. However, multiPQPSO optimized the observation windows and predicted optimal forecasting horizons which resulted in outstanding performance for all datasets.

The statistical test results, mean absolute deviation (MAD), mean rank (MR), and median (MD) for the Exp 1 and Exp 2 are presented in Table 4. The results obtained are statistically significant if MR > critical distance (CD) (Demšar, 2006).

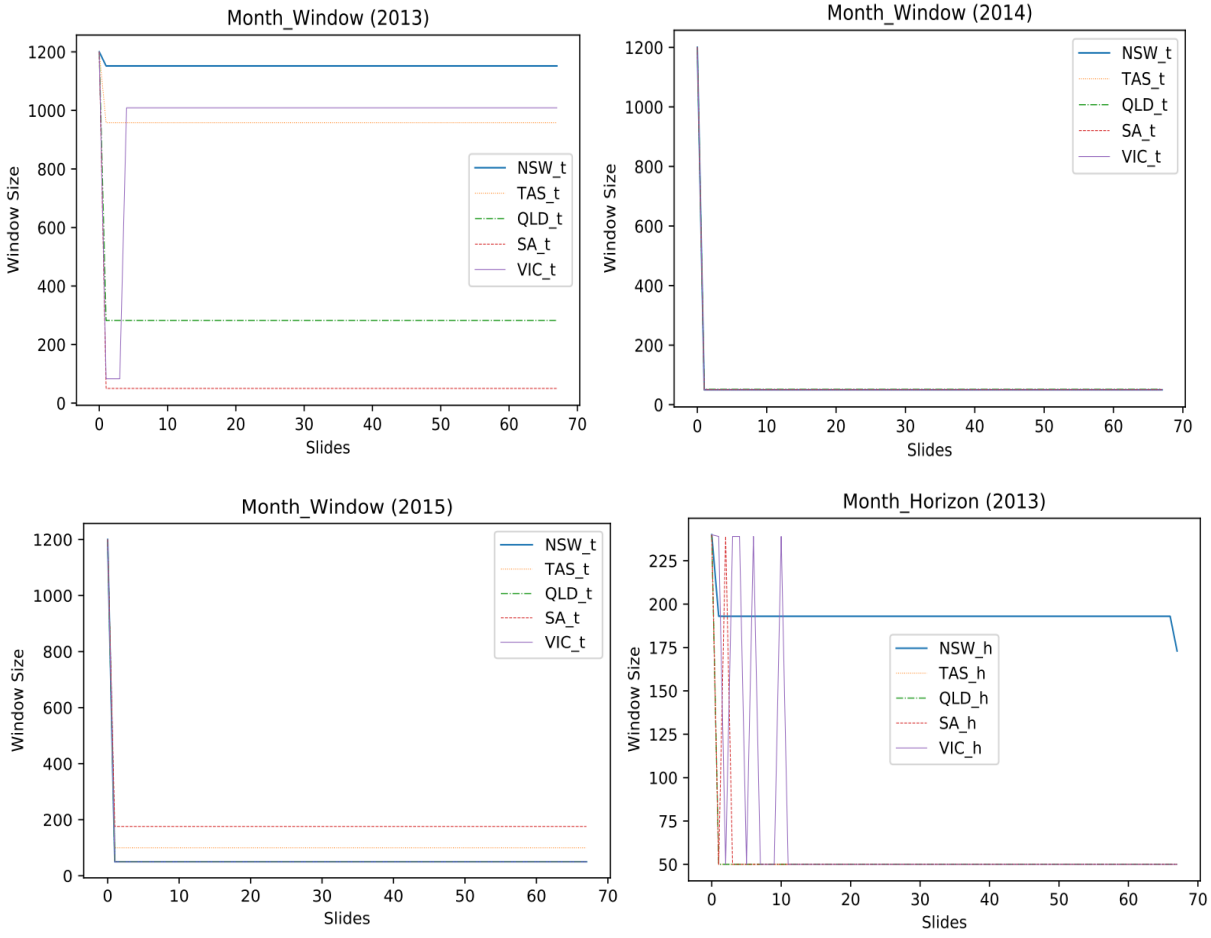
Table 4: Statistical Test Results for AEMO datasets

