energies

MDPI

Article

Adaptive Machine Learning for Automated Modeling of Residential Prosumer Agents

David Toquica [1,*], Kodjo Agbossou [1], Roland Malhamé [2], Nilson Henao [1], Sousso Kelouwani [3] and Alben Cardenas [1]

[1] Department of Electrical and Computer Engineering, Université du Québec à Trois-Rivières, 3351, Boul. des Forges, Trois-Rivières, QC G8Z 4M3, Canada; Kodjo.Agbossou@uqtr.ca (K.A.); Nilson.Henao@uqtr.ca (N.H.); Alben.Cardenasgonzalez@uqtr.ca (A.C.)

[2] Department of Electrical Engineering, Polytechnique Montréal, C.P. 6079, Succ. Centre-Ville, Montréal, QC H3C 3A7, Canada; roland.malhame@polymtl.ca

[3] Department of Mechanical Engineering, Université du Québec à Trois-Rivières, 3351, Boul. des Forges, Trois-Rivières, QC G8Z 4M3, Canada; Sousso.Kelouwani@uqtr.ca

* Correspondence: david.camilo.toquica.cardenas@uqtr.ca

Received: 26 February 2020; Accepted: 15 April 2020; Published: 4 May 2020

check for updates

Abstract: An efficient participation of prosumers in power system management depends on the quality of information they can obtain. Prosumers actions can be performed by automated agents that are operating in time-changing environments. Therefore, it is essential for them to deal with data stream problems in order to make reliable decisions based on the most accurate information. This paper provides an in-depth investigation of data and concept drift issues in accordance with residential prosumer agents. Additionally, the adaptation techniques, forgetting mechanisms, and learning strategies employed to handle these issues are explored. Accordingly, an approach is proposed to adapt the prosumer agent models to overcome the gradual and sudden concept drift concurrently. The suggested method is based on triggered adaptation techniques and performance-based forgetting mechanism. The results obtained in this study demonstrate that the proposed approach is capable of constructing efficient prosumer agents models with regard to the concept drift problem.

Keywords: adaptation; concept drift; data streaming; forecast; modeling; prosumer; regressor; supervised machine learning

1. Introduction

In recent years, the growth of distributed generation and residential prosumers [1] has motivated energy companies to develop new ways of commercializing energy. Their main objective is to reduce the cost and improve the power system management. Accordingly, decentralized optimization processes that enable more participation of final customers expected to help this ambition [2]. In order to facilitate higher cooperation of end customers (including both consumers and prosumers), intelligent decision-making systems are required [3]. In the residential sector, these systems' aim of minimizing the energy cost must also account for customers comfort. For instance, controllable loads, such as heating, ventilation, and air conditioning (HVAC), allow the cost reduction based on dynamic tariffs by taking into account preferable temperature set-points [4].

The intelligent systems are considered as agents since they can perceive the environment and take decisions according to an objective [5]. The quality of the decisions of an agent depends on the information that they have [6]. Consequently, it is crucial to have reliable information about the environment. However, information can become unreliable due to shifting on weather conditions, integration of new devices, change of user preferences, and degradation of appliances.

Generally, for a residential prosumer agent, it is possible to distinguish two environments that are labeled as local and external. The former refers to the behind-the-meter resources [7], while the latter describes situations where the prosumer agent can interact with other agents and information services. In most cases, the external environment only collects data of either weather variables or their forecast, but in a decentralized management scheme, it is possible to consider the external environment as a multi-agent system (MAS) [8]. The agent is able to perceive the local environment by observing the power consumption data of different appliances. In fact, it constructs a time-series database by accumulating new information from a data stream [9]. However, this process is problematic since agents have limitations on memory and processing time [10]. In addition, the data stream can *drift* over time, thus causing previously trained data models of appliances to lose accuracy [11]. Therefore, model adaptation on the basis of recent data is essential [12].

In this regard, several approaches have been developed to address the problems related to non-stationary data streams. In automated machine learning, active and adaptive learning algorithms have been utilized. The active learning techniques query for the information that they need each time for training a model [13]. Beside, the adaptive learning methods only update the models when they detect a drift [14]. Particularly, studies have considered retraining after fixed-size data windows and rule-based models as adaptation methods, applied to residential appliances models [15,16]. Therefore, they have underestimated the importance of drift issues on data management. Moreover, other researches have been conducted on training the models incrementally without analyzing the changes on data streams. For example, Farzan et al. increased the information of a transition matrix of a discrete-time Markov model to simulate both electricity and heating demands of individual households [17]. Yoo et al. trained a Kalman filter, recursively, in order to forecast a household load, considering temperature and occupancy variables [18]. However, adding the information of new samples directly to the models limits the set of models that can be used to represent the behavior of appliances. The applicability of these incremental learning techniques should be examined in detail according to the conditions of prosumer agents [19].

Considering the above restrictions, the main objective of our study is to investigate adaptation methods that can be useful for prosumer agents to have more reliable information. Through extensive analysis, this paper contributes to:

- The definition of the main criteria when choosing an adaptation algorithm in the context of prosumer agents. Here, the issues that the adaptation techniques address are examined to determine the best solution for models management, future concept assumptions, mixed drifts, and selection of training strategies.
- The identification of suitable algorithms for adapting the prosumer agents models to overcome environment changes. The algorithms are estimated with different adaptation strategies and forgetting mechanisms, such as Adaptive Windowing (ADWIN) [20], FISH [21], and Drift Detection Method (DDM) [22], in order to identify their required features in the prosumer agent context.
- Proposition of a new adaptation algorithm based on triggered adaptation techniques and performance-based forgetting mechanisms. The proposed method is non-iterative; thus, it is less computationally complex than other methods in the literature. The suggested approach is capable of training several appliances' models in relatively fast data streams for the prosumer agent context, using a single data window.

This paper is organized as follows: Section 2 presents the problem of prosumer agents obtaining accurate information in a changing environment. The concept drift definition with the implications of the prosumer agent is also discussed in this section. Section 3 explains the main criteria dictating the choice of a particular adaptation algorithm for the agents. Next, Section 4 presents the proposed adaptation algorithm for prosumer agent models. Section 5 presents the experimental setup and the comparison of results obtained by using different algorithms followed by the concluding remarks in Section 6.

2. Problem Statement

The local environment of the prosumer agent is composed of several appliances and local power generation systems. In order to reduce the number of variables, residential appliances are aggregated into two main groups, namely fixed loads and controllable loads. Controllable loads correspond to the appliances that can give some flexibility to the user, allowing him to modify energy consumption according to external signals he receives. Particularly, controllable loads that are examined in this study consist of heating systems because they are the most common flexibility source in the residential sector. In order to forecast both household power demand and know the thermal dynamics, the local environment is modeled with the data window that the agent creates from the data stream, as shown in Figure 1. These models have been generally constructed based on supervised machine learning methods. [23]. For fixed loads and local power generation, the models objective is only to forecast total power since the agent cannot take any action over these two elements. On the other hand, the model of controllable loads is developed to estimate the internal temperature based on the household power consumption profile, decided by the residential agent. As mentioned, this decision has to take into consideration the occupants' comfort levels.

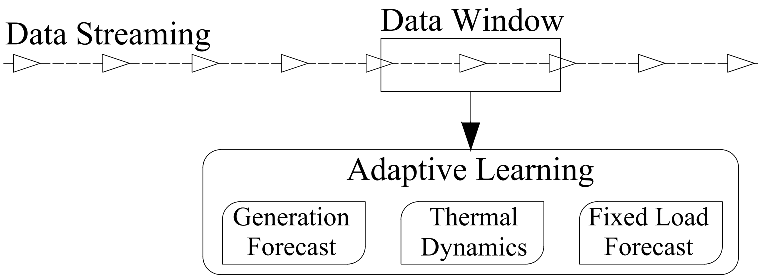


Figure 1. Adaptive learning application.

