IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 27, 2019, accepted May 10, 2019, date of publication May 16, 2019, date of current version May 28, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917322

Energy Conservation in Wireless Sensor Networks Using Partly-Informed Sparse Autoencoder

BABAJIDE O. AYINDE[1,3] **AND ABDULAZIZ Y. BARNAWI**[2]

[1]Department of Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[2]Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[3]Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292, USA

Corresponding author: Babajide O. Ayinde (babajide.ayinde@louisville.edu)

This work was supported by the King Fahd University of Petroleum and Minerals.

ABSTRACT The energy in wireless sensor networks is considered a scarce commodity, especially in scenarios where it is difficult or impossible to provide supplementary energy sources once the initially available energy is used up. Even in cases where energy harvesting is feasible, effective energy utilization is still a crucial step for prolonging the network lifetime. Enhancement of life-time through efficient energy management is one of the essential ingredients underlining the design of any credible wireless sensor network. In this paper, we propose a sensor selection method using a novel and unsupervised neural network structure referred to as partly-informed sparse autoencoder (PISAE) that aims to reconstruct all sensor readings from a select few. The PISAE comprises three submodules, namely: the gate (which selects the most important sensors), encoder (encodes and compresses the data from select sensors), and decoder (decodes the output of the encoder and regenerates the readings of all initial sensors). Our approach relies on the premise that many sensors are redundant because their readings are spatially and temporally correlated and are predictable from the readings of a few other sensors in the network. Thus, overall network reliability and lifetime are enhanced by putting sensors with redundant readings to sleep without losing significant information. We evaluate the efficacy of the proposed method on three benchmark datasets and compare with existing results. The experimental results indicate the superiority of our approach compared with existing approaches in terms of accuracy and lifetime extension factor.

INDEX TERMS Autoencoder, energy conservation, energy management, unsupervised feature extraction, feature importance, feature ranking, lifetime extension, wireless sensor network, deep learning.

I. INTRODUCTION

A wireless sensor network (WSN) is essentially a large collection of networked sensor nodes that are deployed over a sensing field. These sensor nodes are tiny, low-powered, and energy-limited devices with computing, communication, and sensing capabilities. Typically, nodes organize themselves in clusters and networks and sometimes cooperate to perform an assigned monitoring (and/or control) task at scales and resolutions that are difficult, if not impossible, to achieve traditionally. In their assumed structure, nodes monitor environmental and physical conditions/information (e.g., motion, pressure, sound, humidity, vibration, and temperature), locally process their data both in-unit and at cluster level, and communicate the outcome to the cluster head and/or to one or more collection points known as the sinks or base stations [1].

In recent years, scientific and industrial adoptions of WSNs are becoming more pronounced and predictability and longevity are of paramount importance in most applications. Although, many work have sought after balancing lifetime requirements and network performance, however, due to unexpected and unpredictable environmental dynamics, these networks still suffer from premature energy drainage even for network with harvesting and recharging capabilities [2]. Reconciling these two conflicting objectives is commonly achieved via energy management. Energy management is generally conceptualized as strategies and mechanisms by

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Arafatur Rahman.

which total network energy is coordinated, allocated, and effectively consumed by all nodes such that the network stays fully operational for its projected lifetime. Since energy in WSNs is very limited especially in remote terrains where access to supplementary energy is impractical due to environment hostility, it is important to judiciously balance the energy supply and network load through energy conservation to avoid premature energy depletion [3]. As a result, designing a network with lifetime satisfying specific application requirements is still challenging.

Since data collection is one of the main functions of WSN, sensor nodes periodically sense surveillance area and transmit information in a cooperative and distributed fashion [4]. The sensed data is highly correlated due to the nature of the monitored parameters and large number of deployed sensors [5]. It is remarked that the energy used up in communication phase is very significant compared that used for processing or sensing. Hence, reporting readings of individual sensors is not only energy-inefficient, but also memory-inefficient since the data gathered grows exponentially with time [4], [5]. Schemes that can efficiently manage the network energy at the data collection and communication stage are therefore essential to the network sustainability. Reducing the amount of unnecessary data collection and/or communication will help minimize energy waste and extend network lifetime [3], [6], [7].

Intelligent data-driven approaches using Machine Learning (ML) [4], [5], [8]–[10] and evolutionary techniques [11]–[16] are becoming the tenet for addressing some of the aforementioned energy management challenges. As emphasized in [17], the adoption of machine learning in WSNs has immensely contributed to the practicality of these networks partly by fostering energy efficiency and alleviating unneeded redesign issues. More importantly in some applications such as outdoor/environmental monitoring, sensor nodes might not operate as expected because of unexpected environmental behavior. Machine learning-based solutions can overcome such problems by using the newly obtained information to update their knowledge base. Also, the environmental unpredictability can drastically increase the complexity of mathematical model that describes the resulting network, however, machine learning can help proffer accurate and less complex solutions [5].

Enabling systematic end-to-end processing of huge amount of data generated in real-life WSN deployments is challenging. A significant chunk of the problems arises from the discovery of correlations in the network measurements. To circumvent this problem, a plethora of approaches based on ML have been dedicated for identifying and eliminating data attributes that provide no significant information. ML enables the selection of dominant set of features, which have the greater impact on inferring the network performance based on historical data patterns [18]. Over the last few decades, research efforts have focused on strategies that provides solutions to various challenges in WSN, such as routing [16], link-quality estimation [18], energy harvesting [19], link reliability prediction [9], just to mention a few.

Many existing data-driven approaches address the problem of energy conversation in by removing redundancy in sensed data WSN as a data compression problem [20]–[23]. Energy-efficient data acquisition schemes have also sought to reduce energy used for sensing and data transfer to minimize the energy spent on communication [4]. A vast majority of these approaches aim to reduce the amount of data to be sent by the source and delivered to the sink. In data compression paradigm, data in the source node is encoded and compressed before transmission and upon the receipt at the sink, the encoded information is decoded and decompressed. As detailed in [7], [24]–[26], the data transmission sometimes costs approximately the same energy as that needed to process a thousand operations in a typical sensor node. Therefore, it is important to put in place an efficient energy management scheme that prevents the communication of unnecessary/redundant data and minimizes the wastage of useful energy. In addition, efficient energy management protocols can temporarily turn off unused components/circuitories that consume a large amount of energy in idle state to help minimize the excessive amount of energy used by such components.

Of special interest in the data-driven paradigm are approaches that model the phenomenon that describes the network data evolution. A handful of these approaches rest on the premises that the information sensed at nodes can be predicted to a reasonable degree of accuracy at the sink. However, when dealing with high-dimensional data as a result of large number of sensors, the curse of dimensionality is a prominent issue in many practical machine learning problems [27]. Pragmatically, not all sensor are of equal importance, since most of their readings are often highly correlated and ultimately redundant. Sensor nodes with redundant information would unnecessarily drain useful energy out of the network, hence, it is important to put such nodes with redundant data to sleep in order to reduce energy consumption. Many existing work have addressed the selection of important sensors from machine learning perspective [1], [5], [7]. However, these methods are not end-to-end, that it, they rely on frameworks that separate sensor selection stage from the downstream task. This poses a problem because sensors are not selected based on why they are deployed in the first place.

Autoencoder and its variants are increasingly important for nonlinear dimensionality reduction [27]–[30]. Conventionally, autoencoder is trained to learn compressed representation of its input while preserving the most important information. However, it is still challenging to incorporate deterministic input selection into autoencoder structure to distinguish between important from unimportant input readings. To address this issue, we propose a framework known as partly-informed sparse autoencoder (PISAE) to integrate input selection and downstream task-related objectives (e.g., regression, classification, clustering, compression, etc). Specifically, the proposed framework performs a dynamic feature selection by separating the important input features from redundant/irrelevant ones through a joint learning mechanism with autoencoder. It selects few important inputs to reconstructs the entire input.

The problem addressed in this paper is four-fold: (i) an optimized sensor selection algorithm is proposed using a novel and unsupervised neural network structure referred to as partly-informed sparse autoencoder (PISAE) that reconstructs all sensor readings from readings of select few. The proposed method jointly learns input feature selection and extraction in a unified framework to select the most useful sensor readings to reconstruct readings from all sensors, (ii) the feature selection algorithm is regularized for sparsity in order to reduce the number of sensors used for data reconstruction, (iii) comprehensive experiments were performed to show the efficacy of proposed feature extraction for lifetime enhancement, and lastly (iv) the proposed approach is compared and contrasted with existing methods on benchmark tasks and datasets and its superiority is highlighted. The rest of this paper is organized as follows: Section 2 discusses the state-of-the-art-data-driven approaches in the area of energy conservation and/or management in WSNs. Section 3 discusses the proposed data-driven sensor selection method and energy conservation scheme. Section 4 discusses the experimental designs and presents the results. Finally, conclusions with some future work directions are presented in Section 5.

II. RELATED WORK

Energy conservation via data-driven energy management has a long history [1], [3], [7], [8], [25], [31]. A Bayesian-based selection technique was proposed in [32] to determine optimal number of sensors with the intension of reducing the number of sensor nodes and ultimately improving the network lifetime. In [8], lifetime of WSNs has also been enhanced by ranking sensors based on their significance and using least number of select sensors. In similar vein, simultaneous energy consumption minimization and lifetime maximization was proposed in [33]. Energy management in the WSNs has also been addressed at the architectural stage for large-scale network by minimizing both information exchange and energy consumption using support vector machine (SVM) [34].

