A Distributed Support Vector Machine Learning Over Wireless Sensor Networks

Woojin Kim, *Student Member, IEEE,* Miloš S. Stanković, *Member, IEEE,* Karl H. Johansson, *Fellow, IEEE,* and H. Jin Kim, *Member, IEEE*

***Abstract*—This paper is about fully-distributed support vector machine (SVM) learning over wireless sensor networks. With the concept of the geometric SVM, we propose to gossip the set of extreme points of the convex hull of local data set with neighboring nodes. It has the advantages of a simple communication mechanism and finite-time convergence to a common global solution. Furthermore, we analyze the scalability with respect to the amount of exchanged information and convergence time, with a specific emphasis on the small-world phenomenon. First, with the proposed naive convex hull algorithm, the message length remains bounded as the number of nodes increases. Second, by utilizing a small-world network, we have an opportunity to drastically improve the convergence performance with only a small increase in power consumption. These properties offer a great advantage when dealing with a large-scale network. Simulation and experimental results support the feasibility and effectiveness of the proposed gossip-based process and the analysis.**

***Index Terms*—Distributed learning, support vector machine (SVM), wireless sensor networks.**

I. Introduction

DUE TO recent advances in wireless communication and embedded computing, supervised machine learning can address various applications related to wireless sensor networks. Basic supervised learning techniques have been applied to diverse sensor network scenarios. Kernel-based learning [14], [17] has been suggested for simplified localization, object tracking, and environmental monitoring. Also, maximum-likelihood parametric approaches [7], Bayesian networks [13], hidden Markov models [2], statistical regression methods [11] and support vector machines (SVMs) [20], [23] have been employed for source localization, activity recognition, human behavior detection, parameter regression, self-localization and environmental sound recognition, respectively. In particular, SVM is a classification algorithm with the advantages of wide applicability, data sparsity, and global optimality. Training an SVM requires solving a quadratic optimization problem of dimensionality dependent on the cardinality of the training (example) set. The resulting discriminant rule is expressed by a subset of the training set, known as support vectors [21].

In recent studies, due to the tight energy, bandwidth and other constraints on communication capabilities for wireless sensor networks, distributed SVM training has been investigated. A parallel design of centralized SVM is one approach [6], [26]. When the training data set is very large, partial SVMs are obtained using small training subsets and combined at a fusion center. This approach can handle enormous sizes of data, but can be applied only if a central processor is available to combine the partial support vectors, and convergence to the centralized SVM is not always guaranteed for arbitrary partitioning of the data set [10].

On the other hand, there are fully distributed approaches that solve the entire SVM using distributed optimization methods. Because SVM is a quadratic optimization problem, existing convex optimization techniques can be used. In [9], a distributed SVM has been presented, which adopts the alternating direction method of multipliers [4]. This approach is based on message exchanges among neighbors and provably convergent to the centralized SVM. However, since the gradient-based iteration should maintain the connection between nodes until convergence, the intercommunication cost is large. Furthermore, in the nonlinear case, the exchanged message length can become extremely long. These issues render it not suitable to wireless sensor network applications. Another class of distributed SVM, which is not based on the gradient method, relies on gossip-based incremental support vectors obtained from local training data sets [8], [25]. These gossip-based distributed SVM approaches guarantee convergence when the labeled classes are linearly separable. When they are not linearly separable, these approaches can approximate, although not ensure, convergence to the centralized SVM solution.

In this paper, we employ the concept of gossip-based incremental SVM with a geometric representation. The geometric interpretation of SVMs is based on the notion of convex hulls and geometric nearest point algorithms [3], [19]. Unlike the gossip-based incremental support vectors [8], we propose an

Manuscript received August 4, 2013; revised January 7, 2014 and November 3, 2014; accepted November 14, 2014. Date of publication February 24, 2015; date of current version October 13, 2015. This work was supported in part by the National Research Foundation of Korea, the Swedish Foundation for International Cooperation in Research and Higher Education under Grant 2014R1A2A1A12067588 funded by the Ministry of Science, Information/Communication Technology, and Future Planning. This paper was recommended by Associate Editor S. X. Yang.

W. Kim is with the Electronics and Telecommunications Research Institute, Daejeon 305-700, Korea.

M. S. Stanković is with the Innovation Center, School of Electrical Engineering, University of Belgrade, Belgrade 11000, Serbia.

K. H. Johansson is with the ACCESS Linnaeus Center, School of Electrical Engineering, Royal Institute of Technology, Stockholm 100 44, Sweden.

H. J. Kim is with the Institute of Advanced Aerospace Technology, School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-744, Korea (e-mail: hjinkim@snu.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2377123

algorithm based on incremental convex hulls where the nodes gossip only the extreme points of their local convex hulls, initially obtained from local training data sets. Through the join operation of convex hulls, the proposed algorithm guarantees the convergence in finite time to the global solution, i.e., the centralized SVM.

The structure of this paper is as follows. Section II summarizes the contribution of this paper. In Section III, we introduce the geometric SVMs under both the separable and nonseparable cases. The gossip-based distributed SVM training is described in Section IV, with scalability and convergence analysis. In Sections V and VI, simulation and experimental results are presented respectively, which validate the proposed algorithm, and convergence and energy consumption issues are discussed. Finally, the conclusion is given in Section VII.

II. Contributions

This paper focuses on how to make SVM work over the sensor network in a fully distributed manner. Unlike in [26], which deals with the efficient training method in the parallel structure of sensor network topology, this paper assumes that there is no centralized training. Here, training is performed only using one-hop communications between sensor nodes with low computation capability. This includes the nonlinear SVM training, whereas [8] is not applicable to the nonlinear version. Reference [9] has the same structure of training with ours (a fully distributed approach) and is applicable to the nonlinear case, however, the intercommunication cost is large and the exchanged message length can become extremely long in the nonlinear case. These issues render it not suitable to wireless sensor network application.

Deriving inspiration from geometric properties in [3] and [19], we consider the join operation of the convex hull of each labeled data set. The join operation has been introduced in [25] for distributed and incremental SVM learning in linearly separable cases. In this paper, we extend it to nonseparable and nonlinear cases, and theoretically analyze this extension. In order to resolve the nonseparable cases, the concept of reduced convex hull is applied and the convex hull for kernel space is discussed for nonlinear cases.

This paper also contributes to lowering the computational complexity associated with the data fusion process in both memory and computation and to reducing the overall power requirements through coordinating the network connectivity in a fully distributed manner. Furthermore, the convergence time analysis is performed utilizing the concept of a small-world network, for static and random connection topology. A small-world network is a network where the path length between two randomly selected nodes grows logarithmically with the number of nodes [24]. Analysis of the small-world network shows that the average path length of the network topology decreases as reconnection probability increases. From the viewpoint of a trade-off between energy savings and performance improvements, the small-world concept gives the opportunity to drastically increase the performance with only a small increase in energy consumption.

From the overall framework of the proposed distributed SVM training, the following contributions can be obtained.

1) *Fully Distributed Communication:* Basically, the proposed gossip-based algorithm exchanges messages only with neighboring nodes so that the network connection topology is simply determined with one-hop communication.
2) *Guaranteed Convergence to the Centralized SVM Performance:* The local calculation of the convex hull with join operation guarantees finite-time convergence and the global optimality of the solution at each node.
3) *Scalability With Respect to the Communication Packet Length:* As the amount of training data increases, the number of extreme points increases in the worst case. To deal with this, we propose a naive algorithm for convex hulls, where the amount of exchanged information can be controlled, even in the worst case.

III. Geometric Representation of SVMs

In this section, we describe geometric SVMs [19] briefly. In geometric SVMs, the data set is represented using geometric convex hulls, and the classification problem can be converted to a nearest point problem which leads to an elegant and efficient solution to the SVM classification. First, we describe the geometric process for a separable data set, in which two types of labeled data sets are completely divided. Then, we deal with a nonseparable case and formulate centralized SVM training over wireless sensor networks.

A. Separable Case

For the separable cases, the dual form of the original SVM problem is described by

$$\min_{\eta_i} \frac{1}{2}\sum_{i,j} y_i y_j \eta_i \eta_j x_i^T x_j - \sum_i \eta_i \quad \text{such that } \sum_i \eta_i y_i = 0,\ \ \eta_i \geq 0 \tag{1}$$

where, $x_i \in X \subset \mathcal{R}^d$ and $y_i \in \{-1, 1\}$ for $i, j = 1, \ldots, |X|$ are input and output data, respectively, and η_i are the corresponding Lagrangian multipliers. d is the dimension of X and $|X|$ is the cardinality of X. The set X and corresponding set $Y = \{y_1, \ldots, y_{|X|}\}$ are the input–output-paired training sets.

Here, we start the geometric representation of SVMs with some definitions.

Definition 1 (Convex Set): A set is convex if for every pair of points within the set, every point on the straight line segment that joins the pair of points is also within the set.

Definition 2 (Convex Hull): A convex hull $C(X) \subset \mathcal{R}^d$ of data set $X \subset \mathcal{R}^d$ is the smallest convex set containing X such that $C(X) = \{z | z = \sum_i \lambda_i x_i, \sum_i \lambda_i = 1, x_i \in X, \lambda_i \geq 0\}$, for $i = 1, \ldots, |X|$.

Definition 3 (Extreme Point Set): An extreme point set $\mathcal{E}(X)$ is a set of points in $X \in \mathcal{R}^d$ which cannot be represented as a convex combination of any other distinct points in X.

