Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 6250319, 12 pages
http://dx.doi.org/10.1155/2016/6250319

Research Article

Energy Management in Wireless Sensor Networks Based on Naive Bayes, MLP, and SVM Classifications: A Comparative Study

Abdulaziz Y. Barnawi and Ismail M. Keshta

Department of Computer Engineering, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia

Correspondence should be addressed to Abdulaziz Y. Barnawi; barnawi@kfupm.edu.sa

Received 21 October 2015; Revised 4 January 2016; Accepted 5 January 2016

Academic Editor: Kewei Zhang

Maximizing wireless sensor networks (WSNs) lifetime is a primary objective in the design of these networks. Intelligent energy management models can assist designers to achieve this objective. These models aim to reduce the number of selected sensors to report environmental measurements and, hence, achieve higher energy efficiency while maintaining the desired level of accuracy in the reported measurement. In this paper, we present a comparative study of three intelligent models based on Naive Bayes, Multilayer Perceptrons (MLP), and Support Vector Machine (SVM) classifiers. Simulation results show that Linear-SVM selects sensors that produce higher energy efficiency compared to those selected by MLP and Naive Bayes for the same WSNs Lifetime Extension Factor.

1. Introduction

A wireless sensor network (WSN) can be defined as a networked collection of sensor nodes. These sensors are small-scale devices with very limited resources. In a typical sensor network, each node has to monitor environmental and physical conditions such as temperature, sound, humidity, pressure, vibration, motion, and light. Depending on the objective of the sensor network, sensors may cooperate to carry out specific tasks. Also, the generated data is highly correlated due to the nature of the monitored parameters and due to the large number of deployed sensors. Therefore, it is a waste of resources to report each individual sensor reading; hence, energy-efficient protocols and network models, which utilize information fusion, are extremely important [1].

AlTabbakh et al. [2] emphasized that an energy-efficient management scheme has to be employed in order to extend network lifetime. Sensor nodes waste a lot of their energy in data communication. So, reducing the amount of unnecessary communication will help in minimizing energy waste and extending network lifetime. Also, Khan et al. [3] pointed out that designing an effective energy management scheme for sensor networks, especially those deployed in remote areas, is one of the main challenges facing WSNs. Noticeable research efforts have addressed this issue by proposing intelligent models based on machine learning [4]. For example, Abu Alsheikh et al. [5] emphasized the advantage of adopting machine learning algorithms in WSNs since they assist in eliminating unneeded redesign issues. The authors have also indicated that machine learning algorithms have contributed to the development of practical solutions that result in extending the network lifetime. In addition, several reasons were given to stress the importance of adopting machine learning algorithms in WSN environmental monitoring applications. First, sensor nodes might not operate as expected because of unexpected environmental behavior. In such cases, machine learning algorithms can overcome such problems by adjusting themselves to the newly obtained knowledge. Second, due to the unpredictable environments where WSNs are deployed, the resulting mathematical models for the network might be very complex. Machine learning algorithms can assist in developing attractive and

less complex solutions. Third, sensor nodes generate large amounts of correlated and redundant data. Machine learning algorithms are powerful tools that can be used to study and identify correlated data, making decisions, predictions, and data classification [1, 6].

Algorithms such as Naive Bayes, Multilayer Perceptron (MLP), and Support Vector Machine (SVM) are examples of well-known algorithms that are also widely adopted and investigated in the area of machine learning, neural networks, and artificial intelligence. For starters, MLP can perform the classification operation with significant success [7, 8]. On the other hand, MLP neural network training is difficult due to the complexity of its structure [7]. SVM is also considered to be a very powerful algorithm in the field of data mining. It has been successfully applied in a wide range of scientific applications [9].

Despite the importance of machine learning algorithms for WSN applications [5], little attention has been given to provide comparison studies between these algorithms, especially when it comes to energy management of WSNs. In addition, little contributions have been made to specify an appropriate intelligent energy management model for these networks. To our knowledge, there is no previous work that supports the use of a specific intelligent algorithm compared to others for WSNs [10] except the one that we have reported in our previous work [11]. In that work, we propose an intelligent and efficient energy management model for WSNs using MLP, instead of Naive Bayes. A comparative study between MLP and Naive Bayes was provided in order to evaluate their performance in terms of the percentages of accurate classification. Simulation results showed that, given the same Lifetime Extension Factor, MLP achieves a significant improvement in selection accuracy compared to Naive Bayes. However, MLP takes longer time to train the network, due to its complexity and the update process in the series of weights. Consequently, sensor nodes may consume slightly more energy in the deployment phase.

In this work, we extend the work presented in [11] by conducting a detailed and comprehensive comparative study across three intelligent classification algorithms, namely, Naive Bayes, MLP, and Linear-SVM. We chose to include Linear-SVM in this work because it is fast and energy efficient. Unlike MPL, Linear-SVM consumes less energy during the learning process in the deployment phase [11]. Furthermore, both Gupta and Ramanathan [12] and Magno et al. [13] highlighted that Linear-SVM is a classifier with low complexity. Magno et al. [13] have also indicated that Linear-SVM provides a good balance between the percentage of correctly classified data and the computational and memory cost. In addition, Linear-SVM can be effectively implemented within the microcontroller of the sensor node [13, 14]. Sazonov and Fontana [14] presented two powerful characteristics of SVM: high generalization and robustness. Also, Bal et al. [15] deduced that SVM with linear kernel is a promising algorithm in the field of machine learning. Moreover, Yuan et al. [16] were able to conclude that SVM classifier, especially with linear kernel, has the ability to learn and build the required knowledge from fewer training samples and still provides high classification accuracy, unlike other classifiers such as MLP.

Our major contributions in this work are as follows. First, we have provided detailed state-of-the-art related energy management in WSNs based on intelligent machine learning models and algorithms. Second, we carried out a comprehensive comparison among three intelligent neural network classifiers using three real labelled benchmark datasets. Third, the performance of these three intelligent classifiers is examined using a confusion matrix. Finally, we have proposed an intelligent model for efficient energy management in WSNs. This model has been addressed not only at the classification level, but also at the processing level, where we employed the statistical ranking and selection methods. These two methods, in addition to using Linear-SVM as an intelligent classifier, are core elements in building the model. We have also discussed the evaluation of this proposed model.

The rest of this paper is organised as follows. Section 2 reviews the state of the art in the area of energy management in WSNs based on intelligent models. Section 3 provides a background of the classification algorithms used in this work, that is, Naive Bayes, MLP, and SVM. It also presents an overview of the experimental datasets. The main contribution of this paper is discussed in Section 4. Simulation experiments and their setup are discussed in Section 5. Section 6 discusses and summarizes simulation results. Finally, Section 7 concludes the paper and highlights future research directions.

