Learning-based Multivariate Real-Time Data Pruning for Smart PMU Communication

Rushang Gupta, Varun Gupta, Akash Kumar Mandal, and Swades De
Department of Electrical Engineering, IIT Delhi, New Delhi, India

***Abstract*—This paper proposes a novel machine learning-based multivariate real-time data pruning and prediction framework for smart PMU (phasor measurement unit) communication. In an Internet-of-Things (IoT) enabled smart grid monitoring application, the proposed data-driven pruning technique exploits cross- and auto-correlation in multiple attributes sensed by a PMU (IoT node). The attributes are classified into base and non-base groups based on their ability to aid prediction of the remaining attributes. The idea of transmitting only base attributes reduces the data dimensionality significantly. A reconstruction algorithm is designed for the edge node (local Phasor Data Concentrator) for efficient data reconstruction. The performance of the proposed framework is evaluated on large-scale real-time data from the PMUs. Comparison of the proposed technique with the closest state-of-the-art multi-threaded uni-variate data pruning algorithm in literature demonstrates around 40% more bandwidth saving and $\sim 42\%$ reduction in retraining count.**

***Index Terms*—Smart PMU data communication, learning algorithm, multivariate data pruning, support vector regression, low latency**

I. Introduction

Internet-of-things (IoT) has seen huge adoption in recent years for real-time monitoring of high value systems, generating huge volumes of time critical data [1], [2]. Their ability to aid real-time decision-making, makes them highly suitable for time-critical deployments in many diverse fields. In this infrastructure, the smart phasor measurement units (PMUs) constantly sample the power system's data and send it over wireless communication channel to an edge device. This communication strategy lacks in few important aspects of efficient spectral usage and optimized data storage. Moreover, analyzing such a huge bulk of data is a strenuous task for the edge node, especially under the hardware computational constraints. This not only introduces significant computational delays unsuited to time-critical frameworks, but also makes the whole process economically heavy. Also, off-loading the tasks at multiple phasor data concentrators (PDCs) makes the data more susceptible to breaching. Therefore to make the whole process more safe and viable, efficient real-time data pruning and prediction algorithms are required in smart PMU applications, that not only reduce the data size, but also predict much of it with high reliability in low latency frameworks, thus limiting the duration of PMU-to-PDC communication.

A. Literature Review and Motivation

Several approaches [3] have been proposed for compression of sensor node data to reduce the volume exchanged over the wireless link. Most of them emphasize on data compression frameworks that use statistical methods to compress offline or storage data. The work in [4] applies real-time data pruning on storage data received by the edge node but does not compress the data sent over the wireless channel. [5] takes into account the cross-correlation amongst various attributes but uses wavelet compression like in [6], [7] and [8] to compress and reconstruct the transmitted data, introducing significant delays in the system which is incongruous to smart PMU application. Principal component analysis is used by [9] for compression of PDC data and [10] uses a similar approach in smart meter applications. The work in [11] adds a second-stage which uses discrete-cosine transform for compression of PMU data. Such techniques can be efficient in compressing data but their inability to perform this prune in real-time renders them inappropriate in delay-sensitive applications. Also, the approach of sending the model parameters reduces the data redundancy significantly, leading to higher susceptibility of erroneous reconstruction at edge devices (PDCs) in wireless data communication through highly impulsive smart grid channels.

In [12], the authors use a lossy compression framework to achieve high compression ratios in the edge node served by a PDC to control-center data relay, which could be detrimental in real-time control frameworks. While this approach might be suitable in particular scenarios, the strategy to send data only at the occurrence of a disturbance partially solves the problem loosing to incomplete system observability. Some works also harness the abilities of probabilistic models to approximate the data collected and hence minimize the IoT communication overhead [13] but do not exploit multi-attribute correlation. Many researchers have proposed machine learning solutions using decision trees [14] and multi layer perceptron [15] to save bandwidth while addressing this issue of real-time data pruning. In [16], a real-time data compression framework using ϵ-support vector regression (SVR) to dynamically predict the powerline frequency is proposed, but it fails to meet the real-time sensing and processing constraint required in the PMU-PDC communication. Moreover, none of these works consider the cross-correlation amongst multiple attributes measured by a sensor node.

To this end, none of the existing works provide a real-time multivariate data pruning framework for smart IoT communication, exploiting the cross-correlation between different power system attributes measured by the sensing node, without significant loss of information. The development of such a real-time framework will help with a significant reduction in the amount of data transmitted during the PMU-to-PDC

communication, leading to huge amounts of bandwidth saving and storage optimization. This paper proposes one such novel multivariate real-time data pruning algorithm that can efficiently compress and transmit data from the smart PMUs in real-time, while preserving the important characteristics of the raw data required in delay-constrained frameworks.