In all cases, residential prosumer agents need to adapt their models to new environmental conditions. An essential prerequisite to perform effective adaptations is to analyze the causes of data changes and the impacts of these changes on agents knowledge. In this context, it is important to notice that, for data-driven models, the underlying joint probability between features and targets is known as a concept, and it is assumed to be invariant over some interval of time. However, this underlying distribution could evolve over time sufficiently to cause a so-called concept drift [11]. The formal definition of concept drift is presented in Equation (1), where the X stands for the input features (or observations), y is the target variable (or label), and, hence, $\mathcal{P}(X, y)$ is the joint distribution (or concept [21]) evaluated at time t and $t + m$.

$$\mathcal{P}_t(X, y) \neq \mathcal{P}_{t+m}(X, y). \tag{1}$$

Due to causality between the features and the target, it is suitable to consider the following form of the joint probability distribution [24].

$$\mathcal{P}_t(X, y) = \mathcal{P}_t(y|X)\mathcal{P}_t(X), \tag{2}$$

where $\mathcal{P}_t(y|X)$ is the posterior probability of the target given the features, and $\mathcal{P}_t(X)$ is the prior distribution of the features. For example, weather variables as features are subject to changes with seasonal conditions that are explained by the term $\mathcal{P}_t(X)$. Besides, $\mathcal{P}_t(y|X)$ being the conditional distribution on features could also incorporate time-dependent effects, such as degradation of appliances.

As mentioned, the solution to deal with inaccuracy due to concept drift is to adapt the models parameters. However, before choosing an adaptation technique, it is meaningful to look at the characteristics of the concept drifts that can appear in the data streams of residential prosumers. Therefore, a prosumer-oriented study on the quantitative and qualitative measures of drift is provided below based on a general framework, discussed in Reference [25].

2.1. Drift Magnitude

Drift Magnitude $D(t, t+m)$ measures the distance between two probability distributions at two different instants t and $t+m$ for $m > 0$. Generally, drift magnitude is measured with the Hellinger distance H^2 since it has a non-negative value, and the distance between A and B concepts is the same as B and A [26]. However, it is possible to use other distance metrics, such as the total variation distance.

$$\begin{aligned} D(t, t+m) &= H^2(\mathcal{P}_t(X, y), \mathcal{P}_{t+m}(X, y)) \\ &= 1 - \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathcal{P}_t(X, y)\mathcal{P}_{t+m}(X, y)dXdy. \end{aligned} \tag{3}$$

The drift magnitude can be used for drift classification. For example, a minor drift does not necessitate training the model. However, a major drift can imply the need for either retraining or changing the models (in case of ensemble learning mechanisms). In the data streams that a prosumer receives, it is possible to identify several minor drifts during the day at short timesteps that do not necessarily match with a gradual drift, especially for the fixed loads. Besides, when the time step goes bigger, the drift magnitude does not always increase, which means that there could be recurring concepts. For example, for the power generation model, if the prosumer has photovoltaic panels, the concept at dawn may be closer to the concept at night than at noon.

2.2. Drift Duration

The drift duration noted as m can define either a sudden (abrupt) or progressive (gradual or extended) transition between two concepts, depending on its value. As an example, in the local environment of a prosumer agent, a sudden concept drift (small drift duration) occurs when appliances are added, removed, or changed. Other special cases of concept transition like blip drifts and probabilistic drifts are hard to recognize by the prosumer since they can be mistaken for outliers in the data stream.

2.3. Drift Subject

Distinguishing a drift subject is difficult for prosumer agents since similar changes on features distribution can cause different drift subjects. However, it is important to mention that not all the drifts will require retraining the models. Normally, two categories of drift types are considered according to its causes [25]:

- Real concept drift: This type occurs when the posterior probability, $\mathcal{P}(y|X)$ changes over time and requires a retraining of the model. The change can occur in either a portion of the domain of X (sub-concept drift) or all of it (Full-concept drift).

$$\mathcal{P}_t(y|X) \neq \mathcal{P}_{t+m}(y|X). \tag{4}$$

- Virtual concept drift: It happens when, instead, it is the distribution of the features $\mathcal{P}(X)$, which changes over time while the posterior probability, $\mathcal{P}(y|X)$ remains the same. In that case, it is not always necessary to update the model.

$$\mathcal{P}_t(y|X) = \mathcal{P}_{t+m}(y|X) \quad \mathcal{P}_t(X) \neq \mathcal{P}_{t+m}(X). \tag{5}$$

2.4. Drift Predictability

The drift in a data stream can be related to independent events, such as seasons and days of the week. Consequently, it is possible to predict some aspects of the drift if the occurrence of these events is known. Furthermore, a concept drift can be predicted if a known recurrence pattern exists. Nevertheless, in the context of a prosumer agent, there can be different concept drifts coming from different environmental changes that cannot be well differentiated, so it will be barely impossible to anticipate the occurrence of certain types of drift. Moreover, define a magnitude threshold to identify the data that corresponds to previously seen concepts depends on the timestep that the prosumer agent uses.

3. Adaptive Algorithms

In this section, the adaptation algorithms are classified in order to identify suitable methods for residential prosumer agents problems. This classification is preceded by an examination of sub-issues that an adaptation algorithm faces [21]:

- **The appearance of gradual drifts makes it impractical to assume that the concept of future data is always closer to the latest data.** Therefore, instead of assuming the concept of future data, it will be useful to implement an algorithm that recognizes the distribution of the features of arriving data. Besides, in some applications of prosumers, the robustness of the methods is essential to differentiate outliers from concept drifts. However, in this study, the agent trusts the external information he receives from measurement systems and weather information services.
- **In the data stream, there could be different kinds of drifts mixed and outliers data samples.** Another important concern with the concept drift in the specific context of the prosumer agent is the concurrence of the drift types. Thus the agent could be facing sudden drifts, gradual drifts, and incremental drifts within one timeframe [27]. For that reason, adaptation algorithms that were made to solve problems related to specific cases of drift are not the best option for the agent. Here, we test some of those algorithms to validate this affirmation.
- **It is impractical for a prosumer agent to have different models trained with different data sets and ensemble the forecasts.** Therefore, considering the available mechanisms to update the agents knowledge, when using a single model, the only strategy is to adapt the parameters. Nevertheless, in addition to parameters' adaptation, it is also possible to combine the models by weighting them as in ensemble learning [28]. The choice of the method should be made by taking into account the residential agent's restrictions related to the processing time and hardware limitation. Normally, the models of three main groups are used to provide information for other processing systems, such as a Home Energy Management Systems (HEMS), with a practical objective, like either minimizing energy cost or maximizing comfort. These systems usually provide results in five to fifteen minutes intervals, thus limiting convenient exploitation of ensemble learning.
- **Not all adaptation techniques may work for all the models.** The adaptation of parameters depends on the model type since some models can be trained incrementally, for example, by using adaptive linear neuron rule (ADALINE) or recursive least squares (RLS) [29], while others have to start from zero every time. The drawback of incremental learning is that outliers are directly included in the model's knowledge. Notwithstanding, it can reduce memory usage and processing time. Besides, the choice of the learning strategy depends on the nature of data and the rate of data collection. Thus, the residential agent could be receiving and processing the data under both forms of single or batch measurements. For instance, the labels of the models could be given every time while the features could be queried in batches each several hours.

According to the above considerations, the taxonomy of adaptive learning algorithms is presented in Figure 2. This classification is different from other ones, previously provided in Reference [14,30], due to the specificity of the prosumer agent. The new arrangement is to highlight the main concerns of agents over the choice of an algorithm, involving an adaptation strategy, model management, data acquisition, a forgetting mechanism, and a learning strategy. In Figure 2, blind adaptation refers to a case in which the algorithm updates the model parameters at a pre-defined frequency without specific verification of the occurrence of concept drift in the data stream [30]. Alternatively, some algorithms utilize a method to detect concept drift before triggering the adaptation. This mechanism is sometimes referred to as active adaptation.

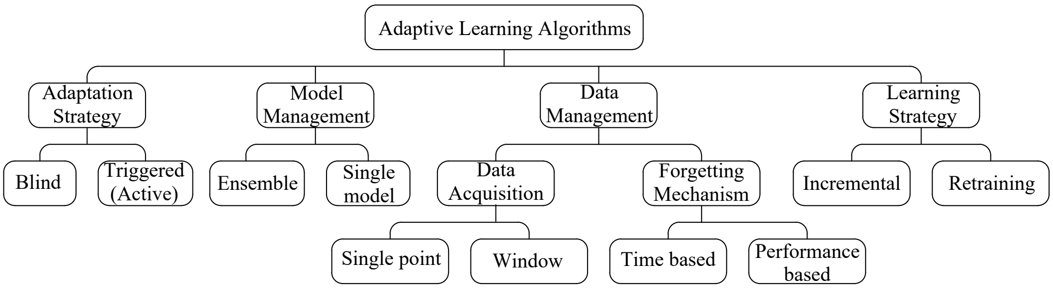


Figure 2. Taxonomy of adaptive learning.

