This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Boundary tracking of continuous objects based on binary tree structured SVM for industrial wireless sensor networks

PLEASE CITE THE PUBLISHED VERSION

<https://doi.org/10.1109/tmc.2020.3019393>

PUBLISHER

Institute of Electrical and Electronics Engineers (IEEE)

VERSION

AM (Accepted Manuscript)

PUBLISHER STATEMENT

Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

LICENCE

All Rights Reserved

REPOSITORY RECORD

Liu, Li, Guangjie Han, Zhengwei Xu, Jinfang Jiang, Lei Shu, and Miguel Martinez-Garcia. 2020. "Boundary Tracking of Continuous Objects Based on Binary Tree Structured SVM for Industrial Wireless Sensor Networks". Loughborough University. <https://hdl.handle.net/2134/13026356.v1>.

Boundary Tracking of Continuous Objects Based on Binary Tree Structured SVM for Industrial Wireless Sensor Networks

Li Liu, Guangjie Han, *Senior Member, IEEE*, Zhengwei Xu, Jinfang Jiang, Lei Shu, *Senior Member, IEEE* and Miguel Martínez-García, *Member, IEEE*

Abstract—Due to the flammability, explosiveness and toxicity of continuous objects (e.g., chemical gas, oil spill, radioactive waste) in the petrochemical and nuclear industries, boundary tracking of continuous objects is a critical issue for industrial wireless sensor networks (IWSNs). In this article, we propose a continuous object boundary tracking algorithm for IWSNs – which fully exploits the collective intelligence and machine learning capability within the sensor nodes. The proposed algorithm first determines an upper bound of the event region covered by the continuous objects. A binary tree-based partition is performed within the event region, obtaining a coarse-grained boundary area mapping. To study the irregularity of continuous objects in detail, the boundary tracking problem is then transformed into a binary classification problem; a *hierarchical soft margin support vector machine* training strategy is designed to address the binary classification problem in a distributed fashion. Simulation results demonstrate that the proposed algorithm shows a reduction in the number of nodes required for boundary tracking by at least 50%. Without additional fault-tolerant mechanisms, the proposed algorithm is inherently robust to false sensor readings, even for high ratios of faulty nodes ($\approx 9\%$).

Index Terms—Industrial wireless sensor networks, continuous objects, boundary tracking, binary tree, support vector machines

1 INTRODUCTION

PRODUCTION activities in petrochemical, plastics, and energy industries are always accompanied with the risk of *normal accidents* [1] which may result in leakage of chemical gases, radioactive contamination, oil spill and other *continuous objects* [2]. These continuous objects tend to spread over a wide region, which may cause poisoning, dangerous fire, or an explosion. Studying the spatio-temporal evolution of continuous objects is of significance in disaster prediction and reduction, including hazard suppression and personnel evacuation [3].

Incorporating Wireless Sensor Networks (WSNs) within industrial automation systems – i.e., Industrial Wireless Sensor Networks (IWSNs) – can assist in mitigating the impact of such accidents. IWSNs embed smart electronics into production systems that can run intelligent algorithms [4][5]. A typical IWSN consists of hundreds of wireless tiny sensor nodes installed on industrial equipment or deployed in the field [6]. The nodes are low-cost, but with the capabilities of sensing, processing,

and communication. By offering sensing services and ubiquitous networking to industrial systems, IWSNs show great potential in serving industrial safety before and after the occurrence of an accident. For example, after the leakage of a toxic gas, using pervasive sensor nodes in IWSNs to acquire the concentration of gas in different positions, the distribution of leak gas can be visualized and emergency relief operation can be carried out orderly [7].

Recently, research efforts around the globe are focusing on exploiting the fine-grained perception of IWSNs to track the movement of continuous objects [8][9]. Detecting the boundary of diffusing continuous objects by identifying the sensor nodes located closest to the evolving front line attracts intensive attention [10]–[12]. The identified nodes are termed as boundary nodes [13]. Boundary nodes report their coordinate positions to a base station for delineating the area affected by the hazard [14].

The challenges behind the above-mentioned idea focus on two aspects. First, the base station monitors either the interior or the exterior boundary by selecting the boundary nodes that are either inside or outside the event region. As shown in Fig. 1, the interior boundary is formed by connecting interior boundary nodes w, x, y, z , while the exterior boundary is formed by connecting exterior boundary nodes m, n, o, p, q . The predicted actual boundary is determined between the interior and the exterior boundary. Secondly, because continuous objects constantly change their sizes and shapes, the number of boundary nodes increases significantly with the expansion of the boundary.

- Li Liu is with the College of Internet of Things Engineering, Hohai University, Changzhou, China, and also with the State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China. E-mail: liulihhuc@gmail.com.
- Guangjie Han, Zhengwei Xu, and Jinfang Jiang are with the College of Internet of Things Engineering, Hohai University, Changzhou, China. E-mail: hanguangjie@gmail.com, xuzhengweihhu@outlook.com, jiangjinfang1989@gmail.com.
- Lei Shu is with the College of Engineering, Nanjing Agricultural University, China. E-mail: lei.shu@ieee.org.
- Miguel Martínez-García is with the Department of Aeronautical and Automotive Engineering, Loughborough University, LE113TU, UK. E-mail: M.Martinez-Garcia@lboro.ac.uk.

Fig. 1. Continuous objects tracking based on boundary node selection.

In addition, low-cost nodes are prone to failure in harsh industrial environments, leading to the sensor readings that contradict reality. Based on false sensor readings, a fault node may yield a spurious event boundary, potentially deceiving the base station and preventing the signaling of an actual event [15] [16].

In this article, a *binary tree structure-based continuous object boundary tracking algorithm* (BTS-COT) is proposed for IWSNs, which makes active use of on-site supervision data. The BTS-COT addresses the problem of boundary tracking by learning a classification hyperplane about the environment, where event and non-event nodes exist. Taking advantage of the sparse representation that support vector machine (SVM) provides for decision boundaries, we regard sensor readings as training samples, and enable sensor nodes to distributively implement SVM learning under the binary tree architecture. As a result, boundary node selection is transformed into support vector searching in a high dimensional feature space. Without specially designed fault-tolerant mechanisms, the algorithm is robust to the abnormal sensor readings caused by node failure. The main contributions of this article are:

- We propose a full binary tree structured network partition mechanism to achieve boundary area mapping, reducing the searching space of boundary nodes.
- We realize boundary tracking using soft margin SVM with binary tree architecture, and design a hierarchical SVM learning strategy.
- The exterior and interior boundaries are used as upper and lower bounds, in order to estimate the real boundary probabilistically.
- Simulation experiments are performed to evaluate the tracking accuracy of the proposed algorithm in scene of uniform/non-uniform diffusion, and to verify the fault-tolerance of the proposed algorithm.

The remainder of this article is organized as follows: Section II reviews related work regarding continuous objects tracking in sensor networks. Section III presents the boundary area mapping under full binary tree structured IWSNs. Section IV introduces the boundary track-

ing algorithm using soft margin SVM with binary tree architecture. Experimental results are given in Section V, and conclusions and future work are shown in Section VI.

2 RELATED WORK

In recent years, multiple research studies have attempted to realize boundary tracking of continuous objects, by partitioning the sensor network into uniform or non-uniform cells. The partition units include grids, Delaunay triangulation cells, Voronoi cells and other types of closed polygons – generated through different planarization algorithms.

2.1 Grid-based Boundary Tracking

Han *et al.* [17] proposed a continuous object tracking scheme with two-layer grid model (TGM-COT) in sensor networks. The TGM-COT divides the network into coarse-grained grids for continuous objects detection. Fine-grained grids are established within the coarse-grained grids, which contain the boundary of the continuous object. Boundary node selection is performed separately per each fine-grained grids. The TGM-COT is easy to implement; it only involves grid-based partition and node clustering. However, frequent cluster formation and deletion downgrade the energy efficiency of the TGM-COT.

Oh *et al.* [18] combined the two-layer grid model in the TGM-COT with a convex hull algorithm to track the boundary of continuous objects. Since convex hull methods cannot work on concave polygons, the authors proposed a recovery mechanism to detect the shape loss between the obtained convex hull and the actual boundary. This method performs well in densely deployed networks, but incurs in high communication overheads when tracking the boundaries of high degree of irregular.