B. Contributions and Significance

The key contributions of this work are as follows: 1) A cross-correlation aware learning-based real-time data pruning algorithm is proposed for smart IoT communication. 2) The proposed approach performs attribute grouping based on their cross-correlation, and communicates the pruned data for the base attributes, thus offering substantial dimensional reduction. 3) This strategy performs real-time data pruning in smart IoT networks without adding any delay to the existing communication delay budget. 4) Results show that the proposed algorithm achieves significant bandwidth savings and thus stands out in the present state-of-the art IoT communications in power system framework. Moreover with reduced data size, it's susceptibility to wireless channel noise decreases, and thus an efficient reconstruction under smart grid wireless channel noise can be ensured at the edge node. Besides wide area monitoring systems, the proposed framework can be applied in other smart IoT applications, namely, wireless body sensor networks and vehicular communication.

Section II presents the system model, Section III contains the multi-variate data pruning algorithm, followed by results and concluding remarks in Sections V and VI, respectively.

II. System Model

A large amount of data is being exchanged over wireless channel between the smart IoT devices, and crucial to these communications is the accuracy and the delay that is experienced in the PMU to PDC communication. Fig. 1 shows the system model the multivariate pruning algorithm bases on.

The proposed system considers optimal placement of the sensor nodes in a massive network composed of multiple IoT sensing nodes [17] and data concentrators (edge nodes) to ensure complete system monitoring. These IoT nodes generate data at high frequencies and transmit to the edge node over a wireless communication channel. The proposed algorithm prunes raw data at the smart IoT nodes to save bandwidth

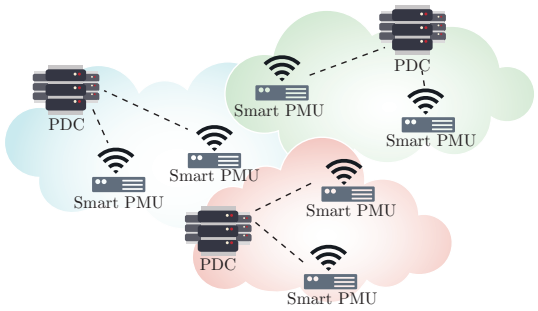


Fig. 1: System model for smart PMU communication.

in low latency frameworks. The edge node works in synchronization with these IoT nodes to receive the pruned data and reconstruct the original data based on a predictive machine learning model described next.

III. Multivariate Data Pruning Framework

Several approaches were discussed in Section I-A which compress sensor data exploiting the auto-correlation in a single attribute. Though these approaches were efficient for data volumes corresponding to one attribute, they neither take into account the transmission of multiple attributes in smart PMU communication, nor account for the cross-correlation between multiple attributes, which can further be exploited to reduce bandwidth consumption in multi-attribute transmission through a wireless communication setup. The proposed framework exploits both cross-correlation and auto-correlation and does real-time data prediction using ϵ-SVR implemented.

ϵ-SVR maps the input features to a higher-dimensional space and performs regression to find the best fit on the given data by creating an ϵ-tube around it, which is particularly useful in cases where the error in prediction has to be contained in a particular range. Let $A_i = \{A_i^1, A_i^2, \cdots, A_i^n\}^T$ be the i^{th} sample of n attributes from the sensor data-set and let us consider estimating l time samples for each of the n attributes measured by the sensor node, such that $i \in \{1, \cdots, l\}$ and $\nu \in \{1, \cdots, n\}$. Then the predictions for the i^{th} sample of these attributes $\hat{A}_i = \{\hat{A}_i^1, \hat{A}_i^2, \cdots, \hat{A}_i^n\}^T$, is expressed as

$$\hat{A}_i = \text{diag}\left\{\omega_{A_i}^T \Phi_{A_i}\right\} + \underline{b}_i \tag{1}$$

where $v = diag(M)$ forms a vector from the diagonal entries of matrix M with their position in the vector v given by their row or column index in the matrix M, b is the $n \times 1$ attribute bias vector, ω_{A_i} and Φ_{A_i}, $i \in \{1, 2, \cdots, l\}$ are given as