2. Related Work

Several intelligent models have been proposed in order to achieve better energy efficiency in WSNs. Thiemjarus et al. [17] proposed a sensor selection technique that can help determine the optimal number of sensor nodes in the network. By doing this, the number of sensor nodes can be reduced without degrading the decision process, and the lifetime of the network can be increased. The Bayesian approach was used for sensor selection, as this filter method helps to find the optimal sensors in the network. In addition, the Self-Organising Map (SOM) was used as a classifier. In [18], the authors proposed a selection scheme for minimizing the energy consumption of WSNs. In their scheme, sensors were ranked from the most to the least significant, based on the significance of their use in the WSNs. Then, the Naive Bayes classification algorithm was used. This approach was tested on three well-known real sensor datasets. The results showed that more energy is consumed if more sensors are used and, subsequently, the lifetime of the sensor network is reduced. However, if the sensors are ranked, the lifetime of the sensor network can be increased, as the selection algorithm is used, followed by the intelligent classifier. This is because the number of selected sensors that are used is less. Similarly, [19] proposed a scheme to both minimize the energy consumption and maximize the lifetime of the sensor network. This is based on a feature/sensor selection that minimizes the number of the used sensors. However, it uses a different selection algorithm and K-Nearest Neighbor (KNN) classification algorithm.

In [20], the clustering problem is viewed as a classification problem such that clusters are built by using Least Squares SVM (LS-SVM) with a hybrid kernel, which is a mixture of polynomial and Radial Basis Function (RBF) kernels. Results showed that the clustering process using LS-SVM with hybrid kernel had better results compared to LS-SVM with a single kernel. This was obvious in the case where we had a multiclass classification problem in the clustering problem. Also, Li et al. [21] addressed the energy management in the WSNs at the architecture level by using SVM classifier. A clustering-based distributed Linear-SVM algorithm, called CDLSVM, was developed, which differs from other parallel SVM algorithms in its ability to obtain a global optimal classifier. The results showed that the proposed algorithm is efficient for large-scale WSN because it saves the information exchange and energy consumption.

Forero et al. [22] addressed distributed classification by developing algorithms in order to train SVM in an environment that has distributed architecture and where the communication between different nodes through a centralized processing unit is prohibited. The distributed modes of operation showed that energy saving can be made. The distributed fix partition SVM (DFP-SVM) and the weighted distributed fix partition SVM (WDFP-SVM) are two energy-efficient distributed learning algorithms for WSN that were proposed based on SVM [23, 24]. The goal was to train an SVM efficiently and in a distributed manner. Hence, better classification results were achieved on the test data with minimum energy cost, compared to traditional SVM.

More recently, more classification algorithms based on SVM are being proposed. For example, Rajasegarar et al. [25] addressed the problem of anomaly detection in WSNs by proposing a Centered Hyperellipsoidal Support Vector Machine (CE-SVM) and a distributed Quarter-Sphere Support Vector Machine (QS-SVM). A comparison study of CE-SVM and QS-SVM was carried out by using four datasets that corresponded to wireless sensor datasets. Four datasets, namely, GDI, Ionosphere, Banana, and Synth, were used for evaluation. In addition, three different kernel functions, that is, RBF, polynomial, and linear, were considered. The results showed that CESVM achieved better detection accuracy, compared to QSSVM. Zhang et al. [26] proposed two distribution and online outlier detection techniques for WSNs, and the main contribution of these techniques is to build real-time intelligent classification techniques in order to identify outliers in the normal behavior of sensor data in real time. The proposed intelligent techniques are based on hyperellipsoid one-class SVM. These two techniques are ellipsoidal SVM-based online outlier detection (EOOD) and ellipsoidal SVM-based adaptive outlier detection (EAOD). Both EOOD and EAOD consider the correlation between sensor data features in order to detect anomalies in a WSN. The simulation results show that EAOD achieved better detection accuracy, compared to EOOD.

Dutta and Terhorst [27] introduced an artificial intelligence-based approach for multisensor fusion and integration. The researchers addressed the intelligent sensor fusion system at the classification level to overcome the problem of common sensor errors and faults. Particularly, intelligent classification techniques for bulk soil moisture estimation using cosmic ray sensor were introduced to achieve optimal fault-tolerant estimates. In this study, four classifiers were evaluated, namely, Adaptive Neuro-Fuzzy Inference System (ANFIS), Probabilistic Neural Network (PNN), Radial Basis Function (RBF) network, and Multilayer Perceptron (MLP) network. Experiments were conducted over three datasets and the results show that ANFIS gives the best performance in terms of correct classification percentages compared to other classifiers. Sazonov and Fontana [14] proposed a sensor system and its related signal processing and pattern recognition methodologies to detect periods of food intake. This study was based on the monitoring and classification of jaw motion. The researchers used the forward selection method to choose the most relevant features, which represented sensor signals. Also, the SVM was used as the classification algorithm for chewing detection. In addition, the kernel activation function used was linear (Linear-SVM). By using such intelligent classifier, the researchers were able to achieve high averaged accuracy, that is, 80.98%, for that application. Furthermore, according to Karatzoglou et al. [28], the linear kernel is the simplest of all the kernel functions for SVM. Ham et al. [29] also indicated that Linear-SVM performs better than other machine learning classifiers. A lot of attention has currently been given to this classifier due to its high performance, in terms of classification accuracy, compared with other machine learning classifiers. Thus, this intelligent classifier has been applied in various applications. Yun and Song [30] proposed a moving human direction detecting system. The system was based on Pyroelectric Infrared (PIR) sensors and intelligence classification techniques. In this study, the dataset was collected by capturing PIR sensor signals while a person was walking. Extracting and selecting distinguishable features from the dataset reduced its size and likewise the memory cost. In addition, seven intelligent classification algorithms, namely, Bayes Net, C4.5, decision table, *K*NN algorithm, Naive Bayes, MLP, and SVM, were used. For SVM, the Linear-SVM was used because its kernel demanded less computation cost compared to other kernels. Experiments which were conducted on the reduced feature sets showed that Linear-SVM is the best classifier for such application.

3. Preliminaries

This section provides a brief background about the classification algorithms used in this study, as well as about the datasets for the experimental comparison.

3.1. Classification Algorithms. The classification algorithms used in this research project are briefly described below.

3.1.1. Naive Bayes Classifier. Naive Bayes is a well-known type of classifier that is based on the application of Bayes' theorem with strong independence assumptions. It is considered to be a simple probabilistic classifier that computes conditional class probabilities and then predicts the most probable class [31]. In other words, it will assign a class for an object based on the values of the descriptive attribute probability model.

3.1.2. Multilayer Perceptron (MLP). MLP is composed of a large number of highly interconnected neurons that are working in parallel to solve a certain problem. It is organised in layers with a feed-forward information flow. The main architecture of an MLP network consists of signals that flow sequentially through the different layers from the input to the output layer. Between the input layer and the output layer are intermediate layers. They are also called hidden layers because they are not visible at the input or at the output. Each unit is first used to calculate the difference between a vector of weights and the vector given by the outputs of the previous layer. To generate the input for the next layer, a transfer function also called activation is applied to the result [32]. Bipolar sigmoid, unipolar sigmoid, and RBF are examples of well-known and commonly used activation functions [33].