The forgetting mechanism can be time-based in case the oldest samples are deleted, while the size of the data window either is kept fixed, changes according to a rule, or assigns fading factors to make old samples irrelevant. In addition, this mechanism can be performance-based in accordance with the adequacy of the samples for training the model or their similarity with future samples considering their statistical properties.

Now, with all the considerations for the prosumer agents' problem, the following algorithms were selected to adapt the models of the local appliances. In addition, they present some remarkable characteristics that can be useful to formulate new methods.

3.1. Drift Detection Method

This method proposed in Reference [22], starts with the premise that more items of the same concept in the data window will reduce prediction error. Consequently, it is possible to take an error increase as a proof of concept drifts [31]. This is under the assumption that the base learner controls over-fitting.

For classifiers, the error can be modeled as a random variable from Bernoulli trials; then, the Binomial distribution gives the general form of the probability for the variable. In that context, if $\mathcal{P}_t$ is the error rate of the learner at time t, then the standard deviation s_t will depend on the window size N_t as follows [22]:

$$s_t = \sqrt{\frac{\mathcal{P}_t(1-\mathcal{P}_t)}{N_t}}. \tag{6}$$

The algorithm storages the minimum value encountered of $\mathcal{P}_{\min}$ and the corresponding $s_{\min}$. Subsequently, two validations are made according to the confidence level:

- Warning level: confidence level is 95%, so it is reached when $\mathcal{P}_t + s_t \geq \mathcal{P}_{\min} + 2s_{\min}$.
- Drift detection: confidence level is 99%, so it is reached when $\mathcal{P}_t + s_t \geq \mathcal{P}_{\min} + 3s_{\min}$.

When the drift is detected, the onset of a new concept is declared starting at the time when the warning first appeared. As can be noted, this method was designed for sudden drifts. Hence, for residential agents, the algorithm will detect a drift several consecutive times when there are gradual drifts. For that reason, in order to apply this method, it was deemed necessary to define a minimum distance in time between a warning level and drift detection. Furthermore, the regressor models of the residential agent need to be considered as multi-label classifiers to implement this method. The method is summarized in Algorithm 1.

Algorithm 1: Drift Detection method.

```
input  : Data stream of labels $(y)$ and features$(X)$
output: Trained model
Set initial conditions of $\mathcal{P}_{min}$ and $s_{min}$;
begin
    if New sample is added at time $t$ then
        Calculate $\mathcal{P}_t$ and $s_t$ with Equation (6);
        Compare $\mathcal{P}_t + s_t$ with the warning level and the drift detection conditions;
        if A drift is detected then
            Resize the window data since the warning level appeared or to a minimum size;
        end
        Train the model with the resulting data window;
    end
end
```

3.2. Gold Ratio Method

This method proposed in Reference [32] was also designed for adapting to sudden drifts. It assumes that the concept has changed if, at any time, the error of the model surpasses a defined level. The significance test must be sensitive enough to discover concept drift as soon as possible and robust to avoid mischaracterizing noise as concept changes. When the accuracy decreases, the oldest examples should be forgotten, and the size of the data window can be optimized by using a search algorithm in one dimension. The search algorithm used here is the Golden Ratio method. Considering a unimodal function, in the interval between a minimum window size (N_{min}) and the current size ($N_{current}$), the accuracy function has only one max at N_{opt}. Then, the algorithm minimizes the number of function evaluations by dividing the range using the golden ratio ($\tau = 0.618$); in so doing, the optimal window size is clustered fast [32].

The stopping criteria for the method is the minimum size of the search interval (100 samples proves to be adequate for the prosumer agent case). The significance test was done by using the root mean squared error (RMSE), and the acceptance level was adjusted according to the base learner. For the fixed load model, it is suitable to accept a higher RMSE since it is the most unpredictable signal. If no concept drift is detected, then the windows continue growing by adding more samples because a bigger training set will improve learning results if the concept is stable. The complete procedure is presented in Algorithm 2.

3.3. Klinkenberg and Joachims' Algorithm

The idea of this method proposed in Reference [33] is to select the window size to minimize the estimated generalization error. In the original case, the base learner is a support vector machine (SVM) because of the residual errors of the training $(\alpha\epsilon - estimator)$ can give an upper bound of the leave-one-out errors without training the learner several times. However, the generalization error can still be minimized by using different window sizes and using k-fold cross-validation, assuming that most recent samples have a higher correlation with future features. In that way, this method can also be seen as a search method as presented in Algorithm 3.

Algorithm 2: Gold ratio method.

```
input  : Data stream of labels $(y)$ and features $(X)$
output : Trained model
begin
    When a sample is added, calculate the $R^2$ coefficient of the regressors;
    if $R^2$ coefficient is lower than expected then
        $N_a = N_{min} + 0.618(N_{current} - N_{min})$;
        $N_b = N_{min} + (1 - 0.618)(N_{current} - N_{min})$;
        Train the model with the $N_a$ and $N_b$ most recent samples;
        if the accuracy of the model with $N_a$ data is bigger than the one with $N_b$ then
            The new search space is between $[N_b, N_{current}]$;
        else
            The new search space is between $[N_{min}, N_a]$;
        end
        Train the model with the resulting data window;
    end
end
```

Algorithm 3: Klinkenberg and Joachims' Algorithm.

```
input  : Data stream of labels $(y)$ and features $(X)$
output : Trained model
begin
    Divide the current data window in k sub-windows;
    Take the last sub-window for testing (cross-validation);
    for $k - 1$ times do
        Train the model adding an older sub-window each time;
        Perform the cross-validation test;
    end
    Train the model with the number of sub-windows that performed better on the test;
end
```

3.4. Fish Method

For sudden concept drift, it should be possible to find the moment of the drift t_D by evaluating extensively for all the samples the likelihood between the distributions before and after that sample, using the Hotelling test, for example [21].

$$\mathcal{T}_D = \max \mathcal{P}(drift|t_D). \tag{7}$$

However, when the drift is gradual, it is not only the distance in time which is relevant to select the samples needed to train the models. For that reason, the Fish method proposes to estimate also the distance in the feature space of the samples respect to the new ones [21]. Thus, for each sample in the window, it will be necessary to calculate the combined distance [34].

$$d_t = d_t^{(T)} + \beta d_t^{(s)}, \tag{8}$$

where $d_t^{(T)}$ is the distance in time, and $d_t^{(s)}$ is the distance in feature space of a sample in time t with respect to the future features. Note that the existence of gradual drifts makes it necessary to give some relevance to the distance on feature space $\beta > 0$. Subsequently, by organizing the samples according to their distance, it is easy to pick up the N closest samples to train the models. This method is best suited for the residential agents' problems since it can account for both sudden and gradual drifts. The problem is to define a good distance function that gives reliable insights without increasing too much the processing time. The prosumer models have few features, so the feature space distance is calculated as the Euclidean distance because it is easy to compute and works well in low-dimensional spaces [35]. The method is summarized in Algorithm 4.

Algorithm 4: Fish method.

```
input  : Data stream of labels $(y)$ and features $(X)$
output : Trained model
begin
    Calculate the distance $d_t$ for each sample in the data window;
    Organize the samples according to their distance;
    Train the model with the N closest samples;
end
```

3.5. ADWIN

This algorithm proposed in Reference [20] is based on hypothesis testing. The idea is that, when two sub-windows have different averages μ, the expected values are different. Like in a Hoeffding test, the null hypothesis is that the expected values are the same. The test can be written as follows for two different sub-windows of sizes N_0 and N_1: If $|\hat{\mu}_0 - \hat{\mu}_1| > \epsilon$, then the null hypothesis is false. Here, $\hat{\mu}$ corresponds to the average value of the sub-window, and ϵ depends on the confidence value δ according to:

$$\epsilon = \sqrt{\frac{N_0 + N_1}{2N_0 N_1} \ln \frac{4(N_0 + N_1)}{\delta}}. \tag{9}$$

For the problem of residential agents, analyzed in this paper, the confidence value is taken as 0.5%. The ADWIN method is summarized in Algorithm 5.