$$\omega_{A_i} = \begin{bmatrix} w_{i-1}^1 & \cdots & w_{i-d_1}^1 & \overset{d_{max}-d_1}{\cdots} & 0 \\ w_{i-1}^2 & \cdots & w_{i-d_2}^2 & \overset{d_{max}-d_2}{\cdots} & 0 \\ \vdots & \ddots & \vdots & \ddots & 0 \\ w_{i-1}^n & \cdots & w_{i-d_n}^n & \overset{d_{max}-d_n}{\cdots} & 0 \end{bmatrix}^T_{n \times d_{max}}$$

$$\Phi_{A_i} = \begin{bmatrix} \phi^1(A_{i-1}^1) & \phi^2(A_{i-1}^2) & \cdots & \phi^n(A_{i-1}^n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi^1(A_{i-d_1}^1) & \phi^2(A_{i-d_1}^2) & \cdots & \phi^n(A_{i-d_1}^n) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \phi^2(A_{i-d_2}^2) & \cdots & \phi^n(A_{i-d_2}^n) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \phi^n(A_{i-d_n}^n) \end{bmatrix}_{d_{max} \times n}$$

are the weight matrix for different attributes and the non-linear mapping from input to feature space for the estimate

of i^{th} attribute sample measured by the IoT node. w^{ν}_{i-f} is the weight of $(i-f)^{th}$ lag value in the auto-regressive model for the ν^{th} attribute estimation from the attribute set A_i and $\phi^{\nu}(A^{\nu}_{i-f}) = \{\phi^{\nu}(A^{\nu}_{i-1}), \phi^{\nu}(A^{\nu}_{i-2}), \cdots, \phi^{\nu}(A^{\nu}_{i-d_\nu})\}$ is the feature mapping for the i^{th} sample estimation of the ν^{th} element in the attribute set A_i, and $f \in \{1, \cdots, d_\nu\}$, with d_ν being the optimum number of lag samples required for estimation of attribute A^{ν} and is based on the variation, importance and tolerance that can be permitted in the estimation of that attribute. All the rows and columns in $\omega^{T}_{A_i}$ and Φ_{A_i} are appropriately zero padded with the number of zeros (noz) appended in them being equal to $noz = d_{\max} - d_p$, where d_p is the number of elements in that row or column and $d_{\max} = \max\{d_1, d_2, \cdots, d_\nu\}$. The values of weights can be obtained by optimizing the problem (2)

$$\begin{aligned} &\mathscr{P}_1\text{: minimize} \left\{ \frac{1}{2}||w_{A^{\nu}_i}||^2 + \Upsilon^{\nu} \sum_{i=1}^{l} (\theta^{\nu}_i + \theta^{*\nu}_i) \right\} \\ &\text{s.t. } \mathscr{C}_1\text{: } A^{\nu}_i - \hat{A}^{\nu}_i \leq \epsilon + \xi^{\nu}_i \\ &\quad \mathscr{C}_2\text{: } \hat{A}^{\nu}_i - A^{\nu}_i \leq \epsilon + \xi^{*\nu}_i;\ \ \xi^{\nu}_i, \xi^{*\nu}_i \geq 0 \end{aligned} \tag{2}$$

for n values of attributes, where $\omega_{A^{\nu}_i}$ is the weight of d_ν lag samples used in the estimation of the i^{th} sample for ν^{th} attribute in A_i, θ^{ν}_is are the slack variables ensuring the feasibility of constraints $\mathscr{C}_1$ and $\mathscr{C}_2$, and Υ^{ν} is the trade-off factor in the curvature of $\omega_{A^{\nu}_i}$ to θ_i for the ν^{th} attribute in A_i. Forming the Lagrangian L for (2) using multipliers π^{ν}_i, $\pi^{*\nu}_i$, υ^{ν}_i, $\upsilon^{*\nu}_i$ and taking its partial derivative with respect to ω, b^{ν}_i (offset bias for the ν^{th} attribute of A_i), θ^{ν}_i and $\theta^{*\nu}_i$, we get the dual as,

$$\begin{aligned} &\mathscr{P}_2\text{: maximize} \left\{ -\frac{1}{2} \sum_{i,j=1}^{l} (\pi^{\nu}_i - \pi^{*\nu}_i)(\pi^{\nu}_j - \pi^{*\nu}_j) \right. \\ &\left. \langle \phi^{\nu}(A^{\nu}_i, \phi^{\nu}(A^{\nu}_j) \rangle - \epsilon \sum_{i=1}^{l} (\pi^{\nu}_i + \pi^{*\nu}_i) + \sum_{i=1}^{l} A^{\nu}_i (\pi^{\nu}_i - \pi^{*\nu}_i) \right\} \\ &\text{s.t., } \mathscr{C}_4\text{: } \sum_{i=1}^{l} (\pi^{\nu}_i - \pi^{*\nu}_i) = 0; \pi^{\nu}_i, \pi^{*\nu}_i \in [0, \Upsilon^{\nu}]. \end{aligned}$$