For the given set X, we can consider the subsets X^+ and X^-, which contain only the points of one class ($y_i = 1$) and the points of another class ($y_i = -1$), respectively, and X satisfies

$X = X^+ \cup X^-$ where $X^+ \cap X^- = \emptyset$. For the separable case, the original SVM problem described in (1) is equivalent to finding the closest points between the convex hulls generated by X^+ and X^- in the feature space [3]. Using the definition of a convex hull, the geometric representation of SVMs in the separable case can be described as follows:

$$\min_{\lambda_i \geq 0} \left\| \sum_{i:y_i=1} \lambda_i x_i - \sum_{i:y_i=-1} \lambda_i x_i \right\|^2 \quad \text{such that} \quad \sum_{i:y_i=1} \lambda_i = 1, \quad \sum_{i:y_i=-1} \lambda_i = 1 \tag{2}$$

where the constraints guarantee that the coefficient λ_is respect the convexity conditions of $C(X^+)$ and $C(X^-)$. From (2), we derive the performance index as

$$\begin{aligned} &\left\| \sum_{i:y_i=1} \lambda_i x_i - \sum_{i:y_i=-1} \lambda_i x_i \right\|^2 \\ &\quad = \sum_{i:y_i=1} \sum_{j:y_j=1} \lambda_i \lambda_j x_i^T x_j + \sum_{i:y_i=-1} \sum_{j:y_j=-1} \lambda_i \lambda_j x_i^T x_j \\ &\qquad - \sum_{i:y_i=1} \sum_{j:y_j=-1} \lambda_i \lambda_j x_i^T x_j - \sum_{i:y_i=-1} \sum_{j:y_j=1} \lambda_i \lambda_j x_i^T x_j \\ &\quad = \sum_i \sum_j y_i y_j \lambda_i \lambda_j x_i^T x_j \end{aligned}$$

and also the constraints can be derived as

$$\sum_{i:y_i=1} \lambda_i - \sum_{i:y_i=-1} \lambda_i = \sum_i y_i \lambda_i = 0$$
$$\sum_{i:y_i=1} \lambda_i + \sum_{i:y_i=-1} \lambda_i = \sum_i \lambda_i = 2.$$

Finally, the following equivalent formulation can be obtained:

$$\min_{\lambda_i} \sum_{i,j} y_i y_j \lambda_i \lambda_j x_i^T x_j \quad \text{such that} \quad \sum_i y_i \lambda_i = 0, \quad \sum_i \lambda_i = 2 \quad \lambda_i \geq 0, \ i, j = 1, \ldots, |X|. \tag{3}$$

According to [3], the above problem leads to the same solution as (1). For a nonlinear case, the inner products of x_i can be replaced by a kernel function in (1), (3), and (11). We mainly consider the Gaussian kernel, $\kappa(x, y) = \phi(x)^T \phi(y) = e^{-(\|x-y\|^2 2\sigma^2)}$, which is the most popular one.

Interestingly, for (3), the data sets X^+ and X^- can be reduced to $\mathcal{E}(X^+)$ and $\mathcal{E}(X^-)$, respectively, because of the following lemma.

Lemma 1: $\mathcal{E}(X)$ is the smallest set to represent $C(X)$.

Proof: See the Appendix. ∎

Remark 1: From Lemma 1, we can redefine the convex hull of set X as

$$C(X) = \left\{ z | z = \sum_i \lambda_i x_i, \sum_i \lambda_i = 1, x_i \in \mathcal{E}(X), \lambda_i \geq 0 \right\} \tag{4}$$

where $i = 1, \ldots, |\mathcal{E}(X)|$, and $|\mathcal{E}(X)|$ is the cardinality of $\mathcal{E}(X)$.

Algorithm 1 Convex Hull Algorithm in Feature Space

Input: set $\mathcal{V}_o = \{x_1\}$ and $\mathcal{V} = \emptyset$, arbitrarily picked $x_1 \in X$
Initialize: $X^* = X - \mathcal{V}_o$
Until X^* is empty,
  Get $x \in X^*$, update $X^* = X^* - \{x\}$
  If CheckPoint$(x, \mathcal{V}_o) = False$, $\mathcal{V} = \mathcal{V}_o \cup \{x\}$
    Until $\mathcal{V}_o$ is empty,
      Get $y \in \mathcal{V}_o$, update $\mathcal{V}_o = \mathcal{V}_o - \{y\}$
      If CheckPoint$(y, \mathcal{V} - \{y\}) = True$, $\mathcal{V} = \mathcal{V} - \{y\}$
    $\mathcal{V}_o = \mathcal{V}$
Output: the extreme point set of X, $\mathcal{E}(X) = \mathcal{V}_o$

The above lemma and remark imply that a compact convex subset of X is the closed convex hull of its extreme points. The Krein–Milman theorem [12] also supports this lemma. Even if X is a subset of kernel space, we can compute the extreme point set of the convex hull of X which can be of an arbitrary dimension [5] according to [15]. The following is a simple procedure for finding the extreme points of a convex hull in the feature space:

For the computation of $\mathcal{E}(X)$ in Algorithm 1, suppose that we have the function CheckPoint(z, X) returning True if z belongs to the interior of $C(X)$. However, in [15], the function CheckPoint(z, X) employs quadratic programming, whose computation complexity is NP-hard. Furthermore, another important issue is that the complexity of the convex hull and the number of extreme points depend on the dimensionality of the feature space, and this can be resolved by the concept of naive convex hull which will be introduced in Section IV-C.

Considering the computational limitations of a sensor node, we propose to use a sufficient condition for the function to return *False*, which is simple and has low computational load, rather than solving the quadratic programming directly, as described in the following lemma.

Lemma 2: Suppose $X \subset \mathcal{R}^d$ is a compact data set and $z \in \mathcal{R}^d$, $X = \{x_1, x_2, \ldots, x_{|X|}\}$. Let $d_{\max} = \sup \|x_j - x_k\|$, $d_{\min} = \inf \|z - x_j\|$. If

$$1 + e^{-d_{\max}^2/2\sigma^2} > 2e^{-d_{\min}^2/2\sigma^2} \tag{5}$$

holds, then CheckPoint(z, X) returns *False* in Gaussian feature space.

Proof: Given the Gaussian kernel, $\kappa(x, y) = \phi(x)^T \phi(y) = e^{-(\|x-y\|^2 2\sigma^2)}$ where $\phi(\cdot)$ denotes the mapping to the feature space, if the distance in the feature space from $\phi(z)$ to the $C(\{\phi(x_1), \phi(x_2), \ldots, \phi(x_{|X|})\})$ is strictly bigger than zero, then $\phi(z)$ does not lie in the interior of $C(\{\phi(x_1), \phi(x_2), \ldots, \phi(x_{|X|})\})$ in the feature space. The distance in feature space is calculated as follows:

$$\begin{aligned} &\left\| \phi(z) - \sum_{j=1}^{|X|} \lambda_j \phi\left(x_j\right) \right\|^2 \\ &\quad = \phi(z)^T \phi(z) + \left(\sum_{j=1}^{|X|} \lambda_j \phi\left(x_j\right) \right)^T \left(\sum_{j=1}^{|X|} \lambda_j \phi\left(x_j\right) \right) \\ &\qquad - 2\phi(z)^T \left(\sum_{j=1}^{|X|} \lambda_j \phi\left(x_j\right) \right) \end{aligned} \tag{6}$$

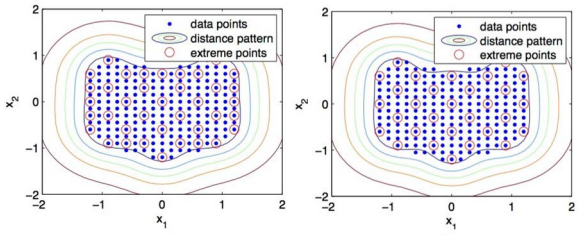


Fig. 1. Computation of extreme points in feature space using (a) convex hull algorithm with quadratic programming described in [15] and (b) sufficient condition (5).

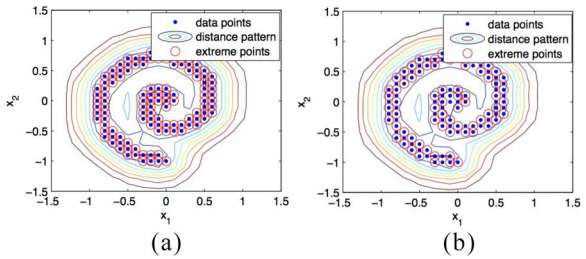


Fig. 2. Computation of extreme points of complex data points in feature space using (a) convex hull algorithm with quadratic programming described in [15] and (b) sufficient condition (5).

where λ_is are the coefficients of the convex combination, satisfying $\sum \lambda_i = 1$ and $\lambda_i \geq 0$. From (6), we can obtain the sufficient condition for positiveness of the distance between $\phi(z)$ and $C(\{\phi(x_1), \phi(x_2), \ldots, \phi(x_{|X|})\})$. Since $\phi(\cdot)$ is the Gaussian mapping, the first term of the right hand side of (6) is $\phi(z)^T\phi(z) = \kappa(z, z) = 1$. The second and third terms satisfy the following inequalities:

$$\left(\sum_{j=1}^{|X|} \lambda_j \phi\left(x_j\right)\right)^T \left(\sum_{j=1}^{|X|} \lambda_j \phi\left(x_j\right)\right) = 1 - \sum_{j\neq k} \lambda_j\lambda_k + \sum_{j\neq k} \lambda_j\lambda_k e^{\frac{-\|x_j - x_k\|^2}{2\sigma^2}} \geq e^{\frac{-d_{\max}^2}{2\sigma^2}} \quad (7)$$

and likewise

$$-2\phi(z)^T \left(\sum_{j=1}^{|X|} \lambda_j \phi\left(x_j\right)\right) \geq -2e^{\frac{-d_{\min}^2}{2\sigma^2}}. \quad (8)$$

Thus, we have

$$\left\|\phi(z) - \sum_{j=1}^{|X|} \lambda_j \phi\left(x_j\right)\right\|^2 \geq 1 + e^{-d_{\max}^2/2\sigma^2} - 2e^{-d_{\min}^2/2\sigma^2}. \quad (9)$$

If $1 + e^{-d_{\max}^2/2\sigma^2} > 2e^{-d_{\min}^2/2\sigma^2}$, then $\phi(z)$ does not belong to the interior of $C(\{\phi(x_1), \phi(x_2), \ldots, \phi(x_{|X|})\})$ in Gaussian feature space. ■

Since the sufficient condition (5) is a simple mathematical computation, we can solve Algorithm 1 in polynomial time. However, because (5) only provides the sufficient condition for a new point not to belong to the interior of convex hull in Gaussian feature space, we need careful observation for validating applicability of the sufficient condition in (5) instead of the function CheckPoint. Figs. 1 and 2 illustrate computation of extreme points using (5) versus CheckPoint.

