# Dynamic Prediction of Powerline Frequency for Wide Area Monitoring and Control

Sharda Tripathi and Swades De

***Abstract*—This paper presents a novel data driven framework based on $\epsilon$-Support Vector Regression to reduce the bandwidth requirement for transmission of Phasor Measurement Unit (PMU) data. This is achieved by judicious elimination of redundant data at the PMU before transmission. Simultaneously, the missing samples are predicted at PDC to ensure faithful identification of impending disturbances in the power system. Due to inherent non-stationary nature of PMU data, the hyper-parameters are dynamically recomputed as necessary, thereby maintaining the accuracy of prediction and robustness of the algorithm. Performance of the proposed algorithm is evaluated via large scale simulations using powerline frequency data. A trade-off between prediction quality and runtime of the algorithm is observed, which is addressed by suitable selection of hyper-parameters. Compared to the competitive data reduction scheme, the proposed algorithm saves around 60% bandwidth and identifies power system disturbances 73% more accurately.**

***Index Terms*—Wide area measurement system, phasor measurement unit, dynamic prediction, $\epsilon$-support vector regression, bandwidth saving**

## I. Introduction

Imparting intelligence, automation, and control to the traditional power grid has enhanced its load handling capabilities. Yet, ensuring an uninterrupted supply of electricity to the end users is a far sight. Stress on power grid causes frequent occurrences of failure events, raising questions on safe and reliable grid operation. To this end, Phasor Measurement Units (PMUs) periodically sample the state of power system and send their data to a remotely located Phasor Data Concentrator (PDC) over a communication network [1]. Fig. 1 shows an example of PMU placement across the power grid in a Wide Area Measurement System (WAMS). It is evident that data from PMU plays a crucial role at the PDC in accurately determining the sequence of events happening in the power system [2]. Thus, reliable and rapid coordination between the PMUs and the PDC is essential to capture the power grid dynamics and provide real-time situational awareness.

### A. Related Works and Motivation

*Transient detection*: Early works emphasized on the use of PMU data for transient prediction via time-domain techniques. But due to time-consuming nature and requirement of accurate network configuration information, they were overridden by robust and computationally-efficient machine learning techniques, such as, multilayer perceptrons [3], decision trees [4], and support vector machines [5], [6]. A recent study in [7] has compared the performance of data-driven versus time-domain models for real-time identification of power system dynamics. Besides transient detection, other key application of data driven models is in monitoring the operation of distributed generation units, especially for integration of renewable energy sources. For instance, in [8] data mining algorithms are studied to identify blade pitch faults in wind turbines. Based on fault diagnosis using PMU data, devising effective control strategies is essential for system-wide protection. In literature, methods for load generation and switching control, oscillation damping, emergency frequency control, and adaptive protection schemes are proposed to address this issue. Since designing of control applications is not in the scope of current work, interested readers are referred to [9]–[11] for further details.

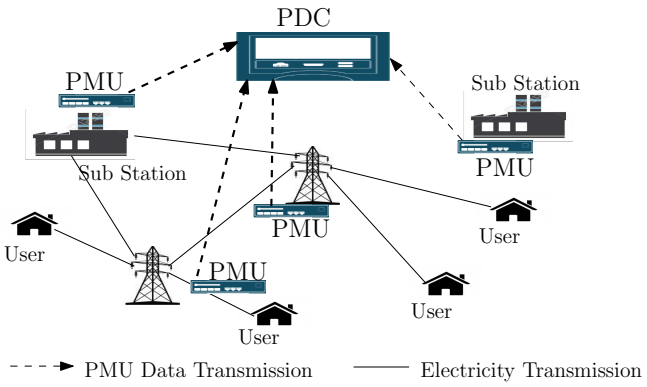



Fig. 1: Basic elements of a WAMS.

It should be noted that, transient detection techniques generally classify the system state as 'stable' or 'unstable' based on fixed-rate PMU data to the PDC, assuming unlimited communication resources. Data reporting rates supported by PMUs are multiples and sub-multiples of nominal system frequency [12]. Presently, standard data reporting rate from PMU to PDC is fixed at 25 and 30 samples/sec, respectively for 50 Hz and 60 Hz systems. While frequent sampling and near-real time data reporting facilitate in timely control actions for preserving power grid stability, huge volume of generated data poses a serious challenge in terms of communication bandwidth and storage needs [13], [14], which further worsens with the increasing number of PMUs installed in a power grid. For instance, a study in [15] has noted that data transmitted from 100 PMUs at 30 samples/sec to a PDC is over 50 GB data per day. With the ever-growing electricity demand, this enormous amount of data is likely to exceed the network transmission capacity in near future.

This work was supported in part by the Department of Science and Technology, Government of India, under Grant DST/RCUK/SEGES/2012/02.

S. Tripathi and S. De are with the Department of Electrical Engineering and Bharti School of Telecommunication, Indian Institute of Technology Delhi, New Delhi, India (e-mail: {sharda.tripathi, swadesd}@ee.iitd.ac.in).

*PMU data reduction techniques:* A few studies have explored stability prediction based on reduced PMU data. In [16], autoregressive modelling of PMU data sequence is studied to identify stress signs from correlation between consecutive samples. Short-term prediction using state-space approach and basis function was studied in [17] to spot measurement errors. Dimensionality reduction of PMU data using linear principal component analysis were studied in [18], [19]. The study in [19] also performed real-time compression using least square curve fitting while archiving the data. Different signal processing algorithms have also addressed the data reduction in WAMS. Discrete cosine transform [20], [21], compressive sampling [22], wavelet packet decomposition [23], [24], preprocessing and lossless encoding [25] (and the references therein) operate offline and also in real-time for data storage. Recently, a fuzzy-based paradigm for efficient processing and compression of smart grid data has been proposed [26]. When the data has low correlation, these methods do not ensure accurate signal reconstruction. Consequently, high-rate sampling is required for a desired accuracy. Besides, complex matrix operations involved in transform-based approaches significantly increase the computational load.

Since transient occurrences in the system are sporadic and PMU data is highly redundant, *fixed-rate* data transmission at all times appears wasteful. As the communication channel bandwidth is limited and expensive, it is critical that this resource is optimally used for power grid communication. Therefore, dynamically preventing redundant PMU data from being transmitted over the channel without compromising on the quality of power grid health monitoring is of current interest. It is notable that, other than [22], all prior approaches investigated data reduction at the PDC. Further, although the objective in [22] has been communication bandwidth reduction, it does not deal with non-stationary nature of PMU data. Thus, in absence of continuous learning and adaptation, quality of compression and hence the quality of power system health monitoring is expected to degrade over time.

### B. Main Contributions

In this work, a novel data-driven framework based on $\epsilon$-Support Vector Regression ($\epsilon$-SVR) is proposed to dynamically reduce the powerline frequency samples at the PMU before transmission and predict the missing samples at the PDC. Performance of the approach is measured in terms of bandwidth saving, retraining count, disturbance identification index, prediction ratio, and root mean square error.

The main contributions of this work are as follows:

1) The devised dynamic prediction algorithm selectively transmits powerline frequency samples achieving up to $90\%$ reduction in channel bandwidth requirement without affecting the quality of stability monitoring of the system.
2) Optimization of the operating parameters, namely, training length and retraining frequency, and performance degradation due to precomputation of parameters are empirically investigated for reduced runtime complexity.
3) Trade-off between accuracy of prediction and runtime of the proposed algorithm is addressed.
4) Computational latency of the proposed algorithm is estimated via its online execution using Simulink model.
5) Comparison of the proposed algorithm with the compressed sampling scheme [22] demonstrates 73% and 60% better performance, respectively, in terms of power system health monitoring and bandwidth saving.