The inner product $\langle \phi^{\nu}(A^{\nu}_i, \phi^{\nu}(A^{\nu}_j) \rangle$ can be replaced with a kernel in input space. Therefore the attribute estimates can be generalized as

$$\hat{A}^{\nu}_i = \sum_{i,j=1}^{l} (\pi^{\nu}_i - \pi^{*\nu}_i) K^{\nu}(A^{\nu}_i, A^{\nu}_j) + b^{\nu}_i \tag{3}$$

In this work, we have used a radial basis kernel function given by, $K^{\nu}(A^{\nu}_i, A^{\nu}_j) = \exp(-\beta||A^{\nu}_i - A^{\nu}_j||)^2\ \forall i, j \in \{1, \cdots, l\}$.

A. Set Allocation

Disjoint sets comprising of various sensor data attributes are constructed such that each set has attributes that are highly correlated and has one base attribute which can be used to predict all other attributes in that set. A set allocation algorithm is put in place that uses the correlation matrix, $\mathbf{M}_{l \times l}$, which is computed for all the attributes using initial data that is exchanged between the smart PMU and PDC. Two groups $\mathbf{G1}$ and $\mathbf{G2}$ are formed to segregate these attributes, with all base attributes in set $\mathbf{G1}$ and the rest in $\mathbf{G2}$. All attributes are initially kept in set $\mathbf{G2}$ and have two parameters associated with them, ρ^{ν}_i representing the value of maximum correlation of the ν^{th} element in $\mathbf{G2}$ for i^{th} sample estimate with the elements in set $\mathbf{G1}$, and δ^{ν}_i is the position index of that attribute in set $\mathbf{G1}$, exhibiting maximum correlation with an attribute in set $\mathbf{G2}$. At the beginning, these parameters are set to 0 and -1, since $\mathbf{G1}$ is empty. A cross-correlation threshold, $\mathbf{c}_t$ is defined using the Pearson's correlation coefficient for a data in a frame width of time T to generate the couple attributes with highest correlation. Using this threshold, we iterate through $\mathbf{G2}$ and find the distance (d^{ν}_i) of ρ^{ν} for i^{th} attribute from $\mathbf{c}_t$ defined as, $d_i = \rho^{\nu}_i - \mathbf{c}_t$. Attributes having $d^{\nu}_i < 0$ are the candidate attributes which can be shifted to set $\mathbf{G1}$. A transfer score (κ) for each candidate attribute is calculated as $\kappa^{\nu}_i = \sum_{A_j \in \mathbf{G2}} ||\mathbf{m}_{kj} - \rho_j||$, where $\mathbf{m}_{kj}$ is an element of the matrix $\mathbf{M}_{l \times l}$. Therefore, one with the maximum score is shifted to $\mathbf{G1}$. Values of ρ and ν_M are then updated for all the attributes in $\mathbf{G2}$. This process is repeated till all the attributes in $\mathbf{G2}$ validate $\rho > \mathbf{c}_t$. After the formation of groups, we form sets $S_l,\ l \in \{1, 2, \cdots, P\}$, with $P \overset{def.}{=} |\mathbf{G1}|$. Each attribute of $\mathbf{G1}$ is allotted a different set and each attribute of $\mathbf{G2}$ is added to the set corresponding to its position index δ^{ν}_i.

B. Dynamic Prediction

Two types of SVR models are maintained at both nodes for each set. An auto-regressive model for the base attribute which uses its self-predicted values for subsequent predictions, and a separate regression model for each remaining attribute, using the base attribute for prediction of the other non-base attributes. We define run-time prediction errors in base and non-base attributes as e_B and e_{NB} respectively. Predictions are carried out for each of these attributes using their respective models and a flag based on their respective errors is compared against the error threshold $\varepsilon_{th,\nu}$. The error flag ($\mathscr{F}_e$) is set to 0 until one the following event triggers:

1) $e_B > \varepsilon_{th,\nu}$: In this case, the SVR model for this base attribute is retrained on both the nodes using the recently predicted good samples from the model. Since both the models will train on previously predicted data which is available at both the nodes, no data is sent over the communication channel. If the error for the next prediction using the retrained models still exceeds the maximum error limit at the PMU, fresh retraining samples that are recorded at the PMU are sent to the PDC and are then used to retrain both the models. These new models are then used to make the subsequent predictions.