Technique	MD	MAD	MR
Exp 1_RMSE			

NARX_SVR	0.088±0.274	0.1310	1.000
NARX_QPSO	0.000±0.000	0.000	2.600
multiPQPSO	0.000±0.000	0.000	2.400
Exp 1_MAPE			
NARX_SVR	0.000±0.004	0.000	1.000
NARX_QPSO	0.000±0.000	0.000	2.600
multiPQPSO	0.000±0.000	0.000	2.400
Exp 2_RMSE			
NARX_SVR	31.940±125.830	47.354	1.867
NARX_QPSO	1421.420±1900.125	1546.026	1.133
multiPQPSO	31.940±125.830	47.354	1.867
Exp 2_MAPE			
NARX_SVR	0.000±0.002	0.000	1.733
NARX_QPSO	0.005±0.008	0.006	1.267
multiPQPSO	0.000±0.000	0.000	3.000

In Exp 1, $CD = 0.605$, and $p = 0.000$ for both RMSE and MAPE which suggests a statistically significant difference in the central tendency of the populations. However, the post-hoc Nemenyi test suggests that there exists no statistically significant difference in the set: {multiPQPSO, NARX_QPSO} for RMSE.

In Exp 2, $CD = 0.605$, and $p = 0.000$ for both RMSE and MAPE which suggests a statistically significant difference in the central tendency of the populations. However, the post-hoc Nemenyi test suggests that there exists no statistically significant difference in the set: {NARX_SVR, NARX_QPSO} for MAPE.



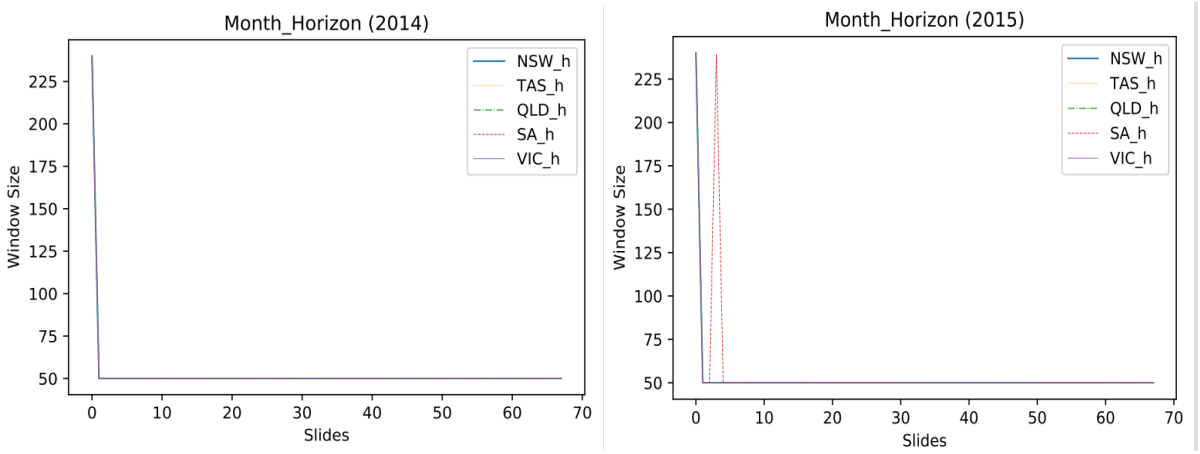


Figure 8 Analysis of Observation Window and Forecasting Horizon for Exp2

Figure 9 illustrates the post-hoc Nemenyi test for RMSE and MAPE for Exp 1 (w_rmse, w_mape) and Exp 2 (m_rmse, m_mape).