Fig. 1 shows an example of computing extreme points in feature space with about 200 data points using the two approaches. Fig. 1(b) is the result when we use Lemma 2, which shows a similar selection of extreme points (red circle markers) with a few missing extreme points compared to the result when we use quadratic programming shown in Fig. 1(a). Since we are interested in the distance parameter from the convex hull to solve SVM problem, we compare the hyper-dimensional distance pattern between Fig. 1(a) and (b). As shown in the contour plots of Fig. 1, they have almost the same distance patterns because the missing extreme points are located very close to the convex hull within the distance of $(1 + e^{-d_{\max}^2/2\sigma^2} - 2e^{-d_{\min}^2/2\sigma^2})^{1/2}$ according to (9), and do not change the distance pattern significantly. Moreover, the computation time of the proposed approach in this example is 0.2001 s, about 100 times faster than that of quadratic programming approach (21.24 s).

Fig. 2 shows a more complicated example of computing extreme points in feature space. When we use quadratic programming shown in Fig. 2(a), all the data points are included in the extreme point set, while Fig. 2(b) shows a similar selection of extreme points with a few missing points. In this complicated case, both approaches yield almost same distance patterns. These results support that the sufficient condition (5) is reasonable and provides computational efficiency.

To avoid the notational complexity, we express the convex hull of X in Gaussian feature space as simply $C(\Phi(X))$ instead of $C(\{\phi(x_1), \phi(x_2), \ldots, \phi(x_{|X|})\})$ in the remaining parts of this paper.

B. Nonseparable Case

For the nonseparable case, the convex hulls of each class overlap and the previous procedure does not make sense, because there are the infinite number of the points in the overlapped area and all these points are the closest points to the convex hulls with zero distance. So, instead of the concept of a convex hull, the reduced convex hull $R(X, \mu)$ of a set X is defined as follows [19].

Definition 4 (Reduced Convex Hull): The reduced convex hull $R(X, \mu)$ of data set $X \subset \mathcal{R}^d$ is the set of all convex combinations of points in X, with the additional constraint that each coefficient λ_i is upper-bounded by a nonnegative number $\mu < 1$: $R(X, \mu) = \{z | z = \sum_i \lambda_i x_i, \sum_{i=1}^{|X|} \lambda_i = 1, x_i \in X, 0 \leq \lambda_i \leq \mu\}$, $i = 1, \ldots, |X|$.

The difference between the convex hull and the reduced convex hull is that the coefficient λ is restricted by $\mu < 1$ and if $\mu = 1$, $R(X, \mu)$ is equivalent to $C(X)$. With the concept of a reduced convex hull, we can extend the scenario of the previous geometric SVM to nonseparable cases with the assumption that the parameter μ has been selected properly so that $R(X^+, \mu) \cap R(X^-, \mu) = \emptyset$. Similar to the separable case, we can obtain a geometric interpretation of SVM that finds

the closest points between the reduced convex hulls generated by X^+ and X^- as follows:

$$\min_{\lambda_i} \sum_{i,j} y_i y_j \lambda_i \lambda_j x_i^T x_j$$
$$\text{such that } \sum_i y_i \lambda_i = 0, \quad \sum_i \lambda_i = 2$$
$$0 \le \lambda_i \le \mu, \ \ i, j = 1, \ldots, |X|. \tag{10}$$

The optimal problem (10) is identical to the Wolfe dual formulation of a modified formulation of SVM which is a scaled version of the ν-SVM [18], [19]. From these setting, we can incorporate projection method such as Theodoridis's algorithm [29] to find the nearest points between the reduced convex hulls. By using that, the decision function can be obtained as (11). For nonlinear cases, the inner product terms in (10) can be replaced by a kernel function. Similar to (3), the data sets X^+ and X^- can be reduced to $\mathcal{E}(X^+)$ and $\mathcal{E}(X^-)$, respectively.

C. Geometric SVM Training and the Decision Function

In the geometric form of the SVM training, we can incorporate the projection method to find the nearest points such as Gilbert's algorithm [27] and Schlesinger–Kozinec's algorithm [28]. Even though the nonseparable case can be handled by the reduced convex hull, Theodoridis's algorithm does not need the reduced convex hull explicitly [29]. Therefore, we can simply use the conventional convex hull instead of the explicit form of the reduced convex hull. By using those algorithms, the decision function can be obtained as follows:

$$f(x) = \sum_{i \in X, \lambda_i \neq 0} \lambda_i y_i x_i^T x + b \tag{11}$$

where

$$b = \frac{1}{2}\left(\sum_{i:y_i=1} \sum_{j:y_j=1} \lambda_i \lambda_j x_i^T x_j - \sum_{i:y_i=-1} \sum_{j:y_j=-1} \lambda_i \lambda_j x_i^T x_j \right).$$

The sign of the decision function $f(x)$ determines whether x lies on the positive or negative side, and $f(x) = 0$ represents the border line in the test phase.

IV. Distributed SVM

Consider data generated by a sensor network as input-output pairs $\{(x, y)\}$ for SVM training, for example, such that $x = [q^T, t]^T$ in some input space $X \subset \mathcal{R}^{d+1}$ consisting of the position measurement $q \in \mathcal{R}^d$ and a timestamp t, and $y \in \{-1, 1\}$ (label) is the measurement corresponding to x or some function of the measurement. For example, in a hazardous area detection scenario, y can be a binary value whether it is hazardous or not on the position q at time t. We assume that all the nodes are synchronized with the time history t. Of course, this restriction is not necessary for monitoring a static quantity.

In a centralized setting, a sensor network has its own fusion center which gathers information from all the nodes and performs massive computation to obtain the global SVM solution. This may incur a heavy communication load, which can cause packet loss, communication delay, and much energy consumption, deteriorating the performance of object localization [30].

In this section, the SVM training is described in a fully distributed fashion. We consider a situation where the centralized fusion is not allowed since we want to comply with the important properties of WSNs such as: low communication complexity, scalability, flexibility, and redundancy. Our goal of distributed SVM training is as follows.

1) Considering the communication complexity, message exchange is allowed only between one-hop neighbors.
2) The exchanged messages should be short enough to reduce the communication costs and battery usage.
3) All nodes keep their local estimate at each time slot and they all converge to a common global estimate.
4) The common global estimate is the same as the result of the centralized training.

In order to satisfy the above goals, we propose a gossip algorithm to solve distributed SVM in the context of geometric SVM as described in Section III. The idea is that if the new data measured by the sensor node lies in the convex hull of its labeled class, that data does not affect the global solution, thus it does not need to be transmitted.

This section is organized as follows. First, the main idea of the gossip process for distributed SVM learning is described in Section IV-A. The scalability analysis in Section IV-B suggests that there may exist the worst case where the message length grows to infinity. To handle this, the concept of the naive convex hull is proposed and its characteristics are analyzed in Section IV-C. From Section IV-D, we find that the convergence time is equivalent to the average path length of the network. To analyze the convergence time in terms of the network topology, the small-world network concept is adopted in Section IV-E.

A. Gossiping Extreme Points for Distributed SVM Training

In order to obtain the global optimum with low energy consumption for solving distributed SVM, we propose to gossip the extreme points with neighboring nodes. Let us suppose that there are n sensor nodes in the connected WSN and each sensor node j has data set $X_j = X_j^+ \cup X_j^-$ for $j = 1, \ldots, n$, where X_j^+ and X_j^- are sets of points of positive and negative classes, respectively. The following is the brief description of the proposed gossiping process, where $\mathcal{B}_j$ is the one-hop-neighbor set of node j.

The final step of the algorithm is performed only when the decision function is actually needed. In static situations, this is done only once after the gossip process converges. The convergence of the gossip process will be discussed in Theorem 1 and Sections IV-D and IV-E.

The above algorithm shows three of the important characteristics.

1) The algorithm is fully distributed. There is no governing fusion center to control the whole network and no matter how complex the network is, the algorithm at each node only uses simple one-hop communication with neighbors only.

