Wireless sensor networks for underwater surveillance systems

Erdal Cayirci, Hakan Tezcan *, Yasar Dogan, Vedat Coskun

Naval Science and Engineering Institute, Naval Academy, Tuzla, 34942 Istanbul, Turkey

Received 1 March 2004; received in revised form 21 September 2004; accepted 8 October 2004
Available online 8 December 2004

Abstract

In this paper, a new wireless sensor network architecture is introduced for underwater surveillance systems where sensors lie in surface buoys when nodes are first deployed. After deployment, sensors are lowered to various depths selected by our scheme such that the maximum coverage of the three dimensional sensor space is maintained. Each node has multiple microsensors of various types; acoustic, magnetic, radiation and mechanical. A classification based data mining scheme based on the readings of these sensor types detects and classifies submarines, small delivery vehicles, mines and divers.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Underwater sensor networks; Coverage, Node placement; Tactical surveillance; Classification mining; Decision tree; Proximity microsensor; Magnetic microsensor; Acoustic microsensor

1. Introduction

Recent advances in wireless communications and electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate untethered in short distances [1]. One of the best application areas for sensor networks made up of these tiny sensor nodes is an underwater surveillance system. We introduce an underwater sensor network architecture where sensors initially lie in the surface buoys. After the deployment of these buoys, sensors can be lowered to various depths such that the maximum coverage of the 3D sensor space is achieved.

The salient features of this architecture can be summarized as follows:

- Each sensor node has a bunch of sensors. These sensors can be lowered to a calculated depth via a cable, i.e., the sensors of the same node stay at the same depth. This is illustrated in Fig. 1.
• The cable that connects sensors to the surface buoy is used for communication between sensors and the buoy.
• By choosing the wired medium as the communication link between sensors and the surface buoy, we eliminate the difficulties of acoustic media such as reverberation, environmental noise, etc. Sensor nodes communicate with each other through the wireless medium over sea surface by using the antenna at the surface buoys.
• The buoys collect the sensed data from their sensors and convey this data to the data collecting buoy (\textit{cbuoy}), i.e., the sink. Please note that we call a buoy and the sensors attached to it as a sensor node.
• Uncovered zones that can exist due to sensor node failures or mobility can be hindered by rearranging the depths of the sensor nodes.
• The \textit{cbuoy} is the gateway between the network and the users of the system.

This architecture can be implemented for a tactical underwater surveillance system. In this paper we present a classification based mining scheme that detects and classifies submarines, small delivery vehicles (SDVs), mines and divers based on the sensed data from mechanical, radiation, magnetic and acoustic microsensors for this tactical underwater surveillance system. Moreover, we introduce a new distributed algorithm that runs on sensor nodes and calculates the depths of sensors according to the locations of buoys such that the maximum three-dimensional (3D) coverage of the sensor space is maintained. Our algorithm is an adaptive one that can rearrange the depths of sensor nodes as they move by currents, winds or other reasons.

The remainder of the paper is organized as follows: In Section 2 we discuss the related work. The formal definition of the 3D sensor space coverage problem and constraints are described in Section 3. The distributed algorithm for 3D sensor space coverage is examined in Section 4. A classification based mining scheme to detect and classify targets is introduced in Section 5. The performance of the scheme is evaluated in Section 6, and the paper is concluded in Section 7.

2. Related work

2.1. Sensor placement

In [2,3] an optimization problem on sensor placement is formulated to provide sufficient grid coverage of sensor field where two polynomial-time algorithms are presented to find out the optimum number of sensors and to place them such that the maximum coverage of the sensor field is achieved. The proposed scheme is for the fixed sensor nodes and runs better for the sensor fields that have obstacles. For the case that there are some obstacles that can hinder the communications between nodes in the field, the knowledge of the terrain is required before deployment.