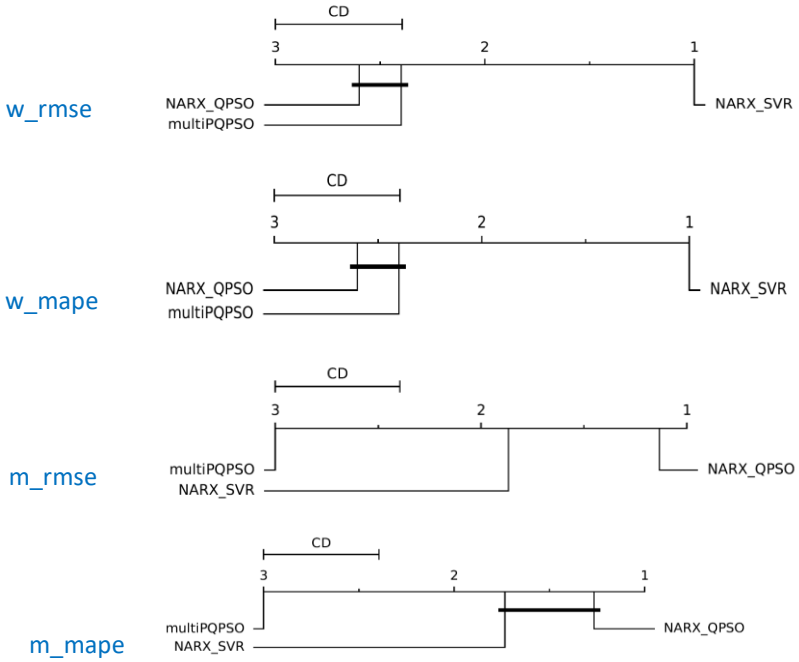


Figure 9 Nemenyi Test for RMSE and MAPE

A comparative study of multiPQPSO on resources utilization estimation to state-of-the-art techniques was performed using Alibaba and Bitbrains datasets. The experiments for multiPQPSO were conducted as described by Iqbal et al., 2019. Table 5 presents the results for the comparative study.

Table 5: Comparative study of multiPQPSO on Resource utilization estimation (Iqbal et al., 2019)

Technique	Alibaba		Bitbrains	
	MAE	RMSE	MAE	RMSE

Kriging	3.99	5.26	6.05	15.80
SVM	4.23	5.63	7.19	19.94
LR	3.87	5.12	6.03	15.01
GBT	3.43	4.57	2.85	9.74
AMS	2.29	3.32	2.57	9.13
Proposed	**1.87**	**1.94**	**1.93**	**5.96**

For the Alibaba dataset, the multiPQPSO enumerated an optimal observation window of 330 minutes and predicted a forecasting horizon of 95 minutes. For the Bitbrains dataset, the multiPQPSO enumerated an observation window of 225 minutes and predicted a forecasting horizon of 85 minutes. The proposed multiPQPSO technique exhibited an outstanding performance on both Alibaba and Bitbrains datasets and significantly outperformed state-of-the-art and baseline techniques presented in Table 5.

The presence of irrelevant data points in the observation window brings skewness to the forecasting model (Kuranga et al., 2020). As such, optimizing the observation window improves the accuracy of the forecasting model as proposed in this work. Also, a user-defined forecasting horizon may not be optimal for the induced predictive model which results in reduced performance.

6. Conclusion and Future Work

The proposed multiPQPSO algorithm decomposes the forecasting problem into three dimensions: model induction; observation window optimization; and prediction of a forecasting horizon. Each dimension was optimized using a given sub-swarm, and sub-swarms were executed in parallel. The results obtained show that an optimal observation window and forecasting horizon improve the predictive and computation performance of the predictive model. Also, the results obtained indicated that a multi-population QPSO approach was effective to induce a forecasting model of improved performance to outperform the benchmark techniques for all datasets. The proposed algorithm outperformed the state-of-the-art techniques for CPU utilization forecasting.