Energy has also been conserved in WSN using a distributed classification algorithms based on SVM, where nodes' communication through a centralized processing unit is prohibited [35], [36]. Also in [37], multi-sensor fusion and integration using artificial intelligence was proposed to address a variety of errors and faults in WSNs. The method was analyzed on bulk soil moisture estimation with cosmic ray sensors and evaluated using four classifiers. Selection of the most relevant sensor signals for detecting periods of food intake has also been addressed using SVM to monitor and classify jaw motion [38]. A system for detecting human direction was proposed in [39] using Pyroelectric Infrared (PIR) sensors to capture signals while walking. Discriminative features was extracted and selected to improve the computational efficiency of the proposed system using multilater perceptron, Naive Bayes, and SVM.

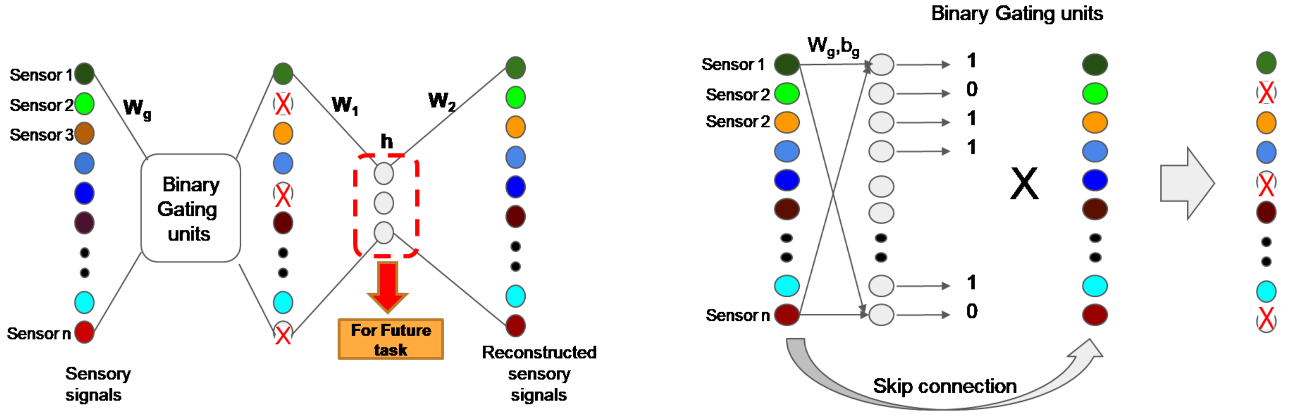
Other related works proposed a wide variety of selection criteria to compute feature importance, such as Laplacian score [40], trace ratio [41], and Fisher score [42]. These methods addressed various combinatorial optimization bottlenecks, such as expensive computational cost and local optimality [41], [42]. In addition, many feature extraction methods have been inspired by conventional autoencoder (AE), which learns to reconstruct the input from its encoded representations [43]. The reconstruction is usually achieved by additive combination through the decoding filters. After training, generating latent encodings for test samples is extremely fast, requiring only a simple matrix-vector multiplication [22], [44]. In general, AEs are forced to unearth the underlying structure in data through imposition of constraints on network parameters. One of such constraints is pushing the response of all hidden units to a small value as in sparse AE (SAE) [45]. Other popular constraints are weight-decay regularization for reducing the effect of overfitting, nonnegativity for extracting interpretable features [30], [46], graph-guidance [47], hidden feature selection for compressing task-relevant and irrelevant information into two groups of hidden units [27]. However, incorporating deterministic input feature selection into autoencoder framework is still challenging.

The most closely related work is denoising AE (DAE) that learns a robust representation through reconstructing a clean input from a corrupted one [48]. Training a DAE is similar to training a basic AE. Generally, some input components are randomly selected and their values forced to 0, while others remain unchanged. DAE uses partially corrupted inputs to learn almost the same representation as the uncorrupted version. This is feasible because data contains stable structures that depend on a combination of many dimensions. High dimensional redundant data are usually recoverable from partial dimensions of the data [4]. However, the fraction (also known as denoising ratio) of input features to be corrupted is pre-determined and features are randomly corrupted based on this predefined ratio. On the other hand, our proposed method is deterministic and the ratio of the input features to be dropped is automatically learned by our novel partly-informed AE. Our proposed algorithm thus aims to select a few input features that are useful for reconstructing the entire input set.

III. PROBLEM FORMULATION

The proposed input feature selection is fully unsupervised and based on AE framework that preserves intrinsic data structures useful for downstream discriminative task. The conventional AE neural network aims to reconstruct input vector using unsupervised learning with mathematical representation as in (1).

$$\hat{\mathbf{x}} = f_{\mathbf{W},\mathbf{b}}(\mathbf{x}) \approx \mathbf{x} \tag{1}$$



(a)

(b)

FIGURE 1. (a) Simultaneous Feature selection and extraction using Partly-Informed Autoencoder architecture (b) Binary gating layer operation.

where $x \in \mathbb{R}^n$ is the scaled input vector (measurements of n sensors), $\mathbf{W} = \{\mathbf{W}_1, \mathbf{W}_2\}$, and $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2\}$ respectively represent the weights and biases of the network. The proposed partly-informed sparse AE (PISAE) is a modified architecture that aims to reconstruct the entire input using partial input information as given in (2) and depicted in Fig. 1.

$$\hat{\mathbf{x}} = f_{\mathbf{W},\mathbf{b},\mathbf{W}_g,\mathbf{b}_g}(\mathbf{x}_{partial}) \approx \mathbf{x} \tag{2}$$

where $\mathbf{x}_{partial} = \mathbf{x} \odot \psi(\mathbf{x})$, $\odot$ is Hadamard product, and

$$\psi(x_i) = \begin{cases} 1 & z_g^{(i)} \geq \tau \\ 0 & otherwise \end{cases} \tag{3}$$

where $z_g^{(i)} = \sigma(\sum_{j=1}^{n}(W_g^{(ij)} x_i + b_g^{(j)}))$; $\mathbf{W}_g \in R^{n \times n}$ and $\mathbf{b}_g \in R^n$ are weight matrix and bias vector of the gating layer, respectively. τ is the threshold of importance of input sensors for reconstruction task and $\sigma(z)$ denotes an element-wise application of the logistic sigmoid of z, $\sigma(z) = 1/(1 + exp(-z))$.

For compressed representation, the partial sensory signals are encoded into a lower dimensional embeddings and the hidden activations are computed as

$$\mathbf{h} = g_{\theta_1}(\mathbf{x}_{partial}) = ReLU(\mathbf{W}_1 \mathbf{x}_{partial} + \mathbf{b}_1) \tag{4}$$

where $\mathbf{h} \in \mathbb{R}^{n'}$, $\mathbf{W}_1 \in R^{n' \times n}$, $\mathbf{b} \in R^{n' \times 1}$, and ReLU denotes a rectified linear unit defined as $ReLU(z) = \max(0, z)$. The resulting hidden representation, $\mathbf{h}$, is then mapped back to a reconstructed vector, $\hat{\mathbf{x}} \in \mathbb{R}^n$, by a similar mapping function, parameterized by $\{\mathbf{W}_2, \mathbf{b}_2\}$,

$$\hat{\mathbf{x}} = g_{\theta_2}(\mathbf{h}) = ReLU(\mathbf{W}_2 \mathbf{h} + \mathbf{b}_2) \tag{5}$$

For the purpose of optimizing the parameters in (2), i.e. $\theta = \{\theta_1, \theta_2\}$, the average reconstruction error is the cost of the optimization objective:

$$J_r(\mathbf{W}, \mathbf{b}, \mathbf{W}_g, \mathbf{b}_g) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} ||\hat{x}^{(i)} - x^{(i)}||^2 \tag{6}$$

where m is the number of examples in the training set.

It must be remarked that most natural signals are not sparse in the original form and it is important to sparsify them through a predefined basis transformation. In order to improve the sparsity, robustness, and richness of the feature extraction step, the hidden representation is constrained to be sparse by penalizing the reconstruction objective in (6) with the Kullback-Leibler (KL) divergence [49] between the hidden activities and desired activity level ρ as computed in (7). The sparsity constraint is essential in WSNs because it helps reduce the amount of transmitted data within the network, thus reducing energy consumption and prolonging the lifetime of whole network.

$$J_{KL}(\rho||\hat{\rho}) = \sum_{j=1}^{n'} \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j} \tag{7}$$

where

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^{m} h_j(x^{(i)}) \tag{8}$$

It must be emphasized that it is important to also constrain the output of the gating layer $\psi(\mathbf{x})$ to be sparse in order to minimize the number of sensors needed to reconstruct the entire input. To ensure this objective is satisfied, an additional sparsity term is added, making the overall cost function for proposed Partly-Informed Sparse Autoencoder (PISAE) to become:

$$J_{PISAE}(\mathbf{W}, \mathbf{b}, \mathbf{W}_g, \mathbf{b}_g) = J_r(\mathbf{W}, \mathbf{b}, \mathbf{W}_g, \mathbf{b}_g) + \beta_1 J_{KL}(\rho||\hat{\rho}) + \frac{\beta_2}{mn} \sum_{j=1}^{m} \sum_{i=1}^{n} ((\mathbf{z}_g)_{ij})^2 \tag{9}$$

where $\beta_1, \beta_2 > 0$ are hyperparameters that control the tradeoff among conflicting objectives.