Algorithm 2 Gossip Process for Distributed SVM Training

Data: Given the initial data set $X_j = X_j^+ \cup X_j^-$ for each sensor node $j = 1, \cdots n$
Compute initial extreme point sets $\mathcal{E}(X_j^+)$ and $\mathcal{E}(X_j^-)$
Replace $X_j^+ = \mathcal{E}(X_j^+)$, $X_j^- = \mathcal{E}(X_j^-)$
for $t = 0, 1, 2, \cdots$
  for all $j = 1, \ldots, n$
    Transmit X^+ and X^- to $\mathcal{B}_j$
  for all $j = 1, \ldots, n$
    Update X_j as $X_j = X_j \cup \left(\cup_{k \in \mathcal{B}_j} X_k \right)$
  for all $j = 1, \ldots, n$
    Compute $\mathcal{E}(X_j^+)$ and $\mathcal{E}(X_j^-)$ with X_j
    Replace $X_j^+ = \mathcal{E}(X_j^+)$, $X_j^- = \mathcal{E}(X_j^-)$
Output: The decision function $f(x)$ by solving the geometric SVM problem (10) with current data set $X_j = X_j^+ \cup X_j^-$ for each node.

2) Node j communicates only the extreme points with one-hop-neighbors $\mathcal{B}_j$ where the transmitted message is as follows:

$$\text{message}_j = \left\{ \mathcal{E}\left(X_j^+\right), \mathcal{E}\left(X_j^-\right) \right\}.$$

The extreme point set for each node $\mathcal{E}(X_j^s)$ is the smallest set to represent the convex hull $C(X_j^s)$ according to Lemma 1. In general, $|\mathcal{E}(X_j^s)| \ll |X_j^s|$, for $s \in \{+, -\}$ and exchanging only extreme points is efficient in terms of energy consumption.

3) Each node keeps only the extreme points at every time slot, so this algorithm is also efficient in terms of memory requirements.

In order to prove the convergence of the gossip process, the join operation is defined and its related property is introduced as follows.

Definition 5 (Join Operations of Convex Hulls): We define the join operation of the two convex set $C(X)$ and $C(Y)$ as the convex hull of the union of two sets $X \cup Y$ as follows:

$$C(X) \vee C(Y) \triangleq C(X \cup Y).$$

Lemma 3 (The Property of the Join Operations): For two data sets X_j^s and X_i^s with $i \neq j$, the following join operation is equivalent to the convex hull of the union of two convex sets, for $s \in \{+, -\}$:

$$C\left(X_j^s\right) \vee C\left(X_i^s\right) = C\left(C\left(X_j^s\right) \cup C\left(X_i^s\right)\right). \tag{12}$$

Proof: See the Appendix. ■

Furthermore, (12) can be directly extended to the case of $n > 2$ data sets as follows:

$$\vee_{j=1}^n C\left(X_j^s\right) = C\left(\cup_{j=1}^n X_j^s\right) = C\left(\cup_{j=1}^n C\left(X_j^s\right)\right). \tag{13}$$

Remark 2: Lemma 3 still holds in the feature space as

$$C\left(\Phi\left(X_j^s\right)\right) \vee C\left(\Phi\left(X_i^s\right)\right) = C\left(C\left(\Phi\left(X_j^s\right)\right) \cup C\left(\Phi\left(X_i^s\right)\right)\right).$$

For notational simplicity, we will denote $\Phi(X_j^s)$ and $\Phi(X_i^s)$ with $\mathcal{X}_j^s$ and $\mathcal{X}_i^s$ respectively. Then, we have

$$C\left(\mathcal{X}_j^s\right) \vee C\left(\mathcal{X}_i^s\right) = C\left(C\left(\mathcal{X}_j^s\right) \cup C\left(\mathcal{X}_i^s\right)\right)$$

which is equivalent to (12).

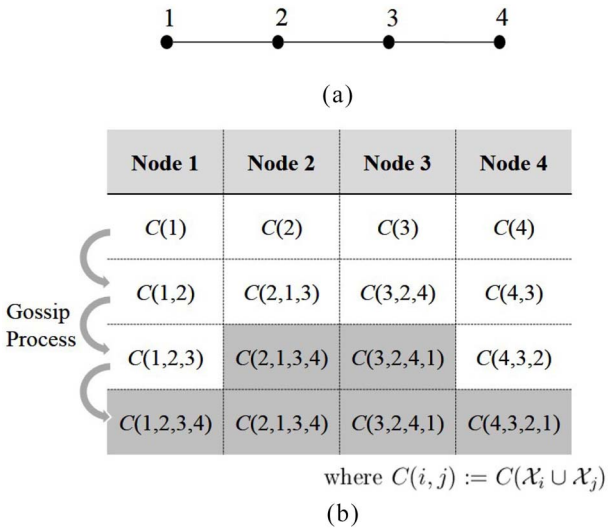


Fig. 3. Example of the gossip process (a) network connected in finite time with simple topology and (b) gossip process with join operation, where the cells with gray background indicate the convergence.

The interactions among the sensor nodes are represented as a graph $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{L})$ such that $\mathcal{V}$ is a set of nodes and an edge $(t, r) \in \mathcal{L}$ if node t can communicate with node $r \neq t$, where $t, r \in \mathcal{V}$. When we allow $\mathcal{L}$ to be time-varying (or random), we can define the connectedness in finite time of $\mathcal{G}$ as follows.

Definition 6: A network topology $\mathcal{G}$ is said to be connected in finite time when $\mathcal{G}$ has a finite-time path from each node to every other node.

Fig. 3(a) is a simple example of a network connected in finite time, with the maximum path length 3. According to Algorithm 2, the gossip process will be performed as shown in Fig. 3(b), which will be over in three time steps.

The following theorem deals with the finite-time convergence of the proposed algorithm.

Theorem 1: If $C(\mathcal{X}_j^s)$ denotes the convex hull of the data set $\mathcal{X}_j^s$ of node j, and the network is connected in finite time, then, by using the proposed gossip process (Algorithm 2) for t large enough, the following holds: $C(\mathcal{X}_1^s) = \cdots = C(\mathcal{X}_n^s) = C(\mathcal{X}^s)$, where $\mathcal{X}^s = \cup_j \mathcal{X}_j^s$, for $s \in \{+, -\}$.

Proof: Consider any two nodes j_0 and j_k both in $\{1, \ldots, n\}$. Since the network is connected in finite time, there exists a finite-time path $\{j_0 j_1 \cdots j_{k-1} j_k\}$ of length at least one, which connects nodes j_0 and j_k. Because $j_{l+1} \in \mathcal{B}_{j_l}$, which is the one-hop-neighbor set of node j_l, for $l = 0, \ldots, k-1$, it is obvious that $C(\mathcal{X}_{j_0}^s) = C(\mathcal{X}_{j_1}^s) = \cdots = C(\mathcal{X}_{j_k}^s) = \vee_{l=0}^k C(\mathcal{X}_{j_l}^s)$, after enough iterations of the gossip process from Lemma 3. Since j_0 and j_k can be picked arbitrarily, it follows readily that $C(\mathcal{X}_1^s) = \cdots = C(\mathcal{X}_n^s) = \vee_{j=1}^n C(\mathcal{X}_j^s) = C(\cup_{j=1}^n \mathcal{X}_j^s) = C(\mathcal{X}^s)$ after enough iterations of the gossip process from (14). ■

Remark 3: The above gossip process is globally optimal; that is, the agreement achieved by exchanging only the extreme points with neighboring nodes guarantees the convergence to the global convex hull identically for each sensor node in a connected network.

Remark 4: Because of the finite-time convergence, it is possible to apply the proposed algorithm and obtain the optimal

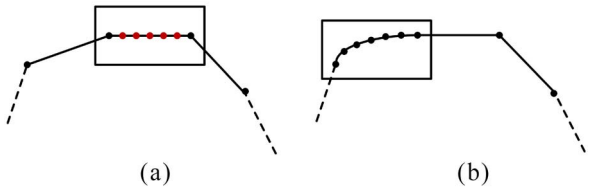


Fig. 4. Extreme points of the convex hull (a) general case of convex hull and (b) worst case of convex hull. The black points indicate the extreme points and the red points indicate the nonextreme points.

solution in a distributed manner, even if the training data sets are time-varying. As long as the collected data varies "slow enough" compared to the convergence time of the algorithm, the above analysis still holds.

B. Scalability Analysis on the Amount of Exchanged Information

The scalability on the amount of exchanged information is important since it is heavily related to the message length of intercommunication which affects the communication performance and energy efficiency. The exchanged information in our proposed gossip algorithm is the extreme point set of data for each node. In general, the cardinality of the extreme point set is finite as shown in Fig. 4(a). If additional points are added in a convex hull or on its boundary, the number of extreme points is not changed as shown using the red dots in the boxed area of Fig. 4(a). However, in the worst case, when the additional points lie on a round convex curve, the number of extreme points increases. In this case, if the number of the additional points lying on the round convex curve increases to infinity, then the number of extreme points, i.e., the message length, will also grow to infinity, as shown in the boxed area of Fig. 4(b). To overcome this scalability problem of the gossip algorithm, we propose the naive convex hull algorithm modifying the convex hull algorithm described in Algorithm 1.