Algorithm 5: ADWIN algorithm.

```
input  : Data stream of labels $(y)$ and features $(X)$
output : Trained model
begin
    Divide the available data in sub-windows;
    Calculate $\epsilon$ with Equation (9);
    while the null hypothesis is false do
        Drop the oldest sample;
        Perform the hypothesis test for every pair of sub-windows;
    end
    Train the model with the resulting data window;
end
```

4. Proposed Algorithm

Given the collection of algorithms we presented above, we propose herein a hybrid approach that we think is best suited for the prosumer agents problem. Similar to the Drift Detection method and Gold Ratio method, the proposal is to use triggered adaptation. The algorithm will reduce the training set of the models only when the expected performance drops below a defined value, not each time that a new sample arrives.

To check the performance of the models, the procedure will be cross-validation using batches of the same size as the forecasting horizon (24 h, in this case), like in the Klinkenberg and Joachim's method [33]. The measurement of fit, in this case, is the RMSE because it gives more weight to bigger deviations; thus, it is better to identify the appearance of concept drifts [36]. It is relevant to mention that the threshold to accept the results of the cross-validation depends on the nature of the target variable of each model [37]. The test data set will be the closest batch to future features. Now, to identify that batch, the distance will be measured as in the FISH method [34] as a combined distance in time and space of the samples. If the result of the cross-validation test is not good enough, then the model will be retrained only with the closest N batches. The parameter N needs to be tuned according to the model to avoid convergence problems in training but knowing that, when a concept drift appears, it is safer to train with a small amount of data to ensure that all samples correspond to the new concept. Figure 3 summarizes the proposed algorithm with the procedure for when new data arrives.

Furthermore, forgetting data is a risky task since there are gradual drifts. The proposal here is to store a data window that contains the N selected samples, as well as the samples that were not selected but fall in between selected samples.

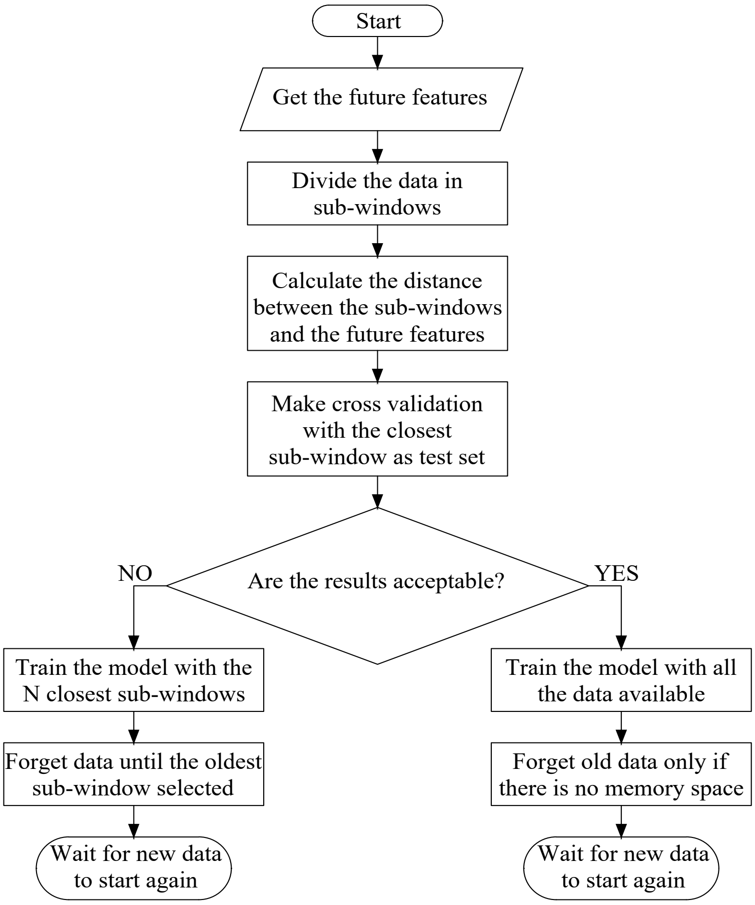

Start
Get the future features
Divide the data in sub-windows
Calculate the distance between the sub-windows and the future features
Make cross validation with the closest sub-window as test set
Are the results acceptable?
NO
YES
Train the model with the N closest sub-windows
Forget data until the oldest sub-window selected
Wait for new data to start again
Train the model with all the data available
Forget old data only if there is no memory space
Wait for new data to start again

Figure 3. Flowchart of the proposed algorithm.

5. Numerical Results

In order to validate our proposal, we now report the results of a numerical experiment where we use a neural network as the power generation model, a decision tree for the fixed load, and a linear model for the thermal load. The details of the models implemented on Scikit-learn [38] are shown here below:

Power Generation: The base learner is a feed-forward neural network that forecasts the power output of a generation system P_{gen}, as shown in Figure 4. The hidden layers of the network have 100 × 5 neurons; the activation function is a hyperbolic tangent, the step size is fixed at 0.0001, the initial state of weights is 1, and the method used to train the model is stochastic gradient descent. The data used in this case was synthetically created by simulating a photovoltaic array in PVlib [39] library with random cloud coverage (from 0 to 100% with transmittance offset of 0.75) and real temperature of Trois-Rivières, QC, in 2018.

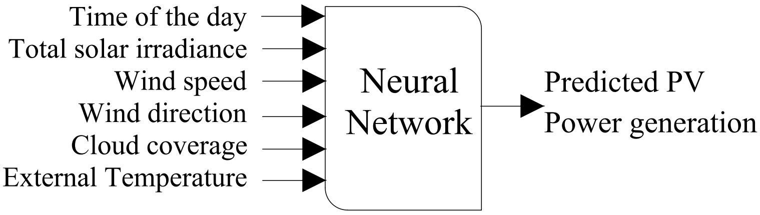


Figure 4. Model of power generation.

Fixed Load: The model to forecast the fixed load consumption P_{fix} is a decision tree where the quality of a split is measured using the mean squared error, and nodes are expanded until all leaves are pure [40]. The variables considered in this model are a cosine signal with a period of 24 h, the number of the day (from 1 to 7), the temperature, and the previous consumption (since the agent obtains data every 5 min, 288 samples correspond to 24 h). This model is presented in Figure 5. The data used in this case is real measurements of the power demand of a house in Trois-rivières, QC, during 2018.

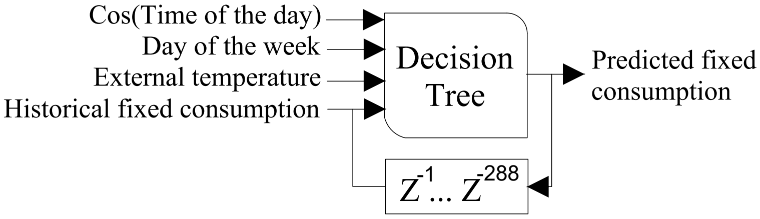


Figure 5. Model of fixed load.

Controllable Load (Thermal model): This model s based on the equivalent circuit 5R1C proposed in the standard ISO 13790:2008 [41]. The inputs in this case are the external temperature T_{ext}, the fixed load consumption P_{fix}, the solar irradiance P_{irr}, and the power demand of the heating system P_{heat}. The output will be the internal air temperature T_{air}. We assume that there is no special ventilation system, so there is only one external temperature as shown in Figure 6.

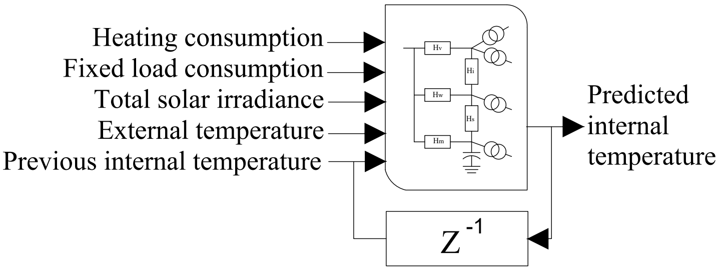


Figure 6. Model of thermal dynamics.