Kim *et al.* [19] proposed an origin-mediated communication scheme to support sink mobility (OCSM) for phenomena detection in IWSNs. When a phenomena occurs in a specific location, the sensor node which has the largest intensity of sensing is selected as a source node. The source node establishes a data delivery path to a mobile sink through a virtual backbone network. The virtual backbone network consists of routing nodes deployed in the form of grids. Although the OCSM utilizes a mobile sink to relieve the problem of energy imbalance, for the backbone nodes in the virtual backbone network, the communication overhead cannot be reduced.

2.2 Voronoi-based Boundary Tracking

Imran *et al.* [20] proposed an efficient continuous object detection approach for Voronoi diagram-based WSNs. In this method, each node in the network is positioned in a Voronoi cell. The node that detects the object within its Voronoi cell utilizes two-hop neighbor detection information to avoid reporting the minimal change in

the phenomenon shape. If a node depletes its energy, a spare node is waken up in the Voronoi cell where the dead node is detected. It is noted that the division of one network using Voronoi cells is non-unique, leading to variable results in boundary tracking.

2.3 Delaunay Triangulation-based Boundary Tracking

Han *et al.* [21] proposed a novel boundary recognition and tracking algorithm for continuous objects (BRTCO) in WSNs. With the help of triangular mesh, the nodes located outside the event region – but close to the boundary – collaborate to determine the geometric characteristics of the boundary line, and filter out less reliable nodes towards improving the tracking accuracy. Similarly to the Voronoi-based partition, Delaunay triangulation-based partitions cannot ensure the uniqueness of the results of boundary tracking.

2.4 Planarization-based Boundary Tracking

Ping *et al.* proposed [22] a two-stage boundary face detection mechanism for duty-cycled IWSNs. When a target event is detected by active nodes, the boundary faces of the continuous object are constructed by planarization algorithms. Boundary nodes are selected within each boundary face so that the size of boundary faces can be refined. The authors adopted four kinds of planarization algorithms including Gabriel graph (GG), relative neighborhood graph (RNG), Yao graph (YG), and k -localized Delaunay graph (LDeLk). Experimental results show that LDeL2 achieves the best accuracy among these planarization algorithms. Rahman *et al.* [23] combined four proximity graphs with different spatial interpolation methods for accurate boundary estimation in a dense duty-cycled WSNs. The four proximity graphs studied in this work are the same as those in [22]. To estimate the sensor values of the geographical points where asleep nodes are positioned, the authors studied four spatial interpolation methods, including Inverse Distance Weighting, Kriging, Spline, and Natural. Then, the inner and outer boundary can be determined without awaking dormant nodes.

2.5 Discussion

Through network partition, the process of boundary node selection is easy to manage. Simultaneously, boundary tracking can be performed energy-efficiently. However, the fixed cells utilized in the existing algorithms cannot adapt to the evolution of continuous objects. To track the boundary of continuous objects adaptively, a dynamic network partition mechanism, which is able to adjust the resolution of partition cells, appears to be more suitable than a static one.

3 BOUNDARY AREA MAPPING UNDER FULL BINARY TREE STRUCTURED NETWORK

This section introduces a full binary tree structured network partition mechanism. The mechanism adaptively adjusts the size of partition cells based on the area of the event region covered by the continuous object. The aim is to localize the boundary area of the object by a collection of non-uniform partition cells in the full binary tree structured network.

3.1 Coarse-grained Event Region Search based on Inter-cluster Collaboration

For the sake of concreteness in the description of the movement of continuous objects, we focus on the diffusion of leakage gas – which is the most common event in petrochemical production. Nowadays, various types of chemical sensor nodes are available to monitor gas leakage. The nodes provide analog quantity outputs to measure the gas concentration. The presence of gas leakage always corresponds to a high sensor reading, while a low reading indicates no event. In order to suppress external interference and measurement noise, a node judges whether a specific event is detected by mapping its numerical sensor reading into a binary variable. If the node is reached by the evolving gas cloud, its numerical sensor reading will be mapped into a positive value – or a negative value otherwise.

Fig. 2. Event region search based on inter-cluster collaboration.

As shown in Fig. 2(a), we consider an IWSN in which N homogeneous nodes are organized into clusters. The nodes in each cluster switch between sleep and active state following a predefined duty cycle. At each time step, the active nodes in the network achieve full coverage to detect the existence of leakage gas. The area covered by leakage gas is termed as the *event region*. In each cluster, the active nodes which detect the existence of gas send *Event_Msg* packets to their cluster head. The cluster head extracts the node ID and corresponding coordinates to infer the upper bound and lower bound of the event region, at the scale of the local cluster. Through an integrated combination of the local bounds determined by all of the cluster heads, a global bound of the event region can be drawn.

An example is showcased in Fig. 2(b); the cluster head CH_A in cluster A extracts coordinates from the

Event_Msg packets received from its cluster members, and then determines the maximum and minimum of x coordinate and y coordinate. The local bound of the event region in cluster A can be represent by a 4-tuples $Edge_A = [Xmin_A, Xmax_A, Ymin_A, Ymax_A]$. CH_A sends $Edge_A$ to its neighboring cluster heads CH_B and CH_C . CH_B compares its local 4-tuples $Edge_B$ with $Edge_A$ to determine the bound of the event region covered by cluster A and B . The 4-tuples keep on exchanging and updating in the network until convergence. Finally, the global bound of the event region convergence to a 4-tuple: $Edge = [Xmin_C, Xmax_B, Ymin_C, Ymax_A]$. It can be observed [Fig. 2(b)] that a rectangle zone can be determined under the constraint of the global 4-tuples $Edge$. The algorithm uses the minimum circumscribed circle of the rectangle as the outmost layer of a full binary tree, and proposes a full binary tree structured network partition mechanism.

3.2 Full Binary Tree Structured Network Partition

Binary trees are defined recursively as a collection of elements (starting at a root), where each element has at most two children. The elements that are directly under an element are called *children*. The element directly above something is called the *parent*. The elements with no children are referred to as *leaves*. A binary tree is considered full if every element in the tree has either two or zero children, and every element without children is on the lowest level of the tree. Inspired by the hierarchy of full binary trees, the organization mode of full binary trees is embedded into network partition in our work.

Once deployed in an area of interest, sensor nodes form a self-organized network. We partition the sensor network into non-uniform cells. Cells are basic partition units which occupy certain areas. Nodes in different locations belong to different cells. As shown in Fig. 3, the sensor network is arranged in a full binary tree structured pattern. The binary tree structured network is made up of cells that have either two or zero children cells. The cell located in the center of the tree structured network is the *root*. Cell 1 has exactly two children, namely, cell 2 and cell 3. Similarly, cell 4 and cell 5 are the children of cell 2, while cell 6 and cell 7 are the children of cell 3. Once the spacing d of two adjacent layers is determined, the area of interest can be occupied by different sizes of cells distributed in h layers. It can be seen that the number of cells increases exponentially with the number of layers. The aims to determine an event region composed of coarse-grained cells from inner layers, while profiling the boundary area using fine-grained cells in outer layers.

3.3 Mathematical Foundation of Full Binary Tree Structured Network

The profile of the full binary tree network structure is determined mathematically by exploiting the knowledge

Fig. 3. Full binary tree structured network partition.

behind Fick's law [24]. Fick's law is a classic theory in fluid mechanics. It works in a two-dimension static aquatic environment where a pollution source diffuses with a constant rate. Let $c(x, y, t)$ denote the diffusion concentration at location (x, y) with diffusion time t . The partial differential equation of the diffusion concentration derived by Fick's law is:

$$\frac{\partial c(x, y, t)}{\partial t} = \lambda \cdot \frac{\partial^2 c(x, y, t)}{\partial x^2} + \lambda \cdot \frac{\partial^2 c(x, y, t)}{\partial y^2} \quad (1)$$

where λ denotes diffusion rate.