2) $e_{NB} > \varepsilon_{th,\nu}$: In this case, if the base attribute is not predicting values within the range, it corrects itself using the method defined in the previous case. Since the attributes of a set are highly correlated, a failing base attribute model fails the non-base attribute model too. Rather than training both the SVR models, training only the base attribute model would lead

Algorithm 1: Pruning algorithm at smart PMU

Result: Pruned data
$\mathbf{M}_{l\times l}$ = Correlation Matrix for l attributes
Initialize set $\mathbf{G1} = \{\}$: Empty set
Initialize set $\mathbf{G2}$: Attribute set
$\rho_i^\nu = 0$ and $\delta_i^\nu = -1 \ \forall \ \nu \in \mathbf{G2}$
Initialize $\mathbf{c}_t$: Correlation threshold
while $d_i^\nu \ (\forall \nu \in \mathbf{G2}) \leq 0$ **do**
  for $\nu \in \mathbf{G2}$ **do**
    if $d_i^\nu > 0$ **then**
      continue
    $\kappa_i^\nu = \sum_{\nu\in\mathbf{G2}} ||\mathbf{m}_{\nu j} - \rho_j||$
  $\{\mathbf{G1}(\nu) \rightarrow \mathbf{G2}[\cdot] : \rho_i^\nu > \rho_j^\nu \forall i \neq j\}$
  for *each* $\nu \in \mathbf{G2}$ **do**
    Update $\rho_i^\nu = \max_{\nu\in\mathbf{G2}}\{\mathbf{m}_{\nu j}\}$
    Update $\delta_i^\nu = \nu \ni \nu \in \mathbf{G2}, \mathbf{m}_{\nu j} = \rho_j$
Form sets, S_v, where $v \in \{1, 2, \cdots, P\}; P = |\mathbf{G1}|$
Allot each attribute of $\mathbf{G1}$ to different sets
for $\forall \nu \in \mathbf{G2}$ **do**
  Allocate $\nu \rightarrow S_k \ni \rho_i^\nu \in S_v$
for $S_v; \forall v \in \{1, 2, \cdots, P\}$ **do**
  Define ϵ-SVR model for base attribute $\in S_v$
  Define ϵ-SVR model for non-base attributes $\in S_v$
  Define ε_i^ν for each attribute
while *True* **do**
  Predict value of each attribute
  if *Prediction error* $\forall \nu < \varepsilon_{th,\nu}$ **then**
    Continue
  if $e_B > \varepsilon_{th,\nu}$ **then**
    Retrain $\epsilon-$ SVR using good predicted samples
    Send updated flags to PDC
    if *Error still persists* **then**
      Send updated flags at edge node
      Retrain $\epsilon-$SVR using fresh samples
      Send model parameters to PDC
  if $e_{NB} > \varepsilon_{th,\nu}$ **then**
    Retrain using good predicted samples
    if *Error still persists* **then**
      Reform the sets, S_v, for $v \in \{1, 2, \cdots, P\}$
      Send updated flags at PDC
      Send sets and model parameters to PDC
      Retrain $\epsilon-$SVR using fresh samples

to an automatic correction in the predictions of the non-base model rendering higher bandwidth saving. If the base model is predicting the values within the error bound, the regression SVR model for the non-base attribute is retrained using the recently predicted good samples of the base attribute. If the error is still greater than the threshold, then we recalculate the correlation matrix and check if it still belongs to its current set. If not, we update it to the correct set and use the actual values to retrain the SVR model on both the nodes.

Algorithm 2: Reconstruction algorithm at PDC

Result: Reconstructed data
Receive initial data points from smart PMU
Receive set information and error thresholds
for *Each* S_v*;* $\forall v \in \{1, 2, \cdots, P\}$ **do**
  Define ϵ-SVR model for base attribute $\in S_v$
  Define ϵ-SVR models for non-base attributes$\in S_v$
while *True* **do**
  Predict value of each attribute
  if *Flags received from smart PMU* **then**
    if *Self-retraining flag received* **then**
      Retrain ϵ-SVR on good predicted samples
    if *Data or model parameters received* **then**
      if *For base attribute* **then**
        Retrain respective model
      if *For non-base attribute* **then**
        Update sets
        Retrain respective model

This casts a trade-off between sending only model parameters leading to more bandwidth saving at the cost of less data redundancy and more prone to wireless channel errors, over sending the retraining data, enhancing the reliability of the wireless transmission scenario. The latter helps in reducing the retransmission counts significantly, and helps to adhere with the ultra reliable low latency communication constraints, however, at the cost of a higher spectral usage. A detailed description of the proposed multivariate algorithm involved in the data pruning and reconstruction is outlined in Algorithm 1 and 2 for the PMU and PDC respectively.