C. Naive Convex Hull Algorithm

According to Algorithm 1, we can obtain the return of the function CheckPoint$(z, \mathcal{X})$ by checking the sufficient condition (5). When the condition is satisfied, i.e., $1+e^{-d^2_{\max}/2\sigma^2}-2e^{-d^2_{\min}/2\sigma^2}>0$, CheckPoint$(z, \mathcal{X})$ returns *False*, which indicates that z belongs to the exterior of $C(\mathcal{X})$ in the feature space. However, since testing the criterion $1+e^{-d^2_{\max}/2\sigma^2}-2e^{-d^2_{\min}/2\sigma^2}>0$ can make the message length overlong as mentioned in the previous section, we propose to relax it by introducing a margin of $\varepsilon>0$, i.e., we test the condition $1+e^{-d^2_{\max}/2\sigma^2}-2e^{-d^2_{\min}/2\sigma^2}>\varepsilon^2$. Fig. 5 shows the geometric interpretation of the naive convex hull algorithm. When we use the strict criterion $1+e^{-d^2_{\max}/2\sigma^2}-2e^{-d^2_{\min}/2\sigma^2}>0$ for CheckPoint(z, X), the number of extreme points is large in the worst case as shown in Fig. 5(a). On the other hand, using the relaxed criterion $1+e^{-d^2_{\max}/2\sigma^2}-2e^{-d^2_{\min}/2\sigma^2}>\varepsilon^2$ shown in Fig. 5(b), the points near to the convex hull $C(\mathcal{X})$ are also included in the interior of $C(\mathcal{X})$. This naive approach reduces the number of extreme points at the expense of introducing a predefined error tolerance ε which is independent of the number of the training data points.

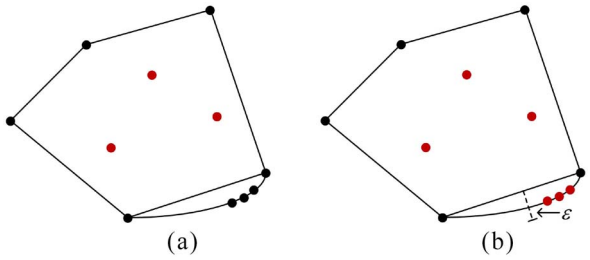


Fig. 5. Comparison between (a) convex hull algorithm and (b) naive convex hull algorithm. The black points indicate the extreme points and the red points indicate the nonextreme points.

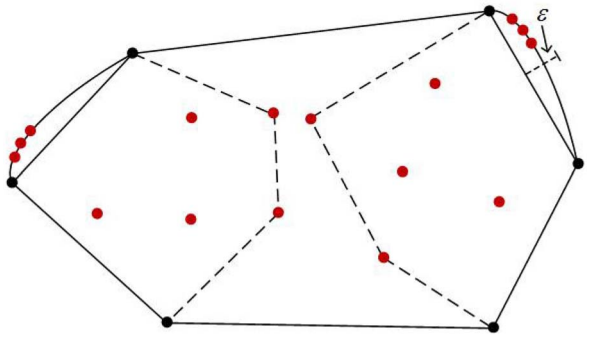


Fig. 6. Join operation of the two naive convex hulls. The black points indicate the extreme points and the red points indicate the nonextreme points.

Remark 5: The naive convex hull algorithm generates a convex hull with an error smaller than ε and this error is independent of the amount of data.

Therefore, by using the naive algorithm, we can find the naive convex hull $C_{nv}(\mathcal{X})$ and reduce the amount of exchanged information. ε is a design parameter, which controls a trade-off relationship between the error of the solution and the exchanged information. In general, we can significantly reduce the length of communication packets by allowing only a small positive error ε. Fig. 6 shows the join operation of two naive convex hulls, i.e., $C_{nv}(\mathcal{X})\vee C_{nv}(\mathcal{Y})$. Since the relaxed criterion affects only the edge points of the convex hulls, the joined convex hull still has an error smaller than ε. However, it is very hard to derive the upper bounds of final error analytically after multiple join operations of naive convex hulls have converged. In order to check whether the result of join operations of ε-naive convex hull maintains the error bound of ε or not, we perform Monte Carlo simulation. On workspace whose size is 2×2, we randomly deploy 50 sensor nodes which measure data points and exchange the extreme points of the ε-naive convex hulls with one-hop neighbors. With various values of ε (100 times for each ε), we figure out what happens in terms of error and exchanged data length with simulation. Fig. 7 shows the results of Monte Carlo simulation. In Fig. 7(a), the numerical error data are plotted in the form of a box plot. The error of naive convex hull is defined as the difference between the conventional convex hull and the naive convex hull. The exact definition is the following:

$$e_{nv}=\max_{z\in C(\mathcal{X})-C_{nv}(\mathcal{X})}\ \min_{z_{nv}\in C_{nv}(\mathcal{X})}\|\phi(z)-\phi(z_{nv})\|. \quad (14)$$

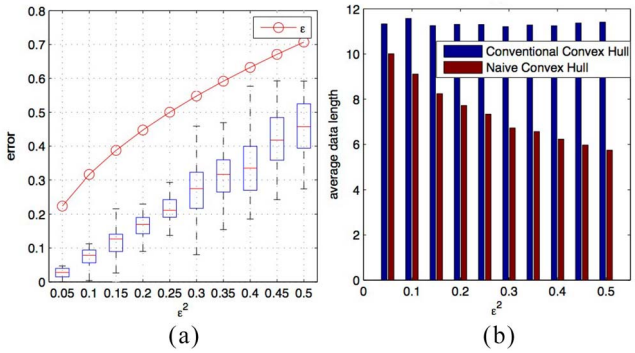


Fig. 7. Results of Monte Carlo simulation of join operations of naive convex hulls with varying ε. (a) Box plots of errors. (b) Average of exchanged data length corresponding to the margin ε.

In Fig. 7, the red circled markers are ε values and all the error values are upper-bounded by the value of ε. Also, in Fig. 7(b), the exchanged data length decreases as ε increases, where the blue bars are the average data length when we use conventional convex hull and the red bars are the average data length when we use naive convex hull. Furthermore, since the worst-case error grows abruptly while data length goes down slowly from $\varepsilon = 0.3$, the trade-off between the error and data length is more efficient when ε is under 0.3. Considering the workspace is sized by 2×2, 0.3 is not a very small value. Simply, the parameter ε controls the trade-off relationship between the communicated data length and the error of the naive convex hull. We selected the value of ε, by comparing the validation performance with different value of ε. A small positive value of ε (rather than $\varepsilon = 0$) can resolve the scalability problem brought up in Section IV-B. From the above simulation results, we can conclude that the naive convex hull algorithm has an error smaller than ε in the computation of the gossip process, introduced as $\vee_{j=1}^{n} C_{nv}(\mathcal{X}_j^s)$.

D. Convergence Time of the Gossip Process

For a connected network topology $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, we define the path from node i to node j as $path_{ij} = \{i, l_2, \ldots, l_{m-1}, j\}$, whose length is $|path_{ij}|$, where $i, j, \{l_k\} \in \mathcal{V}$, $i \neq j$ and $(i, l_1), (l_1, l_2), \cdots, (l_{m-1}, j) \in \mathcal{L}$. By defining the shortest path as $\hat{m}_{ij} = \min_{\{l_k\}} |path_{ij}|$, the average convergence time of the gossip process is denoted by $T_{avg}(\mathcal{G})$ as follows:

$$T_{avg}(\mathcal{G}) = \frac{1}{n(n-1)} \sum_{i,j \in \mathcal{V}, i \neq j} \hat{m}_{ij} \tag{15}$$

where n is the total number of sensor nodes. From (15), we know that the average convergence time of the gossip process is equivalent to the average path length of the network, which is dependent on the network topology $\mathcal{G}$. In order to analyze the performance of the proposed algorithm over sensor networks with random characteristics of the network topology, the small-world network concept is adopted.

E. Small-World Network Analysis

The mathematical model of the small-world network introduced by Watts and Strogatz [24] deals with a class of networks which interpolates between two extremes of the connection topology: completely static or completely random. The representative study of the small-world network is a rewired topology. For n vertices, each vertex is connected to its k nearest neighbors, which we call the degree of the network. Now, each connection is reconnected to another randomly chosen vertex with probability p. This construction of the small-world network introduces occasional long-range connections [1].

According to [24], for the case of a completely static network, i.e., $p = 0$, the average path length of the network $L(n, k, p = 0)$ is proportional to n/k, and for the case of a completely random network, i.e., $p = 1$, $L(n, k, p = 1) \propto \ln n / \ln k$. These properties indicate that the average path length is reduced as p increases for large n. The path length of a network directly indicates the convergence performance by (15). For fast convergence of the gossip process, a completely random topology, i.e., $p = 1$, will be the best choice. However, as mentioned before, rewiring connections causes long range connections which can be a disadvantage in wireless communications.

The range of connections in a graph affects directly the power consumption which is an important issue in wireless sensor networks. The following is a basic power consumption model with respect to distance in wireless communications [22]:

$$P_T(D) = P_{T0} + \frac{\gamma \times D^{\alpha}}{\xi} \tag{16}$$

where P_T is the power of transmitting, P_{T0} is a constant component which does not depend on the transmission range D, γ is a constant determined by characteristics of antennas and the minimum required received power determined by the receiver, ξ is called the drain efficiency, and α is the path loss exponent which is about two for free space and will increase in the presence of obstacles. Equation (16) shows that the transmitting energy consumption of each node grows to the power of α as the receiving node location recedes from the transmitting node.

From the path length analysis on topology and the power consumption model (16), we find an interesting trade-off relationship between energy efficiency and convergence performance on the network topology. For a small rewiring probability p, the frequency of the long range communication is low and vice versa. However, for the small-world network, as p grows from 0, the average path length $L(n, k, p)$ rapidly drops to the average path length of random networks, $L(n, k, 1)$. From this property, we can improve the convergence performance to a similar level as the completely random networks with only a small increase in power consumption.

