Adaptive Multivariate Data Compression in Smart Metering Internet of Things

Mayukh Roy Chowdhury, Sharda Tripathi, and Swades De

***Abstract*—Recent advances in electric metering infrastructure has given rise to the generation of gigantic chunks of data. Transmission of all of these data certainly poses a significant challenge in bandwidth and storage constrained Internet of Things (IoT) where smart meters act as sensors. In this work, a novel multivariate data compression scheme is proposed for smart metering IoT. The proposed algorithm exploits the cross-correlation between different variables sensed by smart meters to reduce the dimension of data. Subsequently, sparsity in each of the decorrelated streams is utilized for temporal compression. To examine the quality of compression, the multivariate data is characterized using Multivariate Normal – Autoregressive Integrated Moving Average (MVN-ARIMA) modeling before compression as well as after reconstruction of the compressed data. Our performance studies indicate that compared to the state of the art, the proposed technique is able to achieve impressive bandwidth saving for transmission of data over communication network without compromising faithful reconstruction of data at the receiver. The proposed algorithm is tested in a real smart metering set-up and its time complexity is also analyzed.**

***Index Terms*—Smart meter, Internet of Things, multivariate data, principal component analysis, compressive sampling**

I. INTRODUCTION

WITH rapid growth in the number of smart objects in the era of Internet of Things (IoT), big data has always been an area of concern. Sensors deployed in an IoT setup generate a massive amount of data which is transmitted over the communication channel to a central entity, e.g. a cloud server. Advanced metering infrastructure (AMI) is an emerging IoT scenario where smart electricity meters behave as sensors. They are installed in domestic or industrial environments and continuously sample values of different variables to monitor electricity consumption. This data is interfaced with various applications, namely billing, demand side management, load forecasting, and dynamic pricing, to facilitate resource optimization by the service provider and quality of service delivery to the consumers. In a typical smart meter set-up, electricity consumption is sampled at high frequency and reported periodically to the collecting node as shown in Fig.1.

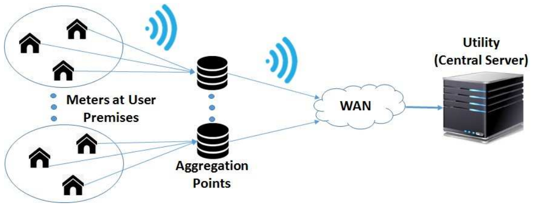


Fig. 1: Data communication framework in AMI.

Millions of smart meters installed throughout the world generate gigabytes of data which is expected to rise even up to the order of hundreds of terabytes per year in near future [1]. Practical examples given in [2], [3] show how a hundred million of such meters recording five kilobytes of data once in 15 minutes can collectively generate as high as 2920 terabytes in one year. Owing to this enormous volume of generated data, the requirement of bandwidth for transmission and storage space for data archival is very high. Also, from the Internet service providers' perspective, loading the network unnecessarily in a constrained scenario needs to be avoided. Hence, compression or pruning of smart meter (IoT) data has attracted the attention of the research community in recent years, as it can lead to saving in transmission bandwidth for the telecom operators and reducing the storage cost [2]–[5].

A. Related works

Smart meter data compression techniques reported in the literature can be broadly categorized based on: a) operating mode: at the meter or at the aggregation point in the AMI framework; b) quality of reconstruction: lossy or lossless. A brief overview of relevant literature is presented below.

Spatial compression techniques are applied at the aggregation point, where consumption data from multiple users are available [6]–[8]. The study in [6] expressed the consumption profile of different customers as different combinations of a set of partial usage patterns and used them for spatial compression. It also exploited sparsity in electricity consumption data of a single customer for temporal compression. The authors in [7] capitalized on the spatial correlation of single variable data collected from metering devices at multiple substations. In [8], load data was characterized using generalized extreme value distribution, and the load features were used for data compression. The main objective of these works is saving of transmission bandwidth between data collectors (aggregation points) and the control center. It is also noted that these techniques have considered low-resolution data.

When data is captured at a high resolution, inconsistencies in consumption patterns bring down compressibility of meter data owing to the difficulty in identifying the underlying patterns. Algorithms applied at the individual device level in smart metering framework have considered data of higher resolution (1 sample in several seconds) and mostly employ

M. Roy Chowdhury 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: {mayukh.roychowdhury, swadesd}@ee.iitd.ac.in).

S. Tripathi is with the Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy (e-mail: sharda2309@gmail.com).

lossless compression [9]–[12]. In [9], the authors compared performance of four lossless compression algorithms, namely, Markov Chain Huffman coding, Lempel - Ziv Markov Chain algorithm, Adaptive Trimmed Huffman method, and Lempel - Ziv - Markov Chain - Huffman method, on smart meter data. A lossless compression technique for smart meter load profiles has been recently reported in [10], which uses Gaussian approximation based on dynamic nonlinear learning technique. A low complexity loss-less compression method with resumability was proposed in [11] based on differential and entropy coding. It was applied on high frequency open dataset REDD [13] with smart meter data logged at every second. In [12], two well-known lossless techniques, namely, Lempel - Ziv Welsh (LZW) and Adaptive Huffman Coding, were assessed in terms of overhead and quality of compression. Datasets with relatively higher sampling period of 10 to 60 minutes were used, and LZW showed better performance.

Lossless compression schemes lead to accurate reconstruction, and they do not incur any information loss. Lossy algorithms, on the other hand, have a higher compression ratio, which makes them suitable for application in error-tolerant scenarios, such as in a wireless communication environment. Moreover, many smart metering applications, e.g., load forecasting, do not require the exact consumption data; rather, a reasonably accurate trend of consumption is sufficient in deciding on an action. Hence, there exists a trade-off between the extent of compression achievable and the information lost due to it. A lossy compression technique proposed in [14] applied piece-wise regression on time series data generated by smart meters and claimed that the compression did not lead to any significant information loss. Sparse coding was also used as lossy compression technique on smart meter data in [15], where the compressed data was further used for load forecasting in residential areas. Symbolic aggregate approximation was used for smart meter data compression [16]; however, this approach was found to suffer from poor recoverability. A lightweight joint authentication and compression scheme for low-cost smart meters was proposed in [17]. A load data compression technique based on Neural Network was proposed in [18]. The authors in [19] highlighted the storage constraints in smart meters and suggested a lossy compression algorithm to address the storage constraint. In a recent study [20], the authors proposed a lossy data compression scheme on a single time-series data that offers higher bandwidth saving compared to the existing schemes.

B. Motivation and major contributions

From the current state-of-the-art, it is observed that most of the existing smart meter data compression techniques consider individual variable only. For example, Adaptive Compressive Sampling (ACS) algorithm proposed in [20] operates on only apparent power (AP) data. A few works that have considered multiple variables, for example [7], applied the compression techniques on each of the variables independently.

In real smart meters, multiple variables, such as power, current, voltage, frequency, energy, and meter-health related parameters, are sensed and transmitted over communication link to a data aggregator. Since the amount of data generated by numerous smart meters is massive and the communication bandwidth is expensive, it is essential to investigate whether continuous transmission of each of the measured variables is required. Some of them are expected to be correlated, and hence there may be redundancy in the information. To this end, characterization of multivariate smart meter data is expected to help in capturing the inter-dependence between different variables. Modeling of the joint distribution of multivariate data will also help to identify whether the distribution of data is preserved after pruning of the content, which in a way would validate the reconstruction accuracy.