Remark 1. *It is worth understanding here that when the SVR model is being trained, retrained or attributes are regrouped at the receiver IoT node (PDC) owing to the predictions going out of error bound, we keep transmitting the actual data. Therefore, the receiver always has system information congruous to any real-time application.*

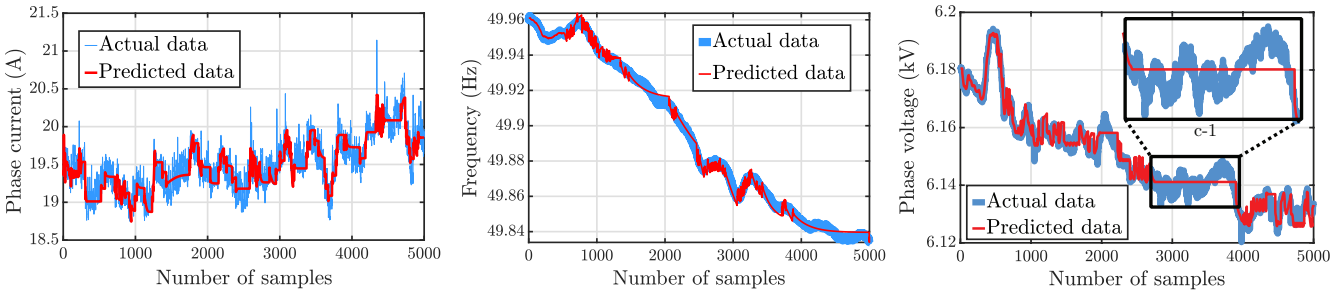
IV. Performance Indices

For the performance analysis of the proposed algorithm, following indices are defined:

1) Retraining count (RC): It is the number of times the model had to be retrained to keep the prediction error bounded by ε_{th}. We have used same $\%\varepsilon_{th}$ for all the attributes in Fig. 4. We define a new performance parameter called effective retraining count η, for analyzing the run-time complexity of the proposed multivariate data pruning algorithm over the state-of-the-art. Mathematically,

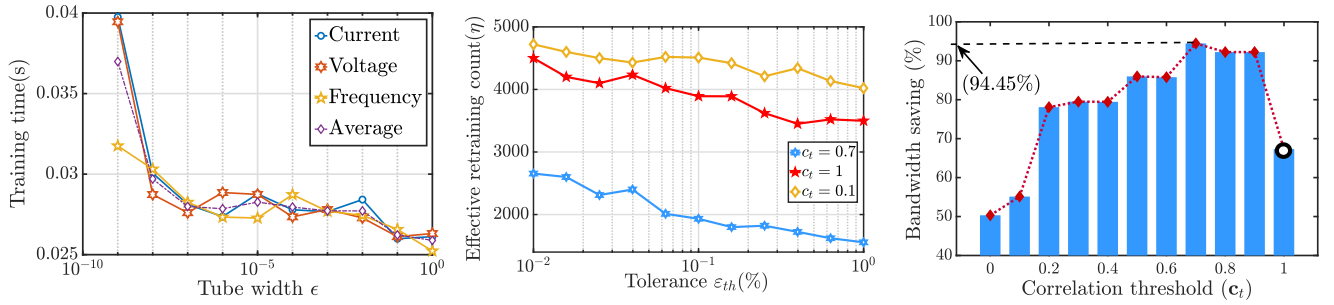
$$\eta = \frac{\sum_{\nu=1}^{n} t_\nu RC_\nu}{\sum_{\nu=1}^{n} t_\nu}$$

where t_ν is the average duration observed in the training of the ν^{th} attribute including all the retraining instances for a



(a) (b) (c)

Fig. 2: Comparison of actual versus predicted samples: (a) phase current; (b) frequency; (c) phase voltage.



(a) (b) (c)

Fig. 3: (a) Training time per attribute vs. tube width; (b) effective retraining count vs. tolerance; (c) bandwidth saving vs. correlation threshold.

given tube width and RC_ν is the total retraining suffered by the attribute.