The main steps of the training phase in an MLP network are as follows: first, given an input pattern X of the dataset, this pattern is forward-propagated to the output of the MLP network and then compared with the desired output; second, the error signal between the output of the network and the desired response is back-propagated to the network; and finally, adjustments are made on the synaptic weights [34]. This process is repeated for the next input vector until all the training patterns are passed through the network.

3.1.3. Support Vector Machine (SVM). SVM splits the dataset into two classes, which are separated by placing a linear boundary between the normal and attack classes in such a way that the margin is maximized. SVM works to find the hyperplane that gives the maximum distance from the hyperplane to the closest positive and negative samples [35, 36]. The basic structure of an SVM network is similar to that of the ordinary RBF network, but instead of the exponential activating function (usually Gaussian activation functions), the kernel activating function is applied. The kernel activating function can be a polynomial kernel, Gaussian radial basis kernel, or two layer feed-forward neural network kernels.

3.2. Dataset. This section provides a brief description of the three datasets used in this work.

3.2.1. Ionosphere. Ionosphere dataset is a radar dataset 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 [18]. This dataset consists of 34 readings and 351 records.

3.2.2. Forest CoverType. Forest CoverType dataset has been developed at the University of Colorado. It is designed to predict the forest cover type of unknown regions. It is used in several research papers on data stream classification. This dataset contains 581, 012 instances and 54 attributes [18, 37]. It is considered to be one of the largest datasets. In our research, the number of records is reduced to 25000 because of limited memory capacity.

3.2.3. Sensor Discrimination. This dataset consists of 12 different numerical values of samples of an unknown substance. It is a labelled dataset that has three classes, namely, group A, group B, and false alarm [38]. The receiver node, which receives 12 numerical values for an unknown sample, is designed to determine which class this 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. This dataset consists of 12 readings and 2211 records.

4. The Proposed System

In this work, an intelligent neural network model for efficient energy management in WSNs is presented with the use of the classification algorithm. Our work adopts the same selection sensor algorithm that was used in [18]. In this study, we have used Ionosphere, Forest CoverType, and Sensor Discrimination datasets to evaluate the performance of the three intelligent algorithms in terms of classification accuracy. In addition, for all conducted experiments, 30% of the dataset is used for training data, and the remaining 70% is used for testing. This allows us to present a fair comparison among the three classifiers, namely, Naive Bayes, MLP, and Linear-SVM, by fixing the shared and common parameters. In addition, as in [18], the Lifetime Extension Factor is given by

$$\text{LTEF} = \frac{\text{Total number of sensors}}{\text{Number of sensor used}}. \quad (1)$$

In [11], two important steps were carried out in order to minimize the energy consumption and extend the lifetime factor. These two steps are as follows: (1) the most dominant sensor nodes in WSNs were selected after ranking them based on the significance of use, from the most significant to the least, and (2) Naive Bayes and MLP classification algorithms were applied. It is important to emphasize that, in our research work, an intelligent model for efficient energy management in WSNs is introduced by means of the classification algorithm by using SVM with linear kernel which is a polynomial kernel with exponent 1. Our proposed scheme operates in four stages, which are as follows:

(1) Preprocessing.

(2) Processing:

  (a) Ranking:

    (i) Calculate significance level of feature/sensor.

    (ii) Sort in a descending order.

  (b) Selection.

(3) Machine learning.

(4) Performance evaluation.

Figure 1 shows the block diagram of the proposed system. Under the dataset block, we have three datasets, namely,

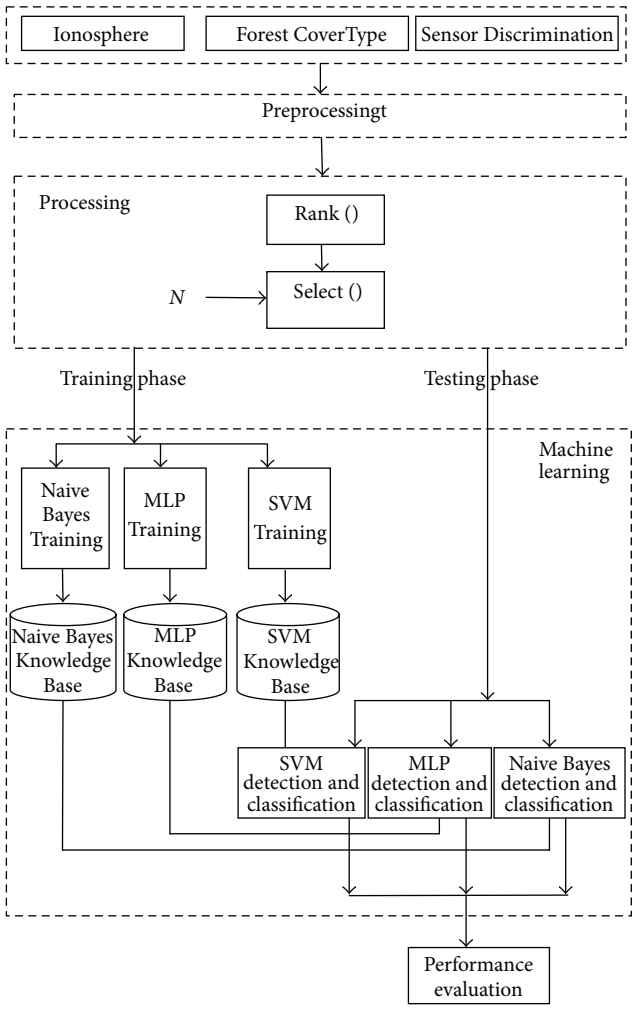

Figure 1: Block diagram of the proposed system.

Ionosphere, Forest CoverType, and Sensor Discrimination. The preprocessing stage is needed to clean the input dataset and put it in the proper format. In the processing block, two functions are performed on the selected dataset. The first is the Rank function that ranks the features/sensors from the most significant to the least significant according to their significance of use in the dataset which corresponds to wireless sensor networks, as proposed in [18]. This function includes two sequential processes, namely, the significance level calculation process and the sorting process. Calculating the significance level of feature/sensor is carried out using a built-in procedure in MATLAB [39] called independent significance features test. Weiss and Indurkhya [40] named this procedure as independent significance features test (IndFeat). This process is usually employed in the field of engineering applications of artificial intelligence to identify and discard weak features quickly. In other words, the number of inputs needed to be considered in the selection process will be significantly decreased. Thus, it can be used as a precursor to the selection process. Moreover, it reduces the computational time required for the final selection process. On the other hand, the sorting process is used to sort the output of the significance level calculation processes in a descending order. The final output of Rank function is an array of features/sensors sorted from the most significant to the least significant. The role of the Rank function is to calculate the significance levels of each feature/sensor. Then, it sorts them in a descending order; that is, features or sensors will be sorted from the most significant to the least significant.