Assuming that an initial amount of pollutant A is discharged at the location (x_0, y_0) at $t = 0$. By integrating Eq. 1, the diffusion concentration $c(x, y, t)$ at location (x, y) with diffusing time t ($t > 0$) can be derived:

$$c(x, y, t) = \frac{A}{4\pi\lambda t} \exp\left(-\frac{d^2(x, y)}{4\lambda t}\right), \quad (2)$$

where $d(x, y)$ denotes the distance between the sampling location (x, y) and the location of the pollution source (x_0, y_0) .

Eq. 1 has been widely used in a number of approaches [25]-[27] and validated by experiments in [28]. According to Eq. 2, the spatial distribution of the concentration at diffusion time t follows a Gaussian distribution. Furthermore, if the concentration threshold detected in gas boundary is c_{th} , the analytic expression of the time-varying gas boundary can be derived by solving Eq. 2:

$$(x - x_0)^2 + (y - y_0)^2 = \ln(4\pi\lambda t c_{th}/A)^{-4\lambda t} \quad (3)$$

Eq. 3 shows that, governed by Fick's law, the boundary of a diffusing object is an circle evolving over time. That is why the proposed full binary tree network structure adopts a set of concentric circles as its basic outline. It is worth noting that the diffusion of continuous objects does not always follow Fick's law in the real world, thus, we further divide the concentric circles into different sizes of cells for fine-grained boundary mapping.

3.4 Boundary Area Mapping under Full Binary Tree Structured Network

As shown in Fig. 3, each cell in the full binary tree structured network is assigned with a unique ID. The

area number is sorted anticlockwise. A node whose polar coordinate is (ρ_i, θ_i) determines the ID of the cell in which it locates as follows:

$$Code(i) = 2^{L_i-1} + \left\lfloor \frac{\theta_i}{2\pi/2^{L_i-1}} \right\rfloor \quad (4)$$

where L_i denotes the layer of the node in the full binary tree. L_i can be calculated as follows:

$$L_i = \left\lceil \frac{\rho_i}{d} \right\rceil \quad (5)$$

We term the cells which contain event nodes as event cells. The cells which contain no event nodes are termed as non-event cells. The cells which can be used to characterize the boundary of event region are termed as boundary mapping cells. We further subdivide boundary mapping cells into non-leaf boundary mapping cells and leaf boundary mapping cells. The definitions of the two types of boundary mapping cells are as follows:

Definition 1. *Non-leaf boundary mapping cell: A cell contains event nodes and at least one of its child cell is non-event cell.*

Definition 2. *Leaf boundary mapping cell: A leaf cell (the cell located in the outmost layer of the tree structured network) which contains event nodes.*

In Fig. 3, cell 9 represents an event cell, and its children cells are event cell 18 and non-event cell 19. Cell 9 can be identified as a non-leaf boundary mapping cell. Cell 16 is an event cell and both of its children cells are non-event cell, cell 16 can be also identified as a non-leaf boundary mapping cell. Cells 35, 36, 47 and 48 are leaf boundary mapping cells. All the mapping boundary cells are marked in green. It can be observed that only a few boundary mapping cells in outer layers are required to profile the border region.

3.5 Rationality Analysis of Full Binary Tree Structured Network

Compared with grid-based, Voronoi-based, Delaunay triangulation-based, and planarization-based network partition mechanisms, the proposed full binary tree structured network partition has the following superiority in boundary tracking.

First of all, the full binary tree structured network partition extracts the features which represent border shape as much as possible. According to Fick's law, the boundary of a diffusing object can be roughly approximated to an evolving circle over time, motivating the proposed network structure to adopt a set of concentric circles as its basic outline. Considering the anisotropy of the boundary, the concentric circles are further divided into different sizes of cells following the rules of full binary tree construction. With parameters control of the network structure, the body of the continuous object will be occupied by coarse-grained cells in inner layers of the full binary tree structured network while the boundary

area will be profiled using fine-grained boundary mapping cells in outer layers. Various combinations of different sizes of cells ensure the irregularity of the boundary can be captured. Based on the above-mentioned analysis, the proposed full binary tree structured network partition is able to provide part of boundary feature information before boundary tracking.

Secondly, the proposed full binary tree structured network partition is dynamically reconfigurable. With the diffusion of the continuous object, the network adaptively adjusts the resolution of partition cells. Thus the cost of maintaining the full binary tree structured network partition will not increase exponentially, as the network partition mechanisms using fixed cells.

Finally, in the next section, we will show how to visualize the peripheries of continuous objects in great detail through boundary node selection. In the phase of boundary node selection, full binary tree structured network partition is an effective method for reducing the search space of boundary nodes from the global network to boundary mapping cells. Further, under two types of cycle cascade architectures, which are the substructures extracted from the binary tree structured network, the nodes in boundary mapping cells are allowed to implement boundary node selection collaboratively.

4 BOUNDARY TRACKING USING SOFT MARGIN SUPPORT VECTOR MACHINES WITH BINARY TREE ARCHITECTURE

4.1 Binary Classification Problem Formulation in Boundary Tracking

Assuming node i locates at $\mathbf{p}_i = [x_i, y_i]^T$ in a two-dimensional Euclidean plane $\mathbf{R}^2$. The observation of node i is a binary variable z_i . z_i is set to positive one if node i detects the object, and negative one otherwise. Each node encapsulates its observation and coordinate information into a packet. If all the nodes upload the packets to a base station, the packets can be viewed as training samples from the perspective of machine learning. The training samples $\mathcal{S} = [\mathbf{p}_i, z_i]_{i=1}^N$ generated in IWSNs can be fed into a classification algorithm to output a decision boundary $f: \mathbf{R}^2 \rightarrow \{-1, +1\}$. It makes sense to draw a particular analogy between the decision boundary and the boundary of continuous objects. The decision boundary divides the feature space into positive and negative sample space while the boundary of continuous objects divides the monitoring area into event region and non-event region. Due to functional equivalence, boundary tracking in IWSNs can be formulated as decision boundary searching in a binary classification problem.

Popular binary classification algorithms include Artificial Neural Networks (ANN), SVM, Naive Bayes, K Nearest Neighbors (KNN), and decision trees. The choice of these algorithms always depends on the task at hand. Considering challenges of continuous object tracking in IWSNs while learning from comparison study of binary

classification algorithms in [29]-[30], we provide the following remarks to explain why we select original soft-margin SVM as a prototype to be reformed for boundary tracking in IWSNs.

Small sample size: For the issue of energy conservation in IWSNs, it is recommended to use as few nodes as possible for generating data samples. Learning machines are required to achieve high classification accuracy with small sample size. Reviewing current classification algorithms, ANN and decision trees pursue high accuracy at the cost of sample size whereas SVM and Naive Bayes is not restricted by the amount of sample. Besides sample size, KNN and Naive Bayes ask for the number of the training examples per class to be balanced.

Robust for imprecise samples: Imprecise training samples are inevitable for low-cost, easy-to-failure nodes working in harsh industrial field. Reviewing current classification algorithms, ANN and SVM are able to provide reasonable classification results using samples with incorrect labels. KNN is sensitive to irrelevant features because its similarity measures can be easily distorted by errors in attribute values. Contrary to KNN, decision trees are resistant to noise because their pruning strategies avoid overfitting. Naive Bayes is naturally robust to missing values which are simply ignored in computing probabilities and hence have no impact on the final decision.

Low time complexity: Boundary tracking of continuous objects, characterized by real time, requires learning machines with low time complexity. Naive Bayes trains quickly due to only a single pass on the data either to count frequencies or to compute the normal probability density function. KNN has a large time complexity on calculating the similarity between the unlabeled sample and all training samples. The calculation complexity of SVM is stemmed from solving a Lagrangian dual problem. Compared with SVM, computational time effort turned out to be much more intensive for ANN. Decision trees work faster than neural networks and SVMs.

Ease of use: The number of model parameters to be tuned by the user is an indicator of an algorithm's ease of use. KNN has only one single parameter which is relatively easy to tune. SVM involves the choice of kernel function and the determination of a regularization parameter. Decision trees require to select a criterion for feature selection and determine a number of parameters for pruning. ANN have the largest number of parameters to be determined, such as the number of layers, the number of units in each layer, maximum training time, learning rate, types of activation functions.