2) Normalized root mean square error (nRMSE): nRMSE for the ν^{th} attribute in $\varepsilon_\nu = A_i^\nu - \hat{A}_i^\nu$ for the proposed algorithm is defined as,

$$nRMSE = \frac{\sum_{\nu=1}^{n} ||\varepsilon_\nu / \sqrt{l}||}{n}$$

3) Bandwidth saving: It is defined as the percentage of actual data that were not transmitted and were predicted within a predefined tolerance ε constrained by the SVR tube.

V. Results and Discussions

A. Experimental Setup for Testing of Proposed Multivariate Data Pruning Algorithm

The proposed algorithm in Section III is implemented and validated on the data measured by a PMU (IoT node) installed at the IIT Delhi MSB substation, reporting data over a wireless channel to a system within a radius of 1 km, behaving as a PDC (edge node). The PMU is connected to the incomer bay of the 11 kV/440 V, 50 Hz substation with maximum load current rating of 600 A. The PMU in its default configuration without the installation of the pruning framework reported data at 25 Hz. The performance of this algorithm is compared to a n single-variate data pruning scenario for a closest fit manifested from literature as explained in following subsections.

B. Determining Optimal Hyper-parameters

Correlation threshold corresponding to the maximum bandwidth saving is used as its optimum value. From Fig. 3, the maximum bandwidth saving is achieved at $\mathbf{c}_t = 0.7$ under the considerations of other parameters used in our algorithm. $\mathbf{c}_t = 1$ corresponds to the case where no set formation happens and each attribute is therefore a base attribute for the PMU data. Hence, with $\mathbf{c}_t = 1$ cross-correlation between the attributes is not exploited, which is analogous to n single-variate data pruning algorithm, and therefore this case gives a valuable insight into the validity as well as strength of the proposed algorithm in pruning the highly sensitive power systems' data. Table I shows the values of the hyper-parameters used in the validation of the proposed algorithm and the performance indices obtained as a by-product. The proposed multivariate algorithm is able to achieve 40% more bandwidth savings than the n single-variate data pruning algorithm (marked with a black hollow dot in Fig. 3) by not transmitting 94.45% of the sampled data to the PDC.

C. Performance of Multivariate Data Pruning Algorithm

Predictions for all the attributes using the algorithm described in section III was tested on the setup in Fig. 4. Upon

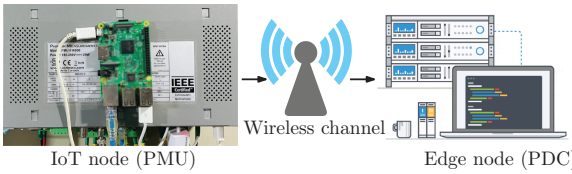


Fig. 4: Experimental setup for validation of the proposed multivariate data pruning algorithm.

TABLE I: Performance indices for experimental setup in Fig. 4

Parameter	Value
Tube width, ϵ	10^{-3}
Tolerance, ε_{th}	10^{-1}
Correlation threshold, c_t	0.7
Training length	50
Lag, d	5
nRMSE	4.34×10^{-5}
Bandwidth saving	40%
Retraining count saving	42.22%

a closer inspection of the actual current samples in Fig. 2(a) (plotted with fat line), we notice small spikes in the data. These spikes last for only one or two samples within 5% of nominal value, accounting for only ~ 0.04-0.08 seconds, at the PMU's default reporting rate (25 Hz). By definition, these can not be considered to constitute a fault [18], owing to a very small power in these impulses. Moreover, these spikes dying quickly do not change the attribute average till the instant, rendering retraining the model unnecessary at such instances. The same inference can be drawn for the zoomed region c-1 in Fig. 2(c).

Algorithm in 1 was run on a Broadcom BCM2837 64-bit quad core processor. The plot in Fig. 3(a) shows the training time for various attributes measured by the PMU. It can be inferred from the plot that the training time for all tolerance values stays below 0.04 seconds, which is less than the reporting rate (25 Hz) used by the PMUs and therefore does not add any delay to the real-time data sensing and communication. Using the defined performance indices in Section IV and the plot in Fig. 3(a), we can infer from Figs. 3(b) and (c) that the proposed algorithm performs much better than the n single-variate real-time data pruning algorithm manifested from existing literature in terms of bandwidth saving ($\sim 40\%$) and effective retraining count reduction (42.22%). The algorithm is applied to the window-averaged data which takes the average of the last 20-25 samples to smoothen the estimated curve. This reduces the retraining count significantly (cf. Fig. 3(b)), while ensuring no exclusion of any critical retraining.