The following equation describes the model.

$$T_{air}^{(t)} = \alpha_1 T_{ext}^{(t)} + \alpha_2 T_{air}^{(t-1)} + \beta_1 P_{heat}^{(t)} + \beta_2 P_{fix}^{(t)} + \beta_3 P_{irr}^{(t)}, \tag{10}$$

where $T_{air}^{(t-1)}$ corresponds to the internal temperature at timestep $t-1$. This linear model is trained by using Ordinary Least Squares (OLS) to find the parameters α_1, α_2, β_1, β_2, and β_3, which are combinations of the original parameters of the standard circuit model. The data used, in this case, corresponds to real measurements in a house in Trois-rivières, QC, during 2018.

The results presented here correspond to simulations starting in two different days when there are suspicions of concept drift: A spring day (19 April 2018) and a summer day (23 August 2018). As mentioned, when the concept changes, old data gives inappropriate data to the models, therefore the error is reduced by training only with the most recent data. In the selected days, there are symptoms of concept drift because when training the models with sliding windows, some times the smaller data window leads to lower errors. Training with sliding windows can be seen as a way to perform adaptation because, every time the agent receives new data, it retrains the models with the most recent data, adding the new samples and forgetting the oldest. Here, the training with sliding windows is used to detect possible concept drifts in data with the evolution of the RMSE. The normalized RMSE (RMSE) is obtained by dividing the RMSE into the range of the label signal.

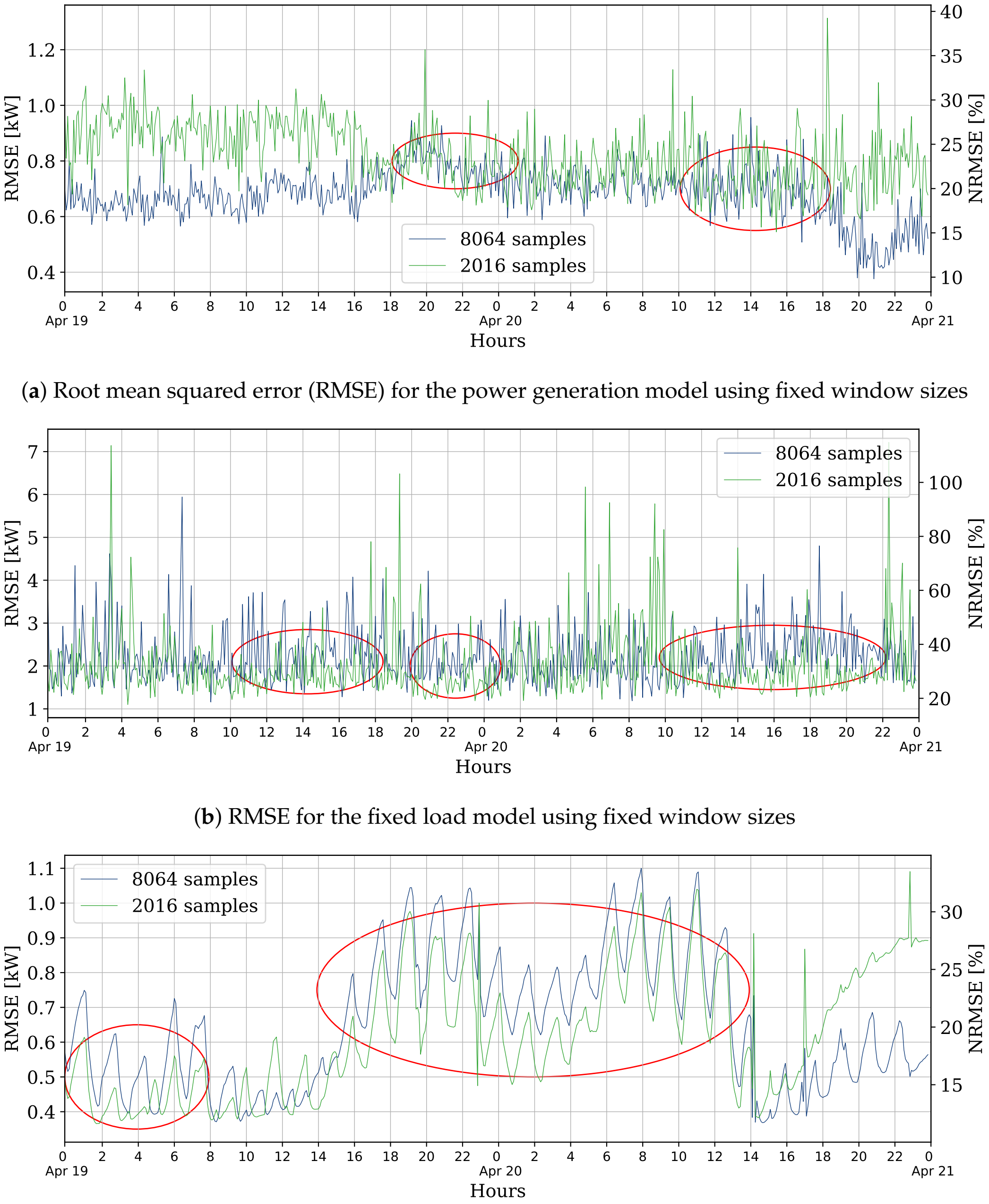
The maximum limit of training data for each model is 8064 samples (28 days sampled every 5 min), and the minimum to ensure convergence is 2016 (7 days). The limits were established according to the models, the linear thermal model can be fitted with less than 2016 samples, but the power generation model does not converge with less than that. This information about the base learners is relevant since it is used to tune parameters of the adaptation methods. Here, the models started with previous knowledge (1 month of data) before beginning the adaptation. In the case that the agent does not have enough information, the parameters of the models can start with prior values [42].

5.1. Spring Day

As can be seen in Figure 7 with the evolution of the RMSE, running over 48 h, some periods seem to present concept drifts. For the power generation model, close to the 19 h of April 19th, it is possible to see an increase of the error when using a window of 8064 samples, that ends close to the 18 h of the next day. Similarly, for the thermal model, the biggest data window lead to a higher error most of the time, until the 14 h of April 20th. For these reasons, these days were chosen to test the adaptation techniques. On the other hand, for the fixed load forecast, the errors with both sizes of sliding windows are close most of the time, and some outliers on the label data may create prominent peaks.

Before presenting the results, it is pertinent to mention some characteristics of the the weather data on these days: the external temperature fluctuates in a range from 1.4 °C and 9.3 °C, with a mean of 6.06 °C; the maximum irradiation of 945 $\mathrm{kJ/m^2/h}$ is reached at 12 h of the first day. The heating system of the house has a capacity of 15 kW, and the internal temperature goes from 17.6 °C to 21.1 °C, with an average of 19 °C.

Afterwards, the models were adapted during the selected period with the techniques presented earlier. The average RMSE (and NRMS) for each case is presented in Table 1. The results obtained with the proposed method are systematically better than the most straightforward adaptation by sliding window, which means that implementing the method gives more reliable information to the prosumer than not doing so, even though the error reduction may be small. Other algorithms will occasionally perform better for adapting specific models. For example, the golden ratio method reduces the error when adapting the power generation model, but it is not suitable to adapt the thermal load model. Furthermore, the proposed method forgets less data, thus making it possible for the prosumer to use a single database for all three models.