The consequences of minimizing (9) are that: (i) the average reconstruction error is reduced (ii) the sparsity of the

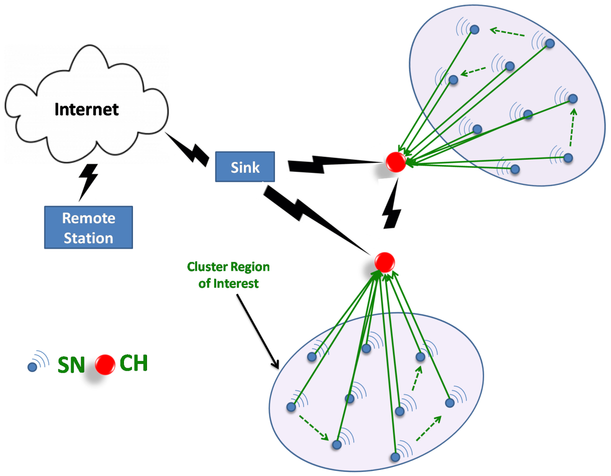


FIGURE 2. Sensor network architecture. The region of interest is assumed to be relatively far from the remote station. All sensor nodes are assumed to have one-hop communication to their respective cluster head (CH). Cluster heads are assumed to have strong power-bank than sensor nodes and are designed to transmit compressed data to the sink.

hidden layer activation is increased, and (iii) the number of sensors required to reconstruct the entire input is minimized. The parameters $\mathbf{W}$, $\mathbf{W}_g$, $\mathbf{b}$, and $\mathbf{b}_g$ are updated using (10), (11), (12), and (13) using the backpropagation algorithm.

$$w_l^{(ij)} = w_l^{(ij)} - \xi \frac{\partial}{\partial w_l^{(ij)}} J_{\mathrm{PISAE}}(W, b, \mathbf{W}_g, \mathbf{b}_g) \quad (10)$$

$$b_l^{(i)} = b_l^{(i)} - \xi \frac{\partial}{\partial b_l^{(i)}} J_{\mathrm{PISAE}}(W, b, \mathbf{W}_g, \mathbf{b}_g) \quad (11)$$

$$w_g^{(ij)} = w_g^{(ij)} - \xi \frac{\partial}{\partial w_g^{(ij)}} J_{\mathrm{PISAE}}(W, b, \mathbf{W}_g, \mathbf{b}_g) \quad (12)$$

$$b_g^{(i)} = b_g^{(i)} - \xi \frac{\partial}{\partial b_g^{(i)}} J_{\mathrm{PISAE}}(W, b, \mathbf{W}_g, \mathbf{b}_g) \quad (13)$$

where $\xi > 0$ is the learning rate.

A. ENERGY CONSERVATION USING PISAE

A hierarchical network depicted in Fig. 2 consisting of k clusters with each cluster containing n sensor nodes and a cluster head is considered. Each i^{th} sensor (where i = 1, ..., n) in a cluster collects a real-valued sample x_i at a predefined sampling period and transmits packets at a configured transmission power to its cluster head but not at power level sufficient to reach the sink due to long distance propagation. Sensor nodes in a cluster are deployed to cover a large geographic area known as the sensing field and are assumed to only transmit for short distances. Cluster heads relay aggregated sensed data to the sink for further analysis and/or compression. The sink processes the aggregated data from many cluster heads and relay to the remote station. The proposed framework assumes a static sensor network, that is, sink, cluster heads, and sensor nodes are all considered to be static.

The proposed energy conservation scheme utilizes a trained PISAE to sparsely encode partial input data with gating and encoding weights at the clusters and sink. The entire signal is decoded at remote station using the decoding weights. In order to train the parameter of PISAE for efficient energy conservation, a database of historical sensed data needs to be utilized. Since the model trained with PISAE needs to generalize to unseen data points, a large quantity of training data is almost always required to enhance model generalization. However, obtaining a large quantity of such data is challenging especially for newly deployed networks. To circumvent this problem, it is therefore necessary to enforce sparsity constrain as in (7) to ensure generalization of model trained on small to mid-sized data.

+ Since sensor nodes are most times densely deployed and their readings spatially correlated, PISAE is able to learn the essential spatial correlation and redundant patterns among the sensors. It must be noted that if the underlying phenomenon becomes different in a way that it alters sensor nodes' spatial correlation, then, it might be difficult for PISAE model trained on previous data to accurately generalize to its new input. In such cases, new data collection and offline model fine-tuning must be performed. As emphasized in [22], the amount of data required for training varies heavily on underlying sensed phenomenon. If the sensors' correlation patterns is complex, more data samples are usually required. In this work, it is assumed that the network designer is able to collect sufficient training data, on cluster basis.

As aforementioned, the radio transceiver is one of the most energy consuming unit in a sensor node. Thus, energy management becomes essential for effective energy consumption and lifetime extension. To this end, an energy conservation scheme is designed using a trained PISAE. The energy conservation scheme is proposed to be implemented on all cluster heads and sink in a distributed fashion for sensor selection and data compression/encoding. Each cluster head is equipped to relay data to the sink on behalf of other cluster heads without processing. Therefore, relay nodes that are adjacent to the sink usually run out of energy faster than those farther away. This condition must be factored into the design of such network by equipping cluster head adjacent to the sink with more powerful battery banks. In the proposed energy conservation scheme, each sensor node sends data to its respective cluster head and awaits instruction whether the energy-consuming circuitries should be put to sleep or shut down. The cluster heads collect all the data from their cluster members and determine the importance of each of the sensors using gating weights $\{\mathbf{W}_g, \mathbf{b}_g\}$.

Cluster heads perform a downward communication to their respective children (sensor nodes) using a binary label. If the label received by a sensor node is 1, such node should remain active until the cluster head assigns a different label. Otherwise, the sensor node is forced to sleep and shut down all energy-consuming circuitries for a designated time interval. In the meantime, the cluster heads set the readings of unimportant sensor nodes to 0. In the upward communication, all cluster heads utilize encoding $\{\mathbf{W}_1, \mathbf{b}_1\}$ to encode and compress their respective members' data and relay to the sink. Cluster heads also send additional information such as whether the data is coming from their cluster members or it is relayed on behalf of other cluster heads. The sink also performs similar operation as the cluster heads by aggregating all the data from the cluster heads and determining if data from some of the clusters are not important to reconstruct all their data. The sink also sends a global cluster label to all the cluster heads to engage in active sensing or put all cluster members to sleep and shut down their energy-consuming circuitries.

Finally, the sink transmits the overall compressed data to the remote station over the backhaul connection. The remote station is equipped with decoding weights $\{\mathbf{W}_2, \mathbf{b}_2\}$ in order to reconstruct back the entire signal. At designated time interval, all sensor nodes are required to wake up, turn on their circuitries, and forward their data to cluster heads to again determine the significance of their sensed data. It is critical to conserve energy using intelligent schemes such as the proposed because by blindly turning OFF the radio during each idle slot, over a period of time to save the energy, more energy might end up been used than if the radio has been left on continuously. So, Power saving is efficient only if the energy conserved is greater than energy expended to transit to and from sleep state [1], [3], [4].

The proposed energy conservation scheme is summarized in Algorithm 1. The online encoding and decoding of sparse codes are practical since both sensor selection and feature encoding for one cluster head has a time complexity of $\mathcal{O}(n^2 + n'n)$ while data decoding at remote station has a complexity of $\mathcal{O}(n' \times n)$. Similar to [5], [8], [50], Lifetime Extension Factor (LEF) is evaluated as given in (14).

$$\begin{aligned} LEF &= \frac{\text{Total number of sensors}}{\text{Number of sensor used for reconstruction}} \\ &= \frac{n}{\sum_{i=1}^{n}(\psi_i)} \end{aligned} \tag{14}$$

IV. EXPERIMENTAL SETUP, RESULTS, AND DISCUSSION

The performance of the proposed energy conservation scheme based on PISAE is evaluated on sensor selection, data reconstruction, and classification using three publicly available databases namely: Ionosphere, Forest CoverType, and Sensor Discrimination datasets. These datasets have been used extensively to benchmark many algorithms on wide variety of tasks [51]–[55]. Each dataset is randomly shuffled and split into 70% and 30% training and testing set, respectively. Adam optimizer [56] with batch size of 100 was used to train the model for 1000 epochs. For sensor selection, the performance of PISAE was compared and contrasted with four existing and well-established feature extraction methods namely: Recursive Feature Extraction (RFE) [57], Independent Significance Features test (IndFeat) [5], [8], Representation Entropy Clustering Feature Selection Algorithm (REC-FSA) [18], and Recursive Feature Elimination with Variable Step Size (VSSRFE) [58]. For classification (referred to as future task in Fig. 1), the hidden activations ($\mathbf{h}$) of PISAE were extracted and fed into a Softmax layer and the performance was compared with four other classifiers (namely, Naive Bayes (NB), Multilayer Perceptron (MLP), Support Vector Machine (SVM) with linear kernel, and Decision Tree (DTree)) using RFE and IndFeat for sensor selection.

TABLE 1. Training hyper-parameters.