Unlike the other signal processing and data compression algorithms, the proposed approach is independent of sparsity of data-set. In this work, the limitations of existing approaches for PMU data reduction are addressed by achieving data pruning at the transmission stage itself. By continuous learning the proposed algorithm is able to identify the system transients and adapt the pruning process by re-estimation of hyper-parameters as needed, thereby increasing accuracy and robustness of prediction and reducing the communication bandwidth requirement. *To the best of the authors' knowledge, dynamically exploiting temporal correlation of PMU data to reduce volume before transmission without trading the quality of power system health monitoring has not been studied yet.*

### C. Paper Organization

Layout of the paper is as follows: Section II briefly describes the application of $\epsilon$-SVR in non-stationary powerline frequency time series prediction. Section III presents the proposed framework for data reduction and dynamic prediction, followed by subsequent discussions on the choice of hyper-parameters and complexity of the proposed algorithm. The performance indices are mentioned in Section IV, and numerical results based on large-scale simulations are discussed in Section V. Finally, the paper is concluded in Section VI.

## II. Use of $\epsilon$-SVR in Prediction of Powerline Frequency Time Series

$\epsilon$-SVR has evolved based on statistical learning theory. Due to its ability to generalize well on unseen data-sets and produce consistent unbiased estimate of the target, in recent years it has been widely applied in varied research areas. Composition monitoring in manufacturing processes [27], traffic signal detection [28], wind speed forecasting [29], and prediction of pollutant emissions [30] are some of the applications.

The statistical pattern of powerline frequency data collected by PMU is analyzed by studying its autocorrelation function. The autocorrelation coefficients are found to have high magnitude and slowly-decaying nature, indicating that the time sequence formed by these samples is stochastically dynamic and non-stationary. $\epsilon$-SVR formulation handles non-stationarity by mapping the data to a high-dimensional feature space produced by a kernel and then linear regression is performed using optimal parameter values. Let $\{f_1, f_2, f_3, \cdots, f_l\}$ be the time sequence of powerline frequency. Due to non-stationarity, each predicted frequency value can be assumed to be a non-linear function of lag values, where '*lag*' is the optimum number of previous samples required to predict the present sample.

From the definition of SVR analysis [31], the predicted value $\hat{f}_i$ corresponding to actual frequency sample $f_i$ is:

$$\hat{f}_i = \sum_{j=i-1}^{i-d} w_j \phi(f_j) + b \quad \forall\ i = (1, \cdots, l) \tag{1}$$

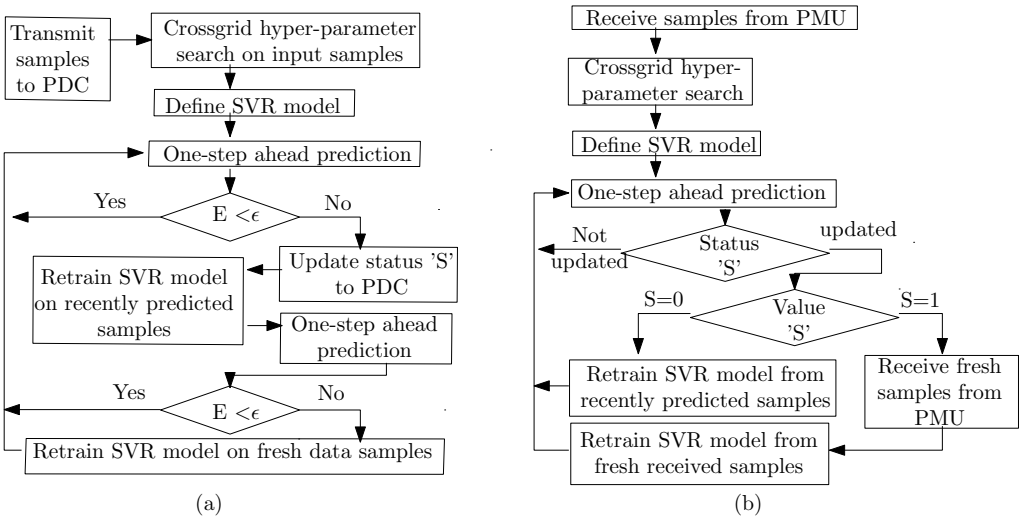



Fig. 2: Flow graph of the proposed dynamic prediction algorithm: (a) at the PMU; (b) at the PDC.

where $d$ is the lag samples, $w_j$ is the model's parameter, $\phi(f_j)$ is the set of basis functions forming a non-linear mapping from input space to a higher-dimensional feature space, and $b$ is the offset value. Essentially, $w_j$ is the weight of participation of $j$th lag value for estimating the present sample. It follows from (1) that sample $f_i$ has an attribute vector $f_{A_i}$ comprising of lag values corresponding to that sample. The training data is organized as $\{(f_{A_1}, f_1), (f_{A_2}, f_2), \cdots, (f_{A_l}, f_l)\} \subset \mathbb{R}^{\mathrm{d}} \times \mathbb{R}$. The input space is $d$-dimensional such that $f_{A_i} = \{f_{i-1}, f_{i-2}, \cdots, f_{i-d}\}$. With this notation, (1) is rewritten as:

$$\hat{f}_i = \langle w_{A_i}, \phi(f_{A_i}) \rangle + b \quad \forall i = (1, \cdots, l) \qquad (2)$$
$$\text{such that} \quad w_{A_i} \in \mathbb{R}^{\mathrm{d}}, b \in \mathbb{R}.$$

In (2), $w_{A_i}$ and $\phi(f_{A_i})$ are respectively the array of weights and non-linear mappings of the vector $f_{A_i}$ corresponding to each sample $\hat{f}_i$ such that $w_{A_i} = \{w_{i-1}, w_{i-2}, \cdots, w_{i-d}\}$ and $\phi(f_{A_i}) = \{\phi(f_{i-1}), \phi(f_{i-2}), \cdots, \phi(f_{i-d})\}$. It is required to find optimal weights $w_{A_i}$ and offset $b$ such that the predicted value $\hat{f}_i$ has at most $\epsilon$ deviation from the actual value $f_i$. These values are obtained as by-product of solution of the optimization problem in (3).

$$\textbf{P1: minimize} \left\{ \frac{1}{2} \parallel w_{A_i} \parallel^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \right\} \qquad (3)$$
$$\text{subject to} \quad f_i - \hat{f}_i \leq \epsilon + \xi_i; \hat{f}_i - f_i \leq \epsilon + \xi_i^* \text{ and } \xi_i, \xi_i^* \geq 0.$$

In (3), $\xi_i$s are real-valued slack variables to ensure feasibility of the optimization problem, and parameter $C$ decides the trade-off between flatness of $w_{A_i}$ and $\xi_i$. The Lagrangian $L$ is obtained by combining the primal objective function with constraints through the multipliers $\alpha_i$, $\alpha_i^*$, $\eta_i$, and $\eta_i^*$. Subsequently, for optimality, partial derivatives of $L$ with respect to primal variables $w$, $b$, $\xi_i$, $\xi_i^*$ vanish from the Karush-Kuhn-Tucker conditions to give dual optimization problem as:

$$\textbf{P2: maximize} \left\{ -\frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \phi(f_{A_i}), \phi(f_{A_j}) \rangle \right.$$
$$\left. -\epsilon \sum_{i=1}^{l} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} f_i (\alpha_i - \alpha_i^*) \right\} \qquad (4)$$
$$\text{subject to} \quad \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \quad \text{and} \quad \alpha_i, \alpha_i^* \in [0, C].$$