(**a**) Root mean squared error (RMSE) for the power generation model using fixed window sizes

(**b**) RMSE for the fixed load model using fixed window sizes

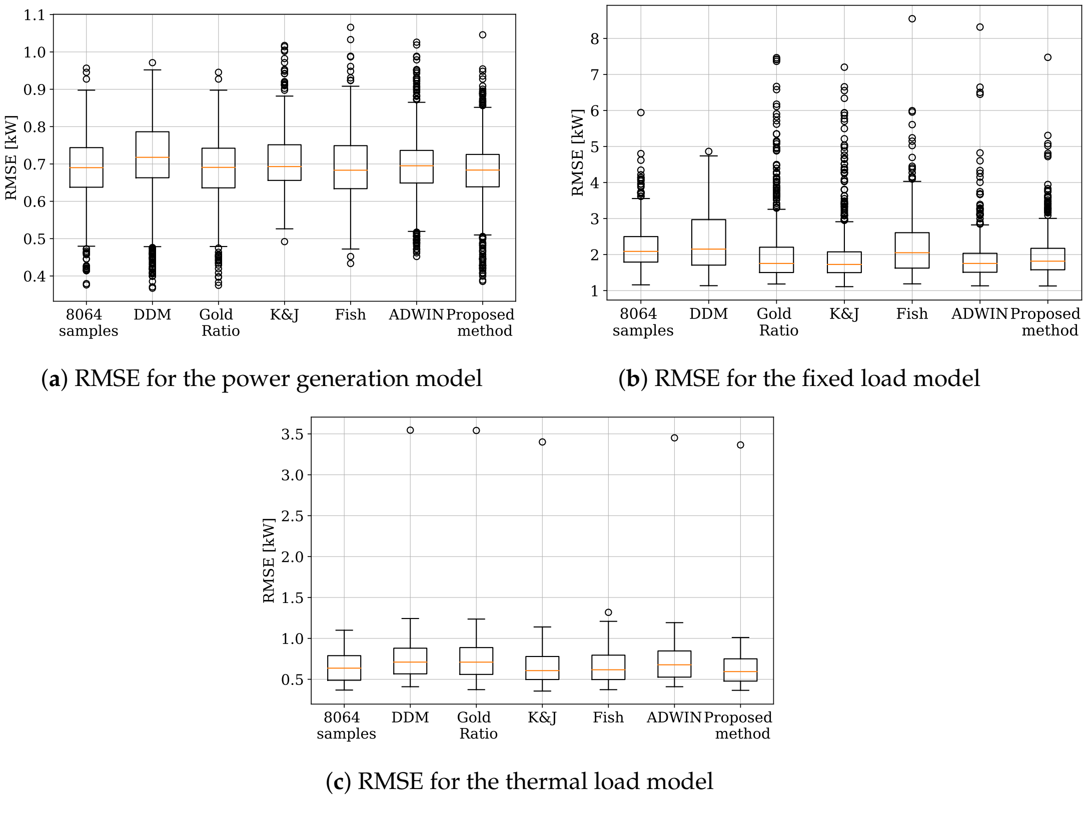
(**c**) RMSE for the thermal model using fixed window sizes

Figure 7. Identification of possible concept drifts in a spring day.

Another aspect that should be analyzed when choosing an algorithm is the dispersion of the error. Algorithms can be deluded by outliers and recurring concepts, leading to an increase in the error. Consequently, choosing a robust algorithm with low average error and low dispersion is important to give reliable information to the prosumer. In this regard, the performance of the proposed method is comparable with the others, as can be seen in Figure 8. One compelling case is the FISH method to adapt the thermal model because it is less sensible than the all the other algorithms.

Table 1. Average RMSE and Average NRMSE for a spring day.

	Power Generation [kW]	Fixed Load [kW]	Thermal Load [°C]
Full memory (8064 samples)	0.68479 (20.89%)	2.22717 (35.71%)	0.65367 (21.57%)
Drift Detection Method	0.70753 (21.65%)	2.36283 (37.90%)	0.74063 (24.37%)
Gold Ratio Method	0.68168 (20.77%)	2.08994 (33.47%)	0.7432 (24.45%)
Klinkenberg Method	0.70976 (21.64%)	1.99740 (32.02%)	0.653353 (22.01%)
FISH Method	0.69418 (21.15%)	2.22050 (35.63%)	0.64908 (21.59%)
Adwin Method	0.69573 (21.19%)	1.89000 (30.31%)	0.70723 (23.33%)
Proposed Method	0.67787 (20.67%)	1.97518 (31.68%)	0.62385 (20.52%)



(**a**) RMSE for the power generation model

(**b**) RMSE for the fixed load model

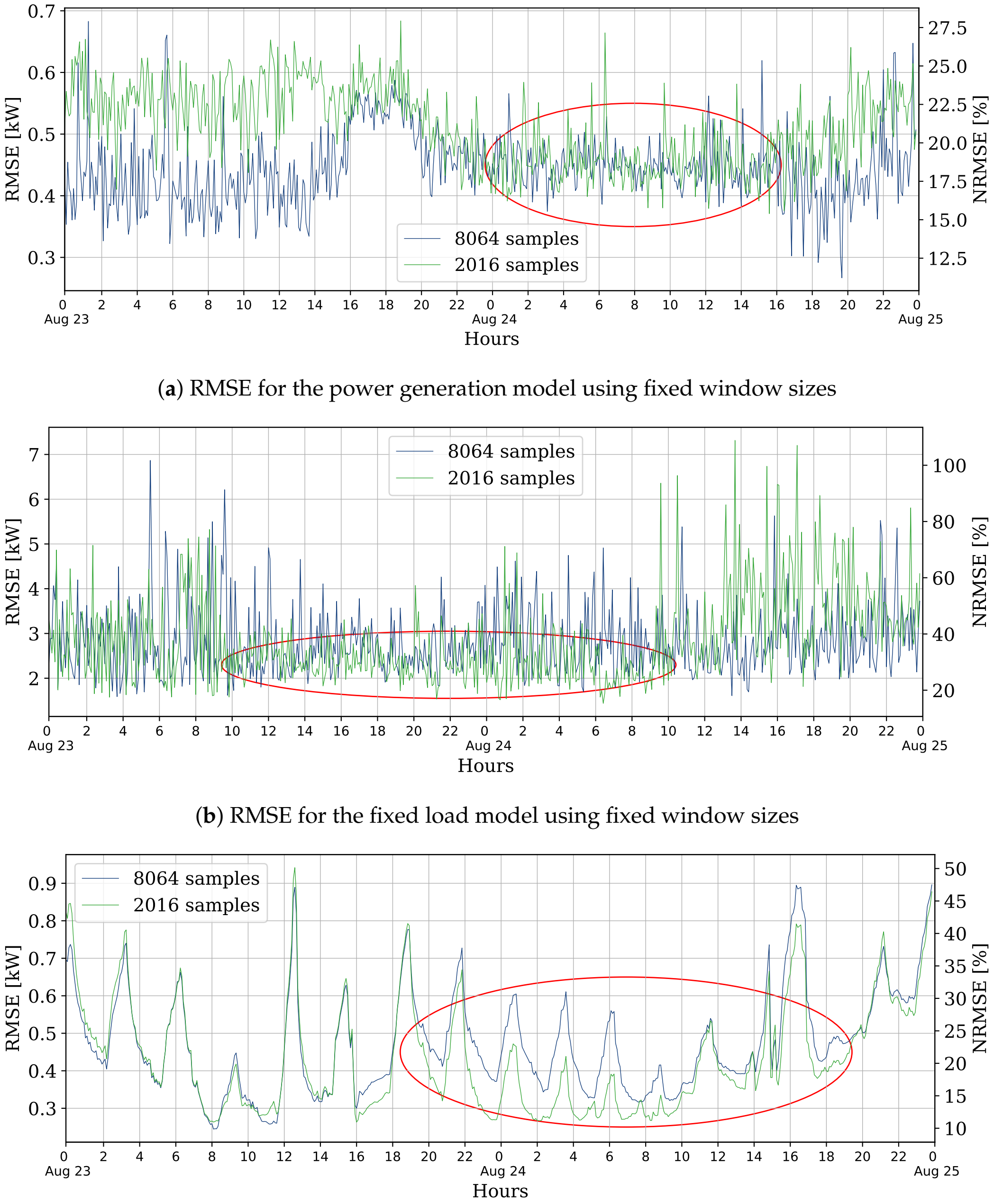
(**c**) RMSE for the thermal load model

Figure 8. RMSE dispersion in the spring day.

5.2. Summer Day

Similarly to the procedure to choose the spring day, we train the models using sliding windows to identify another day with signs of concept drift. On August 23rd, at 16 h, the error of the power generation model goes bigger when using 8064 samples than 2016, as shown in Figure 9. For the thermal load, the error with the bigger data window also surpasses the other at the same time. Again, the results for the fixed load model are close, making it challenging to identify concept drifts. On these selected days, the external temperature has a range from 12.6 °C to 29 °C, with a mean of 22.4 °C;

the maximum irradiation is 2900 kJ/m^2/h; and the internal temperature goes from 24 °C to 28.3 °C, with an average of 25.8 °C.