Paramter	Ionosphere	Sensor Discrimination	Forest CoverType
ξ	0.01	0.9	0.9
β_1	5	5	5
β_2	0.001	0.001	0.01
τ	0.5	0.5	0.1
n'	30	30	10
ρ	0.5	0.5	0.5
batch size	5	100	100

All experiments were performed on Intel(r) Core(TM) i7-6700 CPU @ 3.40Ghz and a 64GB of RAM running a 64-bit Ubuntu 14.04 edition. The software implementation has been in Pytorch library .[1] Naive Bayes, MLP, SVM, and Decision Tree were implemented in Scikit-learn ecosystem.[2] Experiments were repeated 20 times and averaged to mitigate the effect of random initialization in the evaluation. The software implementation and trained models will be publicly available for reproducibility of results.[3] All datasets were normalized for training PISAE model and decoder output mapped back to the original data interval for efficient reconstruction of the original data. Similar to [4], the hidden activations of PISAE are rounded to a low precision during the offline training. Thus, yielding a less memory intensive model practicable for deploying on cluster heads' low precision settings. Other hyper-parameters for training PISAE is summarized in Table 1

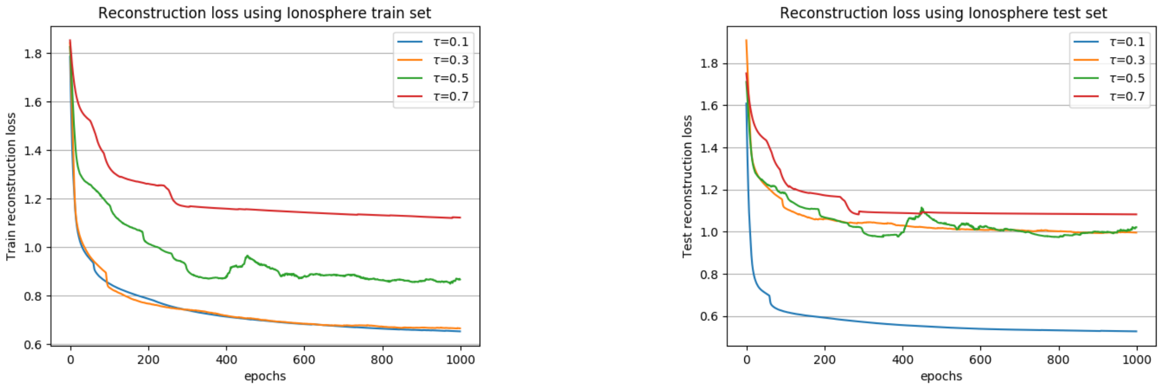
In the first set of large-scale experiments, Ionosphere dataset[4] was used. The dataset consists of radar data collected in Goose Bay, Labrador. It has two classes for radar signals, namely, Good and Bad. Good data are those showing evidence of some type of structure in the Ionosphere. Bad returns are those that do not let their signals pass through the Ionosphere. This dataset consists of 34 readings and 351 records. Figs. 3a and b show the reconstruction performance of PISAE with four importance threshold $\tau = 0.1, 0.3, 0.5,$ and 0.7 for training and testing, respectively. It can be observed that τ of 0.1 outperforms other τ values in terms of both training and testing reconstruction loss. However, in Figs 4a and b, when τ is 0.1, PISAE utilizes all 34 sensors for reconstructing

[1] http://pytorch.org/

[2] https://scikit-learn.org/stable/index.html

[3] Source code will be available at: https://github.com/babajide07/energy-management-in-WSN

[4] https://archive.ics.uci.edu/ml/datasets/ionosphere



(a)

(b)

FIGURE 3. Reconstruction loss against epochs for different τ using Ionosphere (a) Training set (b) Test set.

back its input amounting to no energy conservation. On the other hand, when τ is changed to 0.5, it uses data from less than 10 sensors to reconstruct the entire data from all 34 sensors, but it is more lossy in terms of reconstruction. As τ is increased to 0.7, the reconstruction error increases because PISAE is using, on average, data from only two sensors for reconstruction, which is not enough to capture the dynamics of the data from all 34 sensors.

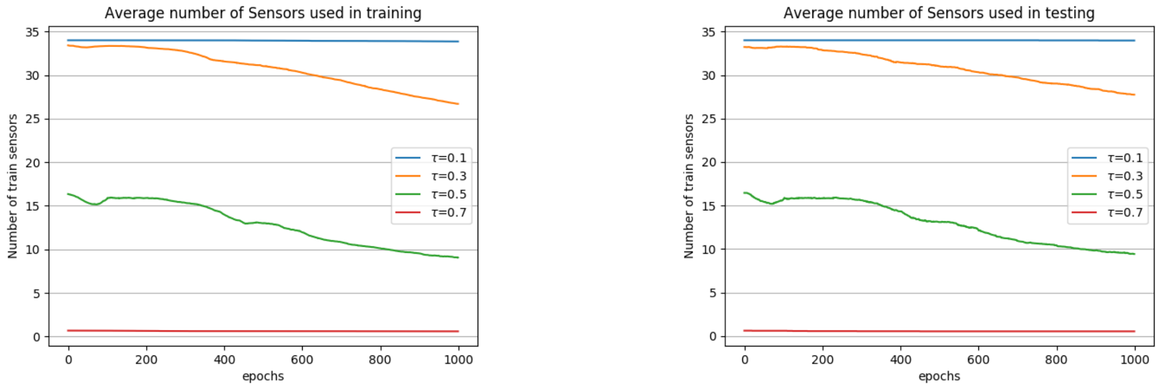
Algorithm 1 Cluster-Level Energy Conservation Using PISAE

```
for every cluster in the network do

    get: readings from all sensor nodes

    set: threshold of importance $\tau$

    select: few important sensors using gating weights
            $\{\mathbf{W}_g, \mathbf{b}_g\}$


    set: redundant sensors to sleep and set their readings
         to 0


    encode: the truncated signal into a lower
            dimensional embedding using $\{\mathbf{W}_1, \mathbf{b}_1\}$


    send: the encoded and compressed data to the sink
end for
```

The performance of PISAE was also compared to both IndFeat and RFE in terms of how their features are distributed in feature space. In this regard, t-distributed stochastic neighbor embedding (t-SNE) [59] was used to project the original data, the hidden activations of PISAE and select sensor features of IndFeat and RFE as shown in Fig.6. The t-SNE projections show that the manifold of embeddings of each class in Fig.6d using PISAE is distinct and more linear than original feature space in Fig.6a. It can also be observed that the manifolds in the projections using IndFeat and RFE are also more convoluted than that of the PISAE as shown in Fig.6b and c. This is an indication that PISAE is able to extract features using data from 12 sensors to represent the entire data. Performance of PISAE was also compared and contrasted with IndFeat, RFE, REC-FSA, and VSSRFE in terms of downstream classification task. The classification accuracy and number of sensors used are reported in Fig. 5a, b, and c for classifiers MLP, NB, and SVM, respectively. The three classifiers were used to classify the data from sensors selected by IndFeat, RFE, REC-FSA, and VSSRFE. Averaged results of 20 independent trials, related mean values and standard deviation are plotted.

As can be observed in Fig. 5 for MLP classifier, IndFeat achieved a test accuracy of 86.79%, 87.0%, and 84.29% for 10, 20, and 30 selected sensors, respectively. RFE, on the other hand, attained test accuracies of 86.42%, 84.0% and 89.19% for the same number of selected sensor. However, both REC-FSA and VSSRFE outperformed MLP and IndFeat for all sensor selections considered. REC-FSA attained test accuracies of 91.60%, 90.19%, and 92.56% whereas VSSRFE achieved 89.15%, 90.47%, 92.55% for 10, 20, and 30 selected sensors, respectively. Sensor selection of 10, 20, and 30 yielded LEF of 3.4, 1.7, and 1.13, respectively. IndFeat outperforms RFE by 0.37% and 3% using only 10 and 20 sensors, respectively. However, when the number selected sensors was increased to 30, RFE outperforms IndFeat by 4.9% with LEF of 1.13. IndFeat outperforms RFE when few number of sensors are selected and this observation is consistent for other classifiers considered, except for SVM with 20 selected sensors. The performance of both REC-FSA and VSSRFE for MLP classifier are comparable when 20 and 30 sensors were selected.

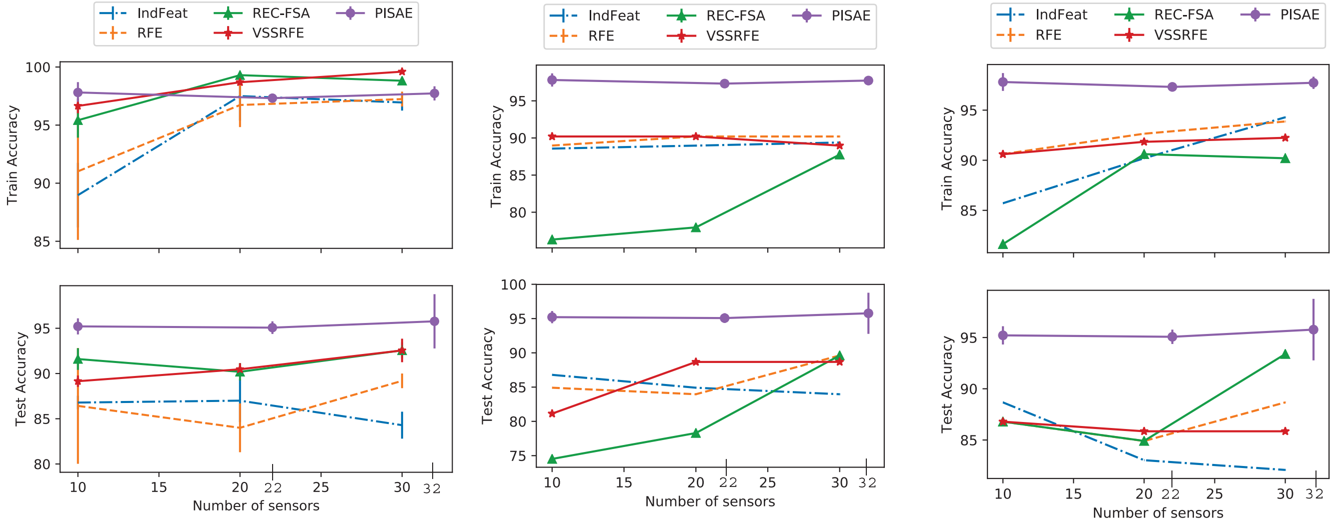
PISAE outperformed all the four feature selection methods considered by 6.54% for IndFeat (SVM with 10 sensors), 8.07% for IndFeat (MLP with 20 sensors), and 6.15% for



(a)

(b)

FIGURE 4. Evolution of sensor selection for different τ using Ionosphere (a) Training set (b) Test set.