Motivated by the above observations, this work presents a multivariate data compression scheme which exploits the cross-correlation among different variables measured by a single smart meter, to reduce the dimension of transmitted data. Subsequently, for each of the chosen dimensions, sparsity is evaluated to compress them temporally. To the best of our knowledge, there is no existing work that exploits the cross-correlation between different variable streams of smart meter to perform multivariate data compression.

The key contributions of this work are as follows:

(i) A novel two-step adaptive compression of multivariate smart meter data is proposed which exploits the temporal correlation in each individual time-series data as well as the cross-correlation among different time series data variables. The proposed technique is shown to achieve up to 36% improvement in bandwidth saving compared to the existing closest competitive approach in [20] without compromising on quality of service (QoS) of the smart metering application.

(ii) A novel method for characterization of multivariate smart meter data using Multivariate Normal Autoregressive Integrated Moving Average (MVN-ARIMA) model is proposed. The proposed characterization model is further used to validate the reconstruction accuracy of adaptive multivariate compression technique proposed in (i).

(iii) Empirical optimization of the operating parameters, namely, batch size, sparsity, and minimum required dimensions is investigated with respect to the data variability for low-complexity online execution of the proposed adaptive compression algorithm.

(iv) Reconstruction accuracy of the proposed adaptive multivariate compression scheme for smart meter data is also validated using a real-life application involving the utility as well as the consumer. It is demonstrated that incorporation of the proposed technique does not have any adverse effect on the applications which use data collected by commercial smart meters.

(v) The proposed algorithm is implemented in real smart meters deployed across the university campus, each with a different data signature. These real implementation performance results show that the proposed algorithm is resource-efficient and computationally inexpensive.

C. Paper organization

The paper layout is organized as follows. Section II briefly gives an overview of the basic techniques used in this work.

Section III describes the proposed algorithm. Modeling of the joint distribution of the multivariate smart meter data is presented in Section IV. Section V contains the numerically obtained compression results on different datasets. In Section VI, real system implementation of the algorithm is described. Section VII concludes the paper.

II. Preliminaries

The proposed algorithm for multivariate data compression uses a two-step mechanism. In the first step, dimensionality of data is reduced by applying Principal Component Analysis (PCA). Exploiting the cross-correlation between different variables in smart meter data and using it for dimensionality reduction is one of the salient contributions of this work. To this end, it is of prime importance to choose the right tool for this task. Other than PCA, the most popular dimensionality reduction techniques available in the literature are: Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), Multidimensional Scaling (MDS), Locally Linear Embedding (LLE), Isometric Mapping (Isomap). Through a comparative study among these competitive techniques, it was inferred in [21] and [22] that, in terms of accuracy as well as processing speed, PCA offers the best trade-off. In the second step, each of those selected components is further compressed temporally using Compressive Sampling (CS). These techniques are briefly discussed next.

A. Principal Component Analysis

PCA [23] is a dimensionality reduction technique which transforms an $n-$dimensional data to p dimensions. With strongly correlated features in input data, $p \ll n$, and, in the transformed space most of the variance of the entire data is preserved in only a few dimensions. PCA involves computing eigen vector $-$ eigen value combinations from covariance matrix of the input data $X \in \mathbb{R}^{m \times n}$, where n is the number of features or variables and m is the number of samples taken from each feature. The covariance matrix is defined as: $Cov(X) = E[(X - E(X))(X - E(X))^T]$. The eigen vectors act as orthogonal basis in the transformed space. Using singular value decomposition, a matrix X is factorized as:

$$X = USV^T, \tag{1}$$

where $S \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the singular values, i.e., square root of the eigen values of X as the diagonal entries. The left and right singular vectors of X are stored in columns of $U \in \mathbb{R}^{m \times n}$ and $V \in \mathbb{R}^{n \times n}$. Hence the projected data in the transformed space, $Y \in \mathbb{R}^{m \times n}$ is expressed as:

$$Y = XV = (USV^T)V = US. \tag{2}$$

V being invertible, X can be easily reconstructed. It can be shown that the matrix U of singular vectors of the data matrix X is same as the matrix of eigen vectors of the $Cov(X)$ [24]. Contrary to various other linear transforms where fixed orthogonal basis vectors are used, the basis vectors in PCA are data-dependent. After PCA is performed, the principal components are uncorrelated and arranged in their decreasing order of variance. A few components are chosen so that most of the variance of data is preserved, leading to compression of data while keeping the information loss at a minimum. This is further elaborated in Section III-A.

B. Compressive Sampling

CS [25] is a technique that can directly acquire a condensed representation without losing much on the information content. If an $m-$dimensional signal y has to be monitored using a sensor, the best case would be to get all the m values. But in constrained environments, only a compressed version $\tilde{y}$ with $\tilde{m} < m$ samples might be available. $\tilde{y}$ is expressed as: $\tilde{y} = \phi y$, where $\tilde{y} \in \mathbb{R}^{\tilde{m}}$ and $y \in \mathbb{R}^m$; $\phi \in \mathbb{R}^{\tilde{m} \times m}$ is called the sampling matrix. The notion of CS says, it is possible to recover y from $\tilde{y}$ if there exists an orthonormal basis ψ that transforms the signal y into a sparse domain. Consequently, y can be expressed as: $y = \psi\alpha$, where the vector of coefficients α corresponding to the sparsifying basis matrix $\psi \in \mathbb{R}^{m \times m}$ is sparse in the sense that, if the magnitudes of α are sorted, they decay considerably fast. Thus,

$$\tilde{y} = \phi\psi\alpha = A\alpha, \tag{3}$$

where $A = \phi\psi$. To recover y from the measurements $\tilde{y}$, the sparsest possible solution for α should be found that satisfies (3). The recovery problem of interest is framed as:

$$P_0 : \quad \min_{\alpha} \quad \|\alpha\|_0 \quad \text{subject to} \quad \tilde{y} = A\alpha. \tag{4}$$

Although l_0-minimization in (4) is an NP-hard problem, various approximation algorithms are proposed in the literature to find the solution to l_0 optimization problem with reasonably low computation complexity. These can broadly be grouped into greedy and relaxation methods. Greedy pursuit algorithms like Orthogonal Matching Pursuit (OMP) [26] have been found to faster than approximation algorithms such as Basis Pursuit [27], which can be handled by linear programming (LP) solvers. In this work, Subspace Pursuit (SP) [28] has been used for reconstruction in CS which is claimed to be nearly as accurate as LP methods while its computational complexity is fairly low, on the same order as that of OMP.

III. Proposed Algorithm

This section presents the proposed Adaptive Multivariate Data Compression (AMDC) scheme for smart meter data.

A. Adaptive multivariate data compression (AMDC)

Multivariate compression of smart meter data involves decorrelating the input variables, having high cross-correlation, using PCA, and then exploiting the sparsity in each of those decorrelated streams to achieve further compression using CS. At the transmitter side, first, PCA is applied on the multivariate data in a batch. This process obtains the principal components by using eigen value-eigen vector combination from correlation matrix of the data. Starting with the eigen vector corresponding to the highest eigen value, the algorithm identifies the required number of principal components that is sufficient for reconstructing the original data at the receiver side. As in PCA projected space most of the variance of the data is preserved in a few dimensions, only those many principal components are considered for an acceptably accurate reconstruction of data at the receiver. Let $X \in \mathbb{R}^{m \times n}$ be the input data matrix, where n is the number of dimensions, i.e., variables measured by the smart meter, and m is the number of samples taken from each variable. PCA operation on X returns the orthogonal basis vectors in V and principal components in Y, as shown in (2).