In (4), inner product in feature space is replaced by an equivalent kernel in input space, i.e., $\langle \phi(f_{A_i}), \phi(f_{A_j}) \rangle = K(f_{A_i}, f_{A_j})$. In this work, radial basis kernel function is used due to its generalization properties and is given by $K(f_{A_i}, f_{A_j}) = \exp\left(-\gamma \parallel f_{A_i} - f_{A_j} \parallel\right)^2, \forall\ i, j = (1, \cdots, l)$. Applying saddle point condition to (4), a sparse representation of $\hat{f}_i$ as a function of support vectors is obtained as:

$$\hat{f}_i = \sum_{\text{SVs}} (\alpha_i - \alpha_i^*) K(f_{A_i}, f_{A_j}) + b. \qquad (5)$$

Support vectors are the data samples characterized by condition $\alpha_i$, $\alpha_i^* < C$ and $\alpha_i \cdot \alpha_i^* = 0$.

## III. Dynamic Prediction Algorithm

In this section, the proposed dynamic prediction algorithm is presented along with the discussion on hyper-parameters for predicting frequency samples $\{\hat{f}_i\}$ and algorithm complexity.

### A. Proposed Dynamic Prediction Algorithm

The dynamic prediction algorithm operates at the PMU to remove the redundant power line frequency samples before transmission, while its counterpart simultaneously operates to estimate the removed samples at the PDC. The key steps involved are: 1) computation of hyper-parameters, 2) making successive one-step ahead prediction of frequency samples using hyper-parameters from step 1. Due to non-stationary nature of data, the hyper-parameters once computed based on prior samples fail to produce consistent output after a while, especially when a disturbance sets in the power system. Thus, it is required to re-compute them whenever the difference between actual sample and the predicted sample exceeds a predefined threshold $\epsilon$. This process is called *retraining*. Since the

TABLE I: Frequency operation limits at 60 Hz [32].

| Range (Hz) | Action |
|---|---|
| 59.9-60.1 | Normal operation, no control action |
| 59.8-59.9, 60.1-60.2 | Primary frequency control active for small dead zone units |
| 59-59.8, 60.2-61.3 | Primary frequency control active for large dead zone units |
| 57.5-59, 61.3-62 | Underfrequency load shedding and secondary frequency control |
| $< 57, > 62$ | Self-protective generator trip or system collapse |

proposed algorithm adapts to changing dynamics in the power grid, it is referred to as *dynamic prediction* algorithm.

Working of the proposed dynamic prediction algorithm is illustrated by a flow graph in Fig. 2. *Optimum Training Length* (OTL) is the length of input samples that is just sufficient for training the SVR model such that statistical reliability is ensured with minimum computations. Lag and OTL are precomputed from offline studies on frequency data collected from PMU before execution of the algorithm in order to reduce the computational complexity during runtime. Of the remaining hyper-parameters, $\epsilon$ is considered predefined and $C$ and $\gamma$ are calculated on-the-fly using cross-validation during the training stage. More details on choice of these hyper-parameters is discussed in the next subsection.

As the PMU has access to the actual frequency samples as well, the predicted value is compared to corresponding actual value following each one-step ahead prediction and the difference between them is termed as error $E$. As long as $E < \epsilon$, one-step ahead prediction process is iterated, else retraining is performed. Note that, in the interest of saving bandwidth and utilizing good predictions already made, this retraining is performed on latest OTL number of *predicted frequency samples*. Following retraining, if the immediate prediction does not satisfy the condition $E < \epsilon$, then process of further prediction is aborted and the next retraining is performed using OTL number of *actual frequency samples*. It is apparent that irrespective of varying transients in the power system, accuracy of the predicted value is not jeopardized due to continuous learning of model parameters, which consequently increases robustness of the proposed algorithm.

At PDC, the algorithm relies on limited data and *status notification* $S$ transmitted from PMU to manage its retraining processes. Intuitively, SVR models trained independently at PMU and PDC with same input values produce similar predictions. Thus, the intuition behind simultaneous execution of the proposed algorithm at PMU and PDC is that, predictions at the PDC help in grid monitoring with significantly reduced bandwidth consumption in transmitting the PMU data while predictions at the PMU identify the data transmission and status notification instants for the PDC.

### B. Choice of Hyper-Parameters

Performance of the proposed algorithm is sensitive to the values of hyper-parameters. This section briefly addresses the selection of lag, OTL, $C$, and $\gamma$ for an optimum performance as a function of the system defined error threshold $\epsilon$.

$\epsilon$ denotes the upper bound of acceptable error. Its value is application-domain specific and depends on the level of accuracy desired in prediction. Important thresholds in power system frequency control are stated in Table I. To estimate optimum lag, $C$, and $\gamma$, $k$-fold cross-validation is performed on training data. To obtain $C$ and $\gamma$, a search space is defined, and $\{C, \gamma\}$ pair is determined that offers minimum cross-validation error. A single course of cross-validation partitions the data-set into $k$-complementary subsets such that the model trained on $k-1$ subsets is validated on the remaining subset. Multiple courses of cross-validation are performed with dissimilar partitions in order to reduce the variability, and all the results are averaged to obtain *cross-validation error*. Despite being exhaustive and computationally intensive, cross-validation and grid search are used for parameter selection in this work because they do not rely on domain knowledge of the user and produce robust results.

Size of the training set plays a crucial role in deciding reliability of any machine learning model. Here, due to non-stationary nature of input data, the hyper-parameters are susceptible to loose their accuracy with time in spite of being trained on a larger data-set. Intuitively, in such a scenario, being parsimonious in deciding OTL may be a practical approach. Thus it is required to bound the training length such that prediction model performs satisfactorily in all scenarios in real-time. Finding true OTL at each retraining instant is tedious and slows down the algorithm. Therefore, it is proposed to use fix value of OTL during the entire training procedure. Numerical results on hyper-parameter selection and impact of model trained on precomputed OTL as against true OTL are further detailed in the Section V. Commercially available machine learning library LibSVM [33] has been used in this work to perform training and prediction of samples using large-scale Matlab simulations.

### C. Complexity of the Proposed Algorithm

To analyze the computation complexity it may be recalled that training and prediction are two essential steps in execution of the proposed algorithm. During training stage, hyper-parameters are determined via $k$-fold cross-validation on input data. Study of step-wise execution of the proposed algorithm reveals that the complexity during training varies as $k.xy.\#itr\mathcal{O}(l.d)$. Here, $x$ number of $C$ values and $y$ number of $\gamma$ values comprise the search space. $\#itr$ is the number of iterations used in the convergence of optimization problem and is specific to the solver used. $l$ and $d$ are respectively, the length of training sequence and lag value. Prediction complexity is given by $\mathcal{O}(l'.d)$, where $l'$ is the number of predictions made at a time. Since the proposed algorithm makes successive 1-step ahead predictions from the moving lag window, $l' = 1$ and $d$ is constant. Therefore prediction complexity is essentially constant. Thus, runtime complexity is basically due to training, and increases linearly with the training length $l$.