(a)

(b)

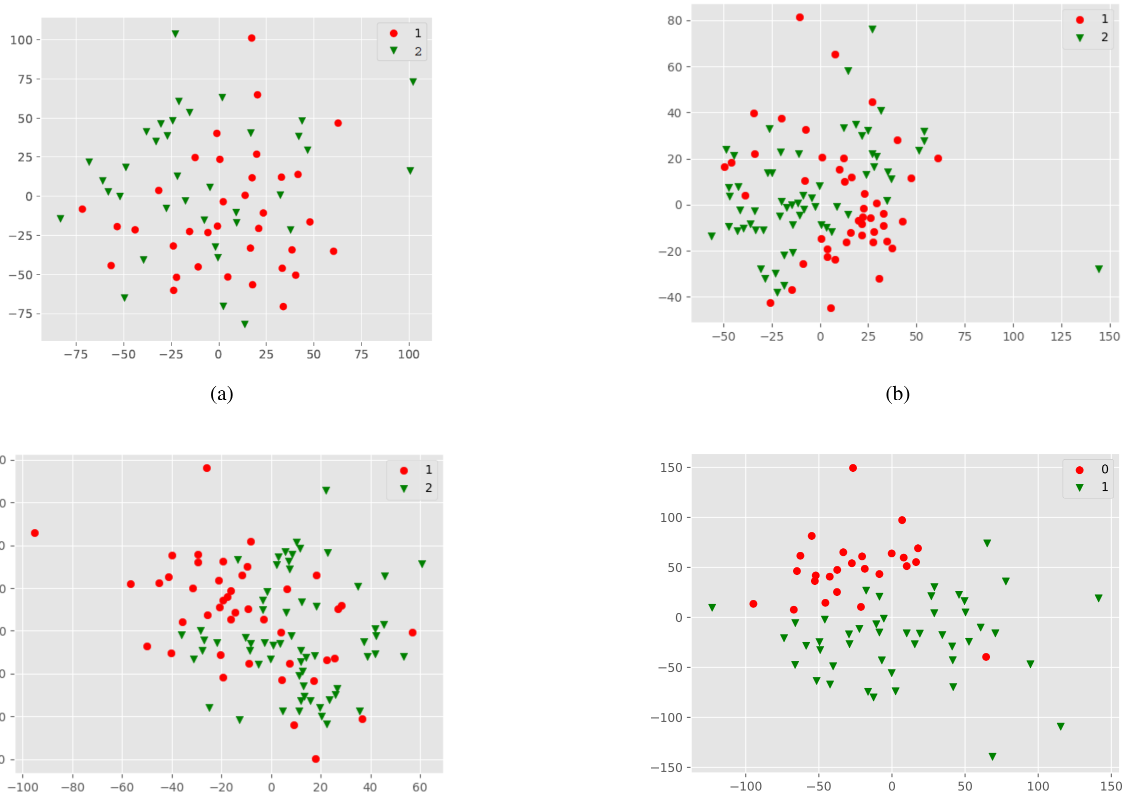
(c)

FIGURE 5. Classification performance of PISAE in comparison with RFE, IndFeat, REC-FSA, and VSSRFE on Ionosphere dataset using (a) MLP (b) Naive Bayes (c) SVM.

RFE (NB with 30 sensors) with 10 ($\tau = 0.5$), 22 ($\tau = 0.3$), and 32 ($\tau = 0.1$) selected sensors, respectively, yielding LEF of 3.4, 1.55, and 1.1. PISAE also outperforms both REC-FSA and VSSRFE by relatively large margins for all three classifiers considered. It is strongly believed that PISAE is able to outperform other methods in terms of test accuracy for comparable number of sensor because it is able to project the original convoluted high-dimensional space into a lower dimensional space with relatively better class separability as shown in Fig.6. The robustness measured by the area under the curve (auc) of receiver operating characteristic curve for PISAE-based classifier in comparison with classifiers based on IndFeat and RFE are shown in Fig. 7a and b, respectively. It can again be observed that PISAE is more robust with auc of 0.9905 compared to the best classifiers based on IndFeat with auc of 0.9381 and RFE with auc of 0.945. It is also observed that RFE is more robust than IndFeat for all benchmark classifiers.

In the second set of large-scale experiments, Sensor Discrimination database[5] was utilized. The database consists of 12 different numerical values of samples of an unknown substance. It is a labeled dataset with three classes, namely, group A, group B, and false alarm. The receiver node, which receives 12 numerical values for an unknown sample, was designed to determine which class each sample belongs to. In other words, the receiver should indicate whether this unknown sample is in group A or group B. Conversely, it falls under false alarm when the sample does not fall into either of the first two groups. The database contains 2211 records of readings from 12 sensors. Every run of the

[5] http://www.technologyforge.net/Datasets/



(a) (b)

(c) (d)

FIGURE 6. 2D t-SNE projection [59] of (a) original 34D test data representation (b) 30D representation using IndFeat (c) 30D representation using RFE and (d) 12D of the activation of hidden units of PISAE using Ionosphere dataset.

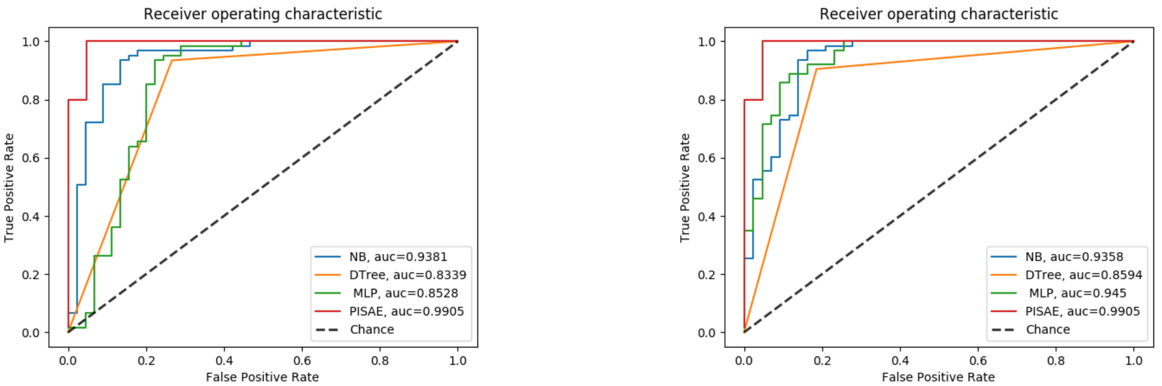
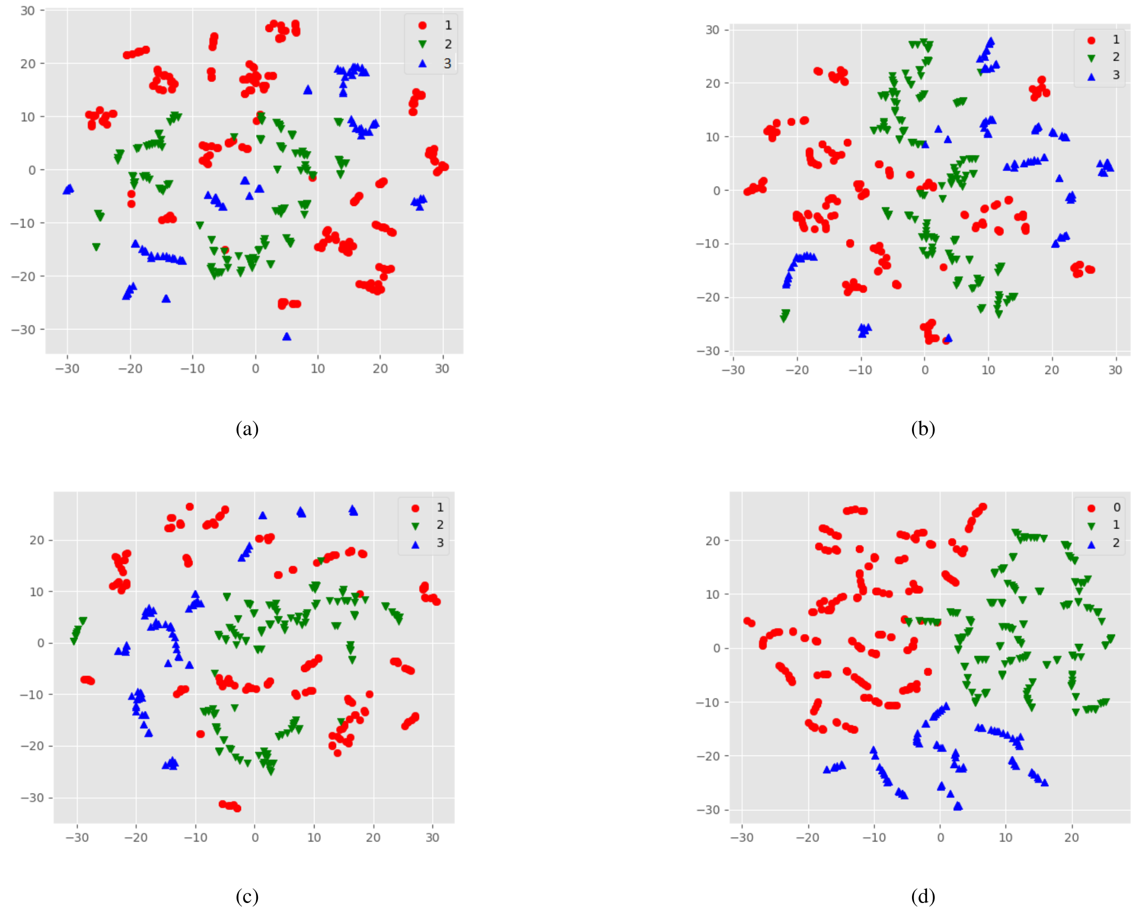