Let p be the number of principal components that preserve more than a certain predefined threshold percentage of the total variance, such that $p < n$. Then, only those p principal components and the corresponding basis vectors are required for an acceptable reconstruction at the receiver. More specifically, to retain ξ % variance, p is chosen such that the following relation holds:

$$p = \left\{ \min\{\kappa\} : \frac{\sum_{i=1}^{\kappa} s_{ii}}{\sum_{i=1}^{n} s_{ii}} \geqslant \frac{\xi}{100}, 1 \leqslant \kappa \leqslant n \right\}, \qquad (5)$$

where s_{ii} is the element of diagonal matrix S at the i^{th} row and i^{th} column. To find the value of p, the matrix S from (1) is used. After the value of p satisfying (5) is chosen, the reduced matrix $V_{red} \in \mathbb{R}^{n \times p}$, which consists of only the first p columns of V, is obtained. If ξ is close to 100, most of the variance in data is captured in the p dimensions, resulting in less error in reconstruction. In this work $\xi = 99$ is considered to ensure faithful reconstruction of all n dimensions of data from p principal components.

The projection of X along those p orthonormal dimensions are given by: $Y_{red} = XV_{red}$. Next, every column of $Y_{red} \in \mathbb{R}^{m \times p}$ is sent to the CS block for temporal compression. Sparsity k of each column is estimated by the number of DFT coefficients that preserve at least 99.99% energy of all the samples.

B. Reconstruction of data at the receiver

Let y_u be the u^{th} column of Y_{red}, and after it is passed through the CS block, let $\tilde{y}_u$ be the temporally compressed output. At the cloud or the data aggregator, $\tilde{y}_u$ vectors are received, from which the corresponding y_u is estimated as $\hat{y}_u$. Thus, the minimization problem in (4) is reframed as:

$$\begin{aligned} P_1 : \min_{\alpha_u} \quad & \|\alpha_u\|_0 \\ \text{subject to} \quad & \tilde{y}_u = A\alpha_u \quad \forall u \in \{1, 2, \cdots p\}. \end{aligned} \qquad (6)$$

The p optimization problems in (6) corresponding to the p principal components are to be solved independently. Since each column represents a smart meter variable projected in transformed space, they may have different sparsity. To adapt to the temporal dynamics, sparsity of each stream is computed at the run time. The solution to the u^{th} problem in (6) is a sparse vector $\bar{\alpha}_u$. From all these $\bar{\alpha}_u$ vectors, first $\hat{y}_u$ vectors are estimated, and then from them, $\widehat{Y}_{red}$ is constructed. Thus $\widehat{Y}_{red}$ is an estimate of Y_{red}. SP algorithm is used for this CS recovery because of its decent reconstruction performance and fast computation. To make sure that the matrix A satisfies restrictive isometry property, the matrices ϕ and ψ have been taken as identity matrix and FFT of identity matrix, respectively [25]. Now, as part of the PCA reconstruction the actual batch data is recovered back from $\widehat{Y}_{red}$ as: $\widehat{X} = \widehat{Y}_{red} {V_{red}}^T$.

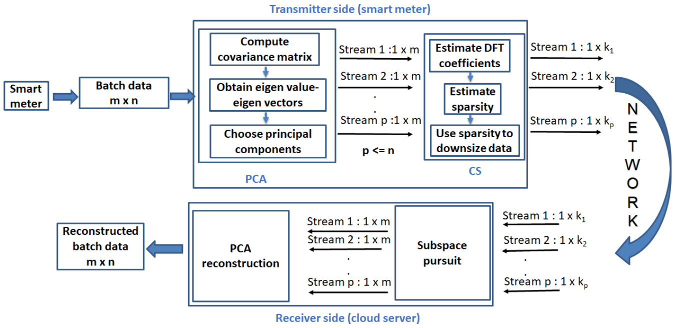


Fig. 2: Flow diagram of the proposed AMDC algorithm.

TABLE I: Components of time complexity

	PCA	**Compressive Sampling**
Transmitter Side	$O(n^3) + O(p)$	$O(mpb^3)$
Receiver Side	$O(mpn)$	$O(p(k(b + k^2)log(k)))$

The proposed algorithm is adaptive in the sense that both sparsity in CS (which decides the number of transmitted samples) and number of principal components in PCA (which decides the number of transmitted variables) are computed during the run time. Hence both of them are adaptively decided based on data variability within a batch. Flow of the proposed algorithm is depicted using a block diagram in Fig. 2.

C. Complexity of the proposed algorithm

The order of complexity is computed by finding the complexity of one iteration and the required number of iterations. Computation complexity of the proposed algorithm is divided primarily into four components as shown in Table I along with the respective orders of complexity, where n is the dimension of original data (number of variables), m is the number of samples of each variable in a batch, p is the number of principal components in each batch, b is the batch size for CS, and k is the sparsity in a batch averaged across all principal components. For a particular dataset, n, m, b are configured offline to certain fixed values before online execution. Hence, combining all four components, complexity of the algorithm, when executed on a meter, is given by: $O(p)$. *Thus, run-time computation complexity of AMDC is linear in the number of principal components in a batch.*

IV. Modeling Multivariate Data

The joint distribution of multivariate smart meter data is modeled in this section which is used in Section V-F to characterize the actual data as well as the reconstructed data. The objective is to evaluate the quality of compression or loss in information.

Time series data in general exhibits temporal correlation. Hence, to model the joint distribution of multiple time series data, first, the individual time series is characterized using signal models, such as AR, ARMA, and ARIMA. Afterward, the

TABLE II: Datasets used for performance evaluation

Meter	Installation Site	Sampling Interval
Meter-01	A computing research lab	30 sec
Meter-02	Power distribution substation of a multi-storey building	30 sec
Meter-03	Mechanized kitchen of a student dormitory	30 sec
Meter-04	Telephone switch room of the university	30 sec
iAWE	A household location	1 sec

inter-dependence is captured by modeling the joint distribution of the decorrelated residuals [29], [30].

In our context, the multivariate smart meter data is characterized using MVN-ARIMA modeling. For modeling of joint distribution, five primary variables, namely, apparent power, power, current, voltage and frequency, which are common amongst all the datasets, are considered in this work. The temporal correlation of all the individual variables are captured using ARIMA models [31] as it is suitable for non-stationary data. An ARIMA(p, d, q) model, when applied on a non-stationary time series data, does a differencing of order d to stationarize before fitting it to ARMA(p, q). Following this, the inter-dependence among the residuals is modeled by MVN distribution. Multivariate normality is tested using squared Mahalanobis distance between an MVN distribution and a random point picked from that distribution [32]. In a way, it generalizes in multiple dimensions the concept of measuring how many standard deviations away a randomly chosen point P is from the mean of a distribution D. For a multivariate normal distribution with mean μ and covariance matrix Σ, the squared Mahalanobis distance is defined as:

$$MD^2(x, \mu) = (x - \mu)^T \Sigma^{-1} (x - \mu). \tag{7}$$

It can be shown that the squared Mahalanobis distance MD^2 of a normally distributed data follow χ^2 distribution. If a random data point is picked from a multivariate normal distributed data, its MD^2 will be smaller than or equal to a critical value with probability p [24]. From the χ^2 table of critical values, it is seen that, for 5 degrees of freedom, the critical value is 15.09 for $p = 0.01$, i.e. at most 1% samples can have $MD^2 \geqslant 15.09$, and those are outliers. The data characterization described here will further be used in Section V-F to validate the performance of the proposed technique.