## IV. Performance Indices

For comprehensive quality assessment of the proposed prediction model the following performance indices are defined:

1) *Bandwidth Saving (BWS)*: It is the percentage of PMU data samples that are not transmitted. These are essentially the samples which are successfully predicted within the error bound $\epsilon$ at the PDC. If $l$ is the length of powerline frequency sequence measured by PMU over a sufficiently large time interval $\Delta$, then, $BWS = \lim_{l\to\infty}(\text{Successful predictions by PMU}/l) \times 100$.

2) *Retraining Count (RC)*: It is the number of retrainings required to make successful predictions over a large time interval $\Delta$. For a given error threshold $\epsilon$, time complexity of the proposed algorithm increases with $RC$. Thus, the hyper-parameters should be chosen such that minimum possible number of retrainings are performed.

3) *Prediction Ratio (PR)*: It quantifies the quality of training. Denoting the number of useful predictions between two consecutive retraining instants as *prediction length*, $PR$ = prediction length/OTL. A high $PR$ reduces $RC$, hence increasing $BWS$.

4) *Disturbance identification Index (DI)*: It is a measure of goodness of the model in identifying a fault scenario. Over a large interval $\Delta$, let $l_{dist}$ and $\hat{l}_{dist}$ be respectively the actual and the estimated number of frequency samples designated to be in disturbed states. Then, $DI = \lim_{\Delta\to\infty}(\hat{l}_{dist}/l_{dist})$. Identifying powerline disturbance can be based on either of the following two criteria: (i) frequency $f_i < 59.55$ Hz, or $> 61$ Hz, (ii) rate of change of frequency $df_i/dt > 0.124$ Hz/sec [34]. This conventional approach tracks only major disturbances. To capture smaller frequency variations, a disturbance can be associated with frequency samples deviating more than $\pm 0.1\%$ (i.e. 0.06 Hz) from the nominal value [23]. However, it is of interest to slightly overestimate the disturbance region (i.e., $\hat{l}_{dist} \geq l_{dist}$) at the PDC as a precautionary measure. Thus, the value 0.06 Hz is further lowered by 1%, to 0.0594 Hz to identify a disturbed instance from the estimated value.

5) *Root Mean Square Error (RMSE)*: It is a standard error metric for predicted values. An acceptable RMSE is always less than $\epsilon$. Mathematically, $RMSE = \sqrt{(1/l)\sum_{i=1}^{l}(f_i - \hat{f}_i)^2}$.

## V. Results and Discussions

In this section, first optimum hyper-parameters of the proposed dynamic prediction model are numerically determined. Next, different performance versus complexity trade-offs are discussed, followed by a comparative performance analysis with respect to a recent competitive approach in [22]. Subsequently, implementation issues are briefly addressed.

### A. Determining Optimum Hyper-Parameter

As discussed in Section III-B, $k$-fold cross-validation error of the training set is used to decide the optimum values of hyper-parameters, namely, lag, $C$, $\gamma$, and OTL. Fig. 3 captures the variation of cross-validation error versus lag value. For a generic conclusion, cross-validation errors are computed on 25 different data-sets from PMU data repository at http://103.7.128.82/rwafms/wafms/, each consisting of several steady states and disturbed states. The plots reveal that the mean error is minimum at lag $= 5$. Hence, this parameter value is chosen for further performance analysis in this work.

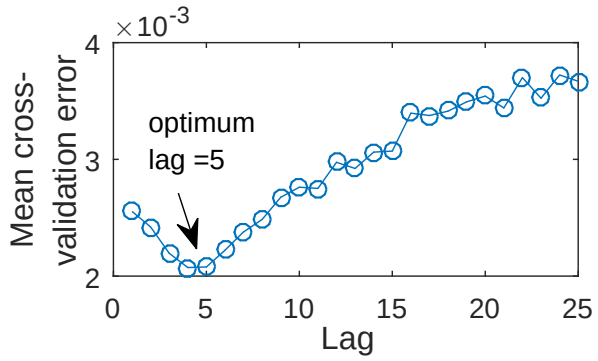



Fig. 3: Cross-validation error versus lag value.

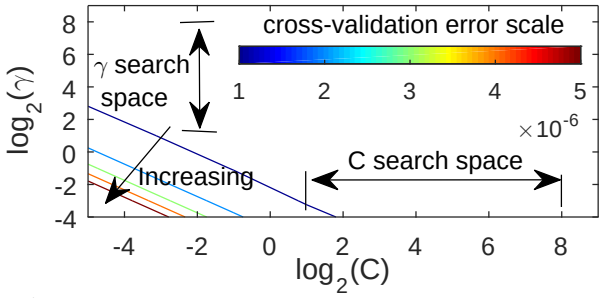
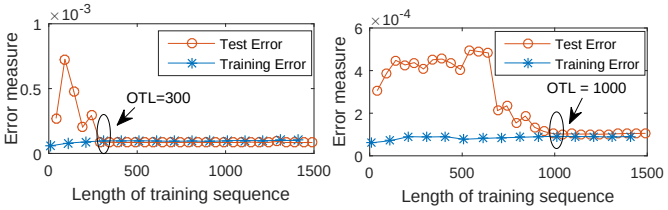



Fig. 4: Cross-validation error versus variation of $C$ and $\gamma$.

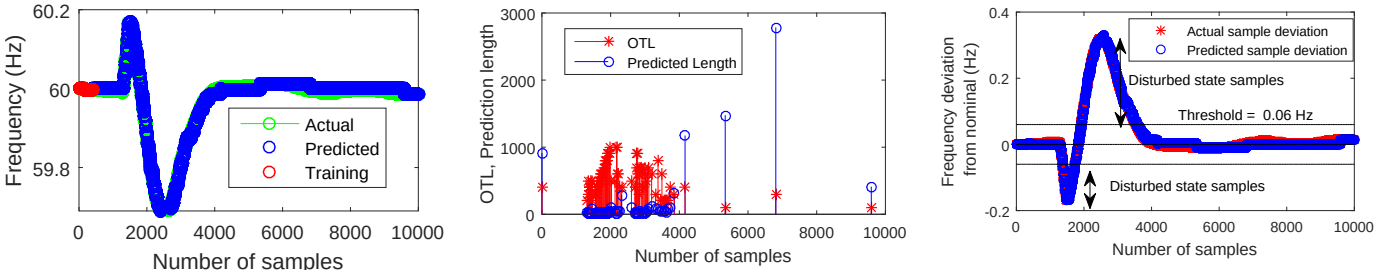



(a) (b)

Fig. 5: OTL at different states: (a) steady state; (b) disturbed state.

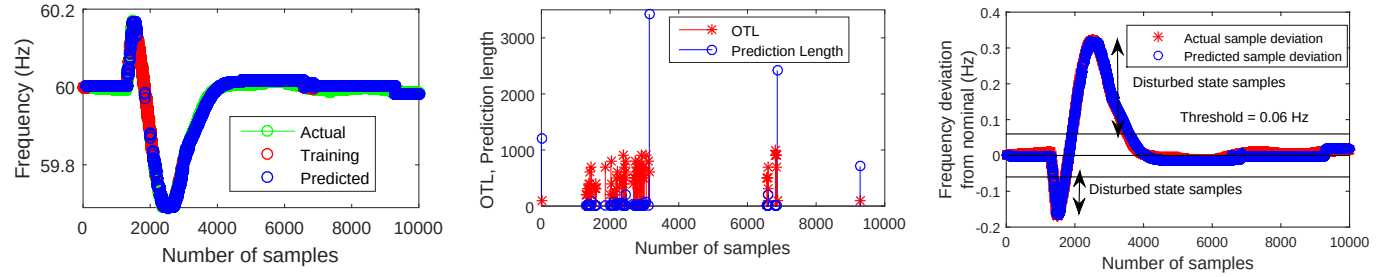
The contour plot in Fig. 4 captures the variation of cross-validation error with respect to $C$, $\gamma$ pairs. From the plot it is evident that lower errors are obtained beyond values $\log_2 C = 2$ and $\log_2 \gamma = 3$. It may be noted that these are not hard boundaries and indicate only average hyper-parameter values. Also, to curb the risk of overfitting, very large values of $C$ and $\gamma$ are avoided in spite of low cross validation error [35]. Thus, grid boundary for $C$ as well as $\gamma$ is fixed at the intermediate values varying from $2^1$ to $2^8$ in this work. This limits the size of search space and increases speed of simulation at runtime.