Remark 2. *It must be noted here that due to unavailability of any mutlivariate real-time data pruning strategies in context to fog and edge node data processing, this paper compares the performance of the proposed algorithm with a n single-variate real-time data pruning scenario manifested from the existing literature. We apply real-time single-variate data pruning on n-data streams corresponding to n attributes sensed by the fog node, corresponding to $\mathbf{c}_t = 1$.*

VI. Conclusion

This paper proposed a novel learning-based multivariate data pruning algorithm for real-time smart IoT communication scenarios. The framework aims at reducing the data volume to be transmitted in PMU-to-PDC (IoT node to edge node) communication. It performs a channel-aware transmission of data and limits the prediction error within an ϵ tube using ϵ-SVR. The various attributes sensed by the IoT node are placed in different sets based on their cross-correlation and its predefined thresholds. The ones exhibiting maximum correlation with others are taken as base attributes and contribute to the estimation of other non-base parameters. The comparison of the algorithm with the closest state-of-the-art on a PMU-to-PDC communication test setup clearly illustrated a superior performance of this algorithm rendering better bandwidth saving with a reduced effective retraining count.

Acknowledgement

This work was supported in parts by the Science and Engineering Research Board under Grant CRG/2019/002293, and the Department of Telecommunications for building end to end 5G Test-bed under Grant 4-23/5G test bed/2017-NT.

References

[1] V. Gupta and S. De, "Collaborative multi-sensing in energy harvesting wireless sensor networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 426–441, 2020.

[2] S. Ghosh, S. De, S. Chatterjee, and M. Portmann, "Learning-based adaptive sensor selection framework for multi-sensing wsn," *IEEE Sensors J.*, vol. 21, no. 12, pp. 13 551–13 563, 2021.

[3] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in *Proc. Int. Conf. Inf. Technol.: Coding Comput.*, vol. 2.

[4] V. Alieksieiev, "One approach of approximation for incoming data stream in iot based monitoring system," in *IEEE Int. Conf. on Data Stream Mining Process.*, 2018.

[5] J. Zhang, K. Yang, L. Xiang, Y. Luo, B. Xiong, and Q. Tang, "A self-adaptive regression-based multivariate data compression scheme with error bound in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 2013, 03 2013.

[6] S. Santoso, E. J. Powers, and W. M. Grady, "Power quality disturbance data compression using wavelet transform methods," *IEEE Trans. on Power Deliv.*, vol. 12, no. 3, 1997.

[7] J. Ning, J. Wang, W. Gao, and C. Liu, "A wavelet-based data compression technique for smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 1, 2011.

[8] 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, 2016.

[9] S. Das and P. S. N. Rao, "Principal component analysis based compression scheme for power system steady state operational data," in *ISGT2011-India*.

[10] M. R. Chowdhury, S. Tripathi, and S. De, "Adaptive multivariate data compression in smart metering internet of things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, 2021.

[11] P. H. Gadde, M. Biswal, S. Brahma, and H. Cao, "Efficient compression of pmu data in wams," *IEEE Trans. on Smart Grid*, vol. 7, no. 5, 2016.

[12] W. Ren, T. Yardley, and K. Nahrstedt, "ISAAC: Intelligent synchrophasor data real-time compression framework for WAMS," in *Proc. IEEE Int. Conf. Smart Grid Comm.*, Dresden, Germany, 2017.

[13] D. Chu, A. Deshpande, J. M. Hellerstein, and Wei Hong, "Approximate data collection in sensor networks using probabilistic models," in *Proc. 22nd Int. Conf. Data Eng.*, 2006.

[14] 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. on Power Syst.*, vol. 22, no. 4, 2007.

[15] C. A. Jensen, M. A. El-Sharkawi, and R. J. Marks, "Power system security assessment using neural networks: feature selection using fisher discrimination," *" IEEE Power Eng. Rev.*, vol. 16, no. 4, 2001.

[16] S. Tripathi and S. De, "Dynamic prediction of powerline frequency for wide area monitoring and control," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, 2018.

[17] L. Huang, Y. Sun, J. Xu, W. Gao, J. Zhang, and Z. Wu, "Optimal pmu placement considering controlled islanding of power system," *IEEE Trans. on Power Syst.*, vol. 29, no. 2, pp. 742–755, 2013.

[18] P. Kundur, "Power system stability," *Power system stability and control*, 2007.