V. Results

In this section extensive analysis of the performance of the proposed algorithm is done by employing it on multiple smart meter datasets. Subsequently, the performance is compared with that of (i) the current practice of transmitting original meter data and (ii) a closest competitive technique for smart meter data compression recently reported in [20].

A. Datasets used for evaluation

The proposed algorithm is applied to multivariate data from four real smart meters (Meter-01, Meter-02, Meter-03, Meter-04) installed at different locations in the university campus where the sampling interval is set to 30 seconds. To test performance of the proposed algorithm on finer granular data, it is also applied on an open dataset namely iAWE [33], which provides a sampled record of the domestic load of a household at 1 second intervals. iAWE has been chosen over other open smart meter datasets available on the web as it includes data corresponding to more than one variable. All the installation sites along with the sampling interval of the corresponding meters are listed in Table II. Each of these locations have multiple appliances with unique power signatures; cross-correlation information of multiple variables are listed in Table III.

TABLE III: Cross-correlation between different variables

Variables	Correlation Coefficient	Category
AP, Current, PF, Power	0.98-0.99	HIGH
Energy, Power Interrupt	0.75	MEDIUM
Frequency, Voltage	0.24	LOW

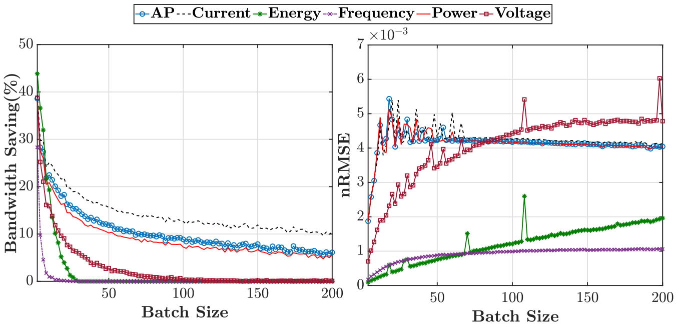


Fig. 3: CS block compression performance versus batch size.

B. Performance indices

The metrics used in performance evaluation of the algorithm are described here. All the variables used here carry the same meaning as mentioned in Section III-C unless explicitly mentioned.

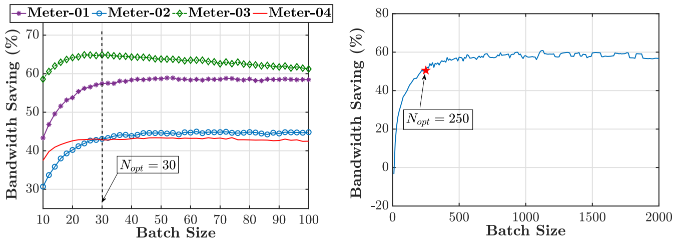
(a) To quantify the gain achieved by the algorithm, *percentage bandwidth saving* is used, which is computed as: $\frac{n\cdot m-(p\cdot k+n\cdot p)}{n\cdot m} \times 100$.

(b) Error induced in the process is measured in terms of *normalized root mean squared error (nRMSE)* [34] of reconstruction for each variable which is calculated as: $\frac{1}{\bar{x}}\sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{x}_i)^2}$,

where x_i and $\hat{x}_i$ are actual and reconstructed data of the variable under consideration; $\bar{x}$ is the maximum of actual data of the same variable, which is used as the normalization factor.

C. Estimation of optimum batch size

While implementing the proposed algorithm on real smart meters, data are fed in batches. Hence estimation of optimum batch size is of interest. In Fig. 3, the variation of bandwidth saving and nRMSE with increasing batch size are shown for different variables of one of the meters (Meter-01). It is observed that for all the variables, as batch size increases, the nRMSE of reconstruction is non-decreasing, whereas the corresponding bandwidth saving decreases. Hence the batch size of CS is kept at the minimum possible value, i.e., two. The performance with the other meters exhibit similar trends, and hence they are not shown here.



(a) 30 sec sampled data (b) 1 sec sampled data

Fig. 4: Batch size estimation for PCA.

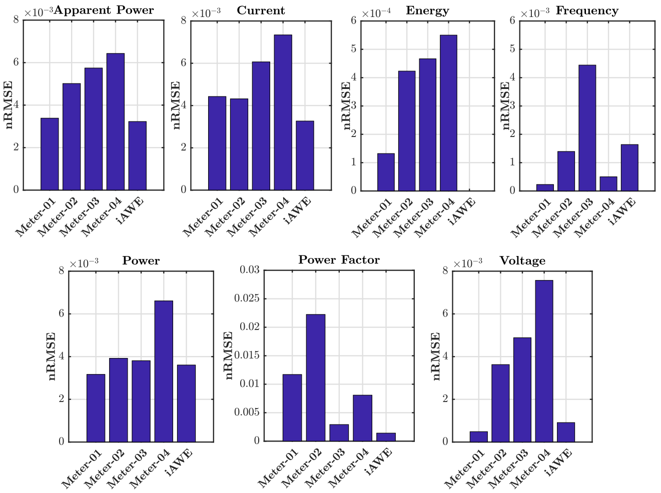


Fig. 5: Reconstruction nRMSE of AMDC for all variables.

In Fig. 4a variation of bandwidth saving with increasing batch size for PCA is presented. It is observed that, for most of the meters with 30 seconds sampling interval, there is no significant variation in bandwidth saving beyond batch size of 30 samples. Also, from the literature, it is noted that the smart meters in domestic or industrial setting typically report accumulated data once every 15 minutes [35]. Hence to make the data collection at the cloud as real-time as possible the batch size is limited to 30 samples which accounts for data over 15 minutes for the meters sampling once in every 30 seconds. Similarly, for the iAWE dataset with 1 second sampling interval, as shown in Fig. 4b, bandwidth saving saturation was observed at batch size of 250 samples, which is close to 4 minutes' data.

Remark 1. *Hence the optimum batch size N_{opt} is chosen as 30 samples for the data sampled at $\frac{1}{30}$ Hz frequency and 250 samples for the data sampled at 1 Hz.*

D. Reconstruction performance evaluation

For the faithful reconstruction of data, nRMSE induced at the optimum batch size should not cross the allowable threshold. This error threshold varies with the kind of application of the smart meter data. In this study, nRMSE below 0.2 is considered to be acceptable, as suggested in [36]. Fig. 5 exhibits nRMSE of reconstruction for all variables across all meters. It is observed that for every case the nRMSE is well below the acceptable threshold 0.2, in fact on the order of 10^{-3}. Intuitively, bandwidth saving decreases with increased reconstruction accuracy, i.e., with reduced nRMSE threshold. Therefore, a trade-off exists between achievable bandwidth saving and induced nRMSE.

A comparison of actual data streams and reconstructed streams at the receiver after decompression are presented respectively in Fig. 6 and Fig. 7 for data sampled at 1 second (iAWE) and 30 seconds (Meter-03), respectively. It is observed that the reconstructed data almost overlap with the actual data in all the cases, owing to high reconstruction accuracy. It is notable that, if instead an overall higher order of nRMSE threshold is chosen, the bandwidth saving with the proposed AMDC is much higher. However that is at the cost of significantly higher mismatch in the reconstructed data, which may be unacceptable for maintaining the QoS.