Figs. 5a and 5b capture the variation of training error and test error versus training length for data-sets in steady state and disturbed state of the power system. It is observed from the plots that increasing the size of training set reduces the test error due to better fit of hypothesis. However, this benefit significantly diminishes beyond a certain training length. This




(a) (b) (c)

Fig. 6: Dynamic model trained on actual frequency samples and true OTL for Case I: (a) prediction performance; (b) variation of OTL and prediction length versus number of samples; (c) disturbance identification.




(a) (b) (c)

Fig. 7: Dynamic model trained on predicted frequency samples and true OTL for Case II: (a) prediction performance; (b) variation of OTL and prediction length versus number of samples; (c) disturbance identification.

TABLE II: Performance indices for Cases I and II.

| Performance Index | Case I | Case II |
|---|---|---|
| BWS (%) | 91.59 | 95 |
| RC | 76 | 85 |
| DI | 1.02 | 1.03 |
| Best PR in steady region | 18 | 20.13 |
| Worst PR in steady region | 1.04 | 0.002 |
| Best PR in disturbed region | 1.39 | 0.265 |
| Worst PR in disturbed region | 0.0113 | 0.001 |
| RMSE | 0.0062 | 0.0069 |

point which denotes a saturation in learning from input data is the OTL of the data-set. For comprehensive study, OTL is evaluated for 25 different training sets from the PMU data repository. It is noted that OTL in steady states is slightly lower, varying from 100 to 400 samples, while for disturbed states it lies in the range of 600-1000. Additionally, maximum OTL amongst all data-sets did not exceed 1000 samples with probability 0.9, and mean OTL is found as 600 samples.

**Remark 1.** Determining the optimal hyper-parameters is vital for every training process to address non-stationarity of data and increase the speed and accuracy of predictions.

### *B. Performance of Dynamic Prediction Trained on Actual Frequency Samples versus Predicted Frequency Samples*

As discussed in Section III-A, although from communication bandwidth saving perspective retraining based on recently predicted frequency samples helps, the quality of prediction is expected to be better when retraining is performed on actual frequency samples. Below, relative performance of the two approaches is investigated. Two cases are considered:

Case I) predictions are made from model trained on actual frequency samples from PMU;

Case II) predictions are made from model trained with latest predicted frequency samples.

Performance in each case is illustrated in Fig. 6 and Fig. 7, where true OTL is found at each training instance. Performance indices are summarized in Table II. It can be observed that bandwidth saving in Case II exceeds that in Case I by approximately 4%. Further, from increased $RC$ in Case II, it can be inferred that if the model is trained using predicted samples, quality of fit degrades, leading to more retraining instances. Despite this, there is no compromise in identification of disturbed states, causing $DI$ to be comparable in both cases I and II. However, Case I offers a better $PR$ performance in the disturbed states, and it has marginally improved RMSE.

**Remark 2.** Case II is good for bandwidth saving as compared to Case I without undermining the disturbance detection process, but at the cost of increased computational complexity.

### *C. Performance of Dynamic Prediction Trained on Precomputed OTL versus True OTL*

As discussed in Section III-B, finding true OTL at each retraining instant is computationally expensive and hazardous, especially in disturbed states when frequent retrainings are required. To improve upon this, use of precomputed OTL is proposed. In this section, relative performance of two precomputed OTL scenarios, namely, maximum OTL and mean OTL are compared with true OTL for Cases I and II.

It is observed from Fig. 8 that use of precomputed OTL causes $RC$ to increase and $PR$s to decrease indicating deterioration in the quality of model fit. Besides, RMSE also

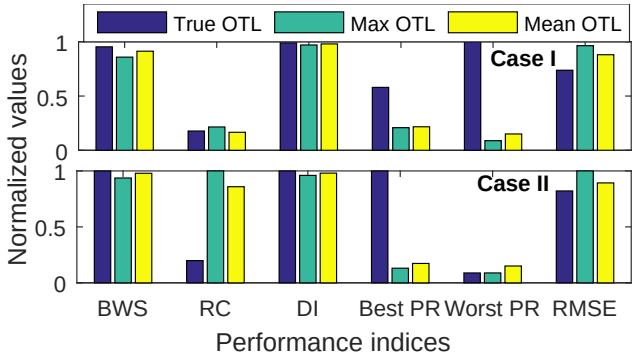



Fig. 8: Performance comparison of dynamic prediction with true, max, and mean OTL for Cases I and II.

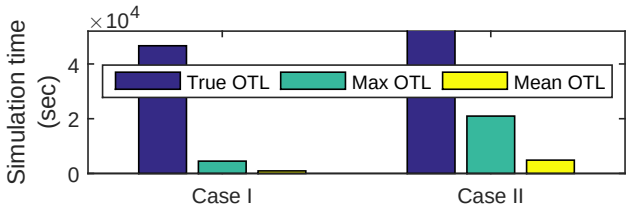



Fig. 9: Simulation time comparison of dynamic prediction with true, max, and mean OTL for Cases I and II.

increases by approximately 20%. This is primarily because true OTL at any retraining instant is rarely same as mean OTL and most likely is exceeded by maximum OTL. As a consequence, more often retraining is performed either on surplus or insufficient data samples leading to poor quality fit. Additionally, subsequent retraining instances on more than required number of samples consumes a larger data chunk thereby reducing the $BWS$. However, faithful detection of all disturbance instances, which is also critical while reducing the data transmission between PMU and PDC, is assured irrespective of the use of maximum or mean OTL. Simulation times of proposed algorithm for 10000 powerline frequency samples with true, maximum and mean OTL are compared in Fig. 9. It can be seen that in spite of higher $RC$, the runtime for model trained on precomputed OTL is considerably small with respect to model trained on true OTL. From the above discussion, it is clear that compared to true OTL, use of precomputed OTL wins in terms of computational burden reduction and consistent fault identification but slightly compromises on the bandwidth savings and fit of the model.

Further, as compared to maximum OTL, model trained with mean OTL seems to be a reasonably better choice. This can be inferred from relative improvement in performance indices. Numerically, for Case I, $BWS$, best $PR$, and worst $PR$ increase by 5.9%, 3.8%, and 41%, while $RC$ and RMSE reduce by 22% and 8.6%, respectively, with mean OTL. Similarly for Case II, $BWS$, best $PR$, and worst $PR$ improve by 4.3%, 24.5%, and 41%, and $RC$ and RMSE lower by 14% and 10%, respectively.

**Remark 3.** Due to non-stationarity of PMU data, large training lengths do not ensure increased accuracy of prediction.