Sensor placement is formulated as an optimization problem and then solved with integer linear programming (ILP) in [4]. This approach deals with sensor fields that have various sensor types different in costs and ranges. It finds out the types and the locations of sensor nodes for maximum coverage of the sensor field in terms of cost minimization.

The art gallery problem (AGP) [5] determines the minimum number of observers needed to cover the interior of an art gallery room such that every point is covered by at least one observer. A polynomial-time algorithm is presented to solve the 3D version of the AGP, which is an NP hard problem in [5].

Coverage was focused as the main criterion on power aware operation strategies in sensor net-
works in [6]. A set of sensor nodes is selected to be active by using an ILP based scheme. Since the sensors not active are placed in a special power-saving sleep mode, the overall energy consumption is reduced while maintaining the guaranteed sensor coverage.

Computational geometry and graph theoretic techniques, specifically the Voronoi diagrams and graph search algorithms are combined and a polynomial-time algorithm is presented for coverage calculation in [7]. The proposed algorithm finds the lowest coverage path, which maximizes the distance of the path to all sensor nodes, and the highest coverage path, which minimizes the distance of the path to the closest sensor nodes. Additional sensor deployment heuristics are also given to improve the stochastic coverage in [7].

All of these strategies are based on a central node and an algorithm that finds out the locations of the sensor nodes such that the maximum coverage is provided and then places the nodes into the selected locations. However in many applications, sensor nodes are randomly deployed and they randomly move around. A tactical underwater surveillance system that can be used to detect enemy submarines, SDVs, mines and divers is one example for such applications. Our scheme is a distributed one that fits the requirements of this application area.

2.2. Detection and classification

There are studies in applying data mining techniques to real time target detection, classification and tracking applications.

In [8] an effective data mining based intrusion detection system is introduced. An anomaly detection algorithm which uses statistics on packet header values is developed. System combines the information from multiple sensors to improve detection accuracy.

A wavelet packets based scheme for classification of underwater targets from the acoustic backscattered signals was developed in [9]. System uses a feature selection scheme and a backpropagation neural-network classifier.

A new adaptive underwater target classification system which uses backscattered acoustic data from targets is presented in [10]. System consists of upper and lower branches. The upper branch works as a memory system to identify the closest matches of an unknown pattern in the feature space and provides decision by using K-NN (K-Nearest Neighbor) algorithm. The lower branch performs feature mapping and classification.

A small underwater robot designed for experiments with sensor networks is described in [11]. Robot consists of a motor driver and various types of sensors. A large tank which contains fresh water was used as test bed. Depth measurement was accomplished by using the pressure sensor. Temperature of the water is tested in different depths.

In [12] a tree-based modeling method for classification of a fault-prone tactical military software module is presented. The system performed real time detection, classification and tracking of mobile and fixed objects in the field.

3. Three dimensional sensor space coverage problem

We use a 3D grid based coordinate system to denote the node locations. According to our coordinate system, a sensor space is a 3D space that covers all sensors. We accept a corner of the sensor space as the origin of our coordinate system. Since sensor nodes may move along with current or tide, the origin corner should be selected according to the expected places of the nodes for the intended sensing period. After assigning the origin point, we partition the whole sensor space into cubes in size. is called the resolution distance which is the edge length of a unit cube that sensor space is partitioned into. Nodes find out their locations based on the origin point and the resolution distance (r). Our coordinate system is shown in Fig. 2. In addition to the resolution distance we use another parameter x, the coordination distance, shown in Fig. 2. Coordination distance indicates the distance in the number of neighboring cubes for a node to coordinate its depth. While a node arranges its depth, it exchanges information with the nodes within the coordination distance.

We also identify each cube along with the z axis from sea surface to the maximum depth with a layer number. Hence all the cubes at the same
depth construct a layer, e.g., all the cubes between depth \(0\) and depth \(r\) construct layer 0.