FIGURE 7. Comparison of receiver operating characteristic curves of classifier with PISAE and four classifiers with (a) IndFeat (b) RFE.

experiments is repeated 10 times and averaged for statistical significance. The 2D t-SNE projections of the 12D test data as well as the projections of features selected by PISAE, IndFeat, and RFE are depicted in Fig.8. Again, the manifolds of original test data, IndFeat, and RFE in Fig.8a, b, and c, respectively, are critically overlapping compared to that of PISAE in Fig.8d which is almost linearly separable. As can be observed in Fig.8d, the separability of the projected embeddings of PISAE features is a testimony of its ability to not only compress the data, but also to extract meaningful discriminative representation. The effect of sensor selection using PISAE is also evaluated and compared with benchmark sensor selection methods based on classification performance. The performance of MLP, Naive Bayes, and SVM are



(a)

(b)

(c)

(d)

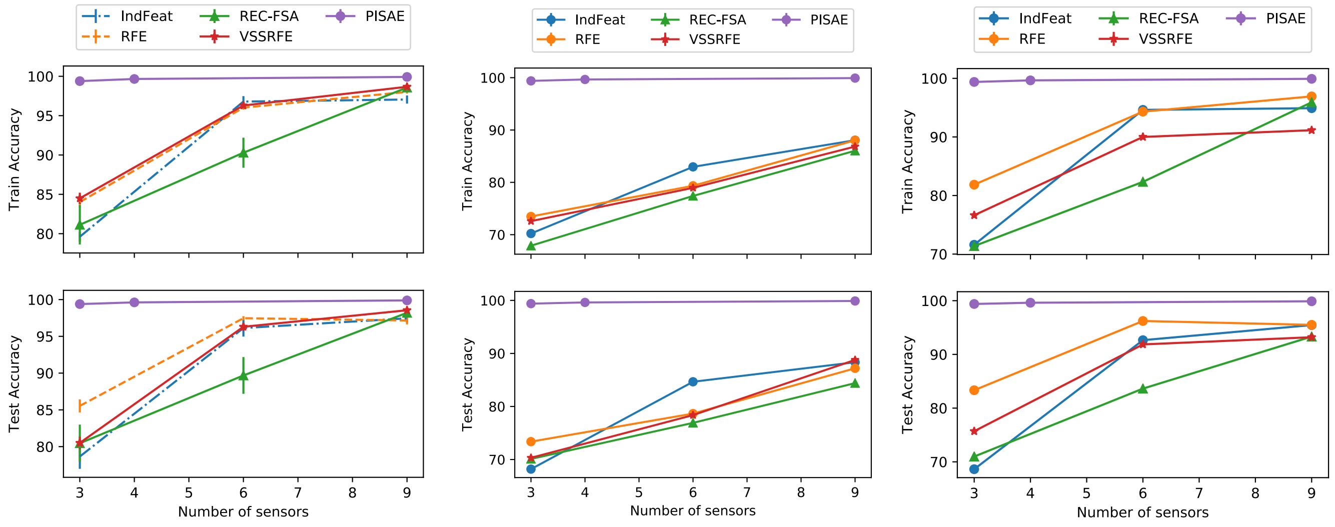
FIGURE 8. t-SNE projection [59] of (a) original 12D test data representation (b) 9D representation using IndFeat (c) 9D representation using RFE and (d) 9D of the activation of hidden units of PISAE using Sensor discrimination dataset.

depicted in Fig 9a, b, and c, respectively. It can be observed that PISAE also outperforms the best classifiers based on IndFeat, RFE, REC-FSA, and VSSRFE. More specifically, PISAE improves the test accuracy by 2.17% and for IndFeat and 2.15% for RFE. PISAE with $\tau = 0.3$ improves the LEF by 200% and 100% in comparison with RFE and IndFeat, respectively. Also, PISAE consistently outperforms both REC-FSA and VSSRFE in terms of test accuracy and LEF. PISAE is able to find a suitable subspace to project the data for better class separability.

In the last set of large-scale experiments, Forest CoverType dataset[6] was used. The dataset was collected and collated at the University of Colorado to predict the forest cover type of unknown regions. It has been used in several research endeavors on data stream classification. This database contains 581, 012 instances and 54 attributes, each belonging to one of 7 classes. The 2747 records were randomly sampled from each class in order to balance data for the training procedure. Hyperparameters were set as given in Table 1. One of the observations in Table 2 is that when $\tau = 0.2$ and 0.3, PISAE-based classifier outperforms the best of both IndFeat and RFE-based classifiers by a small margin. However, the LEF of PISAE is 3.86 and those of IndFeat and RFE is 1.8. This implies that even with fewer number of sensors, PISAE was outshining its counterparts. Another important observation is that MLP consistently outperforms other classifiers for all benchmark feature extraction algorithms in terms of test accuracy. On average, the results of MLP and SVM for RFE, REC-FSA, and VSSRFE were very competitive and outperform those of IndFeat. The observation is similar for NB classifier, except for the VSSRFE which underperforms compared to IndFeat. However, selecting only 17 sensors with PISAE yielded a test accuracy that outperforms almost all of the benchmark methods.

As remarked in [60], it possible to have situation where some sensors report corrupted and very noisy data due to many factors such as external noise, inaccurate sensor calibration, and imperfect node design. In this situation, having over-complete representation might yield improved performance. Data representation is over-complete when $n' > n$,

[6] https://archive.ics.uci.edu/ml/datasets/covertype



(a) (b) (c)

FIGURE 9. Classification performance of PISAE in comparison with RFE, IndFeat, REC-FSA, and VSSRFE on Sensor Discrimination dataset using (a) MLP (b) Naive Bayes (c) SVM classifier.

TABLE 2. Classification performance on Cover-type dataset.

Method		No. of Sensors	train (±std)	test (±std)	p-value	Av. training time (s)	Av. testing time (s)
IndFeat	MLP	30	70.67 (±0.78)	69.39 (± 0.65)	< 0.001	456	3.6×10^{-4}
	Naive Bayes	30	44.10 (± 0.0)	44.74 (± 0.0)	< 0.001		
	SVM	30	61.70 (± 0.0)	60.58 (± 0.0)	< 0.001		
RFE	MLP	30	74.48 (± 0.82)	74.08 (± 0.94)	< 0.001	535.0	3.2×10^{-4}
	Naive Bayes	30	49.42 (± 0.0)	49.29 (± 0.0)	< 0.001		
	SVM	30	66.17 (± 0.0)	66.01 (± 0.0)	< 0.001		
REC-FSA	MLP	30	79.38 (± 0.78)	77.08 (± 0.83)	< 0.001	629.72	3.9×10^{-4}
	Naive Bayes	30	45.85 (± 0.0)	46.94 (± 0.0)	< 0.001		
	SVM	30	64.88 (± 0.0)	64.01 (± 0.0)	< 0.001		
VSSRFE	MLP	30	76.96 (0.75±)	75.34 (± 0.75)	< 0.001	956.75	3.5×10^{-4}
	Naive Bayes	30	40.91 (± 0.0)	40.77 (± 0.0)	< 0.001		
	SVM	30	61.54 (± 0.0)	62.43 (± 0.0)	< 0.001		
PISAE	$\tau = 0.5$	3 (± 1)	67.50 (± 3.029)	66.00 (± 3.10)	-	1014.6	2.3×10^{-4}
	$\tau = 0.3, n' = 30$	14 (± 4)	77.81 (± 0.95)	74.81 (± 0.57)	-		
	$\tau = 0.2, n' = 30$	17 (± 4)	78.16 (± 2.31)	75.40 (± 0.93)	-		
	$\tau = 0.1, n' = 150$	24 (± 2)	83.58 (± 0.82)	**82.64 (±0.81)**	-		

that is, when the number of selected sensors is less than the number of hidden units of PISAE. However, in noise-free networks, using over-complete codes can degrade performance due to high susceptibility to overfitting [22]. In light of this, the number of hidden units in PISAE was increased to 100 and τ set to 0.1. It was observed that test accuracy improved by 7.25% compared to when $\tau = 0.5$, 0.4, and 0.2, thus resulting in LEF of 2.25. With 24 selected sensors, PISAE outperforms all benchmark methods with 30 sensors as shown in Table 2. The p-value was computed between test accuracy of each benchmark classifier and PISAE with comparable number of sensors. The statistical significance of PISAE is also reported as shown by the p-values.

The average training and inference runtime for PISAE and all benchmark feature selection methods were evaluated on CoverType dataset. The timing evaluation of PISAE was conducted in Pytorch version 0.2.0_3 using a mini-batch of size 100. It can be observed that PISAE-based model needed more time to train compared to benchmark methods but inference is competitive, faster and executed in less than 0.25 millisecond.

V. CONCLUSION

Autoencoder-based energy conservation in hierarchical wireless sensor networks is introduced in this paper for prolonging the lifetime of wireless sensor networks. Since many sensors are redundant because their readings are highly correlated, lifetime can be enhanced by putting sensors with redundant readings to sleep without losing significant information. The proposed energy conservation heuristic consists of three submodules: the gate (which selects the most important sensors), encoder (encodes and compresses the data from

TABLE 3. Classification performance on Ionosphere dataset.