TABLE III: Variation of performance indices with $\epsilon$.

| $\epsilon$ | BWS (%) | RC | DI | RMSE |
|---|---|---|---|---|
| 0.1 Hz | 91 | 68 | 4 | 0.1268 |
| 0.01 Hz | 87.70 | 369 | 1.02 | 7.5 x $10^{-3}$ |
| 0.001 Hz | 81 | 951 | 1.001 | 9.03 x $10^{-4}$ |
| 0.0001 Hz | 33.90 | 2889 | 1 | 9.028 x $10^{-5}$ |

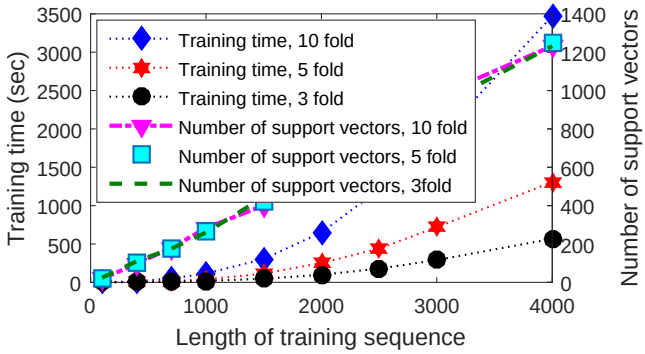



Fig. 10: Training time and number of support vector variation with training length at different folds of cross-validation $k$.

### D. Performance Variation with $\epsilon$ Values

$\epsilon$ is critical to ensure the accuracy in disturbance identification. In this section, suitable choice of $\epsilon$ is investigated.

From Table I it can be inferred that, to preserve operation limits in the predicted frequency samples, $\epsilon$ = 0.01 Hz is sufficient. It accurately characterizes a disturbed state frequency sample, as with this value, first digit after decimal of the predicted frequency matches with the actual one and only a variation of $\pm 1$ in the second digit after decimal is tolerated. That is, for instance, worst prediction for a frequency value 60.14 Hz is 60.15 Hz or 60.13 Hz. Table III compares the performance indices for different $\epsilon$. It is observed that, a lower $\epsilon$ enables more accurate prediction. But, to reach this level of accuracy, the algorithm consumes more samples for training the model, causing reduced $BWS$ and increased $RC$.

### E. Runtime versus Training Length

As discussed in Section III-C, complexity of the proposed algorithm increases linearly with training length. This can be observed from Fig. 10 which shows the variation of training time (i.e., runtime complexity) with respect to training length for different folds of cross-validation. It also captures the variation of number of support vectors required at different training length. It is evident that irrespective of number of folds, support vectors increase almost linearly with the size of training set. Also, from (5), number of support vectors also add to the number of computations required for prediction, thereby increasing the runtime complexity.

Choice of number of folds is also a design parameter for the proposed dynamic prediction algorithm. From the theory of bias-variance trade-off [31], it is well known that for optimal model complexity, (bias)$^2$ and variance of the fit should be minimum. Table IV presents the variation of bias and variance with increasing number of folds for the proposed model over

TABLE IV: Variation of bias and variance of model fit with increasing folds.

| Folds | Bias | Variance |
|---|---|---|
| 3 | 9.23 x $10^{-4}$ | 0.0081 |
| 5 | 7.82 x $10^{-4}$ | 0.0085 |
| 10 | 4.169 x $10^{-4}$ | 0.0087 |
| 25 | 3.17 x $10^{-5}$ | 0.0086 |

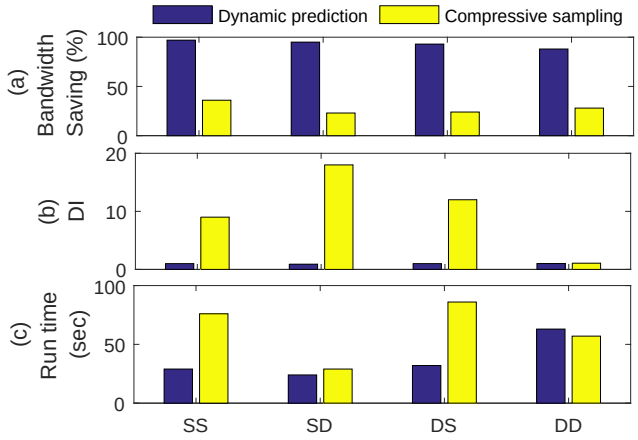



Fig. 11: State-wise comparison of dynamic prediction and compressive sampling at RMSE limit $10^{-3}$: (a) bandwidth saving; (b) disturbance identification index (DI); (c) runtime. SS: entire data-set in steady state; SD: data-set begins in steady state, ends in disturbed state; DS: data-set begins in disturbed state and ends in steady state; DD: entire data-set in disturbed state.

input data. Although complexity increases by a factor of $k$, the change of bias and variance with increasing $k$ is marginal and much less than $\epsilon$. Hence, $k = 3$ is sufficient to obtain desirable accuracy during prediction.

**Remark 4.** To limit the runtime complexity, choice of optimal training length is crucial.

### *F. Comparative Performance Analysis*

Finally, performance of the proposed algorithm is compared with compressive sampling [22] which also aims at network bandwidth reduction by transmitting synchrophasor data at sub-Nyquist rate. Performance of compressive sampling algorithm is sensitive to the choice of window length $N$, sketch length $m$, and sparsity of input data. Here, $N/m$ is a measure of bandwidth saving. Under best parameter settings, performance of compressive sampling is compared with dynamic prediction in different power system states. Performance indices obtained are averaged over tested data-sets for each state. From Fig. 11 it is observed that, for an acceptable upper bound of RMSE on the order of $10^{-3}$, compared to compressive sampling, dynamic prediction consistently saves more bandwidth and at the same time maintains a uniform DI close to 1 across various states. Note that, RMSE of the order $10^{-3}$ corresponds to $\epsilon = 0.01$ for dynamic prediction. Due to low $RC$ in such a scenario, the runtime of dynamic prediction algorithm is lower for SS, SD and DS while it is comparable at DD. From skewed $DI$ for compressive sampling, it is evident that $10^{-3}$ is not sufficient to reconstruct the data with desirable

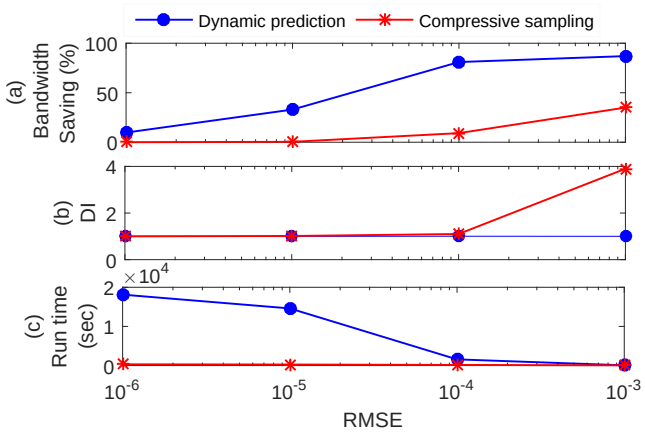



Fig. 12: Performance comparison of dynamic prediction and compressive sampling at different RMSE: (a) bandwidth saving; (b) disturbance identification index (DI); (c) runtime.

accuracy. Misleading predictions in this situation lead to false alarms eventually wasting system resources.