Interpretability: Neural network is a black box model. The knowledge hidden in ANN is hard to be explained. KNN calculates the similarity between the unlabeled samples and all training samples to implement classification. Different definitions of similarity make KNN lack of interpretation. Decision trees have good interpretability due to the interpretation and assessment of rules and the expression of knowledge. Naive Bayes and

SVM also have good interpretability because they are supported by well-defined mathematical theory. Naive Bayes is based on probability theory while its attribute independence assumption is often violated in the real world. SVM is based on statistical learning theory, which focuses on the machine learning of small sample size, and can trade off between the complexity of models and generalization performance. Last but not least, SVM provides the decision function of separation hyperplane, which is unavailable in ANN, KNN, decision trees, and Naive Bayes.

Based on the above-mentioned analysis, it can be concluded that SVM addresses the problem of small sample study of particular advantage, has strict mathematical theory of support and less parameters to be tuned. All these features are desirable for boundary tracking in sensor networks. However, the most fundamental reason for weeding out NN, KNN, decision trees, and Naive Bayes is that all these classifiers only judge discrete points within or outside the region covered by a continuous object. For the points lying on the boundary of the continuous object, they cannot provide a unified, definitive and formal mathematical model.

On the contrary, the decision function provided by SVM can be equivalent to the functional form of the boundary of the continuous object. Moreover, the sparseness representation of the decision boundary indicates that we only need to identify a small number of nodes, whose training samples are support vectors, as boundary nodes. To strengthen the fault-tolerance of SVM, we finally select Soft-margin SVM as the most promising original algorithm. All that remains is how to run Soft-margin SVM in IWSNs in a rapid manner.

4.2 Preliminaries of Soft-margin SVM

Soft-margin SVM is a supervised learning model that attempts to find an optimal hyperplane, denoted by $H(\mathbf{p})$, that maximizes the margin between two classes. When the training data is linearly separable, these two classes can be separated by two margins parallel to the optimal hyperplane. The optimal hyperplane taking the form $H(\mathbf{p}) = \mathbf{w} \cdot \mathbf{p} + b$ can be found by solving the quadratic problem as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i (\mathbf{w} \cdot \mathbf{p}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N \\ & \xi_i \geq 0, \quad i = 1, 2, \dots, N \end{aligned} \quad (6)$$

where $\mathbf{w}$ denotes the normal vector to the hyperplane, b denotes the offset vector that determines offset of the hyperplane from the origin. Slack variable ξ_i measures the amount of violation of the constraints. Penalty parameter C is used to evaluate the cost of constraint violation. By solving the Lagrangian dual of the above problem, one

obtains the simplified problem:

$$\begin{aligned} \min \quad & W(\alpha) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{p}_i \cdot \mathbf{p}_j) - \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N \end{aligned} \quad (7)$$

where α_i denotes Lagrange multipliers. The training samples corresponding to non-zero Lagrange multipliers are called support vectors. By solving Eq. 7, $\mathbf{w}$ can be written as a linear combination of the support vectors, and the hyperplane $H(\mathbf{p})$ can be derived as follows:

$$H(\mathbf{p}) = \sum_{i=1}^N \alpha_i y_i (\mathbf{p} \cdot \mathbf{p}_i) + b \quad (8)$$

Nevertheless, real-world problems are usually nonlinear in nature. The solution derived in the case of linear classification is inapplicable to these problems. In a general formulation, SVM works by mapping the training set that are non-linearly separable in the input space into a high or even infinite dimensional feature space. This higher-dimensional space is called the transformed feature space. In a transformed feature space of sufficient dimensionality, any non-linearly separable training set in the input space can be made linearly separable. The optimal hyperplane determined in the feature space thus corresponds to a nonlinear decision boundary in the input space.

Given a nonlinear mapping $\mathbf{p} \rightarrow \phi(\mathbf{p})$, the classifiers is transformed into the form:

$$H(\mathbf{p}) = \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{p}) \cdot \phi(\mathbf{p}_i) + b \quad (9)$$

This formulation can be extended to general nonlinear functions by using the concepts of kernels. Kernels are symmetric positive definite functions that allow inner products to be calculated directly in feature space without defining mapping operator $\phi(\cdot)$. By kernel trick, we have $K(\mathbf{p}_i, \mathbf{p}_j) = \phi(\mathbf{p}_i) \cdot \phi(\mathbf{p}_j)$, Eq. 9 can be transformed into the form:

$$H(\mathbf{p}) = \sum_{i=1}^N \alpha_i y_i K(\mathbf{p}, \mathbf{p}_i) + b \quad (10)$$

The selection of an appropriate kernel function is important, since the kernel function defines the transformed feature space in which the training set instances will be classified. In our work, Radial basis function (RBF) kernel function is selected for SVM training. Reasons lie in that RBF is a universal kernel functions which can be applied to any of the distribution of the samples through the choice of parameters. Many researches demonstrated that RBF kernel is one of the most popular functions with convincing performance and is a reasonable first choice [31]-[34].

The expression of RBF kernel function is:

$$K(\mathbf{p}_i, \mathbf{p}_j) = \exp(-\gamma \|\mathbf{p}_i - \mathbf{p}_j\|^2) \quad (11)$$

For soft-margin SVM with RBF kernel, there are two parameters, C and γ , remain to be determined. The determination of C and γ , known as hyper-parameter tuning, is usually done by performing a grid search over pairs of C and γ with cross-validation [35]-[37].

We use sigmoid function to assess the probability that a new sample $\mathbf{p}$ belongs to either positive or negative class:

$$\Pr = \frac{1}{1 + e^{-H(\mathbf{p})}} \quad (12)$$

$\mathbf{p}$ belongs to positive class if $\Pr > 0.5$, while negative class if $\Pr < 0.5$. Setting the value of $\Pr$ to be 0.5, we have an equation with one unknown $\mathbf{p}$. The solution of this equation is the set of points on the decision boundary.

4.3 Hierarchical SVM Training Strategy under Binary Tree Structured IWSNs

4.3.1 Overview

Ongoing research in the machine learning community seeks to design statistically sound learning algorithms that scale to large data sets. Massive training samples generated by nodes in IWSNs lay the basis for machine learning algorithms to solve the classification problem of large-scale data. Notice that in traditional centralized SVM, all the training samples should be sent to a base station for decision boundary learning. However, broadcasting communication in an IWSN is highly energy inefficient and easy to cause collisions. Moreover, the time complexity of standard SVM which solves quadratic program is $O(N^3)$, where N is the number of samples fed into SVM [38]. If standard SVM is applied to IWSNs directly, real-time responses will be unavailable to industrial applications. Thus, it is preferable for IWSNs to perform SVM training in a distributed fashion.

Through fully exploiting the collective intelligence of nodes in sensor network, we design a hierarchical SVM training strategy under the proposed binary tree structured IWSNs. By means of domain decomposition, a large-scale SVM training problem can be turned into a number of smaller discrete sub-SVM problems. The sub-SVM problems will be addressed locally in each boundary mapping cell. Our training strategy parallelizes the learning process and diffuses local support vectors generated in each boundary mapping cell over the entire network until convergence.

4.3.2 Search Space Reduction

Eq. 9 which involves a total of N training samples is a normal form of the separation hyperplane. This equation indicates that Standard SVM regards the complete training set as search space to output support vectors. However, from the mathematical point of view, the non-support vectors whose Lagrange multipliers are zero are superfluous items. A compact expression of the separation hyperplane can be obtained by removing non-support vectors in Eq. 9. Inspired by the compact expression, the original search space can be reduced by

safely removing those training samples which are non-support vectors.

Considering that non-support vectors do not contain any knowledge for classifier training while accounting for a large portion in the training set, we are allowed to build a representative subset by eliminating non-support vectors out of the original training set as much as possible. The representative subset is small in volume but covers all the knowledge contained in the complete set. Then, a reduced SVM can use this subset to formulate a less-scale quadratic programming problem. The computation for solving this new quadratic programming problem is far less than the standard SVM.