E. Comparison with adaptive compressive sampling (ACS)

Performance of the proposed AMDC algorithm is now compared with that of a recently proposed ACS algorithm [20], which claims to perform better than a lossless compression method in the presence of communication errors. Since ACS was intended for compression of single variable only, each smart meter variable is independently compressed using ACS and stacked together for evaluation of bandwidth saving and reconstruction error. In Table IV, nRMSE comparison across all locations are shown. From Table IV it may be noted that the order of nRMSE induced in both the algorithms is same.

Fig. 8 shows the offered bandwidth saving with AMDC and ACS with respect to non-compressed transmission. It is observed that AMDC achieves 10% to 36% gain in bandwidth saving over ACS for the meter data sampled at $\frac{1}{30}$ Hz, whereas it offers 11% gain in case of data with sampling frequency 1 Hz. In both cases, error induced with AMDC in terms of nRMSE are still in the same order as that of ACS, thereby verifying fairness of relative bandwidth saving performance.

Remark 2. *The bandwidth saving is highly dependent on data variability and hence it varies with installation locations.*

F. Validation of reconstruction using distribution modeling

To further validate the reconstruction performance of the proposed AMDC algorithm, multivariate distribution modeling discussed in Section IV is used. First, the variables from the actual data are fitted individually to ARIMA models. The best-fit model is found using auto.arima function of forecast package in R [37] where Akaike Information Criterion (AIC) is used to compare models, and the order of differencing d is computed based on Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. The parameters (p, d, q) corresponding to the best-fit ARIMA models of all the individual time series are listed in Table V for all the datasets. Next, the joint distribution of residuals from best-fit models of the actual data is modeled using multivariate normal distribution. The goodness of fit of the multivariate distribution is validated by Mahalanobis distance between the actual data and the modeled distribution. Similarly, the variables from the reconstructed data are fitted

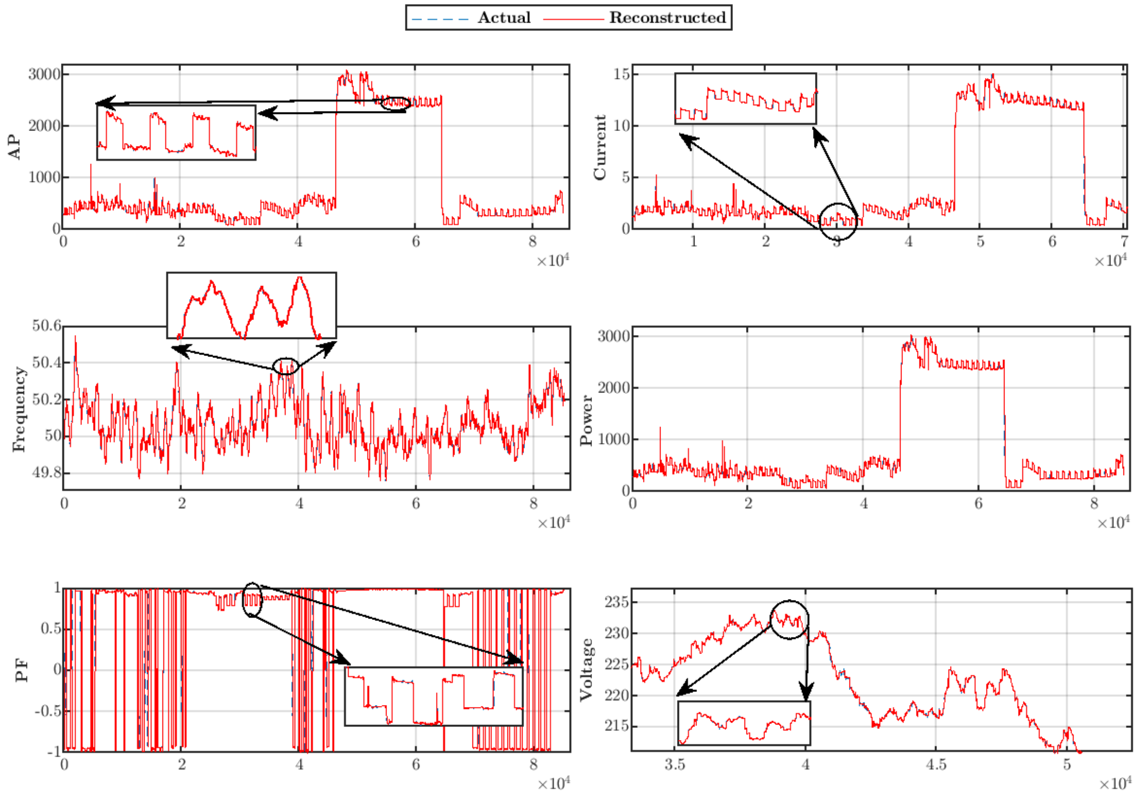


Fig. 6: Reconstruction of data sampled at 1 Hz (iAWE).

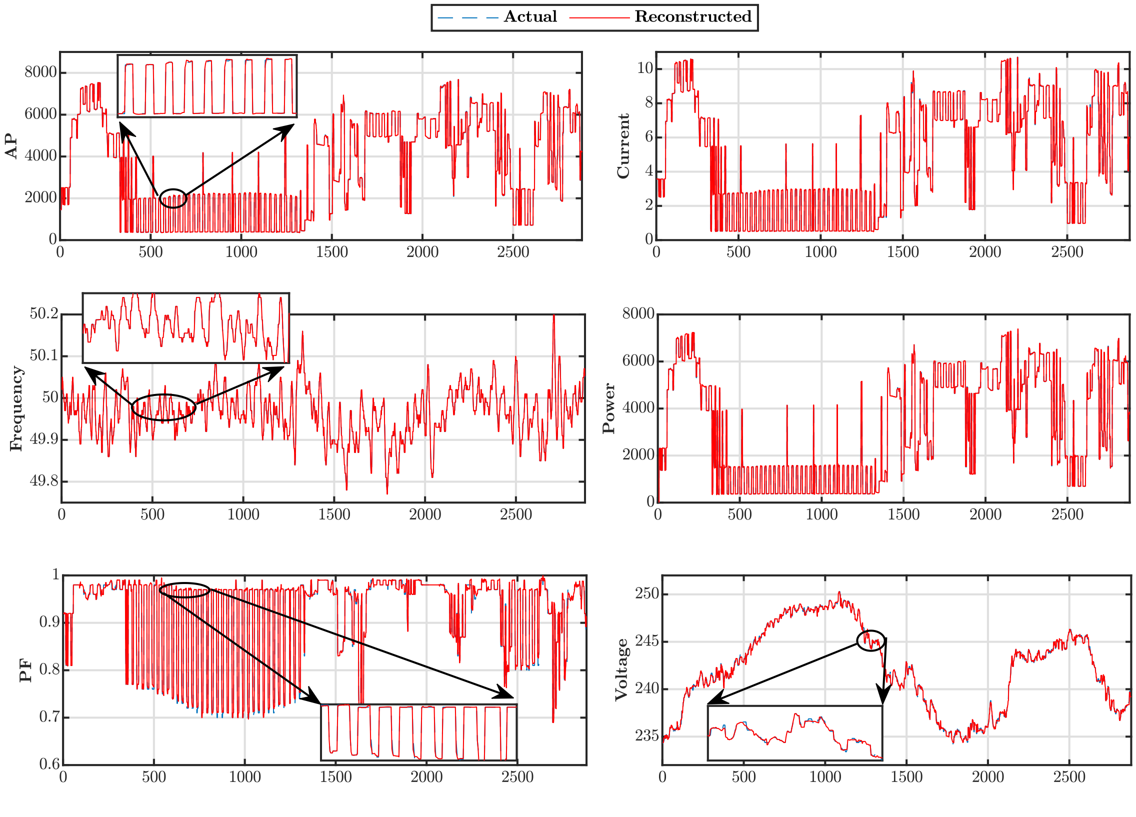


Fig. 7: Reconstruction of data sampled at $\frac{1}{30}$ Hz (Meter-03).