Fig. 12 further elaborates the response of both algorithms for varying RMSE. It can be observed that, in order to achieve an acceptable $DI$ the bandwidth savings of compressive sampling is almost exhausted. On the other hand, dynamic prediction maintains a constant $DI$ and trades off between accuracy of prediction and runtime for the considered values of RMSE. Further, as reducing RMSE does not lead to increase in accuracy of predictions in dynamic prediction, upper error bound of $10^{-3}$ represents a reasonably optimum value. At this point, bandwidth saving of dynamic prediction is 60% more as compared to compressive sampling, with an appreciable improvement of 73% in correctly identifying all disturbance instances with a comparable runtime.

**Remark 5.** With acceptable prediction accuracy and comparable run time complexity, dynamic prediction algorithm outperforms the competitive compressive sampling scheme in terms of bandwidth saving and power system health monitoring.

### *G. Implementation Issues*

For real-time execution of dynamic prediction algorithm, the processing time of each sample and transmission delay should be within the acceptable latency limits. This is typically in the range of 20 ms – 10 sec, depending on the kind of application feeding upon the data [12]. Because of non-stationary nature of PMU data, in the proposed framework it is crucial to occasionally re-estimate the hyper-parameters, which requires a significant fraction of runtime. As observed in Section V-E, training time is considerably higher for longer training lengths, although they do not necessarily guarantee better performance. This is also pointed out in Section V-C that compared to maximum OTL, prediction model trained on mean OTL performs better in terms of all specified performance measures.

In the current study, online implementation of dynamic prediction algorithm has been tested using Simulink based model in Windows 7 operating system. With the optimally chosen parameter settings: OTL = 600 samples, $\epsilon$ = 0.01, and 3-fold cross-validation during training process, we have noted

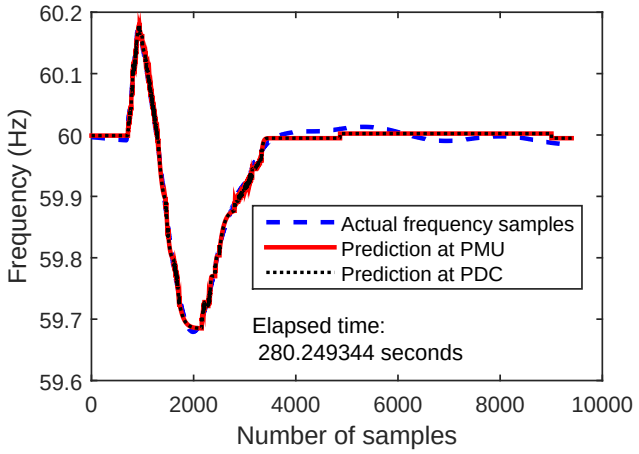



Fig. 13: Online implementation of Dynamic Prediction algorithm.

that the average training and prediction time for each sample in the test data-sets from PMU data repository is 12.7 ms. Typical communication delay is 3-5 ms for a distance of 500 miles between PMU and PDC [12]. Thus, the total delay involved in processing and communication in the proposed dynamic prediction algorithm is less than the lower bound of latency specification, i.e., 20 ms. Fig. 13 shows a test case simulation of 10000 samples in Simulink. It can be observed that the predictions at PDC very closely follow the PMU predictions, and the total execution time is 280.249344 sec. It is expected that by optimizing the code on real-time operating system, processing time of the algorithm can be further reduced.

Fig. 14 shows a possible real-time hardware implementation of the proposed dynamic prediction algorithm. The basic modules comprise of a buffer for storing training samples, a processing unit to perform training and prediction, and a signalling unit to initiate retrainings at the PMU whenever required and to manage status updates and control information flow between PMU and PDC. Logically, the processing unit can be configured to perform 4 primary functions: (a) sequential minimal optimization during the training phase, (b) cache for storing support vectors and temporary hyper-parameter values, (c) estimation of predicted values, and (d) comparator to validate the accuracy of prediction. Since the PDC relies on updates from PMU for its retraining processes, comparator operation is not required at the PDC. Here, the predicted frequency values are exported to the control applications and data archival units through peripherals. Owing to easy re-programmability and real parallel processing, System on Chip built using Field Programmable Gate Arrays are preferred as processing units over the other embedded platforms for SVR hardware implementation. Altera Cyclone II, Cyclone III, Xilinx Virtex-4, and Zynq are a few options considered in state-of-the-art [36]–[38], as they facilitate high speed computations in a time constrained scenario.

For ensuring timely data delivery, careful network design and thorough evaluation of all communication aspects are required. If a future smart grid application requires prediction accuracy $\epsilon \leq 0.01$, then the functionality of dynamic prediction algorithm can be limited by large processing time. However, with advanced performance optimization techniques

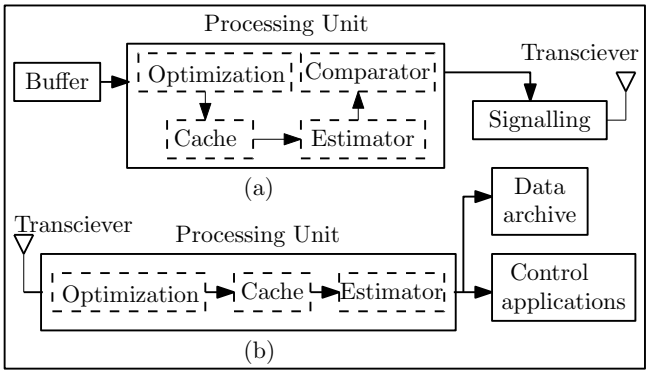



Fig. 14: Hardware implementation schematic of dynamic prediction algorithm: (a) at the PMU; (b) at the PDC.

proposed for future smart grid computations [39], processing time of big data is expected to be considerably small.

## VI. Concluding Remarks

In this paper, a novel algorithm for dynamic prediction of powerline frequency samples based on $\epsilon$-SVR has been proposed for WAMS. The algorithm exploits temporal correlatedness in powerline frequency samples to eliminate transmission of redundant data from PMU. The proposed approach adapts to non-stationarity of power grid transients by re-estimating the hyper-parameters whenever required, in order to ensure high accuracy and robustness during online prediction. The results demonstrate that, with the proposed dynamic prediction around 90% saving in communication channel bandwidth can be achieved without impacting the power system health monitoring process. With suitable choice of hyper-parameters, execution complexity of the proposed algorithm is considerably low and it can be effectively implemented in real-time scenarios. Future works will be aimed at extensive application of the proposed model to different types of data collected by PMUs in multi-machine context, more efficient detection, as well as classification of faults and other transients in the grid.

## References

[1] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, Nov. 2011.

[2] D. Ghosh, T. Ghose, and D. K. Mohanta, "Communication feasibility analysis for smart grid with phasor measurement units," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1486–1496, Aug. 2013.

[3] C. A. Jensen, M. A. El-Sharkawi, and R. J. Marks, "Power system security assessment using neural networks: feature selection using fisher discrimination," *IEEE Trans. Power Syst.*, vol. 16, no. 4, pp. 757–763, Nov. 2001.

[4] K. Sun, S. Likhate, V. Vittal, V. S. Kolluri, and S. Mandal, "An online dynamic security assessment scheme using phasor measurements and decision trees," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1935–1943, Nov. 2007.

[5] F. R. Gomez, A. D. Rajapakse, U. D. Annakkage, and I. T. Fernando, "Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1474–1483, Aug. 2011.

[6] L. S. Moulin, A. P. A. da Silva, M. A. El-Sharkawi, and R. J. Marks, "Support vector machines for transient stability analysis of large-scale power systems," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 818–825, May 2004.