3.1. The formal definition of our problem

Our goals are to

1. Maximize coverage efficiency (\(d\))

\[
\delta = \frac{n_c}{n_n} \tag{1}
\]

where \(n_c\) is the number of the cubes covered by at least one sensor and \(n_n\) is the total number of nodes in the sensor space. This aims to provide maximal coverage.

2. Maximize average distance (\(\theta\))

\[
\theta = \frac{\sum_{i=1}^{k} e_i}{k} \tag{2}
\]

where \(e_i\) is the distance between two nodes in node pair \(i\). If there are \(n_N\) nodes in a sensor field we can have as many as \(k\) node pairs given by

\[
k = \frac{n_N \cdot (n_N - 1)}{2} \tag{3}\]

This goal aims to reduce the probability that all nodes are concentrated on certain depths. Thus we hinder the existence of corridors so that an intruder cannot find an uncovered depth.

Our goals are subject to the following constraints:

1. We cannot move nodes in \(x\) and \(y\) axes. Nodes are randomly deployed. We can arrange only the depths of nodes.
2. Nodes may move with currents or tide. They most probably move in the same direction with the same speed. Therefore their relative locations to each other normally do not change. However if there are currents in different directions for different depths, then even the relative locations of nodes may change.
3. Our scheme needs to be distributed. We cannot rely on a central node that knows the up to date locations of nodes and arranges the depth of each node centrally.

4. Distributed 3D space coverage scheme

We introduce a new distributed heuristic to solve the underwater coverage problem described in the previous section. In this algorithm a node coordinates its depth with the other nodes in their coordination space which is the 3D space that include all the cubes whose \(x\) and \(y\) coordinates satisfy the following conditions:

\[
\begin{align*}
x_i - \alpha &\leq x_n \leq x_i + \alpha, \\
y_i - \alpha &\leq y_n \leq y_i + \alpha,
\end{align*} \tag{4}
\]

where \(x_n\) and \(y_n\) are the coordinates of a given cube and \(x_i\) and \(y_i\) are the coordinates of the node. This is also depicted in Fig. 2. A node exchanges information with the nodes in its coordination space and finds out the maximum distance in depth between

Fig. 2. The coordinate system for a sensor space with coordination distance \((z) = 1\).
these neighboring nodes and then arranges its depth to a cube in the middle of that distance. The packet formats and the block diagram of the algorithm are shown in Figs. 3 and 4 respectively.

After the deployment, a global organize packet is broadcasted by the cbuoy. When a node receives an organize packet, it listens info packets of neighbor nodes in its coordination space for a randomly selected time period. An info packet contains the coordinates of the node that sends the info packet, i.e., x, y and z coordinates in cubes according to the origin point as explained in Section 2. Therefore nodes learn about the coordinates of neighbors as they receive info packets and store these data in their neighbor table. An instance of neighbor table is illustrated in Table 1. The records in this table are ordered according to z value which is the depth of the related node. After waiting a random time period for info packets, the node finds out an appropriate depth based on the data available in its neighbor table. Please note that the node may not hear from all of its neighbors by that time. Basically the node checks its neighbor table and runs the algorithm shown in Fig. 4 to calculate its depth and sends an info packet to inform its neighbors and then tunes itself to the calculated cube. For example, in a sensor field that has 100 m depth the appropriate cube (depth) for the node that has the neighbor table shown in Table 1 will be “7” when resolution distance is 10 m. According to these parameters the sensor space has 10 layers. When we look at the neighbor table it can easily be seen that the total number of entries is smaller than the total number of layers. For such a case, the node finds the maximum distance in depth between the entries that follow each other and then tunes its depth to the cube in the middle of that distance. The distance between the nodes in the 5th and 9th cubes is the maximum comparing to the other distances between the nodes, therefore the node locates itself in the middle of that distance which is the 7th cube.

Nodes may change their x and y locations due to some reasons such as wind, current etc. When a node leaves its cube it sends an update packet that contains new location data. Thus neighbor nodes of the migrating node learn that the node left its cube. When a node receives an update packet it replies with an info packet if the sender node of the update packet is in its coordination space. Thus the migrating node learns about the coordinates of new neighbors and then arranges its depth according to the new neighbor table information.