TABLE IV: nRMSE values for different meters and different variables

		AP	Current	Energy	Frequency	Power	PF	Voltage
Meter-01	**AMDC**	0.0034	0.0044	0.0001	0.0002	0.0031	0.0117	0.0005
	ACS	0.0034	0.0041	0.0001	0.0002	0.0030	0.0069	0.0005
Meter-02	**AMDC**	0.0050	0.0043	0.0004	0.0014	0.0039	0.0222	0.0036
	ACS	0.0057	0.0036	0.0001	0.0011	0.0036	0.0174	0.0015
Meter-03	**AMDC**	0.0058	0.0060	0.0004	0.0044	0.0038	0.0029	0.0049
	ACS	0.0051	0.0061	0.0001	0.0044	0.0038	0.0029	0.0023
Meter-04	**AMDC**	0.0064	0.0073	0.0005	0.0005	0.0066	0.0081	0.0076
	ACS	0.0060	0.0074	0.0001	0.0005	0.0063	0.0083	0.0073
iAWE	**AMDC**	0.0032	0.0033	N/A	0.0016	0.0036	0.0014	0.0009
	ACS	0.0033	0.0032	N/A	0.0013	0.0029	0.0012	0.0009

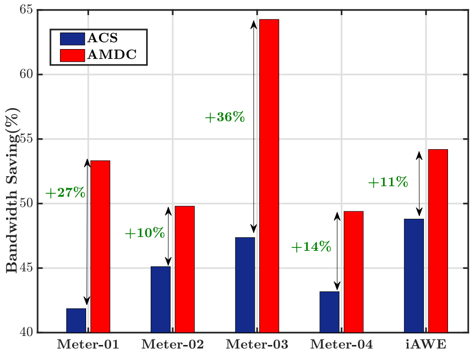


Fig. 8: Bandwidth saving comparison for different meters.

TABLE V: ARIMA modeling parameters for best fit

	AP	Current	Frequency	Power	Voltage
Meter-01	(1,1,1)	(1,1,1)	(3,1,2)	(1,1,2)	(5,1,4)
Meter-02	(5,1,5)	(5,1,5)	(3,1,3)	(5,1,5)	(1,1,3)
Meter-03	(5,1,5)	(3,1,5)	(3,1,3)	(4,1,2)	(3,1,3)
Meter-04	(5,1,5)	(1,1,3)	(3,1,2)	(5,1,5)	(4,1,4)
iAWE	(1,1,1)	(2,1,0)	(5,1,3)	(0,1,1)	(2,1,1)

TABLE VI: Outliers in multivariate normality

	Meter-01	Meter-02	Meter-03	Meter-04	iAWE
Actual	0.0099	0.0096	0.0096	0.0098	0.0095
Reconstructed	0.0101	0.0099	0.0105	0.0101	0.0097

to the same ARIMA models with parameters listed in Table V, which were found in the previous step with the actual dataset. This is followed by modeling the joint distribution of the residuals using a multivariate normal distribution. Subsequently, reconstruction performance is validated by Mahalanobis distance between the residuals of reconstructed variables and the modeled multivariate distribution. The fraction of outliers among random sample points drawn from the fitted multivariate distributions for both the actual and reconstructed data is shown in Table VI. It can be noted from the table that, in all cases, the number of outliers is limited to 1%. It may be recalled that, this validation is done with batch size equal to N_{opt} of the corresponding dataset.

Remark 3. *Multivariate smart meter data characteristics are preserved in the compression process; thus, the bandwidth saving is attained with minimum loss of information.*

G. Validation of reconstruction using real-life application

TABLE VII: Comparison of monthly bill calculated with actual versus reconstructed data

	Meter-01	Meter-02	Meter-03	Meter-04
Actual	\$ 57.3989	\$ 4921.9	\$ 205.0368	\$ 105.5123
Reconstructed	\$ 57.3989	\$ 4921.9	\$ 205.0273	\$ 105.5342

In the previous subsection reconstruction accuracy of the proposed AMDC algorithm was validated using distribution modeling. We now demonstrate through a practical example that the proposed technique can be implemented in commercial smart meters without affecting the real-life user experience. One most pertinent application that affects both utility companies and consumers in the power sector is electricity billing. Utility companies use the month-long data collected from the smart meters installed at household/industrial locations for monthly billing. Table VII presents a comparison of monthly bill calculated using the actual data and that after applying the proposed AMDC algorithm at the smart meters installed in the campus, assuming each of them belong to different type of consumers. Monthly bill is calculated using the power data from all meters and then following the bill calculation logic of a local government-controlled power service provider. Final billed amounts in USD are compared in the two cases. The iAWE dataset is not considered for billing, as it does not have month-long data. From Table VII, it can be noted that using AMDC data instead of the actual data practically does not affect the total monthly bill calculated from any of the meters. In fact, use of reconstructed data leads to a maximum deviation of 0.02%, which is quite acceptable. This result strengthens the claim that, while AMDC reduces the bandwidth requirement, loss of information is practically negligible.

Remark 4. *The proposed technique does not have any adverse effect in terms of monetary loss or compromise of QoS. While at the same time, it leads to effective resource optimization for the service provider by not loading the network unnecessarily.*

VI. Implementation in a real smart meter

In this section, implementation of the proposed AMDC algorithm on real smart metering setup deployed in the university campus is presented.

A. Hardware setup

The metering hardware setup is shown in Fig. 9. It comprises of an ENERSOL MFR2810 energy meter which records multiple data variables at a sampling interval of 30 seconds.

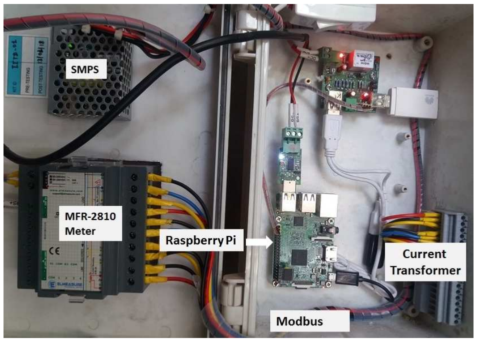
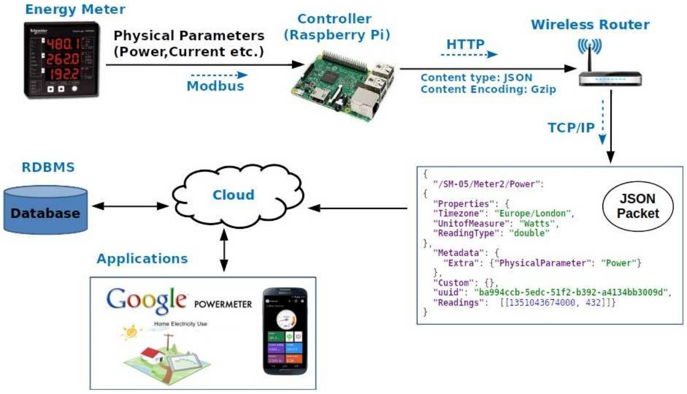


Fig. 9: Smart metering IoT setup installed in the campus.



```
{
  "/SM-05/Meter2/Power":
{
  "Properties": {
  "Timezone": "Europe/London",
  "UnitofMeasure": "Watts",
  "ReadingType": "double"
},
  "Metadata": {
    "Extra": {"PhysicalParameter": "Power"}
  },
  "Custom": {},
  "uuid": "ba994ccb-5edc-51f2-b392-a4134bb3009d",
  "Readings":  [[1351043674000, 432]]}
}
```

Fig. 10: Architecture of the smart metering system.