[7] S. Brahma, R. Kavasseri, H. Cao, N. R. Chaudhuri, T. Alexopoulos, and Y. Cui, "Real-time identification of dynamic events in power systems using PMU data, and potential applications-models, promises, and challenges," *IEEE Trans. Power Del.*, vol. 32, no. 1, pp. 294–301, Feb. 2017.

[8] A. Kusiak and A. Verma, "A data-driven approach for monitoring blade pitch faults in wind turbines," *IEEE Trans. Sustain. Energy*, vol. 2, no. 1, pp. 87–96, Jan. 2011.

[9] V. Terzija, G. Valverde, D. Cai, P. Regulski, V. Madani, J. Fitch, S. Skok, M. M. Begovic, and A. Phadke, "Wide-area monitoring, protection, and control of future electric power networks," *Proc. IEEE*, vol. 99, no. 1, pp. 80–93, Jan. 2011.

[10] X. Liu, Z. Gao, and M. Z. Q. Chen, "Takagi-Sugeno fuzzy model based fault estimation and signal compensation with application to wind turbines," *IEEE Trans. Ind. Electron.*, vol. 64, no. 7, pp. 5678–5689, July 2017.

[11] S. Simani and P. Castaldi, "Data-driven and adaptive control applications to a wind turbine benchmark model," *Control Engineering Practice*, vol. 21, no. 12, pp. 1678 – 1693, Dec. 2013.

[12] "IEEE standard for synchrophasor data transfer for power systems," *IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–53, Dec. 2011.

[13] P. Kansal and A. Bose, "Bandwidth and latency requirements for smart transmission grid applications," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1344–1352, Sep. 2012.

[14] C. Liu, S. McArthur, and S. Lee, *Smart Grid Handbook*. John Wiley & Sons, 2016, vol. 1, ch. 10–11.

[15] M. Patel, S. Aivaliotis, E. Allen *et al.*, "Real-time application of synchrophasors for improving reliability," Princeton, NJ, USA,, Tech. Rep., 2010.

[16] E. Cotilla-Sanchez, P. D. H. Hines, and C. M. Danforth, "Predicting critical transitions from time series synchrophasor data," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1832–1840, Dec. 2012.

[17] J. Dong, X. Ma, S. M. Djouadi, H. Li, and Y. Liu, "Frequency prediction of power systems in FNET based on state-space approach and uncertain basis functions," *IEEE Trans. Power Syst.*, vol. 29, no. 6, pp. 2602–2612, Nov. 2014.

[18] L. Xie, Y. Chen, and P. R. Kumar, "Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis," *IEEE Trans. Power Syst.*, vol. 29, no. 6, pp. 2784–2794, Nov. 2014.

[19] Y. Ge, A. J. Flueck, D. K. Kim, J. B. Ahn, J. D. Lee, and D. Y. Kwon, "Power system real-time event detection and associated data archival reduction based on synchrophasors," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 2088–2097, Jul. 2015.

[20] D. Salomon, *Data Compression: The Complete Reference*. Springer London, 2007, ch. 4.

[21] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. San Diego, CA, USA: Academic Press Professional, Inc., 1990, ch. 7.

[22] S. Das and T. S. Sidhu, "Application of compressive sampling in synchrophasor data communication in WAMS," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 450–460, Feb. 2014.

[23] P. H. Gadde, M. Biswal, S. Brahma, and H. Cao, "Efficient compression of PMU data in WAMS," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2406–2413, Sep. 2016.

[24] J. Khan, S. Bhuiyan, G. Murphy, and J. Williams, "Data denoising and compression for smart grid communication," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 200–214, Jun. 2016.

[25] J. E. Tate, "Preprocessing and golomb -rice encoding for lossless compression of phasor angle data," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 718–729, Mar. 2016.

[26] V. Loia, S. Tomasiello, and A. Vaccaro, "Fuzzy transform based compression of electric signal waveforms for smart grids," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 121–132, Jan. 2017.

[27] L. Song, W. Huang, X. Han, and J. Mazumder, "Real-time composition monitoring using support vector regression of laser-induced plasma for laser additive manufacturing," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 633–642, Jan. 2017.

[28] T. Chen and S. Lu, "Accurate and efficient traffic sign detection using discriminative adaboost and support vector regression," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4006–4015, Jun. 2016.

[29] Y. Ren, P. N. Suganthan, and N. Srikanth, "A novel empirical mode decomposition with support vector regression for wind speed forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1793–1798, Aug. 2016.

[30] N. Li, G. Lu, X. Li, and Y. Yan, "Prediction of pollutant emissions of biomass flames through digital imaging, contourlet transform, and support vector regression modeling," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 9, pp. 2409–2416, Sep. 2015.

[31] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer New York, 1999.

[32] J. Machowski, J. Bialek, and J. Bumby, *Power System Dynamics: Stability and Control*. Wiley, 2011, ch. 9.

[33] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[34] "Disturbance monitoring and reporting requirements," Atlanta, GA, USA, Tech. Rep., Nov. 2014.

[35] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[36] X. Pan, H. Yang, L. Li, Z. Liu, and L. Hou, "FPGA implementation of svm decision function based on hardware-friendly kernel," in *International Conference on Computational and Information Sciences*, June 2013, pp. 133–136.

[37] P. Å koda, B. M. Rogina, and V. Sruk, "FPGA implementations of data mining algorithms," in *Proceedings of the 35th International Convention MIPRO*, May 2012, pp. 362–367.

[38] M. Ruiz-Llata, G. Guarnizo, and M. YÃ©benes-Calvino, "FPGA implementation of a support vector machine for classification and regression," in *International Joint Conference on Neural Networks (IJCNN)*, July 2010, pp. 1–5.

[39] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid– The new and improved power grid: A survey," *IEEE Commun. Surveys Tutorials*, vol. 14, no. 4, pp. 944–980, Apr. 2012.

**Sharda Tripathi** received her B. Tech. degree from Rajiv Gandhi Technical University, Bhopal, India, in 2007 and the M.Tech. degree in Digital Communication Engineering from Department of Electronics and Telecommunication Engineering, Maulana Azad National Institute of Technology, Bhopal, India in 2011. She is currently pursuing the Ph.D. degree in Department of Electrical Engineering, Indian Institute of Technology Delhi, India. Her current research interests include application of machine learning in smart grid communication networks.

**Swades De** (S'02-M'04-SM'14) received his B.Tech. in Radiophysics and Electronics from the University of Calcutta, India, in 1993, his M.Tech. in Optoelectronics and Optical Communication from IIT Delhi in 1998, and his Ph.D. in Electrical Engineering from the State University of New York at Buffalo in 2004. He is currently a Professor in the Department of Electrical Engineering at IIT Delhi. Before moving to IIT Delhi in 2007, he was a Tenure-Track Assistant Professor of Electrical and Computer Engineering at the New Jersey Institute of Technology (2004-2007). He worked as an ERCIM post-doctoral researcher at ISTI-CNR, Pisa, Italy (2004), and has nearly five years of industry experience in India on telecom hardware and software development (1993-1997, 1999). His research interests are broadly in communication networks, with emphasis on performance modeling and analysis. Current directions include energy harvesting sensor networks, broadband wireless access and routing, cognitive/white-space access networks, and smart grid networks. He currently serves as a Senior Editor of IEEE Communications Letters, and an Associate Editor of IEEE Wireless Communications Letters, Springer Photonic Network Communications, and the IETE Technical Review Journal. He is a Senior Member of the IEEE Communications and Computer Societies.