The future direction of this work may consider an empirical analysis of the search and solution spaces of multiPQPSO, and to perform a fitness landscape analysis of the proposed technique. Taking the improvement/worsening behavior of the induced predictive model to predict the forecasting horizon may be misleading. Therefore, it can be useful to consider the statistical features of the optimized observation window to predict the forecasting horizon. Deep learning techniques could have obtained results competitive to the proposed model. As such, a comparative study of the multiPQPSO to state-of-the-art deep learning model can be experimented on.

References

Alberg, D., & Last, M. (2018). Short-term load forecasting in smart meters with sliding window-based ARIMA algorithms. *Vietnam Journal of Computer Science*, *5*(3), 241–249.

Alibaba Cluster Log (Alib). Available: https://github.com/alibaba/clusterdata. Accessed: Jun. 1, 2022. [Online].

Arunkumar, P. M., & Ramasamy, L. K. (2021). Time-Series Forecasting and Analysis of COVID-19 Outbreak in Highly Populated Countries: A Data-Driven Approach. *International Journal of E-Health and Medical Communications (IJEHMC)*, *13*(2), 1–17.

*Australian bureau of meteorology (*AusMet*).* [Online]. Available http://www.bom.gov.au/. Accessed: Jun. 1, 2022.

Australian energy market operator (AEMO). [Online]. Available: http://www.aemo.com.au/. Accessed: Jun. 1, 2022.

Baek, Y., Yun, U., Kim, H., Nam, H., Kim, H., Lin, J. C.-W., Vo, B., & Pedrycz, W. (2021). RHUPS: Mining recent high utility patterns with sliding Window–based arrival time control over data streams. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(2), 1–27.

Blackwell, T. M., & Bentley, P. J. (2002). Dynamic search with a charged swarm. *In Proceedings of the 4th annual conference on genetic and evolutionary computation*, (pp. 19-26).

Bitbrains Cluster Log (BitB). Accessed: Jun. 1, 2022. [Online]. Available: http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains. Accessed: Jun. 1, 2022.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2016). *Time Series Analysis: Forecasting and Control. John Willey and Sons, New Jersey*.

Ciaburro, G., & Iannace, G. (2021). Machine Learning-Based Algorithms to Knowledge Extraction from Time Series Data: A Review. *Data*, *6*(6), 55.

Cuaresma, J. C., Hlouskova, J., Kossmeier, S., & Obersteiner, M. (2004). Forecasting electricity spot-prices using linear univariate time-series models. *Applied Energy*, *77*(1), 87–106.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7, 1-30.

Fentis, A., Bahatti, L., Tabaa, M., & Mestari, M. (2019). Short-term nonlinear autoregressive photovoltaic power forecasting using statistical learning approaches and in-situ observations. *International Journal of Energy and Environmental Engineering*, *10*(2), 189–206.

Georgieva, K., & Engelbrecht, A.P, "Dynamic Differential Evolution Algorithm for Clustering Temporal Data," *Large Scale Scientific Computing, Lecture Notes in Computer Science*, vol. 8353, pp. 240-247, 2014.

Gomes, H. M., Read, J., Bifet, A., Barddal, J. P., & Gama, J. (2019). Machine learning for streaming data: state of the art, challenges, and opportunities. *ACM SIGKDD Explorations Newsletter*, *21*(2), 6–22.

Gunning, D., & Aha, D. (2019). DARPA's explainable artificial intelligence (XAI) program. *AI Magazine*, *40*(2), 44–58.

Hannan, M. A., How, D. N. T., Lipu, M. S., Mansor, M., Ker, P. J., Dong, Z. Y., Sahari, K. S. M., Tiong, S. K., Muttaqi, K. M., & Mahlia, T. M. (2021). Deep learning approach towards accurate state of charge estimation for lithium-ion batteries using self-supervised transformer model. *Scientific Reports*, *11*(1), 1–13.