Remember that all the training samples are provided by sensor nodes, search space reduction can be carried out by limiting the number of sensor nodes allowed to provide training samples. After narrowing the possible distribution area of the hyperplane to the collection of the boundary mapping cells, the sensor nodes outside the boundary mapping cells do not need to generate training samples because they are far away from the hyperplane. The search space of support vectors can be reduced to the set of training samples generated inside the boundary mapping cells. Assume the number of nodes in the boundary mapping cells is M . Only the M nodes are required to generate training samples for a reduced SVM learning.

4.3.3 Hierarchical SVM Training under Binary Tree Structure

After search space reduction, nodes in different boundary mapping cells will be organized into different numbers of equal-sized learning clusters. Assuming an individual node is able to train support vectors from a training set of K samples ($K \ll M \ll N$), the size of each learning cluster is less than K . As shown in Fig. 3, cells 35, 36, 47, and 48 are leaf boundary mapping cells. If the number of learning clusters in these cells are quantified as one unit, the number of learning clusters in parent cells 17, 18, 23, 24 are two units. Similarly, the number of learning clusters in cells 16, 20-22, 29-31 are two units while in cells 9, 12-14 are four units.

Learning clusters in the boundary mapping cells perform distributed learning under two types of cycle cascade architectures, which are the substructures extracted from the binary tree structured network. Fig. 4 illustrates the vertical cycle cascade architecture where the boundary mapping cells are in a parent-child relationship from top to bottom. For example, in Fig. 3, there is a parent-child relationship between cells 9 and 18, and cells 18 and 36. The learning clusters in the three cells will perform distributed SVM training under a three-layer vertical cycle cascade architecture.

As shown in Fig. 4, LC_{i-1} , $LC_{i-2}, \dots, LC_{i-4}$ are learning clusters formed in boundary cells located in the i -th layer of the full binary tree. Each learning cluster outputs local support vectors and then uploads the support vectors to the learning clusters formed in

Fig. 4. Vertical cycle cascade architecture.

its descendant boundary cell in $i+1$ -th layer. LC_{i+1-1} and LC_{i+1-2} will receive the support vectors from their ancestors and add the support vectors in their local support vectors sets to output corresponding support vectors. Afterwards, LC_{i+1-1} and LC_{i+1-2} upload their support vectors to the learning clusters formed in its descendant boundary cell in $i+2$ -th layer. LC_{i+2-1} will receive the support vectors and add the support vectors in its local support vectors set to output corresponding support vectors. We introduce a feedback loop that enters the resulting support vectors in the last layer into the first layer. The training procedure will keep cycling under the vertical cascade architecture until the convergence of the final support vectors.

Fig. 5. Horizontal cycle cascade architecture.

Fig. 5 illustrates the horizontal cycle cascade architecture where the boundary mapping cells are in a sibling relationship. For example, in Fig. 3, there is a sibling relationship between cells 20 and 21, and cells 21 and 22. As shown in Fig. 5, at the beginning, LC_{i-1} outputs local support vectors and then transmits the support vectors to LC_{i-2} formed in LC_{i-1} 's sibling cell. LC_{i-2} receives the support vectors from LC_{i-1} and adds the support vectors in its local support vectors set to output corresponding support vectors. Afterwards, LC_{i-2} transmits its support vectors to LC_{i-3} formed in LC_{i-2} 's sibling cell. We also introduce a feedback loop that enters the resulting support vectors in the last layer into the first layer. The training procedure will keep cycling under the horizontal cascade architecture until the convergence of the final support vectors.

A layered cascade architecture is guaranteed to converge to the global optimum if we keep the best set of support vectors produced in one layer, and use it in at

least one of the subsets in the next layer [39]. This is the case shown in Fig. 4 and Fig. 5.

Finally, the function of the decision boundary is as follows:

$$\tilde{H}(\mathbf{p}) = \sum_{i=1}^{N_{sv}} \alpha_i^* y_i K(\mathbf{p}, \mathbf{p}_{sv_i}) + b^* \quad (13)$$

where $\mathbf{p}_{sv_i}$ denotes support vectors generated under the binary tree architecture, α_i^* denotes the non-zero Lagrange multiplier of $\mathbf{p}_{sv_i}$. N_{sv} denotes the number of support vectors. The nodes whose training samples are support vectors can be regarded as boundary nodes. After the base station receives support vectors and corresponding Lagrange multipliers from boundary nodes, the analytical solution of the boundary line can be derived by substituting $\tilde{H}(\mathbf{p})$ for $H(\mathbf{p})$ in Eq. 12.

4.4 Complexity Analysis

The basic idea behind the proposed BTS-COT is to use the coordinates of sensor nodes as input, sensing labels as output to find an optimal hyperplane. BTS-COT employs Soft-margin SVM to train the hyperplane due to its generalization performance in small sample size, solid mathematical foundation, ease of use, and fault-tolerance. However, traditional soft-margin SVM requires large memory requirement and computation time when dealing with large data sets. Thus, BTS-COT trains soft-margin SVM in a hierarchical framework so that training time can be dramatically reduced meanwhile the burden of training can be tolerable for sensor nodes with limited memory and computing capacity. The complexity of BTS-COT within the context of hyperplane training is discussed as follows.

Review that N sensor nodes are deployed in the IWSN, traditional soft-margin SVM tends to scale with the cube of the number of training vectors $O(N^3)$. After boundary area mapping, a certain number of boundary mapping cells will be selected to profile the border region. From the perspective of SVM training, boundary area mapping eliminates non-support vectors early from the optimization. Review that the total number of sensor nodes in the boundary cells is symbolized by M , the complexity of traditional soft-margin SVM can be reduced from $O(N^3)$ to $O(M^3)$.

For further savings in computation and speeding up training SVM, nodes in the boundary mapping cells are grouped into learning clusters. The learning clusters split soft-margin SVM with $O(M^3)$ complexity into a number of smaller quadratic programming subproblems and learn in parallel these subproblems under either horizontal or vertical cascade architecture. Review that the maximum number of samples can be trained by a node is K , the number of sensor nodes in each learning cluster is symbolized by P ($P < K$). In initial, multiple soft-margin SVM with $O(P^3)$ complexity are trained in parallel in each learning cluster. Under either horizontal or vertical cascade architecture, a learning cluster will

transmit its local support vectors to its descendant or sibling learning cluster. The learning cluster which receives the support vectors will train a soft-margin SVM with $O((P - Q + R)^3)$ complexity. Q is the number of local non-support vectors separated from P . R is the number of received support vectors. The set of the support vectors are optimized iteratively, until the global optimum is reached.

5 SIMULATION EXPERIMENTS AND PERFORMANCE ANALYSIS

5.1 Simulation Setting

Simulations were performed in a square area using Matlab. For a network with fixed area, we use node density to control the number of nodes deployed in it. Node density is the ratio of the number of nodes to the network area, thus the number of nodes is the product of node density and network area. We set two experimental environments, where the gas diffused evenly and unevenly, respectively. The source of leak was initiated at the center of the sensor field and leakage gas diffused at a rate of 1 m/s in average. Boundary tracking was performed every 10s. When carrying out hierarchical SVM Training in each learning cluster, a grid search over pairs of C and γ is performed for hyper-parameter tuning. Both C and γ range from 2^{-8} to 2^8 with a step of 1. The remaining simulation parameters and their default values are listed in Table 1.

TABLE 1
Simulation parameters

Parameter	Value
Network Size	133 * 133 m ²
Node density	0.2
Communication Radius	10m
Number of binary tree's layer	7
Tracking interval	10s
Mean diffusion rate	1 m/s
Degree of irregular	0 - 0.2
Ratio of faulty nodes	0 - 0.15
Penalty parameter C	2^{-8} to 2^8
Width of RBF kernel γ	2^{-8} to 2^8

BTS-COT was compared with VFPOD, TGM-COT and BRTCO, which are all proposed for continuous objects tracking in sensor networks. Performance indicators include the number of boundary nodes, related contour accuracy, and overlap ratio. The number of boundary nodes reflects the energy-efficiency while related contour accuracy and overlap ratio evaluate the tracking accuracy. The mathematical definition of related contour accuracy is:

$$RCA = 1 - \frac{|A_{est} - A_{act}|}{A_{act}} \quad (14)$$

where A_{act} denotes the actual area of the continuous object, A_{est} denotes the area of the zone bounded by tracking boundary.