V. Simulation Results

In this section, simulation results are presented to validate the proposed algorithm. We perform the simulations to verify the join operation of convex hulls and convergence of the gossip process which guarantees global optimality. We also check the small-world phenomenon of the example to discover a potential methodology that improves the performance while maintaining the energy consumption level.

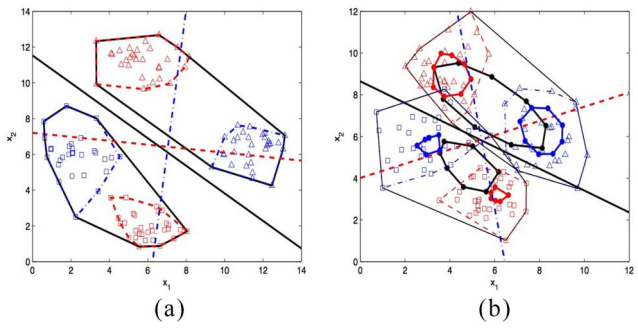


Fig. 8. Result of the join operation of convex hulls in (a) linearly separable case and (b) linearly nonseparable case: for the positive (square) and negative (triangle) datasets obtained by node 1 (red) and node 2 (blue), the local convex hulls and SVM solutions of nodes 1 and 2 are represented as the red-dashed (node 1) and blue dash-dotted (node 2) polygons and lines, before exchanging data. Then, nodes 1 and 2 exchange the extreme point sets with each other to obtain the global (reduced) convex hull and SVM solution (the black solid polygon and line).

A. Linear Example: Join Operation of Convex Hulls

In this example, we simulate and test the join operation of convex hulls using the Ripley data set [16] which consists of two classes where the data for each class have been generated by a mixture of two Gaussian distributions. We assume that there are two nodes with a 2-D data set of two classes, and they deliver only the extreme point set to each other to get the global SVM solution. Fig. 8(a) shows the result of the join operation of convex hulls, which are linearly separable. As a result of performing the join operation on two classes separately, we obtain a pair of global convex hulls and also a linear discriminant function that is the solution of the global SVM. Similarly, the process above is also applicable to linearly nonseparable cases. The Ripley data set shown in Fig. 8(b) is densely distributed, such that there is no linear solution which separates the two classes completely. As mentioned in Section III-B, for this linearly nonseparable case, we employ the reduced convex hull ($\mu = 0.2$) instead of the convex hull. In Fig. 8(b), important results are represented such as the global convex hull, the global reduced convex hull, the local convex hulls, and the local reduced convex hulls. As shown in the results, the reduced convex hulls do not satisfy the join operation. However, after obtaining a pair of convex hulls in the same manner as in the linearly separable case, we can have the SVM solution of the linearly nonseparable case. Note that the reduced convex hull does not have to be calculated explicitly because the information about it does not show up in the middle of the communication process.

B. Example: Centralized and Distributed SVM Training Over WSNs

In this example, we perform centralized and distributed SVM training over wireless sensor networks. We assume that each node has a binary sensor and the goal of the SVM training is to divide the workspace into two regions, one for positive and one for negative values. We also assume that the communication topology is directed and has n vertices with a degree of six, i.e., $k = 6$. This intercommunication topology is utilized for the distributed SVM training.

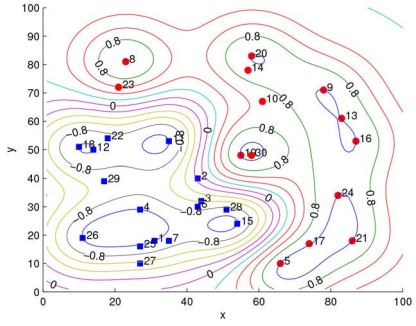


Fig. 9. Result of the centralized geometric SVM: the contour of the discriminant value is plotted over the workspace. The zero-valued contour is the discriminant function of the SVM.

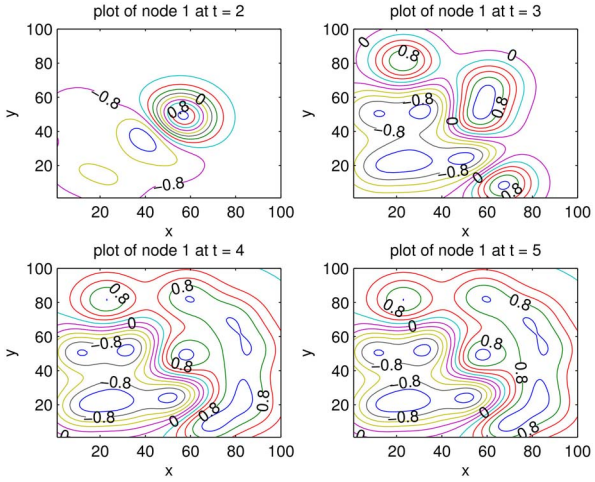


Fig. 10. Result of the distributed SVM: evolution of the contour of the discriminant value for node 1 is plotted over the workspace. The zero-valued contour is the discriminant function of the SVM.

1) Centralized SVM Training: In order to check the performance of geometric SVM training and set up the global reference results for the distributed SVM, we perform the centralized SVM training first. Since the data distribution is nonlinear, we apply the convex hull algorithm with the Gaussian kernel and geometric SVM to obtain the solution. Fig. 9 shows the result of the centralized SVM training and it shows the well-separated solution of the centralized geometric SVM.

2) Distributed SVM Training: In this simulation, we test the proposed gossip-based distributed SVM training over the wireless sensor network setup in the same manner as the centralized one, but with the intercommunication topology described in the beginning of Section V-B. The result of the distributed SVM training is shown in Fig. 10. As time advances, node 1 collects more data from its neighbors to converge to the global solution. At $t = 4$, node 1 obtains the same solution as the global solution from the centralized SVM training described in Fig. 9, and maintains the global solution at $t = 5$. The important thing is that the proposed algorithm yields the identical global solution of the SVM at every node. Fig. 11 shows the agreement of SVM training results of nodes 5, 10, 15, 20, 25, and 30. They show identical contour patterns that are the same as the centralized SVM in Fig. 9. We can

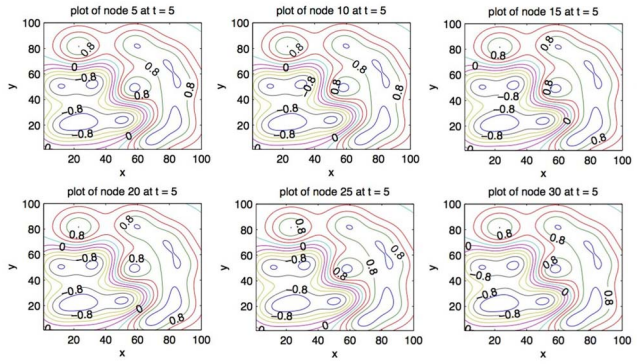


Fig. 11. Agreements of the distributed SVM for nodes 5, 10, 15, 20, 25, and 30: the contour of the discriminant value is plotted over the workspace. The zero-valued contour is the discriminant function of the SVM.

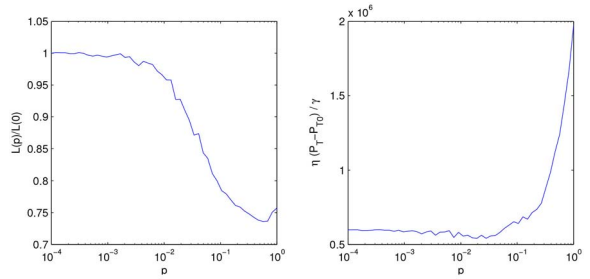


Fig. 12. (a) Average path length. (b) Power consumption analysis for various rewiring probabilities: a logarithmic horizontal scale has been used to resolve the rapid changes in $L(p)$ and P_T.

also confirm the agreement of SVM training results for the other nodes.

C. Small-World Properties in Wireless Sensor Networks

In Section IV-E, we discuss the small-world network properties to enhance the convergence performance with only a small loss of energy. The key issues of this topic are the average path length depending on the rewiring probability and the power consumption by the long-range communications. In this section, the path length analysis and the power consumption analysis are performed using the same setup and process as in Section V-B.

Fig. 12(a) shows the average path length $L(n, k, p)$ for the randomly rewired graphs during the distributed SVM simulation performed in Section V-B, where $n = 30$ and $k = 6$. The plot of $L(n, k, p)$ shown in Fig. 12(a) is the average over 30 random realizations of the rewiring process, and has been normalized by the value of $L(n, k, 0)$. As shown in the figure, $L(n, k, p)$ drops rapidly as the rewiring probability p increases.