The meter is connected to a Raspberry Pi (R-Pi) 3 board with 1.2 GHz 64-bit quad-core ARM v8 processor and 1 GB RAM, which acts as the meter controller. After recording the multivariate data, all of them are stacked together in JavaScript Object Notation (JSON) format and then zipped using gzip compression technique to transmit over TCP/HTTP link to the storage cloud through a TP-LINK TL - MR3020 wireless router. Simple Measurement and Actuation Profile (sMAP) 2.0 [38] protocol is used to read different time series data from the meter in a simple and configurable setting, and to publish it on the web or a central cloud. It uses Modbus protocol to read data from energy meter. The whole system architecture used for implementation is presented in Fig. 10. The proposed AMDC algorithm is implemented in Python 3.6 and configured to operate in R-Pi controller in the meter between the data collection and data transmission blocks. The frequency of reporting compressed data to the cloud is decided based on the optimum batch size for a particular meter, which is governed by the location-specific meter data pattern.

B. Packet traffic comparison

Bandwidth saving with compressed batch transmission over the network is obtained by measuring the size of the link layer packets sent over TCP/HTTP link from the client (meter) to the server (cloud). This evaluation has been conducted over a period of one month and averaged to obtain per-day traffic due to this data communication. Packet size is acquired by sniffing packets over the connection between the client and the server access point using Wireshark software, which is an open-source packet analyzer.

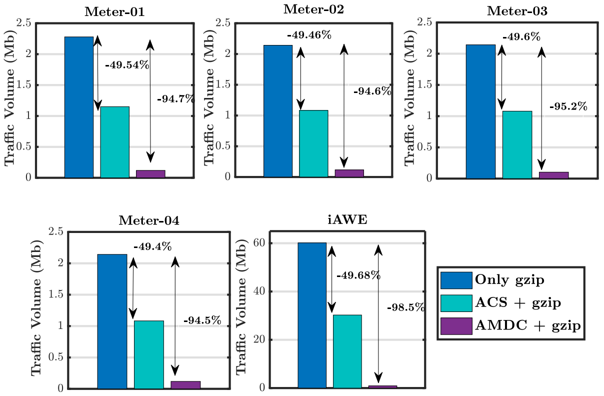


Fig. 11: Relative packet traffic volume generated per day.

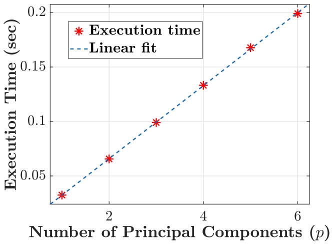


Fig. 12: Linear fit of time complexity of AMDC.

It may be noted that, in the current practice the controller does not use any intelligence to decide on which variables or how many samples of each variable are to be sent. Rather, it sends all of them over the wireless access network. The commercial meters installed in the university campus inherently employ gzip format [39] for loss-less data compression before transmission. The underlying mechanism of gzip is based on DEFLATE [40], which is a combination of Huffman coding and LZ77 [41]. Hence gzip is insensitive to the data dynamics and cross-correlations, which are taken into consideration in AMDC. In Fig. 11, the traffic volume generated without data-driven compression (i.e., with gzip only, as done in a standard smart meter) is compared with the traffic generated by the proposed AMDC algorithm followed by gzip as well as by employing ACS [20] followed by gzip. It is observed that, compared to the conventional scheme in standard smart meters, ACS reduces the traffic by about 50%, whereas AMDC is able to reduce it by about 95% for $\frac{1}{30}$ Hz data and 98.5% for 1 Hz data. It may be noted that the percentage bandwidth savings shown in Fig. 8 and those in Fig. 11 are different. This is because, unlike in Fig. 11, the results in Fig. 8 did not account for the network overhead; bandwidth saving was evaluated only based on data reduction after compression.

C. Computation time complexity

The proposed algorithm is executed on a real smart meter, and its per-batch execution time is noted to be 0.108 second for a batch size of 30 samples, collected over 15 minutes at a sampling rate of $\frac{1}{30}$ Hz. Thus run time overhead of the proposed algorithm is negligible compared to the data collection time granularity. Further, run time corresponding to different number of principal components (p) is recorded. For a given p, execution time is averaged over different batches. Fig. 12 shows the variation of execution time with p. Curve-fitting is used to study the nature of variation of execution time with p. The parameters of curve fitting are: Execution Time, $\tau(p) = \alpha_1 \cdot p + \alpha_2$; coefficients: $\alpha_1 = 0.033$, $\alpha_2 = -0.001$; goodness of fit: $R^2 = 0.9999$, RMSE $= 0.00085$.

Remark 5. *It is observed that run time complexity is linear in the number of principal components in a batch, p, which also validates the complexity analysis in Section III-C.*

VII. Conclusion

In this work a novel two-step compression scheme called AMDC has been proposed for multivariate smart meter data. The proposed AMDC algorithm exploits cross-correlation between different variables to reduce the dimensionality of input data. Subsequently, it exploits temporal correlation in the individual streams to increase the bandwidth saving without any significant information loss. Through exhaustive testing on real smart meter data it has been demonstrated that AMDC can reduce the bandwidth requirement for transmission of multivariate smart meter data over actual communication network by up to 98.5% while ensuring faithful reconstruction of data in the aggregator within an acceptable error threshold. Further, performance of the proposed multivariate compression algorithm has been shown to offer about 36% reduction of bandwidth requirement with respect to a nearest individual-stream based ACS scheme. It has been noted that, different smart meter data can have widely differing data dynamics, which is accounted in the proposed algorithm by online parameter tuning according to the individual smart meter data pattern. The proposed algorithm has been implemented in a real smart meter and its execution time overhead has been shown to be very small in comparison with the typical interval between two consecutive data batch transmissions. Considering a real-life application that uses smart meter data, it has been validated that implementation of AMDC leads to significant reduction in network resource requirement by the service provider while the QoS of the electricity consumer remains unaffected.

References

[1] M. Aiello and G. A. Pagani, "The smart grid's data generating potentials," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Warsaw, Poland, Sep. 2014, pp. 9–16.

[2] L. Wen, K. Zhou, S. Yang, and L. Li, "Compression of smart meter big data: A survey," *Renew. Sustain. Energy Rev.*, vol. 91, pp. 59 – 69, Aug. 2018.

[3] S. Tripathi and S. De, "Data-driven optimizations in IoT: A new frontier of challenges and opportunities," *Springer CSI Trans. ICT*, vol. 7, no. 1, pp. 35–43, Mar. 2019.

[4] M. P. Tcheou, L. Lovisolo, M. V. Ribeiro, E. A. B. da Silva, M. A. M. Rodrigues, J. M. T. Romano, and P. S. R. Diniz, "The compression of electric signal waveforms for smart grids: State of the art and future trends," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 291–302, Jan. 2014.

[5] S. Tripathi and S. De, "Dynamic prediction of powerline frequency for wide area monitoring and control," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 2837–2846, Jul. 2018.