The mathematical definition of overlap ratio is:

$$OR = \frac{A_{est} \cap A_{act}}{A_{act}} \quad (15)$$

5.2 Experiment results in Scene of Uniform Diffusion

Fig. 6. Snapshots of boundary tracking using VFPOD, TGM-COT, BRTCO, and BTS-COT.

We first simulated an ideal environment where the gas diffused evenly and there was no risk of node failure. We established Cartesian coordinates on the deployed sensor field, the source of the gas was set in ordinate origin initially, and then kept expanding. By gradient transportation, the boundary of the gas exhibited circle shape with a growing radius over time. Fig. 6 shows the snapshots of BTS-COT, VFPOD, TGM-COT and BRTCO in the first quadrant when the diffusion radius of the gas was 30m. In the first quadrant, the boundary of the gas is shown as a quarter of circle arc in red. Black dots are non-event nodes while red dots are event nodes. It can be observed that VFPOD works in a network partitioned by Voronoi diagram, TGM-COT works in a network partitioned by uniform grids, and BRTCO works in a network partitioned by Delaunay triangles. For BTS-COT, the full binary tree network partition only works on the sensor field covered by continuous objects rather than the entire network area. The boundary nodes selected by each algorithm are marked by squares.

Fig. 7 shows the number of boundary nodes selected by each algorithm in the process of diffusion. The diffusion radius of the gas was the product of tracking interval and mean diffusion rate. It can be observed that, with the growth of gas boundary, more boundary nodes are involved in boundary tracking. BTS-COT always introduces the least number of boundary nodes, because support vectors typically account for a little portion of

the total samples. Through searching support vectors in a high dimensional feature space, the impact of expanding boundary line on the number of boundary nodes is weakened.

Fig. 7. The number of boundary nodes versus diffusion time.

In Fig. 8, we used curve fitting of selected boundary nodes to estimate the boundary line. The similarities between the estimated boundary and the actual boundary is measured by RCA and OR. Fig. 8(a) shows the RCA obtained by each algorithm. BRTCO and TGM-COT select non-event nodes as boundary nodes. The area enclosed by the resulting fitting curve is larger than that enclosed by the actual boundary. VFPOD selects event nodes as boundary nodes so that the fitting curve is enclosed by the actual boundary. BTS-COT fitted curves probabilistically using support vectors. Support vectors consist of both event and non-event nodes. The fitted curves is in between the upper bound and lower bound determined by non-event nodes and event nodes, respectively. Therefore, the RCA of BTS-COT outperforms that of the other three algorithms.

Fig. 8. Tracking accuracy versus diffusion time.

Fig. 8(b) shows the OR obtained by each algorithm. It can be observed the OR obtained by BRTCO and TGM-COT is always 1. It is because the boundary estimated by BRTCO and TGM-COT is delimited by non-event nodes so that actual boundary is within the estimated boundary. The boundary estimated by VFPOD is bound-

ed by event nodes. Thus, the estimated boundary is within the actual boundary. BTS-COT also shows high OR compared with BRTCO and TGM-COT. However, BTS-COT does not exaggerate the area of the continuous object as BRTCO and TGM-COT do.

Fig. 9 depicts the relationship between the number of boundary nodes and the node density. In this experiment, the radius of the leakage gas was set to 30m. It can be observed that, with the increasing of the node density, the number of boundary nodes selected by each algorithm increases in varying degrees. BTS-COT is the least sensitive to the node density among the algorithms. Reasons lie in that the number of boundary nodes selected by BTS-COT is equal to the number of support vectors. With the increase of node density, if all the added nodes are far away from the hyperplane, the Lagrange multipliers of the added nodes are zero, no new support vectors will be produced.

Fig. 9. The number of boundary nodes versus node density.

5.3 Experiment Results in Scene of Non-uniform Diffusion

In a real environment, the leakage gas is susceptible to outside influences and always follows non-uniform diffusion. We use degree of irregular (DOI) to evaluate the irregularity of the boundary. In this set of experiment, we set the node density to be 0.2. The radius of the leakage gas in different directions follows a Gaussian distribution $N(R, R * DOI)$. R is the product of tracking interval and mean diffusion rate. As the gas spreads, its boundary line suffers serious external interference and becomes more irregular.

Fig. 10 depicts the relationship between DOI and the number of boundary nodes at 30s. It can be observed that with the increasing of DOI, the number of boundary nodes selected by BTS-COT increases linearly. The reason is that the irregular boundary indicates a hyperplane which requires more support vectors to characterize. For VFPOD, TGM-COT and BRTCO, more localization

Fig. 10. The number of boundary nodes versus DOI.

information of boundary nodes is introduced to profile a boundary of high DOI. Thus, a great number of boundary nodes will be involved for boundary tracking.

Fig.11 shows the impacts of DOI on tracking accuracy. It can be observed that with the DOI varying, the tracking accuracy of each algorithm is stable. The RCA achieved by BTS-COT is about 0.9 while the OR achieved by BTS-COT is about 1. The satisfying performance of BTS-COT is guaranteed by the increasing number of support vectors.

Fig. 11. Tracking accuracy versus DOI.

5.4 Convergence Speed of the BTS-COT

In this section, numerical experiments are carried out to show the convergence speed of BTS-COT. Specifically, the convergence speed of BTS-COT depends on the number of feedback iterations required by soft-margin SVM training, under the horizontal and the vertical cascade architecture. In this set of experiments, each learning cluster starts with 500 training samples for its local SVM training.

For the case of uniform diffusion, most of the learning clusters train their local SVM under horizontal cascade architecture. We vary the number of learning clusters from three to six to build different cluster sizes of horizontal cascade architecture. Table 2 shows that, SVM training under horizontal cascade architecture always converges with one feedback iteration. Compared with

standard SVM training, horizontal cascade architecture obtains more than 50% reduction in training time, and with the increase of training samples, such reduction becomes increasingly clear.

TABLE 2
Convergence speed of SVM training under horizontal and vertical cascade architecture

Size of horizontal cascade architecture	Feedback iterations	Training time	Speed up
3 LCs	1	3.24 ms	58.96%
4 LCs	1	4.42 ms	70.24%
5 LCs	1	6.01 ms	82.13%
6 LCs	1	11.22 ms	95.15%
Size of vertical cascade architecture	Feedback iterations	Training time	Speed up
2-layer	1	1.39 ms	82.50%
3-layer	1	2.14 ms	92.46%

For the case of non-uniform diffusion, learning clusters train their local soft-margin SVM under horizontal and vertical cascade architecture jointly. Because BTS-COT profiles the boundary area of the continuous object using boundary mapping cells in outer layers, the number of layers of the vertical cascade architecture is always in a low level. Experiment results show that the number of layers of the vertical cascade architecture is usually two or three. Similar to horizontal cascade architecture, SVM training under vertical cascade architecture always converges with one feedback iteration. Compared with standard SVM training, the reduction in training time obtained by vertical cascade architecture is remarkable, even under the two-layer vertical cascade architecture.

The numerical experiments show that BTS-COT has great advantage in convergence. This is likely to be caused by the fact that the BTS-COT decomposes the traditional SVM training problem into a number of independent smaller optimization tasks, and the partial results are combined in later stages in a hierarchical way. All the subproblems are processed in parallel at each step, and non-support-vector data are filtered out sequentially.

5.5 Experiment Results in Scene of Node Failure

Besides the influence of the external environment on the network, a high ratio of faulty nodes in the industrial field can also affect the tracking accuracy. The ratio of faulty nodes is defined as the ratio of the number of faulty nodes to the total number of nodes. A faulty node might produce a false positive, which is a high reading indicating an event occurred when it did not, or a false negative, which is a low reading indicating the absence of event when one did occur. Lacking of fault-tolerant mechanisms, TGM-COT, VFPOD, and BRTCO cannot handle a failure of nodes but blindly trust the sensor readings. Consequently, some false boundary nodes will be selected to fabricate a non-existent boundary. Soft margin SVM minimizes the structural risk minimization and provides good performances in terms of global op-

Fig. 12. Snapshots of fault-tolerant boundary tracking using BTS-COT.

timization and good generalization abilities. Thus, BTS-COT has inherent ability to eliminate the influence of abnormal sensor readings.