On the other hand, Fig. 12(b) shows the estimated average transmitting power of the distributed SVM simulations performed in Section V-B. The estimation has been achieved using the transmitting power model (16). Because P_{T0}, γ and η are constants in the (16), the data plotted in the figure has been calculated by

$$\eta \left(P_T - P_{T0}\right)/\gamma = d^{\alpha} \tag{17}$$

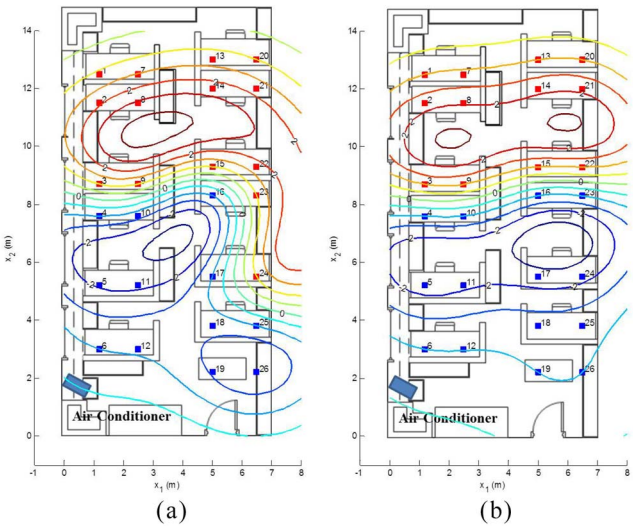


Fig. 13. Environment of the indoor experiment and the results: the blue squares indicate nodes detecting a low temperature and the red squares indicate nodes detecting a high temperature. Part (a) and (b) show the results of independent trials.

and we set $\alpha = 2$ by assuming that the environment of wireless communication is a free space. As shown in the figure, the average transmission power P_T maintains a low level in the small probability region, but drastically increases in the high probability region.

Comparing Fig. 12(a) and (b), there is a region where both the average path length and the estimated power consumption are small. In this region, we can substantially improve the performance of our proposed algorithm in terms of the convergence speed with only a negligible increase in energy consumption.

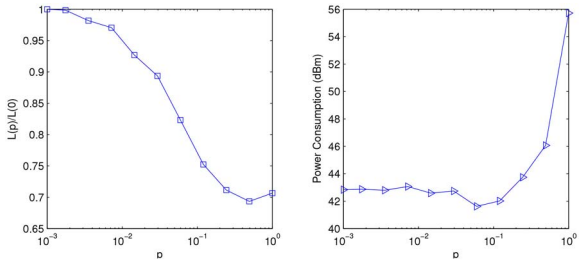
VI. Experimental Results

In this section, we describe the experimental results for analyzing the proposed algorithm and testing its feasibility in practical environment. As shown in Fig. 13, we try to determine the actual cooling region of an air conditioner located at the corner of an office using the proposed algorithm over the wireless sensor network. A total of 26 sensor nodes are deployed in the indoor environment. For these experiments, each sensor node employs TinyOS with TelosB platform and communicates using the IEEE 802.15.4 standard. Furthermore, the network topology is implemented such that each node has a degree of six, i.e., a node receives radio packets from six neighbors, with rewiring probability which we can control.

The radio transmission power also can be controlled by the geometric interpretation. In Table I, there are seven different output powers and their corresponding current consumptions for a single transmission, provided by the manual of the CC2420 radio chip installed in the sensor nodes. Through the prestudy with theses values, we obtain the minimum transmission power for the communication ranges with 95 percent reliability (the success rate of the communication) as written in the first column of Table I.

TABLE I
RADIO TRANSMISSION POWER

Range (m)	Output Power (dBm)	Current Consumption (mA)
0∼1	-25	8.5
1∼2	-15	9.9
2∼3	-10	11.2
3∼4	-7	12.5
4∼6	-3	15.2
6∼10	-1	16.5
10∼	0	17.4



(a) (b)

Fig. 14. Experimental results of (a) average path length and (b) power consumption for various rewiring probabilities: a logarithmic horizontal scale has been used to resolve the rapid changes in $L(p)$ and P_T.

The contour plot described in Fig. 13 shows the distributed SVM results of the experiment. The blue squares indicate nodes detecting a low temperature and the red squares indicate nodes detecting a high temperature. The results of the distributed SVM training show well-separated solutions.

We also analyze the small-world effects with the various rewiring probabilities. Fig. 14 shows the experimental results of $L(p)$ and P_T for various rewiring probabilities. The path length $L(p)$ can be obtained from the convergence time and the transmission power P_T is calculated based on the data in Table I.

The results have a similar pattern to the simulation results in Fig. 12; there exists a region around $p = 10^{-1}$ where both the average path length and the power consumption are small. According to this, the wireless sensor network can have its best performance with the rewiring probability $p = 10^{-1}$ in terms of the convergence speed and energy consumption.

VII. CONCLUSION

In this paper, we propose a distributed SVM training algorithm. Based on geometric SVM, we propose to gossip only the extreme point sets with neighboring nodes. The proposed algorithm has a simple communication mechanism and guarantees to converge to a common global solution. In order to prove the global optimality by exchanging only the extreme points with neighboring nodes, we utilize the join operation of the convex hulls and show that the gossip process can achieve the agreement.

Furthermore, we analyze the scalability in terms of the amount of exchanged information and the convergence time. The exchanged information of the proposed algorithm is the extreme point set and in general, the number of extreme points is finite. However, in the worst case where the extreme points lie on a round convex curve, it is possible that the number of extreme points grows to infinity. To resolve this possibility, we propose the naive convex hull algorithm which bounds the number of the extreme points.

The convergence time is strongly related to the average path length of the network topology. In order to analyze the performance of the proposed algorithm over sensor networks, we adopt the concept of the small-world network. From the analysis of the path length and the power consumption, we find a trade-off between energy efficiency and convergence performance. Moreover, due to the small-world phenomenon, the average path length is reduced by using long-range rewiring with a small probability, which gives an opportunity to drastically improve the convergence performance of the algorithm with only a small increase in power consumption.

APPENDIX

Proof of Lemma 1

For any $z \in C(X)$, $z = \sum_i \lambda_i x_i$, with $x_i \in X$, $\sum_i \lambda_i = 1$, and $\lambda_i \geq 0$, $i = 1, \ldots, N$, where $N = |X|$. If there is an element x_k of $X = \{x_i\}_{i=1}^N$ which can be represented as a convex combination of any other distinct points in X, we can substitute $x_k = \sum_{j \neq k} \beta_j x_j$ where, $x_k \neq x_j \in X$, $\sum_{j \neq k} \beta_j = 1$, $\beta_j \geq 0$. Then, the point z in $C(X)$ can be formulated as follows:

$$z = \sum_{i=1, i\neq k}^{N} \lambda_i x_i + \lambda_k \left(\sum_{i=1}^{k-1} \beta_i x_i + \sum_{i=k+1}^{N} \beta_i x_i \right)$$
$$= \sum_{i=1}^{k-1} (\lambda_i + \lambda_k \beta_i) x_i + \sum_{i=k+1}^{N} (\lambda_i + \lambda_k \beta_i) x_i$$

and the coefficients are obviously nonnegative, furthermore, their sum is 1 as follows:

$$\sum_{i=1}^{k-1} (\lambda_i + \lambda_k \beta_i) + \sum_{i=k+1}^{N} (\lambda_i + \lambda_k \beta_i)$$
$$= \sum_{i=1, i\neq k}^{N} \lambda_i + \lambda_k \sum_{i=1, i\neq k}^{N} \beta_i = 1 = \sum_{i=1}^{N} \lambda_i = 1.$$

Let X' be a new convex set defined as $X' = X \backslash x_k$ which still holds that $C(X) = C(X')$ and $|X'| < |X|$. Same as above, we can continue setting X' to a new X iteratively until all the elements of X' cannot be a convex combination of any other distinct points in X'. Then, by definition, the final X' is the extreme point set $\mathcal{E}(X)$ which has the minimum size to represent $C(X)$.

Proof of Lemma 3

From the definition of a convex hull, we can set $z \in C(C(X_j^s) \cup C(X_i^s))$ as $z = \sum_k \lambda_k x_k$, where $\lambda_k \geq 0$, $\sum_k \lambda_k = 1$, and $x_k \in C(X_j^s) \cup C(X_i^s)$. Let $x_k \in C(X_j^s)$ for $k = 1, \cdots K_1$, and $x_k \in C(X_i^s) - C(X_j^s)$ for $k = K_1 + 1, \ldots, K_2$, where $K_1 = |X_j^s|$ and $K_2 = |X_j^s \cup X_i^s|$, then $z \in C(C(X_j^s) \cup C(X_i^s))$ can

be represented using the definition of a convex hull as follows:

$$z = \sum_{k=1}^{K_1} \lambda_k \left(\sum_{n:p_n \in X_j^s} \alpha_n^{(k)} p_n \right) + \sum_{k=K_1+1}^{K_2} \lambda_k \left(\sum_{m:q_m \in X_i^s} \beta_m^{(k)} q_m \right)$$
$$= \sum_{n\ :\ p_n \in X_j^s} \left(\sum_{k=1}^{K_1} \lambda_k \alpha_n^{(k)} \right) p_n + \sum_{m\ :\ q_m \in X_i^s} \left(\sum_{k=K_1+1}^{K_2} \lambda_k \beta_m^{(k)} \right) q_m$$

where, $\alpha_n^{(k)} \geq 0$ and $\beta_m^{(k)} \geq 0$ with $\sum_n \alpha_n^{(k)} = \sum_m \beta_m^{(k)} = 1$ and the superscript (k) denotes that the convex combination coefficients $\alpha_n^{(k)}$ $(n = 1, \ldots, |X_j^s|)$ and $\beta_m^{(k)}$ $(m = 1, \cdots, |X_i^s|)$ construct x_k. Here, it is obvious that $\sum_{k=1}^{K_1} \lambda_k \alpha_n^{(k)} \geq 0$ and $\sum_{k=K_1+1}^{K_2} \lambda_k \beta_m^{(k)} \geq 0$. Moreover

$$\sum_n \left(\sum_{k=1}^{K_1} \lambda_k \alpha_n^{(k)} \right) + \sum_m \left(\sum_{k=K_1+1}^{K_2} \lambda_k \beta_m^{(k)} \right)$$
$$= \sum_{k=1}^{K_1} \lambda_k \left(\sum_n \alpha_n^{(k)} \right) + \sum_{k=K_1+1}^{K_2} \lambda_k \left(\sum_m \beta_m^{(k)} \right)$$
$$= \sum_{k=1}^{K_1} \lambda_k + \sum_{k=K_1+1}^{K_2} \lambda_k = \sum_k \lambda_k = 1.$$

The above properties show that $z \in C(X_j^s \cup X_i^s)$, and that means (12) holds.