Method/classifier		No. of Sensors	train (±std)	test (±std)	p-value
IndFeat	MLP	10	88.97 (± 2.78)	86.79 (±0.75)	< 0.001
		20	97.50 (± 2.11)	87.0 (±2.37)	< 0.001
		30	96.95 (±0.7)	84.29 (±1.49)	< 0.001
	Naive Bayes	10	88.57 (±0.0)	86.79 (±0.0)	< 0.001
		20	88.97 (± 0.0)	84.91 (±0.0)	< 0.001
		30	89.39 (±0.0)	83.96 (± 0.0)	< 0.001
	SVM	10	85.71 (±0.0)	88.67 (± 0.0)	< 0.001
		20	90.20 (± 0.0)	83.02 (± 0.0)	< 0.001
		30	94.29 (± 0.0)	82.08 (± 0.0)	< 0.001
	Decision Tree	30	100 (±0.0)	83.82 (± 1.98)	< 0.001
RFE	MLP	10	91.02 (± 5.9)	86.42 (± 6.4)	< 0.001
		20	96.73 (± 1.9)	84.0 (± 2.7)	< 0.001
		30	97.24 (± 0.64)	89.19 (± 0.81)	< 0.001
	Naive Bayes	10	88.98 (± 0.0)	84.91 (± 0.0)	< 0.001
		20	90.20 (± 0.0)	83.96 (± 0.0)	< 0.001
		30	90.20 (± 0.0)	89.62 (± 0.0)	< 0.001
	SVM	10	90.61 (± 0.0)	86.79 (± 0.0)	< 0.001
		20	92.65 (± 0.0)	84.9 (± 0.0)	< 0.001
		30	93.87 (± 0.0)	88.67 (± 0.0)	< 0.001
	Decision Tree	30	97.24 (± 0.0)	89.20 (± 1.2)	< 0.001
REC-FSA	MLP	10	95.42 (± 1.5)	91.60 (±1.2)	< 0.001
		20	99.3 (± 0.26)	90.19(± 0.96)	< 0.001
		30	98.82 (± 0.22)	92.56 (± 1.3)	< 0.001
	Naive Bayes	10	77.32 (±0.0)	74.52 (±0.0)	< 0.001
		20	77.96 (± 0.0)	78.30 (±0.0)	< 0.001
		30	87.76 (±0.0)	89.62 (± 0.0)	< 0.001
	SVM	10	81.63 (±0.0)	86.79 (± 0.0)	< 0.001
		20	90.61 (± 0.0)	84.91 (± 0.0)	< 0.001
		30	90.20 (± 0.0)	93.39 (± 0.0)	< 0.001
VSSRFE	MLP	10	96.65 (± 0.63)	89.15 (±0.632)	< 0.001
		20	98.69 (± 0.24)	90.47 (± 0.67)	< 0.001
		30	99.6 (±0.12)	92.55 (±1.3)	< 0.001
	Naive Bayes	10	90.2 (±0.0)	81.13 (±0.0)	< 0.001
		20	90.2 (± 0.0)	88.67 (±0.0)	< 0.001
		30	88.97 (±0.0)	88.68 (± 0.0)	< 0.001
	SVM	10	90.60 (±0.0)	86.79 (± 0.0)	< 0.001
		20	91.84 (± 0.0)	85.85 (± 0.0)	< 0.001
		30	92.24 (± 0.0)	85.85 (± 0.0)	< 0.001
PISAE	$\tau = 0.5$	10 (± 2)	97.81 (± 0.89)	95.21 (± 0.89)	-
	$\tau = 0.3$	22 (± 3)	97.32 (± 0.18)	95.07 (± 0.70)	-
	$\tau = 0.1$	32 (± 1)	97.73 (± 0.61)	**95.77** (± 3.0)	-

select sensors), and decoder (decodes the output of the encoder and regenerates the readings of all sensors). The proposed scheme is evaluated on three benchmark datasets, namely, Ionosphere, Sensor Discrimination, and Forest CoverType. The energy conservation was measured in terms of lifetime extension. Experimental results show that network

TABLE 4. Classification performance on Sensor Discrimination dataset.

Method/classifier		No. of Sensors	train (±std)	test (±std)	p-value
IndFeat	MLP	3	79.61 (± 0.63)	78.65 (± 1.67)	< 0.001
		6	96.79 (± 0.69)	96.13 (± 1.16)	< 0.001
		9	97.05 (± 0.52)	97.45 (± 0.6)	< 0.001
	Naive Bayes	3	70.22 (± 0.0)	68.19 (± 0.0)	< 0.001
		6	82.96 (± 0.0)	84.66 (± 0.0)	< 0.001
		9	88.05 (± 0.0)	88.34 (± 0.0)	< 0.001
	SVM	3	71.59 (± 0.0)	68.63 (± 0.0)	< 0.001
		6	94.62 (± 0.0)	92.65 (± 0.0)	< 0.001
		9	94.91 (± 0.0)	95.46 (± 0.0)	< 0.001
RFE	MLP	3	84.02 (± 0.67)	85.54 (± 0.89)	< 0.001
		6	96.02 (± 0.24)	97.47 (± 0.27)	< 0.001
		9	98.02 (± 0.21)	97.16 (± 0.53)	< 0.001
	Naive Bayes	3	73.48 (± 0.0)	73.37 (± 0.0)	< 0.001
		6	79.34 (± 0.0)	78.66 (± 0.0)	< 0.001
		9	88.04 (± 0.0)	87.20 (± 0.0)	< 0.001
	SVM	3	81.85 (± 0.0)	83.33 (± 0.0)	< 0.001
		6	94.32 (± 0.0)	96.22 (± 0.0)	< 0.001
		9	96.92 (± 0.0)	95.50 (± 0.0)	< 0.001
REC-FSA	MLP	3	81.13 (± 2.5)	80.46 (± 2.53)	< 0.001
		6	90.29 (± 1.91)	89.68 (± 2.5)	< 0.001
		9	96.95 (±0.7)	98.2 (± 0.54)	< 0.001
	Naive Bayes	3	67.89 (±0.0)	70.11 (±0.0)	< 0.001
		6	77.40 (± 0.0)	76.91 (±0.0)	< 0.001
		9	86.04 (±0.0)	84.40 (± 0.0)	< 0.001
	SVM	3	71.37 (±0.0)	70.98 (± 0.0)	< 0.001
		6	82.33 (± 0.0)	83.63 (± 0.0)	< 0.001
		9	95.91 (± 0.0)	93.20 (± 0.0)	< 0.001
VSSRFE	MLP	3	84.49 (± 0.72)	80.5 (± 0.85)	< 0.001
		6	96.29 (± 0.45)	96.3 (± 0.56)	< 0.001
		9	98.67 (± 0.18)	98.55 (± 0.18)	< 0.001
	Naive Bayes	3	72.59 (±0.0)	70.32 (±0.0)	< 0.001
		6	78.95 (± 0.0)	78.37 (±0.0)	< 0.001
		9	86.84 (±0.0)	88.78 (± 0.0)	< 0.001
	SVM	3	76.63 (±0.0)	75.70 (± 0.0)	< 0.001
		6	90.00 (± 0.0)	91.86 (± 0.0)	< 0.001
		9	91.15 (± 0.0)	93.20 (± 0.0)	< 0.001
PISAE	$\tau = 0.5$	2 (± 1)	99.39 (± 0.055)	99.39 (± 0.37)	-
	$\tau = 0.3$	3 (± 1)	99.66 (± 0.21)	**99.62** (± 0.11)	-

lifetime can be enhanced by a factor of 3.4 for Ionosphere, 4.0 for Sensor Discrimination, and 2.25 for Forest CoverType datasets. It has also be shown that the proposed approach consistently outperforms four existing methods on select datasets because of its ability to select most important features in the input space and extract more separable low-dimensional embeddings in the feature space.

In future, we intend to extend this work by collecting new data both at cluster and sink level to enable us train and deploy PISAE model at the sink. We hypothesize that this will tremendously improve network lifetime extension factor since energy of a whole cluster can potentially be conserved by putting all members to sleep provided they are not contributing additional information.

APPENDIX

See Table 3 and 4.

REFERENCES

[1] R. Shelke, G. Kulkarni, R. Sutar, P. Bhore, D. Nilesh, and S. Belsare, "Energy management in wireless sensor network," in *Proc. UKSim IEEE 15th Int. Conf. Comput. Modelling Simulation (UKSim)*, Apr. 2013, pp. 668–671.

[2] X. Jiang *et al.*, "An architecture for energy management in wireless sensor networks," *ACM SIGBED Rev.*, vol. 4, no. 3, pp. 31–36, 2007.

[3] J. A. Khan, H. K. Qureshi, and A. Iqbal, "Energy management in wireless sensor networks: A survey," *Comput. Elect. Eng.*, vol. 41, pp. 159–176, Jan. 2015.

[4] G. Li, S. Peng, C. Wang, J. Niu, and Y. Yuan, "An energy-efficient data collection scheme using denoising autoencoder in wireless sensor networks," *Tsinghua Sci. Technol.*, vol. 24, no. 1, pp. 86–96, 2018.

[5] A. Y. Barnawi and I. M. Keshta, "Energy management in wireless sensor networks based on Naive Bayes, MLP, and SVM classifications: A comparative study," *J. Sensors*, vol. 2016, Jan. 2016, Art. no. 6250319.

[6] S. M. AlTabbakh, E. M. Elshahed, R. A. Ramadan, and H. M. Elzahed, "Energy management of wireless sensor network based on modeling by game theory approaches," in *Proc. IEEE Eur. Modelling Symp. (EMS)*, Oct. 2015, pp. 389–395.

[7] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537–568, May 2009.

[8] M. D. Alwadi and G. Chetty, "Feature selection and energy management for wireless sensor networks," *Int. J. Comput. Sci. Netw. Secur.*, vol. 12, no. 6, p. 46, 2012.

[9] W. Sun, Q. Li, J. Wang, L. Chen, D. Mu, and X. Yuan, "A radio link reliability prediction model for wireless sensor networks," *Int. J. Sensor Netw.*, vol. 27, no. 4, pp. 215–226, 2018.

[10] J. K. Sandhu, A. K. Verma, and P. S. Rana, "A novel framework for reliable network prediction of small scale wireless sensor networks (SSWSNs)," *Fundam. Inform.*, vol. 160, no. 3, pp. 303–341, 2018.