In Fig. 12, we validated the fault-tolerant capability of BTS-COT. We set the average radius of leakage gas to be 30 m, DOI to be 0.05, and node density to be 0.2. As shown in Fig. 12, red solid line denotes the actual boundary while blue solid line denotes the tracking boundary. Faulty nodes are marked by green squares. It can be observed that, when the ratio of faulty node is no more than 0.09, BTS-COT is able to eliminate the influence of false sensor readings. The result of boundary tracking agreed well with the actual boundary. However, when the ratio of faulty node is 0.12, the BTS-COT is not robust enough to defense the faults. As shown in Fig. 12(c), the hyperplane learned by BTS-COT is determined by both real boundary nodes and faulty nodes. When the ratio of faulty node is up to 0.15, the real hyperplane has been distorted by high percentage of faulty nodes. It can be demonstrated that BTS-COT shows fault tolerance in IWSNs where a small number of nodes is allowed to report false sensor readings.

6 CONCLUSIONS AND FUTURE WORK

In this article, a boundary tracking algorithm of hazardous continuous objects was proposed for IWSNs. Through organizing the network nodes into a full binary tree structured network, the border region of the continuous objects can be profiled by a small number of boundary mapping cells. To study the geometry characteristic of boundary line in great detail, we transformed the

boundary tracking problem into a binary classification problem, and designed a hierarchical soft margin SVM training strategy – under the binary tree structured IWSNs. The boundary node selection was realized by searching support vectors in a high dimensional feature space. Simulation results demonstrated that the proposed algorithm achieves high tracking accuracy with a small number of boundary nodes. Even without specially designed fault-tolerant mechanisms, the algorithm displays inherent ability to eliminate the influence of spurious sensor readings caused by faulty nodes.

The proposed boundary tracking algorithm is designed for continuous objects with planar motion. Future research will be directed towards studying a variant configuration of the full binary tree structure, which can be applied for boundary mapping in 3D environments.

ACKNOWLEDGMENTS

The work is supported by the National Key Research and Development Program, No.2017YFE0125300, the Jiangsu Key Research and Development Program, No.BE2019648, the Fundamental Research Funds for the Central Universities, B200201035, the State Key Laboratory of Acoustics, Chinese Academy of Sciences, SKLA202004.

REFERENCES

- [1] M. Martínez-García, Y. Zhang, K. Suzuki and Y. Zhang, "Deep Recurrent Entropy Adaptive Model for System Reliability Monitoring," *IEEE Trans Ind. Informat.*, doi: 10.1109/TII.2020.3007152.
- [2] L. Shu, M. Mukherjee, and X. Wu, "Toxic gas boundary area detection in large-scale petrochemical plants with industrial wireless sensor networks," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 22-28, 2016.
- [3] P. P. Ray, M. Mukherjee, and L. Shu, "Internet of things for disaster management: State-of-the-art and prospects," *IEEE Access*, vol. 5, pp. 18818-18835, 2017.
- [4] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724-4734, 2018.
- [5] X. Li, D. Li, J. Wan, A. Vasilakos, C. Lai, and S. Wang, "A review of industrial wireless networks in the context of industry 4.0," *Wireless Netw.*, vol. 23, no. 1, pp. 23-41, 2018.
- [6] G. Han, J. Wu, H. Wang, M. Guizani, J. A. Ansere, and W. Zhang, "A Multi-Charger Cooperative Energy Provision Algorithm Based on Density Clustering in the Industrial Internet of Things," *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 9165-9174, 2019.
- [7] B. Rashid, and M. H. Rehmani, "Applications of wireless sensor networks for urban areas: A survey," *J. Netw. Comput. Applicat.*, vol. 60, pp. 192-219, 2016.
- [8] M. Y. Aalsalem, W. Z. Khan, W. Gharibi, M. Khan, and Q. Arshad, "Wireless Sensor Networks in oil and gas industry: Recent advances, taxonomy, requirements, and open challenges," *J. Netw. Comput. Applicat.*, vol. 113, pp. 87-97, 2018.
- [9] D. V. Manatakis, and E. S. Manolakis, "Estimating the spatiotemporal evolution characteristics of diffusive hazards using wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 9, pp. 2444-2458, 2015.
- [10] C. Zhang, Y. Zhang, and Y. Fang, "Localized coverage boundary detection for wireless sensor networks," *Wireless Netw.*, vol.15, no.1, pp.3-20, 2009.
- [11] C. Zhang, Y. Zhang, and Y. Fang, "A coverage inference protocol for wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol.9, no.6, pp.850-864, 2010.
- [12] L. Shu, M. Mukherjee, X. Xu, K. Wang, and X. Wu, "A survey on gas leakage source detection and boundary tracking with wireless sensor networks," *IEEE Access*, vol. 4, pp. 1700-1715, 2016.
- [13] C. Zhang, Y. Zhang, and Y. Fang, "Detecting coverage boundary nodes in wireless sensor networks," *IEEE Int. Conf. Netw., Sensing and Control*, pp. 23-25, 2006.
- [14] H. J. Lee, M. T. Soe, S. H. Chauhdary, S. Rhee, and M. Park, "A data aggregation scheme for boundary detection and tracking of continuous objects in WSN," *Intelligent Automation & Soft Computing*, vol. 23, no. 1, pp. 135-147, 2017.
- [15] E. Ould-Ahmed-Vall, B. H. Ferri, and G. F. Riley, "Distributed fault-tolerance for event detection using heterogeneous wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 12, pp. 1994-2007, 2012.
- [16] L. Liu, G. Han, Y. He, and J. Jiang, "Fault-Tolerant Event Region Detection on Trajectory Pattern Extraction for Industrial Wireless Sensor Networks," *IEEE Trans Ind. Informat.*, vol. 16, no. 3, pp. 2072-2080, 2020.
- [17] G. Han, J. Shen, L. Liu, A. Qian, and L. Shu, "TGM-COT: energy-efficient continuous object tracking scheme with two-layer grid model in wireless sensor networks," *Personal and Ubiquitous Computing*, vol. 20, no. 3, pp. 349-359, 2016.
- [18] S. Oh, J. Lee, and S. Park, "Energy efficient and accurate monitoring of large-scale diffusive objects in internet of things," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 612-615, 2016.
- [19] M. E. Kim, Y. Son, C. Kim, Y. Yim, and S. H. Kim, "Origin-Mediated Sink Mobility Support for Large-Scale Phenomena Monitoring in IWSNs," *IEEE 32nd Int. Conf. AINA*, 2018, pp. 109-114.
- [20] S. Imran, and Y. Ko, "A continuous object boundary detection and tracking scheme for failure-prone sensor networks," *Sensors*, vol. 17, no. 2, pp. 1-17, 2017.
- [21] G. Han, J. Shen, L. Liu, and L. Shu, "BRTCO: A novel boundary recognition and tracking algorithm for continuous objects in wireless sensor networks," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2056-2065, 2016.
- [22] H. Ping, Z. Zhou, Z. Shi, and T. Rahman, "Accurate and energy-efficient boundary detection of continuous objects in duty-cycled wireless sensor networks," *Personal and Ubiquitous Computing*, vol. 22, no. 3, pp. 597-613, 2018.
- [23] T. Rahman, Z. Zhou, and H. Ning, "Energy efficient and accurate tracking and detection of continuous objects in wireless sensor networks," *IEEE Int. Conf. SmartIoT*, 2018, pp. 210-215.
- [24] B. P. Van Milligen, P. D. Bons, B. A. Carreras, and R. Sanchez, "On the applicability of Fick's law to diffusion in inhomogeneous systems," *European journal of physics*, vol. 26, no. 5, pp. 913-925, 2005.
- [25] C. Xiang, P. Yang, X. Wu, H. He, B. Wang, and Y. Liu, "istep: A step-aware sampling approach for diffusion profiling in mobile sensor networks," *IEEE Trans. Veh. Tech.*, vol. 65, no. 10, pp. 8616-8628, 2015.
- [26] T. Zhao, A. Nehorai, "Detecting and estimating biochemical dispersion of a moving source in a semi infinite medium," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2213-2225, 2006.
- [27] S. Vijayakumaran, Y. Levinbook, and T. F. Wong, "Maximum likelihood localization of a diffusive point source using binary observations," *IEEE Trans. Signal Process.*, vol. 55, no. 2, pp. 665-676, 2007.
- [28] Y. Wang, R. Tan, G. Xing, J. Wang, and X. Tan, "Profiling aquatic diffusion process using robotic sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 880-893, 2014.
- [29] B. Sekeroglu, S. S. Hasan, and S. M. Abdullah, "Comparison of Machine Learning Algorithms for Classification Problems," *Science and Information Conf.*, Springer, Cham, pp. 491-499, 2019.
- [30] R. Sathya, A. Abraham, "Comparison of supervised and unsupervised learning algorithms for pattern classification," *Int. J. Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34-38, 2013.
- [31] S. Han, C. Qubo, and H. Meng, "Parameter selection in SVM with RBF kernel function," *World Automation Congress 2012 IEEE*, pp. 1-4, 2012.
- [32] H. Guo, J. Yin, J. Zhao, Z. Huang, and Y. Pan, "Prediction of fatigue life of packaging EMC material based on RBF-SVM," *Int. J. Materials and Product Technology*, vol. 49, no. 1, pp. 5-17, 2014.
- [33] B. C. Kuo, H. H. Ho, C. H. Li, C. C. Hung, and J. S. Taur, "A kernel-based feature selection method for SVM with RBF kernel for hyperspectral image classification," *IEEE J. Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 1, pp. 317-326, 2013.