REFERENCES

[1] A. Barrat and M. Weight, "On the properties of small-world network models," *Eur. Phys. J. B, Condens. Matt. Complex Syst.,* vol. 13, no. 3, pp. 547–560, 2000.

[2] O. Brdiczka, M. Langet, J. Maisonnasse, and J. Crowley, "Detecting human behavior models from multimodal observation in a smart home," *IEEE Trans. Autom. Sci. Eng.,* vol. 6, no. 4, pp. 588–597, Oct. 2009.

[3] K. P. Bennett and E. J. Bredensteiner, "Duality and geometry in SVM classifiers," in *Proc. 7th Int. Conf. Mach. Learn.,* Stanford, CA, USA, 2000, pp. 57–64.

[4] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA, USA: Athena Scientific, 1997.

[5] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.,* vol. 22, no. 4, pp. 469–483, 1996.

[6] K. Zhu *et al.*, "Parallelizing support vector machines on distributed computers," *Adv. Neural Inf. Process. Syst.*, vol. 20, pp. 257–264, 2008.

[7] E. Gilbert, "An iterative procedure for computing the minimum of a quadratic form on a convex set," *SIAM J. Control*, vol. 4, no. 1, pp. 61–80, 1966.

[8] K. Flouri, B. Beferull-Lozan, and P. Tsakalides, "Distributed consensus algorithms for SVM training in wireless sensor networks," in *Proc. IEEE 16th Eur. Signal Process. Conf.,* Laussane, Switzerland, 2008, pp. 1048–1054.

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

[10] H. Graf, E. Cosatto, L. Bottou, I. Dourdanovic, and V. Vapnik, "Parallel support vector machines: The cascade SVM," in *Advances in Neural Information Processing Systems,* vol. 17. Cambridge, MA, USA: MIT Press, 2005.

[11] D. Gu and Z. Wang, "Distributed regression over sensor networks: A support vector machine approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS),* Nice, France, 2008, pp. 3286–3291.

[12] M. Krein and D. Milman, "On extreme points of regular convex sets," *Stud. Math.,* vol. 9, no. 1, pp. 133–138, 1940.

[13] C. Lu and L. Fu, "Robust location-aware activity recognition using wireless sensor network in an attentive home," *IEEE Trans. Autom. Sci. Eng.,* vol. 6, no. 4, pp. 598–609, Oct. 2009.

[14] X. Nguyen, M. Jordan, and B. Sinopoli, "A kernel-based learning approach to ad-hoc sensor network localization," *ACM Trans. Sensor Netw.,* vol. 1, no. 1, pp. 134–152, 2005.

[15] E. Osuna and O. D. Castro, "Convex hull in feature space for support vector machines," in *Proc. 8th Ibero-Amer. Conf. Artif. Intell.,* Seville, Spain, 2002, pp. 411–419.

[16] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[17] S. Simić, "A learning theory approach to sensor networks," *IEEE Pervasive Comput.,* vol. 2, no. 4, pp. 44–49, Oct./Dec. 2003.

[18] M. L. Stone, P. R. Armstrong, D. D. Chen, G. H. Brusewitz, and N. O. Maness, "Peach firmness prediction by multiple location impulse testing," *Trans. ASAE,* vol. 47, no. 1, pp. 115–119, 1998.

[19] S. Theodoridis and M. Mavroforakis, "Reduced convex hulls: A geometric approach to support vector machines," *IEEE Signal Process. Mag.,* vol. 24, no. 3, pp. 119–122, May 2007.

[20] D. Tran and T. Nquyen, "Localization in wireless sensor networks based on support vector machines," *IEEE Trans. Parallel Distrib. Syst.,* vol. 19, no. 7, pp. 981–994, Jul. 2008.

[21] V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[22] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *Proc. IEEE Sensor Ad Hoc Commun. Netw.,* Reston, VA, USA, 2006, pp. 286–295.

[23] J. Wang, H. Lee, J. Wang, and C. Lin, "Robust environmental sound recognition for home automation," *IEEE Trans. Autom. Sci. Eng.,* vol. 5, no. 1, pp. 25–31, Jan. 2008.

[24] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature,* vol. 393, pp. 440–442, Jun. 1998.

[25] D. Caragea, A. Silvescu, and V. Honavar, "Agents that learn from distributed dynamic data sources," in *Proc. Workshop Learn. Agents Agents 2000/ECML*, Barcelona, Spain, pp. 53–61.

[26] W. Kim, J. Park, J. Yoo, H. Kim, and C. Park, "Target localization using ensemble support vector regression in wireless sensor networks," *IEEE Trans. Cybern.,* vol. 43, no. 4, pp. 1189–1198, Aug. 2013.

[27] E. Gilbert, "Minimizing the quadratic form on a convex set," *SIAM J. Control,* vol. 4, pp. 61–80, 1966.

[28] V. Franc and V. Hlaváč, "An iterative algorithm learning the maximal margin classifier," *Pattern Recognit.,* vol. 35, pp. 1985–1996, Sep. 2003.

[29] M. Mavroforakis and S. Theodoridis, "A geometric approach to support vector machine (SVM) classification," *IEEE Trans. Neural Netw.,* vol. 17, no. 3, pp. 671–682, May 2006.

[30] J. Yoo, W. Kim, and H. Kim, "Event-driven Gaussian process for object localization in wireless sensor networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS),* San Francisco, CA, USA, 2011, pp. 2790–2795.

Woojin Kim (S'10) received the B.S. degree in electrical engineering from the Korean Advanced Institute of Science and Technology, Daejeon, Korea, and the Ph.D. degree in mechanical and aerospace engineering from Seoul National University, Seoul, Korea.

In 2014, he joined the Electronics and Telecommunications Research Institute, Daejeon, as a Researcher. His current research interests include control, automation, coordination and applications of systems including sensor networks, mobile robots, and industrial machines.

Miloš S. Stanković (M'04) received the bachelor's (Dipl.Ing.) and master's degrees from the School of Electrical Engineering, University of Belgrade, Belgrade, Serbia, in 2002 and 2006, respectively, and the Ph.D. degree in systems and entrepreneurial engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, IL, USA, in 2009.

He was a Research and Teaching Assistant at the Control and Decision Group, Coordinated Science Laboratory, UIUC, from 2006 to 2009. From 2009 to 2012, he was a Post-Doctoral Researcher at the Automatic Control Laboratory and the ACCESS Linnaeus Centre, KTH Royal Institute of Technology, Stockholm, Sweden. In 2012, he joined the Innovation Center, School of Electrical Engineering, University of Belgrade, as an EU Marie Curie Research Fellow. His current research interests include decentralized decision making, networked control systems, dynamic game theory, optimization, and machine learning.

Karl H. Johansson (F'13) received the M.Sc. and Ph.D. degrees in electrical engineering from Lund University, Lund, Sweden.

He is a Director of the ACCESS Linnaeus Centre and a Professor at the School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden. He is a Wallenberg Scholar and has held a six-year Senior Researcher Position with the Swedish Research Council. He is also the Head of the Stockholm Strategic Research Area ICT The Next Generation. He has held visiting positions at the University of California, Berkeley, Berkeley, CA, USA, from 1998 to 2000 and at the California Institute of Technology, Pasadena, CA, USA, from 2006 to 2007. His current research interests include networked control systems, hybrid and embedded system, and applications in transportation, energy, and automation systems.

Mr. Johansson was the recipient of the Best Paper Award from the IEEE International Conference on Mobile Ad-hoc and Sensor Systems in 2009, the Best Theory Paper Award from the World Congress on Intelligent Control and Automation in 2014, the Wallenberg Scholar as one of the first ten scholars from all sciences by the Knut and Alice Wallenberg Foundation in 2009, an Individual Grant for the Advancement of Research Leaders from the Swedish Foundation for Strategic Research in 2005, the triennial Young Author Prize from IFAC in 1996, the Peccei Award from the International Institute of System Analysis, Austria in 1993, and the Young Researcher Award from Scania in 1996 and from Ericsson in 1998 and 1999. He has been a Governing Board Member of the IEEE Control Systems Society and the Chair of the IFAC Technical Committee on Networked Systems. He has also been an Editorial Board Member of several journals, including *Automatica*, the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and the *IET Control Theory and Applications*. He is currently on the Editorial Board of the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS and the *European Journal of Control*. He has been a Guest Editor for special issues, including two issues of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He was the General Chair of the ACM/IEEE Cyber-Physical Systems Week 2010 in Stockholm and an IPC Chair of several conferences. He has served on the Executive Committees of several European research projects in the area of network embedded systems.

H. Jin Kim (M'02) received the B.S. degree in mechanical engineering from the Korean Advanced Institute of Science and Technology, Daejeon, Korea, in 1995, and the M.S. and Ph.D. degrees from the University of California, Berkeley, Berkeley, CA, USA, in 1999 and 2001, respectively.

From 2002 to 2004, she was a Post-Doctorial Researcher and a Lecturer with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. In 2004, she joined the School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Korea, as an Assistant Professor, where she is currently a Professor. Her current research interests include robotics and intelligent control.