(**a**) RMSE for the power generation model using fixed window sizes

(**b**) RMSE for the fixed load model using fixed window sizes

(**c**) RMSE for the thermal model using fixed window sizes

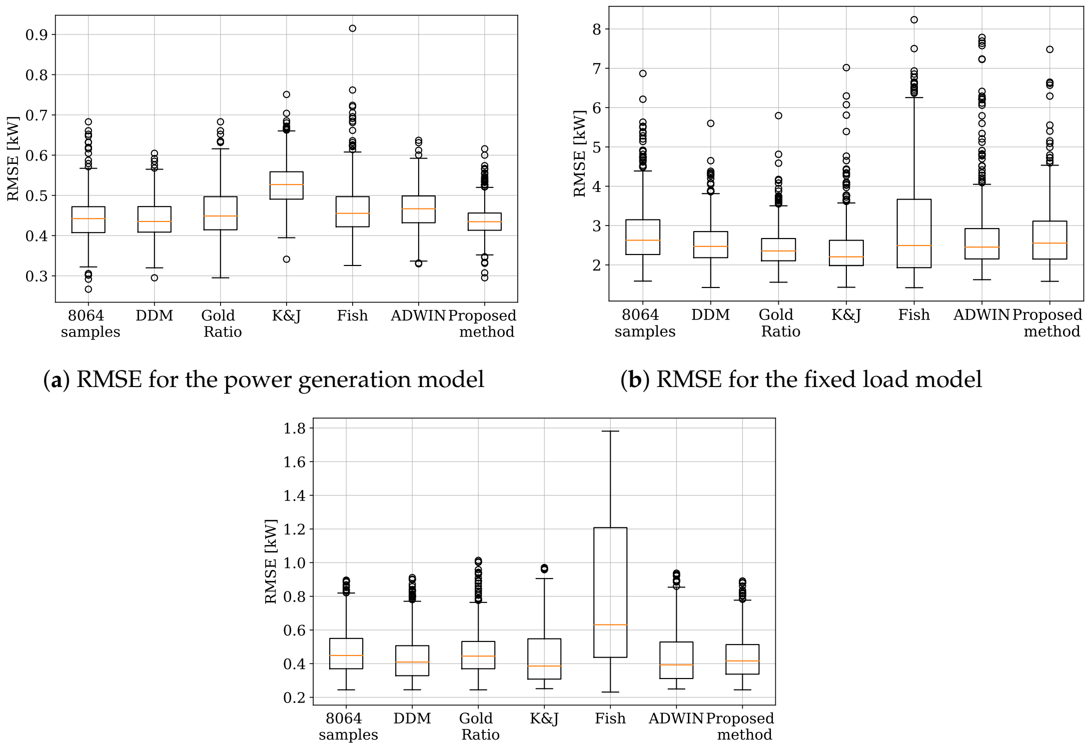
Figure 9. Identification of possible concept drifts in a summer day.

In Table 2, the average RMSE for the selected summer days is displayed. Again, the proposed method performs better than the sliding window for all models. Nevertheless, for the fixed load model, the Klinkenberg and Joachims method achieves a lower average error. This result suggests that adapting different models with different techniques is preferable in some cases, even if that implies managing different databases for each model.

Table 2. Average RMSE and average NRMSE for a summer day.

	Power Generation [kW]	**Fixed Load [kW]**	**Thermal Load [°C]**
Full memory (8064 samples)	0.44479 (17.32%)	2.80095 (34.55%)	0.47161 (19.82%)
Drift Detection Method	0.44250 (17.24%)	2.55943 (31.58%)	0.44122 (18.775%)
Gold Ratio Method	0.45597 (17.80%)	2.43693 (30.04%)	0.47094 (19.79%)
Klinkenberg Method	0.53091 (20.64%)	2.39652 (29.55%)	0.44375 (19.04%)
FISH Method	0.46619 (18.17%)	2.95768 (36.87%)	0.79645 (32.82%)
Adwin Method	0.46240 (18.05%)	2.69773 (33.26%)	0.43769 (18.81%)
Proposed Method	0.43797 (17.06%)	2.71268 (33.52%)	0.4423 (18.68%)

In this case, the dispersion of the error is comparable for all the methods, but, for the fixed load model, Drift Detection and Gold Ratio Methods seem to be less sensitive to outliers and recurrent concepts. On the other hand, in Figure 10, it is shown that the FISH method has the biggest dispersion in all three models, which means it is not suitable for the kind of drift that appears on these days.



(**a**) RMSE for the power generation model

(**b**) RMSE for the fixed load model

(**c**) RMSE for the thermal load model

Figure 10. RMSE dispersion in the summer day.

5.3. Results Analysis

As mentioned before, for the case of the prosumer agents, concept drift can appear in different ways, and they are hard to predict. As a consequence, forgetting data becomes a difficult decision because the agent could lose relevant information for the forecast. In this context, the proposal of keeping in memory more data than only the training set seems to be adequate for the agent. However, for some models, other methods obtain a better result using fewer data. For example, for the fixed load model, ADWIN and Klinkenberg's methods had the lowest average errors. It is relevant to mention that these methods assume that future samples are more strongly related to the most recent data in memory, and it is not possible to keep that assumption when there are gradual drifts.

Another key finding of the experiment is in comparison with the average error when using a fixed data window. In particular, it is possible to observe that some adaptation techniques were not adequate since they lead to an increase in the error. In contrast, the proposed method always had a better performance than using a simple sliding data window. As for the parameters of the proposed algorithm, the error threshold for accepting the result of the cross-validation was 0.8 kW for the power generation model, 2 kW for the fixed load model, and 0.8 °C for the thermal model. For the parameter N, the used value was 2016 since it ensures convergence on the training of all the models in the tested periods. Tuning algorithms parameters depends on knowledge about the performance of the base learners, so the agents should have validated the models beforehand.

A final remark is that most of the algorithms have a fixed number of operations or training processes. The only method that has an indefinite amount of operations is the gold ratio search, for which the complexity is estimated at $\mathcal{O}(\log \frac{1}{100})$ (100 is the minimum search interval in this case) [43]. Inevitably, this method takes more processing time than the others presented here. Since the models are trained with different amounts of samples every time, it is complicated to establish a metric to compare processing times.

6. Conclusions

Prosumer agents will play an essential role in energy management systems as they move towards decentralization. Then, it will be necessary to address their information needs to achieve adequate decisions. In particular, it is important to recognize that the agent environment is subject to change, and the models of different appliances need to be adapted to overcome different kinds of concept drift.

In order to formulate strategies to give the residential prosumers more reliable information upon which to base their decisions, this paper presented relevant criteria to account for when choosing adaptive algorithms to train the models of the local environment. Some of the existing algorithms in the literature were tested here with satisfactory results as they occasionally performed better than having a fixed window size.

Additionally, in this paper, we proposed another method that works for the information needs of the agents. The method depends on the number of batches selected to train the models; this parameter should be small to ensure that most of the samples correspond to the new concept but big enough to ensure the convergence when training the model. In experimental results, this algorithm was the only one that systematically produced better results than that obtained with a fixed data window size.

Author Contributions: Conceptualization, D.T., K.A., N.H., and A.C.; methodology, D.T. and N.H.; validation, R.M. and S.K.; formal analysis, D.T., K.A., R.M., and N.H.; writing–original draft preparation, D.T.; writing–review and editing, R.M., S.K., and A.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Laboratoire des technologies de l'énergie d'Hydro-Québec, the Natural Science and Engineering Research Council of Canada, and the Foundation of Université du Québec à Trois-Rivières.