[6] Y. Wang, Q. Chen, C. Kang, Q. Xia, and M. Luo, "Sparse and redundant representation-based smart meter data compression and pattern extraction," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2142–2151, May 2017.

[7] J. C. S. de Souza, T. M. L. Assis, and B. C. Pal, "Data compression in smart distribution systems via singular value decomposition," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 275–284, Jan. 2017.

[8] X. Tong, C. Kang, and Q. Xia, "Smart metering load data compression based on load feature identification," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2414–2422, Sep. 2016.

[9] M. Ringwelski, C. Renner, A. Reinhardt, A. Weigel, and V. Turau, "The hitchhiker's guide to choosing the compression algorithm for your smart meter data," in *Proc. IEEE Intl. Energy Conf. Exhibit.*, Florence, Italy, Sep. 2012, pp. 935–940.

[10] A. Abuadbba, I. Khalil, and X. Yu, "Gaussian approximation-based lossless compression of smart meter readings," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5047–5056, Sep. 2018.

[11] A. Unterweger and D. Engel, "Resumable load data compression in smart grids," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 919–929, Mar. 2015.

[12] M. Zeinali and J. S. Thompson, "Impact of compression and aggregation in wireless networks on smart meter data," in *Proc. IEEE Wksp. Sig. Process. Advances in Wireless Commun.*, Edinburgh, UK, Jul. 2016, pp. 1–5.

[13] J. Z. Kolter and M. J. Johnson, "REDD: A Public Data Set for Energy Disaggregation Research," in *Proc. Wksp. Data Min. Appl. Sustain*, San Diego, CA, USA, Sep. 2011, pp. 1–6.

[14] F. Eichinger, P. Efros, S. Karnouskos, and K. Böhm, "A time-series compression technique and its application to the smart grid," *VLDB J.*, vol. 24, no. 2, pp. 193–218, Apr. 2015.

[15] C. Yu, P. Mirowski, and T. K. Ho, "A Sparse Coding Approach to Household Electricity Demand Forecasting in Smart Grids," *IEEE Trans. on Smart Grid*, vol. 8, no. 2, pp. 738–748, Mar. 2017.

[16] A. Notaristefano, G. Chicco, and F. Piglione, "Data size reduction with symbolic aggregate approximation for electrical load pattern grouping," *IET Gene. Transm. Distrib.*, vol. 7, no. 2, pp. 108–117, Feb. 2013.

[17] Y. Lee, E. Hwang, and J. Choi, "A Unified Approach for Compression and Authentication of Smart Meter Reading in AMI," *IEEE Access*, vol. 7, pp. 34 383–34 394, 2019.

[18] C. Li and R. Zheng, "Load Data Compression Based on Integrated Neural Network Model," in *Proc. Chin. Control Decision Conf.*, Nanchang, China, Jun. 2019, pp. 6203–6209.

[19] A. Joshi, A. Yerudkar, C. D. Vecchio, and L. Glielmo, "Storage Constrained Smart Meter Sensing using Semi-Tensor Product," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Bari, Italy, Oct. 2019, pp. 51–56.

[20] S. Tripathi and S. De, "An efficient data characterization and reduction scheme for smart metering infrastructure," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4300–4308, Oct. 2018.

[21] T. Lan, D. Erdogmus, L. Black, and J. Van Santen, "A comparison of different dimensionality reduction and feature selection methods for single trial ERP detection," in *Proc. IEEE Engg. in Med. and Bio.*, Aug. 2010, pp. 6329–6332.

[22] J. Zubova, O. Kurasova, and M. Liutvinavičius, "Dimensionality reduction methods: the comparison of speed and accuracy," *Inf. Technol. Control*, vol. 47, no. 1, pp. 151–160, 2018.

[23] I. Jolliffe, *Principal Component Analysis*. 2nd ed. New York, NY, USA: Springer-Verlag, 2002.

[24] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*. 6th ed. Upper Saddle River, NJ: Prentice-Hall, 2008.

[25] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[26] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[27] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[28] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[29] J.-D. Fermanian, "Goodness-of-fit tests for copulas," *J. Multivariate Anal.*, vol. 95, no. 1, pp. 119 – 152, 2005.

[30] T.-H. Lee and X. Long, "Copula-based multivariate garch model with uncorrelated dependent errors," *J. Econometrics*, vol. 150, no. 2, pp. 207 – 218, 2009.

[31] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis Forecasting and Control*. Third ed. Englewood Cliffs, NJ: PrenticeHall, 1994.

[32] G. Gallego, C. Cuevas, R. Mohedano, and N. García, "On the Mahalanobis distance classification criterion for multidimensional normal distributions," *IEEE Trans. Signal Process.*, vol. 61, no. 17, pp. 4387–4396, Sep. 2013.

[33] N. Batra, M. Gulati, A. Singh, and M. B. Srivastava, "It's different: Insights into home energy consumption in india," in *Proc. ACM Wksp Embedded Systems For Energy-Efficient Buildings (BuildSys)*, Roma, Italy, Nov. 2013, pp. 3:1–3:8.

[34] L. Larsson, M. Nyström, and M. Stridh, "Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit," *IEEE Trans. Biomed. Engg.*, vol. 60, no. 9, pp. 2484–2493, Sep. 2013.

[35] S. Wang, L. Cui, J. Que, D. H. Choi, X. Jiang, S. Cheng, and L. Xie, "A randomized response model for privacy preserving smart metering," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1317–1324, Sep. 2012.

[36] J. Qin, Q. Zhao, H. Yin, Y. Jin, and C. Liu, "Numerical simulation and experiment on optical packet header recognition utilizing reservoir computing based on optoelectronic feedback," *IEEE Photon. J.*, vol. 9, no. 1, pp. 1–11, Feb. 2017.

[37] R Core Team, "R: A language and environment for statistical computing," R Foundation for Statistical Computing, Vienna, Austria, 2019.

[38] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler, "sMAP: A simple measurement and actuation profile for physical information," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, Zurich, Switzerland, Nov. 2010, pp. 197–210.

[39] P. Deutsch, "GZIP file format specification version 4.3. Technical Report RFC 1952, Network Working Group," May 1996.

[40] ——, "DEFLATE Compressed Data Format Specification version 1.3. Technical Report RFC 1951, Network Working Group," May 1996.

[41] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

Mayukh Roy Chowdhury received the B. Tech. degree in Electronics and Communication Engineering from West Bengal University of Technology, Kolkata, India, in 2012 and the M.Tech degree in Communication Systems Engineering from Indian Institute of Technology Patna, India, in 2016. He is currently pursuing the Ph.D. degree with Department of Electrical Engineering, Indian Institute of Technology Delhi, India. His research interests include applied machine learning in wireless networks, data driven techniques for smart IoT systems, edge computing, massive machine type communication in 5G, energy and bandwidth efficiency in communication networks.

Sharda Tripathi received her B. Tech. degree in Electronics and Communication Engineering from Rajiv Gandhi Technical University, Bhopal, India, in 2007, 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, and the Ph.D in Electrical Engineering from IIT Delhi, New Delhi, India, in 2019. She is currently a Postdoc Research Fellow at the Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy. Her current research interests are broadly in data-driven resource optimization strategies for smart IoT applications, design and analysis of next-generation 5G networks, application of machine learning for predictive modeling of non-stationary processes.

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.

Dr. De currently serves as an Area Editor for the IEEE COMMUNICATIONS LETTERS, and an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE WIRELESS COMMUNICATIONS LETTERS, the IEEE NETWORKING LETTERS, and the IETE Technical Review journal.