Figure 2 shows an example of the output when calculating the significant level process and sorting process on Sensor Discrimination dataset. The figure clearly explains what processes will be performed sequentially in the Ranking function. Thereafter, the Select function is used to select the first N features/sensors. For example, if $N = 10$, that means this method returns the numbers of the first 10 features/sensors. Similarly, if $N = 20$, it returns the numbers of the first 20 features/sensors and so on. The fourth block is the machine learning block, which runs the selected classifier Naive Bayes, MLP, or Linear-SVM on the N selected features/sensors. In this block, two phases are performed; the first is the training or learning phase which allows the intelligent system to build the right knowledge base. The intelligent system learns relationships/correlations existing in the constructed

Table 1: Experimental parameters.

Parameter	Value
% training	30%
% testing	70%
Learning rate for MLP	0.3
Number of epochs for MLP	500
Number of hidden layers for MLP	1
Number of hidden neurons for MLP	(# selected sensors + # classes)/2
Kernel for SVM	Linear kernel (polynomial kernel with exponent being 1)

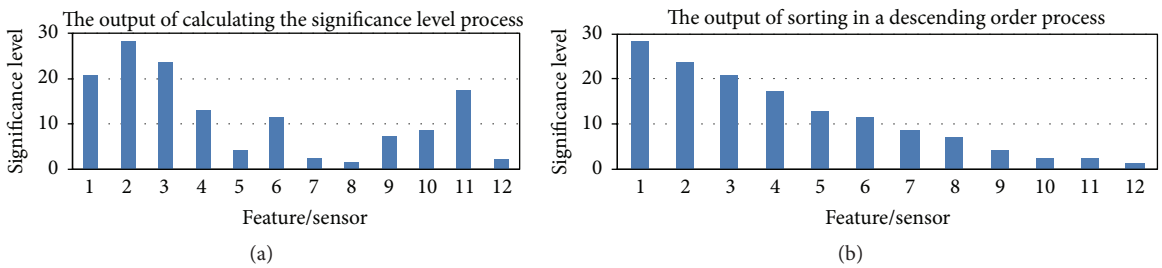

Figure 2: The output of Rank process on the Sensor Discrimination dataset.

training dataset. Simply, the training phase is considered as adaptation process to the intelligent system to give the best possible response in the testing phase. The testing phase is used to test the proposed system using test datasets. The output results of this phase show the performance of the system and its effectiveness in terms of classification accuracy and computational speed.

5. Experimental Work

Three different experiments were conducted on the three datasets. In each experiment, we have adopted the same algorithm in [18] for selecting and ranking dominant sensor nodes. In addition, for all experiments, we have used 30% of the dataset for training and the remaining 70% for testing. Also, WEKA simulator v3.6 [41] is used in the classification process. For the MLP and Linear-SVM algorithms, Table 1 lists the values of the important parameters such as learning rate, number of epochs (number of passes through data), number of hidden layers, number of hidden units in hidden layers, and kernel type used in the SVM. All other parameters were set to their default settings in WEKA.

5.1. Experiment 1. This experiment was conducted on Ionosphere dataset, which is a radar dataset collected from 34 different real sensor nodes. MATLAB was used to perform the selection algorithm proposed in [18]. This selection algorithm (see Figure 3) ranked the sensors on the significance of their use, from the most to the least significant.

Table 2 shows the results of this sensor ranking on the Ionosphere dataset. The first row of the table shows the number of the 10 most significant sensors. We also selected this case in order to provide it with a NN structure (as in Figure 3), since it is small in size. The numbers in the first row of Table 2 are exactly the same as those in the NN structure in Figure 3. Therefore, the inputs into the NN structure for the selection of 10 in Figure 3 are S2, S3, S5, S7, S1, S9, S31, S33, S29, and S21, which are exactly the same numbers as the first row in Table 2. It is important to point out here that there is one hidden layer in Figure 3 with six hidden units. This is because if we apply the formula in Table 1, given by (# selected sensors + # classes)/2, the number of selected sensors is 10. It is also clear from the specification of the Ionosphere dataset that the number of classes/outputs is 2. Therefore, there are $\lfloor (10+2)/2 \rfloor = 6$ hidden units.

If we consider the first and the second rows together, it is clear which 20 sensors were the most significant. Consider the best 20, for example. If we look at Table 2, it is clear that the feature/sensor that had number 2 was the most significant, while that which had number 6 was the least significant. Similarly, the best selection of 30 features/sensors was calculated by considering the first, second, and third rows, with the feature/sensor that has number 2 being the most significant and that with number 32 being the least significant.

5.2. Experiment 2. This experiment was conducted on the Forest CoverType dataset, which has been developed at the University of Colorado. By examining the received data from the sensors, the Forest CoverType of unknown regions, which contains 581,012 records and 54 incoming values, can be predicted. The same steps that were taken during Experiment 1 were also performed here (see Figure 4) and the selection results are shown in Table 3. The first row of Table 3 represents the best selection of 10. The first and the second rows represent the best selection of 20. The first, second, and third rows represent the best selection of 30, and so on. For example, for the case of the best 30 features/sensors, feature/sensor number 15 was the most significant, while that which had number 49 was the least significant. It is

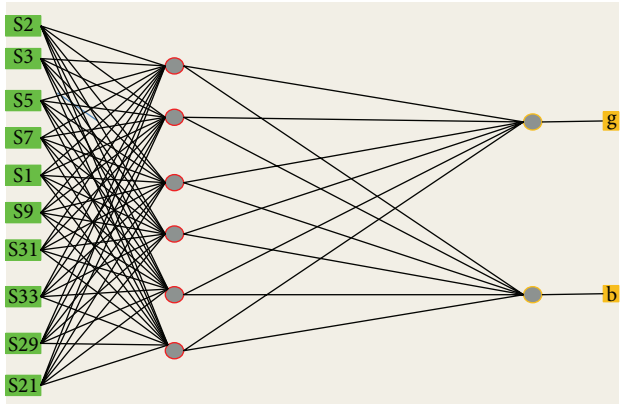

Figure 3: Experiment 1, NN structure for the selection of 10.

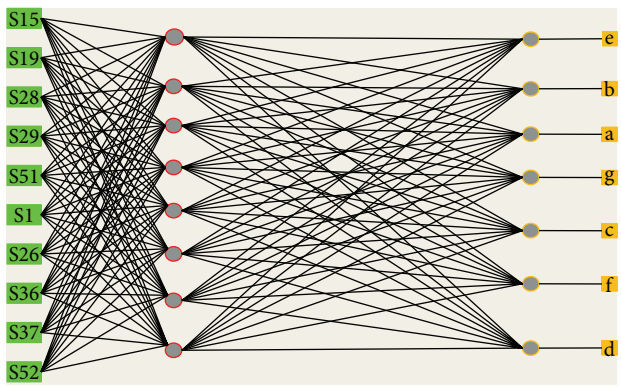

Figure 4: Experiment 2, NN structure for the selection of 10.

Table 2: Selection and ranking on Ionosphere dataset.