Herbold, S. (2020). Autorank: A python package for automated ranking of classifiers. *Journal of Open Source Software*, 5(48), 2173.

Iqbal, W., Berral, J. L., & Carrera, D. (2019). Adaptive sliding windows for improved estimation of data center resource utilization. *Future Generation Computer Systems*, *104*, 212–224.

Kandananond, K. (2012). A comparison of various forecasting methods for autocorrelated time series. *International Journal of Engineering Business Management*, *4*, 4.

Kang, Y., Hyndman, R. J., & Li, F. (2020). GRATIS: GeneRAting TIme Series with diverse and controllable characteristics. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *13*(4), 354–376.

Kuranga, C., & Pillay, N. (2022). A comparative study of nonlinear regression and autoregressive techniques in hybrid with particle swarm optimization for time-series forecasting. *Expert Systems with Applications*, *190*, 116163.

Kuranga, C., & Pillay, N. (2020). Nonlinear Regression in Dynamic Environments Using

Particle Swarm Optimization. *International Conference on the Theory and Practice of Natural Computing*, 133–144.
Li, H., & Wang, L. (2017). A variable size sliding window based frequent itemsets mining algorithm in data stream. *AIP Conference Proceedings*, *1839*(1), 20146.
Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media.
Ma, T., Antoniou, C., & Toledo, T. (2020). Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast. *Transportation Research Part C: Emerging Technologies*, *111*, 352–372.
Madsen, H. (2007). *Time series analysis*. Chapman and Hall/CRC.
Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018a). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PloS One*, *13*(3), e0194889.
Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018b). The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*(4), 802–808.
Maurya, S., & Shrivastava, S. K. (2015). Kalman filter based flexible sliding window algorithm for mining frequent itemset over data stream. *International Journal of Computer Applications*, *111*(9).
Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., & Talagala, T. S. (2020). FFORMA: Feature-based forecast model averaging. *International Journal of Forecasting*, *36*(1), 86–92.
Munawar, M. A., & Ward, P. A. S. (2007). A comparative study of pairwise regression techniques for problem determination. *Proceedings of the 2007 Conference of the Center for Advanced Studies on Collaborative Research*, 152–166.
Nti, I. K., Teimeh, M., Nyarko-Boateng, O., & Adekoya, A. F. (2020). *Electricity load forecasting: a systematic*.
Osman, N., Torki, M., ElNainay, M., AlHaidari, A., & Nabil, E. (2021). Artificial Intelligence-Based Model for Predicting the Effect of Governments' Measures on Community Mobility. *Alexandria Engineering Journal*, *60*(4), 3679–3692.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning research*. 12, 2825-2830.
Preece, A., Harborne, D., Braines, D., Tomsett, R., & Chakraborty, S. (2018). Stakeholders in explainable AI. *ArXiv Preprint ArXiv:1810.00184*.
Qiao, H., Wang, T., Wang, P., Qiao, S., & Zhang, L. (2018). A time-distributed spatiotemporal feature learning method for machine health monitoring with multi-sensor time series. *Sensors*, *18*(9), 2932.
Ristanoski, G., Liu, W., & Bailey, J. (2013). A time-dependent enhanced support vector machine for time series regression. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 946–954.
Smyl, S. (2017). Ensemble of specialized neural networks for time series forecasting. *37th International Symposium on Forecasting*.
Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, *36*(1), 75–85.
Struye, J., & Latré, S. (2020). Hierarchical temporal memory and recurrent neural networks for time series prediction: An empirical validation and reduction to multilayer perceptrons.

Neurocomputing, *396*, 291–301.
Surakhi, O., Zaidan, M. A., Fung, P. L., Hossein Motlagh, N., Serhan, S., AlKhanafseh, M., Ghoniem, R. M., & Hussein, T. (2021). Time-Lag Selection for Time-Series Forecasting Using Neural Network and Heuristic Algorithm. *Electronics, 10*(20), 2518.