[11] N. M. A. Latiff, C. C. Tsimenidis, and B. S. Sharif, "Energy-aware clustering for wireless sensor networks using particle swarm optimization," in *Proc. IEEE 18th Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2007, pp. 1–5.

[12] B. A. Attea and E. A. Khalil, "A new evolutionary based routing protocol for clustered heterogeneous wireless sensor networks," *Appl. Soft Comput.*, vol. 12, no. 7, pp. 1950–1957, 2012.

[13] B. O. Ayinde and H. A. Hashim, "Energy-efficient deployment of relay nodes in wireless sensor networks using evolutionary techniques," *Int. J. Wireless Inf. Netw.*, vol. 25, no. 2, pp. 157–172, 2018.

[14] H. A. Hashim, B. O. Ayinde, and M. A. Abido, "Optimal placement of relay nodes in wireless sensor network using artificial bee colony algorithm," *J. Netw. Comput. Appl.*, vol. 64, pp. 239–248, Apr. 2016.

[15] R. Asorey-Cacheda, A.-J. García-Sánchez, F. García-Sánchez, and J. García-Haro, "A survey on non-linear optimization problems in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 82, pp. 1–20, Mar. 2017.

[16] Y. Chang, X. Yuan, B. Li, D. Niyato, and N. Al-Dhahir, "Machine-learning-based parallel genetic algorithms for multi-objective optimization in ultra-reliable low-latency WSNs," *IEEE Access*, vol. 7, pp. 4913–4926, 2019.

[17] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1996–2018, 4th Quart., 2014.

[18] A. Panousopoulou, M. Azkune, and P. Tsakalides, "Feature selection for performance characterization in multi-hop wireless sensor networks," *Ad Hoc Netw.*, vol. 49, pp. 70–89, Oct. 2016.

[19] M. K. Lagha, F. A. Aoudia, M. Gautier, and O. Berder, "Feature selection framework for multi-source energy harvesting wireless sensor networks," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–5.

[20] C. Tang and C. S. Raghavendra, "Compression techniques for wireless sensor networks," in *Wireless Sensor Networks*. Boston, MA, USA: Springer, 2004, pp. 207–231.

[21] M. Wu and C. W. Chen, "Multiple bit stream image transmission over wireless sensor networks," in *Sensor Network Operations*. Hoboken, NJ, USA: Wiley, 2006, pp. 677–687.

[22] M. A. Alsheikh, S. Lin, H.-P. Tan, and D. Niyato, "Toward a robust sparse data representation for wireless sensor networks," in *Proc. IEEE 40th Conf. Local Comput. Netw. (LCN)*, Oct. 2015, pp. 117–124.

[23] A. Aziz, K. Singh, W. Osamy, and A. M. Khedr, "Effective algorithm for optimizing compressive sensing in iot and periodic monitoring applications," *J. Netw. Comput. Appl.*, vol. 126, pp. 12–28, Jan. 2019.

[24] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[25] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 40–50, Mar. 2002.

[26] W.-H. Liao, Y. Kao, and C.-M. Fan, "Data aggregation in wireless sensor networks using ant colony algorithm," *J. Netw. Comput. Appl.*, vol. 31, no. 4, pp. 387–401, 2008.

[27] S. Wang, Z. Ding, and Y. Fu, "Feature selection guided auto-encoder," in *Proc. AAAI*, 2017, pp. 2725–2731.

[28] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[29] S. Wang, Z. Ding, and Y. Fu, "Coupled marginalized auto-encoders for cross-domain multi-view learning," in *Proc. IJCAI*, 2016, pp. 2125–2131.

[30] B. O. Ayinde and J. M. Zurada, "Deep learning of constrained autoencoders for enhanced understanding of data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 3969–3979, Sep. 2018.

[31] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Des. Test Comput.*, vol. 18, no. 2, pp. 62–74, Mar./Apr. 2001.

[32] S. Thiemjarus, B. Lo, K. Laerhoven, and G. Yang, "Feature selection for wireless sensor networks," in *Proc. 1st Int. Workshop Wearable Implant. Body Sensor Netw. Imperial*, London, U.K., 2004, pp. 1–2.

[33] J. Csirik, P. Bertholet, and H. Bunke, "Pattern recognition in wireless sensor networks in presence of sensor failures," in *Proc. WSEAS Conf. Recent Res. Neural Netw., Fuzzy Syst., Evol. Comput. Autom.*, N. B. Lupulescu, S. Yordanova, and V. Mladenov, Eds., 2011, pp. 104–109.

[34] Y. Li, Y. Wang, and G. He, "Clustering-based distributed support vector machine in wireless sensor networks," *J. Inf. Comput. Sci.*, vol. 9, no. 4, pp. 1083–1096, 2012.

[35] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. Mach. Learn. Res.*, vol. 11, pp. 1663–1707, Jan. 2010.

[36] K. Flouri, B. Beferull-Lozano, and P. Tsakalides, "Energy-efficient distributed support vector machines for wireless sensor networks," in *Proc. Adjunct 3rd Eur. Workshop (EWSN)*, Zürich, Switzerland, Feb. 2006, pp. 10–11.

[37] R. Dutta and A. Terhorst, "Adaptive neuro-fuzzy inference system-based remote bulk soil moisture estimation: Using CosmOz cosmic ray sensor," *IEEE Sensors J.*, vol. 13, no. 6, pp. 2374–2381, Jun. 2013.

[38] E. S. Sazonov and J. M. Fontana, "A sensor system for automatic detection of food intake through non-invasive monitoring of chewing," *IEEE Sensors J.*, vol. 12, no. 5, pp. 1340–1348, May 2012.

[39] J. Yun and M.-H. Song, "Detecting direction of movement using pyroelectric infrared sensors," *IEEE Sensors J.*, vol. 14, no. 5, pp. 1482–1489, May 2014.

[40] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. IEEE 10th Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Oct. 2005, pp. 1208–1213.

[41] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace ratio criterion for feature selection," in *Proc. AAAI*, vol. 2, 2008, pp. 671–676.

[42] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," in *Proc. 27th Conf. Uncertainty Artif. Intell. (AUAI)*, 2011, pp. 266–273.

[43] J. Wang, H. He, and D. V. Prokhorov, "A folded neural network autoencoder for dimensionality reduction," *Procedia Comput. Sci.*, vol. 13, pp. 120–127, Oct. 2012.

[44] B. O. Ayinde and J. M. Zurada, "Discovery through constraints: Imposing constraints on autoencoders for data representation and dictionary learning," *IEEE Syst., Man, Cybern. Mag.*, vol. 3, no. 3, pp. 13–24, Jul. 2017.

[45] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.

[46] A. Lemme, R. F. Reinhart, and J. J. Steil, "Online learning and generalization of parts-based image representations by non-negative sparse autoencoders," *Neural Netw.*, vol. 33, pp. 194–203, Sep. 2012.

[47] W. Yu, G. Zeng, P. Luo, F. Zhuang, Q. He, and Z. Shi, "Embedding with autoencoder regularization," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2013, pp. 208–223.

[48] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ACM 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.

[49] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area V2," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 873–880.

[50] A. Y. Barnawi and I. M. Keshta, "Energy management of wireless sensor networks based on multi-layer perceptrons," in *Proc. 20th Eur. Wireless Conf. Eur. Wireless*, 2014, pp. 1–6.

[51] G. R. L. Silva, P. C. Neto, L. C. B. Torres, and A. P. Braga, "A fuzzy data reduction cluster method based on boundary information for large datasets," in *Neural Computing and Applications*. London, U.K.: Springer, 2019, pp. 1–10.

[52] H. Pant, M. Sharma, and S. Soman, "Twin neural networks for the classification of large unbalanced datasets," *Neurocomputing*, vol. 343, pp. 34–49, May 2019.

[53] A. Zakeri and A. Hokmabadi, "Efficient feature selection method using real-valued grasshopper optimization algorithm," *Expert Syst. Appl.*, vol. 119, pp. 61–72, Apr. 2019.

[54] Y. Chen, Y. Wang, L. Cao, and Q. Jin, "CCFS: A Confidence-based Cost-effective feature selection scheme for healthcare data classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, to be published.

[55] J. Prajapati and S. C. Jain, "Machine learning techniques and challenges in wireless sensor networks," in *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, 2018, pp. 233–238.

[56] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[57] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.

[58] Z. Li, W. Xie, and T. Liu, "Efficient feature selection and classification for microarray data," *PLoS ONE*, vol. 13, no. 8, 2018, Art. no. e0202167.

[59] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[60] K. Ni *et al.*, "Sensor network data fault types," *ACM Trans. Sensor Netw.*, vol. 5, no. 3, p. 25, 2009.

BABAJIDE O. AYINDE received the M.Sc. degree in engineering systems and control from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, and the Ph.D. degree from the University of Louisville, Louisville, KY, USA. His current research interests include unsupervised feature learning, and deep learning techniques and application in wireless systems. He was a recipient of the University of Louisville Grosscurth Fellowship.

ABDULAZIZ Y. BARNAWI received the B.Sc. and M.Sc. degrees in computer engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 1999 and 2001, respectively, and the Ph.D. degree from Carleton University, Ottawa, ON, Canada, in 2009. He joined the Computer Engineering Department, KFUPM, as an Assistant Professor, in 2009. He has authored and coauthored more than 10 refereed journal and conference publications. His research interests include ad hoc and sensor networks, modeling and simulation of computer networks, and iterative optimization heretics. He contributed in several funded research grants related to wireless sensor networks as a principal investigator and as a member. He is also serving as the Director of the Office of Planning and Quality in KFUPM.