	1	2	3	4	5	6	7	8	9	10
1	2	3	5	7	1	9	31	33	29	21
2	15	23	8	13	25	14	11	12	16	6
3	19	10	18	22	27	4	17	34	28	32

Table 3: Selection and ranking on Forest CoverType dataset.

	1	2	3	4	5	6	7	8	9	10
1	15	19	28	29	51	1	26	36	37	52
2	24	53	12	25	27	54	44	14	18	43
3	10	6	32	8	40	17	48	38	20	49
4	16	35	42	7	33	5	23	3	13	31
5	30	4	45	2	11	21	41	9	39	22

important to highlight here that the features/sensors were ranked according to their significance of use in the Forest CoverType dataset.

As in Experiment 1, the selection of 10 was chosen to provide its NN structure, as shown in Figure 4. The inputs to the NN structure for the selection of 10, as shown in Figure 4, are S15, S19, S28, S29, S51, S1, S26, S36, S37, and S52, which are exactly the same as the first row in Table 3. In this case, the feature/sensor that has the number 15 is the most significant, while that which has the number 52 is the least significant one. It is clear from the specification of the Forest CoverType dataset that the number of selected sensors was 10 and the number of classes/outputs was 7. By applying the formula in Table 1, we get $\lfloor (10+7)/2 \rfloor = 8$. This means there are 8 hidden units in the hidden layer between the input layer and output layer, which is exactly the same (as indicated in Figure 4) and shows the NN structure for the selection of 10 on the Forest CoverType dataset.

5.3. Experiment 3. This experiment was conducted on the Sensor Discrimination dataset, which consisted of 12 readings and 2,213 records. The receiver node should be able to determine the unknown sample and indicate which class the unknown sample belongs to. The full records were used in this research work. As in Experiments 1 and 2, MATLAB was used to run the selection algorithm (see Figure 5). Table 4 shows the best selection of 3, 6, and 9, respectively. In other words, the first row shows the best selection of 3. The first and second rows provide the best selection of 6. Finally, the first, second, and third rows show the best selection of 9. For example, the features/sensors were ranked as follows for the selection of 6:

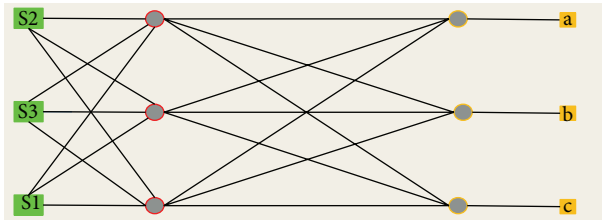

Figure 5: Experiment 3, NN structure for the selection of 3.

Table 4: Selection and ranking on Sensor Discrimination dataset.

	1	2	3
1	2	3	1
2	11	4	6
3	10	9	5

Table 5: Confusion matrix.

		Predicted class	
		Positive	Negative
Actual class	Positive	TP	FP
	Negative	FN	TN

S2, S3, S1, S11, S4, and S6, according to the significance of their use in this sensor dataset.

The NN structure for the selection of 3, as performed in Experiments 1 and 2, is provided in Figure 5. This figure shows the inputs to NN, which are S2, S3, and S1. These are exactly in the same row as in Table 4. Since the number of selections was 3 and the number of classes/outputs was 3, as indicated in the specifications for the Sensor Discrimination dataset, the number of hidden units in the hidden layer was $(3+3)/2 = 3$, which is exactly the same, as shown in Figure 5.

It is important to point out that Tables 2, 3, and 4 show the results of the sensor ranking approach that is proposed in [8], which ranks the features/sensors in decreasing order with respect to their significance of use in the Ionosphere, Forest CoverType, and Sensor Discrimination datasets, respectively. These public datasets correspond to wireless sensor networks.

6. Discussion of Results

The performance of the three intelligent classification algorithms is measured by using the confusion matrix [30, 42]. This matrix gives visualization of how the classifier has performed on the input dataset. Different performance metrics, such as accuracy, recall, and specificity, can be derived from this matrix. Table 5 shows the structure of the confusion matrix. There are four possible cases/outcomes, which are false positive (FP), true positive (TP), false negative (FN), and true negative (TN) [42, 43], where

(1) FP happens when actual class of test sample is negative and is classified incorrectly as positive;

(2) TN happens when actual class of test sample is negative and is classified correctly as negative;

(3) FN happens when actual class of test sample is positive and is classified incorrectly as negative;

(4) TP happens when actual class of test sample is positive and is classified correctly as positive.

We used the accuracy as the performance metric in this study to evaluate these algorithms. Accuracy represents the overall correctness of the intelligent classification of a dataset and it is given by

$$\text{Accuracy} = \frac{(\text{TN} + \text{TP})}{(\text{TN} + \text{TP} + \text{FP} + \text{FN})}. \quad (2)$$

For example, Tables 6, 7, and 8 show the confusion matrixes for Naive Bayes, MLP, and Linear-SVM, respectively, over the Ionosphere dataset with a selection of 10. They reveal how many instances are assigned to each class. The number of correctly classified samples is represented by the sum of the diagonals. For example, the total number of correctly classified samples for Naive Bayes is 187, that is, the sum of 123 and 64. The accuracy in this case is $187/246 * 100 = 76.02\%$, where the total number of instances in the testing dataset is 246. However, the accuracy is 84.14% and 85.36% for MLP and Linear-SVM, respectively. The same procedure is applied to the remaining dataset with a different selection over the three classifiers.

Figure 6 shows the overall results of Experiment 1, which was conducted on the Ionosphere dataset. It shows that both Linear-SVM and MLP give better results for all three of the selection cases, that is, the selection of 10, 20, and 30 sensors compared to Naive Bayes, in terms of accuracy over the same Lifetime Extension Factor. For example, when 10 sensors out of 34 were selected, the Lifetime Extension Factor was $10/34 = 3.4$ (see Figure 10) and the accuracy was 85.4% and 84.1%, using Linear-SVM and MLP, respectively. However, using Naive Bayes resulted in there being 76% accuracy for the same Lifetime Extension Factor. Moreover, it can be seen that Linear-SVM achieves the best result in the first and second selection, especially in the case of selecting 20. When it comes to Linear-SVM, this powerful classifier maintains its high performance in the first and second selection. Even when 30 were selected, there was a slight drop of around 2%, which is negligible.

For MLP, it can be noted that the accuracy of MLP does not increase significantly, despite the growth in the number of selected sensors. For example, in the case of selecting 30 sensor nodes, MLP achieved 84.9% accuracy, which is a negligible 0.8% improvement on selecting 10. There was

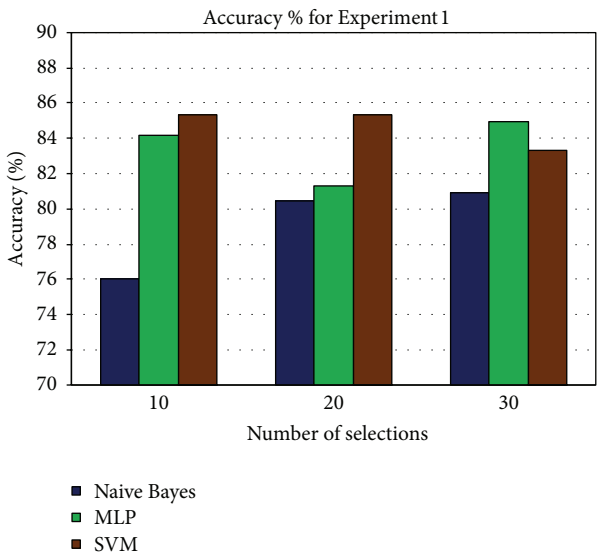

Figure 6: Selection accuracy of Experiment 1.