- [34] H. L. Jian, K. T. Tang, "Improving classification accuracy of SSVEP based BCI using RBF SVM with signal quality evaluation," *2014 IEEE ISAPCS*, pp. 302-306, 2014.
- [35] P. Lameski, E. Zdravevski, R. Mingov, and A. Kulakov, "SVM parameter tuning with grid search and its impact on reduction of model over-fitting," *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Springer, Cham, pp. 464-474, 2015.
- [36] Duarte E, Wainer J. "Empirical comparison of cross-validation and internal metrics for tuning SVM hyperparameters," *Pattern Recognition Letters*, vol. 88, pp. 6-11, 2017.
- [37] T. Wang, X. Ye, L. Wang, and H. Li, "Grid search optimized SVM method for dish-like underwater robot attitude prediction," *2012 Fifth Int. Joint Conf. Computational Sciences and Optimization IEEE*, pp. 839-843, 2012.
- [38] I. W. Tsang, J. T. Kwok, P. M. Cheung, "Core vector machines: Fast SVM training on very large data sets," *Journal of Machine Learning Research*, vol. 6, pp. 363-392, 2005.
- [39] H. P. Graf, E. Cosatto, L. Bottou, I. Durdanovic, and V. Vapnik, "Parallel support vector machines: The cascade svm," *Advances in Neural Inform. Process. Syst.*, pp. 521-528, 2005.

Li Liu received the Ph.D. degree in Internet of Things Technology from Hohai University, Nanjing, China, in 2019. He was a Senior Research Assistant in Department of Electrical Engineering in City University of Hong Kong from October 2017 to February 2018. He is currently a Lecturer with the College of Internet of Things Engineering, Hohai University, Changzhou, China. He has published over 20 papers in related international conferences and journals, including IEEE TII, IEEE IOTJ, IEEE Communications

Magazine, etc. His research interests include wireless sensor networks, industrial internet of things, and machine learning.

Guangjie Han [S'03, M'05, SM'18] received the Ph.D. degree from Northeastern University, Shenyang, China, in 2004. He is currently a Professor with the Department of Information and Communication System, Hohai University, Changzhou, China and a Distinguished Professor of Dalian University of Technology, Dalian, China. He is the author of over 330 papers published in related international conference proceedings and journals, including the IEEE JSAC, IEEE COMST, IEEE TMC, IEEE TIE, IEEE TII,

IEEE TCC, IEEE TPDS, IEEE TVT, IEEE IoT Journal, IEEE Systems, IEEE Wireless Communications, IEEE Communications, IEEE Network, etc, and is the holder of 130 patents. His current research interests include Internet of Things, Industrial Internet, Machine Learning and Artificial Intelligence, Mobile Computing, Security and Privacy.

Dr. Han has served as a Co-chair for more than 50 international conferences/workshops and as a Technical Program Committee member of more than 150 conferences. He has served on the Editorial Boards of up to 16 international journals, including the IEEE JSAC, IEEE Network, IEEE Systems, IEEE ACCESS, IEEE/CAA JAS, Telecommunication Systems, etc. He has guest edited a number of special issues in IEEE Journals and Magazines, including the IEEE Communications, IEEE Wireless Communications, IEEE Transactions on Industrial Informatics, Computer Networks, etc. He has served as a Reviewer of more than 60 journals. He had been awarded the ComManTel 2014, ComComAP 2014, Chinacom 2014 and Qshine 2016 Best Paper Awards. He is a Senior Member of IEEE.

Zhengwei Xu received his B.S. degree in Computer Science and Technology from Henan Normal University, China, in 2019. He is currently pursuing his Ph.D. degree from the College of Computer Science and Information at Hohai University. His current research interest include outlier detection in Industrial Internet of Things.

Jinfang Jiang received the Ph.D. degree from the Department of Computer Science, Hohai University, Nanjing, China, in 2015. She is currently an associate professor with the Department of Information and Communication System, Hohai University, Nanjing, China. She has published over 40 papers in related international conferences and journals, including IEEE TPDS, IEEE TMC, IEEE Communications Magazine, etc. Her research interests include wireless sensor networks, mobile cloud computing, and information

security technologies.

Lei Shu [M'07, SM'15] received the Ph.D. degree from the Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland, in 2010. He is currently a Distinguished Professor with Nanjing Agricultural University, China, and a Lincoln Professor with the University of Lincoln, U.K. He is also the Director of the NAU-Lincoln Joint Research Center of Intelligent Engineering. His main research fields are wireless sensor networks and Internet of Things. He has published over 400 papers in

related conferences, journals, and books in the areas of sensor networks and Internet of Things. He was a recipient of 2014 Top Level Talents in Sailing Plan of Guangdong Province, China, the 2015 Outstanding Young Professor of Guangdong Province, and the GLOBECOM 2010, ICC 2013, ComManTel 2014, WICON 2016, SigTelCom 2017 Best Paper Awards, the 2017 and 2018 IEEE Systems Journal Best Paper Awards, the 2017 Journal of Network and Computer Applications Best Research Paper Award, and the Outstanding Associate Editor Award of 2017, 2018 IEEE ACCESS. He has also served over 50 various Co-Chair for international conferences/workshops, such as IWCMC, ICC, ISCC, ICNC, Chinacom, especially the Symposium Co-Chair for IWCMC 2012, ICC 2012, the General Co-Chair for Chinacom 2014, Qshine 2015, Collaboratecom 2017, DependSys 2018, and SCI 2019, the TPC Chair for InisCom 2015, NCCA 2015, WICON 2016, NCCA 2016, Chinacom 2017, InisCom 2017, WMNC 2017, and NCCA 2018; TPC member of over 150 conferences, such as ICDCS, DCOSS, MASS, ICC, GLOBECOM, ICCCN, WCNC, and ISCC.

Miguel Martínez-García received a BSc degree in Mathematics from the Polytechnic University of Catalonia (UPC), Spain in 2013, a MSc in Advanced Mathematics and Mathematical Engineering (MAMME) from the same university in 2014, and a PhD in Engineering from the University of Lincoln, UK, in 2018. He also worked as a researcher since 2017, both at the University of Lincoln and at the Advanced Virtual Reality Research Centre (AVRRC) in Loughborough. In 2020 he became a Lecturer in Human-machine

Systems at Loughborough University. His research interests include human-machine integration, machine learning, artificial intelligence, intelligent signal processing, and complex systems, with particular focus in the analysis of non-linear signals representing phenomena of interest between humans and machines.