Acknowledgments: The authors would like to thank the Laboratoire des technologies de l'énergie d'Hydro-Québec, the Natural Science and Engineering Research Council of Canada, and the Foundation of Université du Québec à Trois-Rivières.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. U.S. EIA. Annual Energy Outlook 2019 with projections to 2050. Available online: https://www.eia.gov/aeo (accessed on 20 December 2019).
2. GWAC. GridWise Transactive Energy Framework. *GridWise Archit. Counc. Trans. Energy* **2013**, *1*, 1–23.
3. Lian, J.; Zhang, W.; Sun, Y.; Marinovici, L.; Kalsi, K.; Widergren, S. *Transactive System. Part I: Theoretical Underpinnings of Payoff Functions, Control Decisions, Information Privacy, and Solution Concepts*; Technical Report December; Pacific Northwest National Laboratory: Richland, WA, USA, 2017.
4. Yoon, J.H.; Baldick, R.; Novoselac, A. Dynamic demand response controller based on real-time retail price for residential buildings. *IEEE Trans. Smart Grid* **2014**, *5*, 121–129. [CrossRef]
5. Nagi, K. *Transactional Agents: Towards a Robust Multi-Agent System*; Springer Berlin Heidelberg: Karlsrule, Germany, 2001.
6. Odell, J.; Giorgini, P.; Müller, J. *Agent-Oriented Software Engineering V*; Springer: Berlin/Heidelberg, Germany, 2004.
7. Damisa, U.; Nwulu, N.I.; Sun, Y. Microgrid energy and reserve management incorporating prosumer behind-the-meter resources. *IET Renew. Power Gener.* **2018**, *12*, 910–919. [CrossRef]
8. Zhang, Y.; Huang, T.; Bompard, E.F. Big data analytics in smart grids: A review. *Energy Inform.* **2018**, *1*, 1–24. [CrossRef]
9. Khamphanchai, W.; Saha, A.; Rathinavel, K.; Kuzlu, M.; Pipattanasomporn, M.; Rahman, S.; Akyol, B.; Haack, J. Conceptual Architecture of Building Energy Management Open Source Software (BEMOSS). In Proceedings of the 2014 5th IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), Istanbul, Turkey, 12–15 October 2014; pp. 1–6. [CrossRef]
10. Bifet, A.; Gavaldà, R. Adaptive Learning from Evolving Data Streams. In *Advances in Intelligent Data Analysis VIII*; Springer: Berlin, Germany, 2009; pp. 249–260. [CrossRef]
11. Loeffel, P.X. Adaptive Machine Learning Algorithms for Data Streams Subject to Concept Drifts. Ph.D. Thesis, Iniversité Pierre et Marie Curie, Paris, France, 2018.
12. Zaknich, A. *Principles of Adaptive Filters and Self-Learning Systems*; Springer: Murdoch, Australia, 2005.
13. Balasubramanian, V.; Ho, S.S.; Vovk, V. *Conformal Prediction for Reliable Machine Learning*; Elsevier: Waltham, MA, USA, 2016. [CrossRef]
14. Gama, J.; Zliobaité, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A Survey on Concept Drift Adaptation. *ACM Comput. Surv.* **2013**, *46*. [CrossRef]
15. Al-Ali, A.; Zualkernan, I.A.; Rashid, M.; Gupta, R.; Alikarar, M. A smart home energy management system using IoT and big data analytics approach. *IEEE Trans. Consum. Electron.* **2017**, *63*, 426–434. [CrossRef]
16. Keshtkar, A.; Arzanpour, S. An adaptive fuzzy logic system for residential energy management in smart grid environments. *Appl. Energy* **2017**, *186*, 68–81. [CrossRef]
17. Farzan, F.; Jafari, M.A.; Gong, J.; Farzan, F.; Stryker, A. A multi-scale adaptive model of residential energy demand. *Appl. Energy* **2015**, *150*, 258–273. [CrossRef]
18. Yoo, J.; Park, B.; An, K.; Al-Ammar, E.A.; Khan, Y.; Hur, K.; Kim, J.H. Look-ahead energy management of a grid-connected residential PV system with energy storage under time-based rate programs. *Energies* **2012**, *5*, 1116–1134. [CrossRef]
19. Bouchachia, A.; Gabrys, B.; Sahel, Z. Overview of some incremental learning algorithms. In Proceedings of the IEEE International Conference on Fuzzy Systems, London, UK, 23–26 July 2007. [CrossRef]
20. Bifet, A.; Gavaldà, R. Learning from Time-Changing Data with Adaptive Windowing. In Proceedings of the 2007 SIAM International Conference on Data Mining, Minneapolis, MN, USA, 26–28 April 2007; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2007; pp. 443–448. [CrossRef]
21. Zliobaite, I. Adaptive Training Set Formation. Ph.D. Thesis, Vilnius University, Vilnius, Lithuania, 2010.
22. Gama, J.; Medas, P.; Castillo, G.; Rodrigues, P. Learning with Drift Detection. In *Advances in Artificial Intelligence-SBIA 2004*; Springer: Berlin, Germany, 2004; Volume 3171, pp. 286–295. [CrossRef]
23. Murphy, K. *Machine Learning: A Probabilistic Perspective*; The MIT Press: Cambridge, MA, USA, 2012; pp. 27–71. [CrossRef]
24. Moreno-Torres, J.G.; Raeder, T.; Alaiz-Rodríguez, R.; Chawla, N.V.; Herrera, F. A unifying view on dataset shift in classification. *Pattern Recognit.* **2012**, *45*, 521–530. [CrossRef]
25. Webb, G.I.; Hyde, R. Characterizing Concept Drift. *Data Min. Knowl. Discov.* **2016**, 30, 964–994, arXiv:1511.03816v6.

26. Cieslak, D.A.; Hoens, T.R.; Chawla, N.V.; Kegelmeyer, W.P. Hellinger distance decision trees are robust and skew-insensitive. *Data Min. Knowl. Discov.* **2012**, *24*, 136–158. [CrossRef]
27. Widmer, G.; Kubat, M. Learning in the presence of concept drift and hidden contexts. *Mach. Learn.* **1996**, *23*, 69–101. [CrossRef]
28. Zhou, Z.H. *Ensemble Methods: Foundations and Algorithms*; CRC Press: Boca Raton, FL, USA, 2012.
29. Palnitkar, R.; Cannady, J. A Review of Adaptive Neural Networks. In Proceedings of the IEEE SoutheastCon, Greensboro, NC, USA, 26–29 March 2004; pp. 38–47. [CrossRef]
30. Kuncheva, L.I.; Gunn, I.A. A concept-drift perspective on prototype selection and generation. In Proceedings of the International Joint Conference on Neural Networks, Vancouver, BC, Canada, 24–29 July 2016; pp. 16–23. [CrossRef]
31. Bifet, A.; Gavaldà, R.; Holmes, G.; Pfahringer, B. *Machine Learning for Data Streams: With Practical Examples in MOA*; The MIT Press: Cambridge, MA, USA, 2018.
32. Koychev, I.; Lothian, R. Tracking Drifting Concepts by Time Window Optimisation. In *Research and Development in Intelligent Systems XXII*; Springer: London, UK, 2006; pp. 46–59. [CrossRef]
33. Klinkenberg, R.; Joachims, T. Detecting Concept Drift with Support Vector Machines. In Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA, USA, 29 June–2 July 2000; pp. 487–494.
34. Aggarwal, C.C. Towards systematic design of distance functions for data mining applications. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-KDD '03, Washington, DC, USA, 24–27 August 2003; ACM Press: New York, NY, USA, 2003; p. 9. [CrossRef]
35. Aggarwal, C.C.; Hinneburg, A.; Keim, D.A. On the Surprising Behavior of Distance Metrics in High Dimensional Space. *Am. J. Dermatopathol.* **2001**, *30*, 420–434. [CrossRef]
36. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [CrossRef]
37. Touzani, S.; Granderson, J.; Fernandes, S. Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy Build.* **2018**, *158*, 1533–1543. [CrossRef]
38. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Müller, A.; Nothman, J.; Louppe, G.; et al. Scikit-learn: Machine Learning in Python. *Psychol. Sci.* **2012**, *25*, 1682–1690. [CrossRef]
39. Holmgren, W.; Hansen, C.; Mikofski, M. pvlib python: A python package for modeling solar energy systems. *J. Open Source Softw.* **2018**, *3*, 884. [CrossRef]
40. Alpaydin, E. *Introduction to Machine Learning Second Edition*, 3rd ed.; The MIT Press: Cambridge, MA, USA, 2010; pp. 350–380. [CrossRef]
41. Jokisalo, J.; Kurnitski, J. Performance of EN ISO 13790 utilisation factor heat demand calculation method in a cold climate. *Energy Build.* **2007**, *39*, 236–247. [CrossRef]
42. Gelman, A.; Carlin, J.; Stern, H.; Rubin, D. *Bayesian Data Analysis*; Chapman & Hall: London, UK, 2004.
43. Chong, E.; Zak, S. *An Introduction to Optimization*, 4th ed.; Wiley and Sons: Hoboken, NJ, USA, 2013.