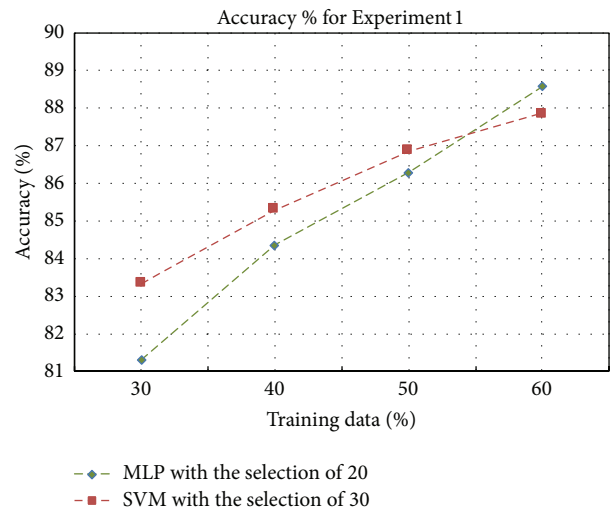

Figure 7: Percentages of the training data versus the accuracies for Experiment 1.

Table 6: Confusion matrix of Naive Bayes for the Ionosphere dataset with selection of 10.

		Predicted class	
		Good (g)	Bad (b)
Actual class	Good (g)	123	37
	Bad (b)	22	64

Table 7: Confusion matrix of MLP for the Ionosphere dataset with selection of 10.

		Predicted class	
		Good (g)	Bad (b)
Actual class	Good (g)	148	12
	Bad (b)	27	59

Table 8: Confusion matrix of SVM for the Ionosphere dataset with selection of 10.

		Predicted class	
		Good (g)	Bad (b)
Actual class	Good (g)	152	8
	Bad (b)	28	58

also a drop in accuracy for MLP when 20 sensor nodes were selected, compared to 10. There was a slight drop in accuracy for MLP and Linear-SVM when selecting 20 and 30, respectively. Apparently, this was because the training dataset was not large enough to build the right knowledge base for these two classifiers. In other words, the result obtained could be explained by taking into account the number of training samples used, so that the capabilities of both MLP and Linear-SVM are not completely utilized. This point can be verified by conducting multiple test runs for different percentages of the training data, as is illustrated in Figure 7. In this figure, we were able to show that the accuracy increases with the number of training samples; that is, when the number of training samples grows, the classification accuracy will increase. This issue is also confirmed and corroborated by Sakar et al. [44], Fernandez-Rodriguez et al. [45], and Murty and Devi [46].

It is important to note here that if percentages of the training data increase, both computational complexity and the training time will increase as well. Therefore, 30% of the dataset is used for training in order to have good balance between the percentage of correctly classified data and the computational and memory cost of WSN applications. For Experiment 1, it can therefore be concluded that the selection of the 10 nodes with Linear-SVM as an intelligent classifier is the best choice, as it results in a higher Lifetime Extension Factor. Hence, it is more energy efficient. Even though MLP and Linear-SVM were closer to each other when it came to the accuracy levels, Linear-SVM was selected as the best classifier, instead of MLP in particular, because it took significantly less time, on average, to build the classifier model on the full training set for Linear-SVM than it did for MLP.

Experiment 2, on the other hand, was performed on the Forest CoverType dataset. Simulation results showed that MLP produces better accuracy over the same Lifetime Extension Factor, compared to Naive Bayes and Linear-SVM, when 20, 30, 40, and 50 were selected. In these four cases, the accuracy of Naive Bayes and Linear-SVM was almost at the same level, at around 70%. However, MLP achieved 68%, 76.7%, 77.7%, and 78.5%, respectively, in the four selection cases (see Figure 8). Despite this improvement in MLP, Naive Bayes behaved slightly better when 10 sensor nodes were selected. Its accuracy was higher by 3%, compared to MLP. It is important to say that the difference in accuracy levels between MLP and Linear-SVM was around 5.5% on average. This cannot be considered as a huge, significant improvement if we take into account the fact that MLP had the highest running time in building the classifier model in all of the selection cases. The selection of 30, with Linear-SVM as a

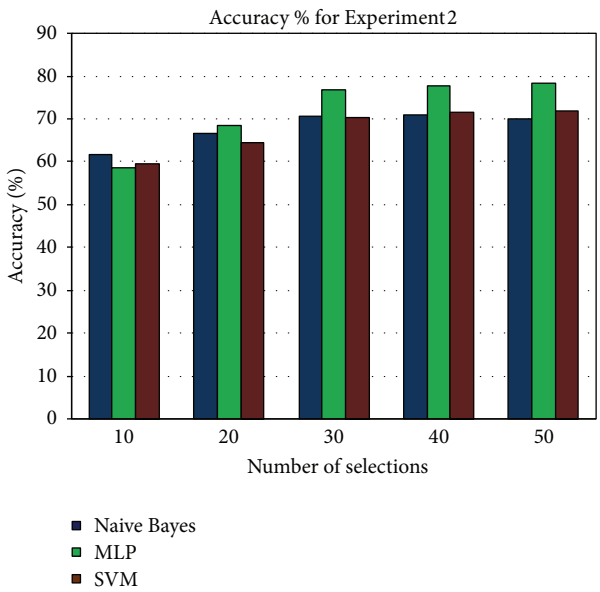

Figure 8: Selection accuracy of Experiment 2.

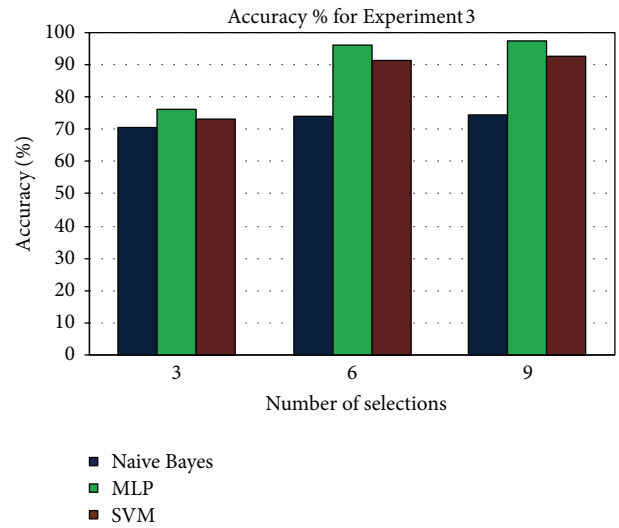

Figure 9: Selection accuracy of Experiment 3.