In the reception of an update packet, the neighbor nodes of the migrating node in the left coordination space decide if there is a need for local organization. This decision is taken according to the local coverage efficiency (δ) value for their coordination space described in the previous section. If δ value for a node in the left coordination space is smaller than the threshold value, it sends a local_org packet, and then a local organization process which is similar to global organization starts. Nodes prefer not changing their depth because changing depth comes with a price of consuming extra energy.

![Table 1. The records in this table are ordered according to z value which is the depth of the related node.](image)

![Fig. 3. Control packet formats.](image)
The origin coordinates should be selected such that the sensor space is covered for the required time period as explained in Section 2. Nodes may move out of the sensor field in time which means the nodes cannot be addressed by using the current parameters such as origin coordinates, width (w),...
length \( (l) \), and depth \( (d) \). In this case a new organize packet that contains new values for these parameters is broadcasted by the cbuoy and a global organization process restarts.

It is still possible that two nodes in the same coordination space may use the same depth. When a node notices such a case from its neighbor table and there exists an uncovered cube in its coordination space, it sends a conflict packet that contains conflicted node data to one of these nodes. The node that receives the conflict packet updates its neighbor table, sends an info packet to inform its neighbors, and rearranges its depth.

The info, update and local_org packets include also a time to live \( (ttl) \) field. When a node takes one of these control packets, it decrements the \( ttl \) field and forwards it if the sender node of the packet is in its coordination space, and the \( ttl \) field is greater than zero. Since energy consumption is critical in wireless sensor networks, the number of the transmitted packets is very important. Therefore \( ttl \) value must be selected carefully.

The parameters, resolution distance \( (r) \) and coordination distance \( (z) \) are the key factors of our algorithm. Bigger \( z \) value brings more neighbor nodes to exchange information and causes more control message overhead. On the other hand bigger \( r \) value means lower resolution. We can trade off between resolution and power consumption by selecting balancing values for these parameters. We examine the impact of these parameters on the performance of our algorithm in detail during our experiments.

Our scheme can also be adapted for the case where nodes are not location aware. In such a case the nodes coordinate their depths with \( z \) hop neighbors. Note that \( z \) is the coordination distance in our scheme. In non location aware case of our scheme, nodes initialize \( ttl \) values in their info packet with the coordination distance \( (z) \). Every node which receives these packets decrements the \( ttl \) field, and relay them if \( ttl \) is greater than 0. They also reply these packets as explained above. Apart from this, there is no need to change our scheme to make the nodes coordinate with their \( z \) hop neighbors when they are not location aware.

5. Classification based data mining for underwater sensor networks

Sensors can be defined as devices that convert a non-electrical, physical or chemical quantity into an electrical signal [10]. Reduction in the size of sensors leads to increase its applicability. Advantages of microsensors are low weight, low manufacturing cost and wide range of application areas. Since microsensors are cheap and tiny, using them in sensor networks is feasible. Microsensors can be classified into five categories according to their principle form of signal: radiation, mechanical, magnetic, thermal and chemical [13].

5.1. Microsensors

Tactical underwater surveillance systems have three major tasks: detection, classification and tracking submerged targets. In this paper we focus on detection and classification of targets by using microsensors. Detection is to recognize the presence of a target in the proximity, and classification is to identify the target type. In our model mechanical and radiation microsensors are used to detect the proximity of the targets listed above. Magnetic and acoustic microsensors are used for the classification of the detected targets.