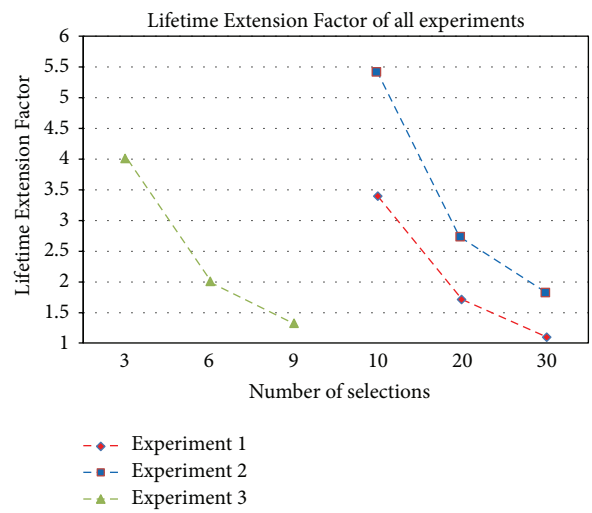

Figure 10: Lifetime Extension Factor of all experiments.

classifier, is therefore the best selection, if we consider both the accuracy and saving in computational time that was taken.

Experiment 3 was conducted on the Sensor Discrimination dataset and the results are shown in Figure 9. Both the MLP and Linear-SVM algorithms have better results compared to Naive Bayes for the three selection cases, which was the selection of 3, 6, and 9 sensors. MLP achieved accuracy levels of 76%, 95.9%, and 97.2% for these selection cases, respectively. For Linear-SVM, it is 73%, 91%, and 93%, respectively. Naive Bayes, however, has around 73% accuracy on average for all of the cases. Therefore, even though MLP is slightly better than Linear-SVM for this dataset, the best selection scenario is the selection of 6 sensor nodes with Linear-SVM as a classifier. This is because Linear-SVM was faster than MLP in building the model out of the full training set. In addition, there was a selection of 6 results in a Lifetime Extension Factor of 2 (see Figure 10).

It is worth mentioning that, despite its high accuracy, MLP takes more time to train the network compared to Naive Bayes and Linear-SVM as shown in Tables 9, 10, and 11. For example, the longest training time for Experiment 1 occurred when 30 features/sensors were selected (see Table 9). In this case, MLP increased the time by a factor of 206 and 30 compared to Naive Bayes and Linear-SVM, respectively. MLP was even worse compared to Naive Bayes, as shown in Table 10. This was due to its complexity and the update process of the series of weights. Furthermore, as the number of selected nodes increased, so did the size of the inputs to the model.

All of the above simulation times are based on our experiments which were carried out on a PC with Intel Atom CPU (N450) @ 1.66 GHz and 2.00 GB of RAM running 32-bit Windows 8 Enterprise Edition. Therefore, such models are very applicable in sensor networks with an end system that can perform the training process in a reasonable time. However, if the training process is executed by normal sensor nodes, even the powerful ones, then we expect a considerable amount of energy consumption and delays, especially in the case of MLP. In this case, we believe that SVM is a viable choice. This is because it is characterized by fast training times and high scalability compared to MLP. In addition, it has better accuracy compared to Naive Bayes.

7. Conclusions and Future Work

This paper presents an intelligent model for efficient energy management in WSNs. It provides a comparative study among Naive Bayes, MLP, and Linear-SVM. The aim is to determine the most appropriate intelligent classification model for efficient energy management in WSNs. In order to evaluate the performance of the three classifiers, different experiments were conducted on three benchmarking

Table 9: Average time taken, in seconds, to build the models in Experiment 1.

Number of selections	Naive Bayes	MLP	SVM
10	0.12	7.16	1.03
20	0.14	18.39	1.13
30	0.17	35.08	1.17

Table 10: Average time taken, in seconds, to build the models in Experiment 2.

Number of selections	Naive Bayes	MLP	SVM
10	1.49	820.13	64.27
20	2.45	1746.78	405.77
30	5.36	3546.74	703.71

Table 11: Average time taken, in seconds, to build the models in Experiment 3.

Number of selections	Naive Bayes	MLP	SVM
3	0.14	16.24	2.05
6	0.17	29.51	2.11
9	0.59	45.22	2.29

datasets, namely, Ionosphere, Forest CoverType, and Sensor Discrimination. These datasets correspond to different WSNs that are used in various types of applications.

Simulation results on both Ionosphere and Sensor Discrimination datasets show that, given the same Lifetime Extension Factor, Linear-SVM and MLP algorithms achieve a significant improvement in selection accuracy compared to Naive Bayes. On the Forest CoverType dataset, MLP has the best accuracy, on average, compared to all classifiers. However, MLP takes the longest running time in building the classifier model. Overall results suggest that Linear-SVM is the best classifier that can be used as an intelligent classification model for efficient energy management in WSNs. This is because the average time it takes to train the classifier model over the full training set is significantly shorter than that of MLP while keeping the accuracy level closer to what is achievable by MLP or even better.

For future work, we intend to evaluate the performance of SVM under other benchmarking datasets. Moreover, a performance comparison of SVM with different kernels, such as Gaussian or sigmoid kernels, will be conducted.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors acknowledge the Computer Engineering Department and King Fahd University of Petroleum and Minerals (KFUPM) for supporting this research work.

References

[1] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: methods, models, and classifications," *ACM Computing Surveys*, vol. 39, no. 3, article 9, 2007.

[2] 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 *Proceedings of the IEEE European Modelling Symposium (EMS '15)*, Madrid, Spain, October 2015.

[3] J. A. Khan, H. K. Qureshi, and A. Iqbal, "Energy management in wireless sensor networks: a survey," *Computers and Electrical Engineering*, vol. 41, pp. 159–176, 2015.

[4] Y. Yu, "A survey of intelligent approaches in wireless sensor networks for efficient energy consumption," in *Proceedings of the International Conference on Software and Computer Applications (ICSCA '12)*, Singapore, June 2012.

[5] M. Abu Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: algorithms, strategies, and applications," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[6] N. Enami, R. A. Moghadam, K. Dadashtabar, and M. Hoseini, "Neural network based energy efficiency in wireless sensor networks: a survey," *International Journal of Computer Science & Engineering Survey*, vol. 1, no. 1, pp. 39–53, 2010.

[7] Y. M. M. Hassim and R. Ghazali, "Training a functional link neural network using an artificial bee colony for solving a classification problems," *Journal of Computing*, vol. 4, no. 9, pp. 110–115, 2012.

[8] A. K. Pal and S. Pal, "Classification model of prediction for placement of students," *International Journal of Modern Education & Computer Science*, vol. 5, no. 11, p. 49, 2013.

[9] S. W. Purnami, J. M. Zain, and T. Heriawan, "An alternative algorithm for classification large categorical dataset: k-mode clustering reduced support vector machine," *International Journal of Database Theory and Application*, vol. 4, no. 1, pp. 19–30, 2011.

[10] A. Mitrokotsa, M. Tsagkaris, and C. Douligeris, "Intrusion detection in mobile ad hoc networks using classification algorithms," in *Advances in Ad Hoc Networking*, vol. 265 of *IFIP International Federation for Information Processing*, pp. 133–144, Springer, New York, NY, USA, 2008.

[11] A. Y. Barnawi and I. M. Keshta, "Energy management of wireless sensor networks based on multi-layer perceptrons," in *Proceedings of the 20th European Wireless Conference, European Wireless*, pp. 1–6, VDE, Barcelona, Spain, May 2014.

[12] G. R. Gupta and P. Ramanathan, "Level set estimation using uncoordinated mobile sensors," in *Ad-Hoc, Mobile, and Wireless Networks*, vol. 4686 of *Lecture Notes in Computer Science*, pp. 101–114, Springer, Berlin, Germany, 2007.

[13] M. Magno, D. Brunelli, P. Zappi, and L. Benini, "Energy efficient cooperative multimodal ambient monitoring," in *Smart Sensing and Context*, pp. 56–70, Springer, Berlin, Germany, 2010.

[14] E. S. Sazonov and J. M. Fontana, "A sensor system for automatic detection of food intake through non-invasive monitoring of

chewing," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1340–1348, 2012.

[15] M. Bal, M. F. Amasyali, H. Sever, G. Kose, and A. Demirhan, "Performance evaluation of the machine learning algorithms used in inference mechanism of a medical decision support system," *The Scientific World Journal*, vol. 2014, Article ID 137896, 15 pages, 2014.

[16] S. Yuan, D. Liang, L. Qiu, and M. Liu, "Mobile multi-agent evaluation method for wireless sensor networks-based large-scale structural health monitoring," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 164527, 15 pages, 2012.

[17] S. Thiemjarus, B. P. L. Lo, K. V. Laerhoven, and G. Z. Yang, "Feature selection for wireless sensor networks," in *Proceedings of the 1st International Workshop on Wearable and Implantable Body Sensor Networks (BSN '04)*, London, UK, April 2004.

[18] M. Alwadi and G. Chetty, "Feature selection and energy management for wireless sensor networks," *IJCSNS International Journal of Computer Science and Network Security*, vol. 12, no. 6, 2012.

[19] J. Csirik, P. Bertholet, and H. Bunke, "Pattern recognition in wireless sensor networks in presence of sensor failures," in *Proceedings of the 12th WSEAS International Conference on Neural Networks (NN '11)*, Brașov, Romania, April 2011.

[20] J. Martyna, "Least squares support vector machines for clustering in wireless sensor networks," in *Proceedings of the 2nd National Scientific Conference on Data Processing Technologies*, Poznań, Poland, September 2007.

[21] Y. Li, Y. Wang, and G. He, "Clustering-based distributed support vector machine in wireless sensor networks," *Journal of Information and Computational Science*, vol. 9, no. 4, pp. 1083–1096, 2012.

[22] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, pp. 1663–1707, 2010.

[23] K. Flouri, B. Beferull-Lozano, and T. Tsakalides, "Energy-efficient distributed support vector machines for wireless sensor networks," in *Proceedings of the 3rd European Workshop on Wireless Sensor Networks (EWSN '06)*, Zürich, Switzerland, February 2006.

[24] K. Flouri, B. Beferull-Lozano, and T. Tsakalides, "Distributed consensus algorithms for SVM training in wireless sensor networks," in *Proceedings of the 16th European Signal Processing Conference (EUSIPCO '08)*, Lausanne, Switzerland, 2008.

[25] S. Rajasegarar, C. Leckie, J. C. Bezdek, and M. Palaniswami, "Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 518–533, 2010.

[26] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1062–1074, 2013.

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

[28] A. Karatzoglou, D. Meyer, and K. Hornik, "Support vector machines in R," *Journal of Statistical Software*, vol. 15, no. 9, pp. 1–32, 2006.

[29] H.-S. Ham, H.-H. Kim, M.-S. Kim, and M.-J. Choi, "Linear SVM-based android malware detection for reliable IoT services," *Journal of Applied Mathematics*, vol. 2014, Article ID 594501, 10 pages, 2014.

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

[31] Y. Guo, G. Bai, and Y. Hu, "Using bayes network for prediction of type-2 diabetes," in *Proceedings of the 7th International Conference for Internet Technology and Secured Transactions (ICITST '12)*, pp. 471–476, London, UK, December 2012.

[32] R. Battiti, M. Brunato, and F. Mascia, *Reactive Search and Intelligent Optimization*, Operations Research/Computer Science Interfaces, Springer, 2008.

[33] B. Karlik and A. VehbiOlgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011.

[34] H. Bouzgou and N. Benoudjit, "Multiple architecture system for wind speed prediction," *Applied Energy*, vol. 88, no. 7, pp. 2463–2471, 2011.

[35] K. P. Bennett and C. Campbell, "Support vector machines: hype or hallelujah?" *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 2, pp. 1–13, 2000.

[36] V. Vapnik, *The Natural of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[37] M. Alwadi and G. Chetty, "A novel feature selection scheme for energy efficient wireless sensor networks," in *Algorithms and Architectures for Parallel Processing: 12th International Conference, ICA3PP 2012, Fukuoka, Japan, September 4–7, 2012, Proceedings, Part II*, vol. 7440 of *Lecture Notes in Computer Science*, pp. 264–273, Springer, Berlin, Germany, 2012.

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

[39] MathWorks, *MATLAB and Statistics Toolbox Release 2012b*, MathWorks, Natick, Mass, USA, 2012.

[40] S. H. Weiss and N. Indurkhya, *Predictive Data Mining: A Practical Guide*, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 1998.

[41] http://www.cs.waikato.ac.nz/ml/weka/.

[42] A. Ahmim and N. Ghoualmi-Zine, "A new fast and high performance intrusion detection system," *International Journal of Security and Its Applications*, vol. 7, no. 5, pp. 67–80, 2013.

[43] A. K. Santra and C. Josephine Christy, "Genetic algorithm and confusion matrix for document clustering," *International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 322–328, 2012.

[44] C. O. Sakar, O. Kursun, and F. Gurgen, "Feature extraction based on discriminative alternating regression," in *XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013*, pp. 730–733, Springer, 2014.

[45] J. J. Fernandez-Rodriguez, A. Cañas, E. Roca, F. J. Pelayo, J. Fernandez-Mena, and A. Prieto, "Application of artificial neural networks to chest image classification," in *New Trends in Neural Computation*, vol. 686 of *Lecture Notes in Computer Science*, pp. 542–549, Springer, Berlin, Germany, 1993.

[46] M. N. Murty and V. S. Devi, *Pattern Recognition: An Algorithmic Approach*, Springer Science and Business Media, 2011.