In Table 2 the microsensors that can be used to detect a given type of target are shown by a plus

<table>
<thead>
<tr>
<th>Sensor type</th>
<th>Target type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radiation</td>
<td>Diver +</td>
</tr>
<tr>
<td></td>
<td>Mine +</td>
</tr>
<tr>
<td></td>
<td>Submarine +</td>
</tr>
<tr>
<td></td>
<td>SDV +</td>
</tr>
<tr>
<td></td>
<td>Sea animals +</td>
</tr>
<tr>
<td>Mechanical</td>
<td>Diver +</td>
</tr>
<tr>
<td></td>
<td>Mine +</td>
</tr>
<tr>
<td></td>
<td>Submarine +</td>
</tr>
<tr>
<td></td>
<td>SDV +</td>
</tr>
<tr>
<td></td>
<td>Sea animals +</td>
</tr>
<tr>
<td>Magnetic</td>
<td>Diver +</td>
</tr>
<tr>
<td></td>
<td>Mine +</td>
</tr>
<tr>
<td></td>
<td>Submarine +</td>
</tr>
<tr>
<td></td>
<td>SDV +</td>
</tr>
<tr>
<td></td>
<td>Sea animals +</td>
</tr>
<tr>
<td>Acoustic</td>
<td>Diver +</td>
</tr>
<tr>
<td></td>
<td>Mine +</td>
</tr>
<tr>
<td></td>
<td>Submarine +</td>
</tr>
<tr>
<td></td>
<td>SDV +</td>
</tr>
<tr>
<td></td>
<td>Sea animals +</td>
</tr>
</tbody>
</table>
(+ ) sign. Please note that we use high number of randomly deployed sensor nodes. After deployment, our nodes adjust their depth such that the best coverage of the 3D sensor space is achieved. Therefore we assume that there is at least one microsensor bunch, i.e., every sensor node has several types of sensors and all of the sensors of a node are at the same depth, within the detection range of every point in the 3D sensor space.

Various types of radiation microsensors and their characteristics are shown in Table 3. Radiation sensors are non-conducting, because they detect emission. They can be classified into three categories: Ultraviolet (UV) wavelength is between 0.002 and 0.4 μm, Visible (Vis) wavelength is between 0.4 and 0.7 μm, and Infrared (IR) wavelength is between 0.7 and 500 μm. Since IR thermal changes can be used to detect proximity we can use radiation microsensors as proximity sensors. Proximity detection sensors that we can use in our model are photo-transistors (IR), photo-diodes (IR) and photo-conductive CdS photocells (Vis). Since the cost of photo-multiplier tubes and other photo-conductors can be as high as $100, and photo-multiplier-tubes are mechanically fragile, they do not fit our requirements. The cost of a photo-transistor or a photo-diode sensor is about 25 cents. A photo-conductive CdS is even cheaper, i.e., about 10 cents. Their operating temperature is between −55 and +100 °C. The size of a photo-transistor or a photo-diode is around 5 cm., and the size of a photo-conductive CdS cell is 0.5 cm. The characteristics of radiation microsensors are in Table 3.

The detection of a target that does not have thermal radiation such as mine needs other types of sensors such as visible light and NIR (Near Infrared) phototransistors to detect the proximity. Since mines lie in visible area (about 30 m depth) we can use these microsensors to detect the proximity of targets that do not have thermal radiation.

For proximity detection we can use also mechanical microsensors such as microswitches (touch sensor) shown in Table 4.

Since divers use specially designed clothes, which prevent heat transfer, and also the heat of a mine is equalized with the heat of water, it is hard to detect a diver or a mine with a thermal microsensor. So we use other types of microsensors to classify targets. One of these is magnetic microsensors. Magnetic microsensors are used to control underwater traffic, to monitor magnetic properties of the volcanic rock on the seafloor and to find underwater wrecks. We can use a magnetic hall effect device which consists of attached wire to each side of a thin square or rectangular plate. The plate is often fixed to a ceramic substrate that provides mechanical support and thermal stability. Characteristics of a hall device are shown in Table 5. Divers wear special clothes, use diver tube and carry special metallic weights. Mines have metallic components. SDVs and submarines also contain lots of metal. All of these cause difference in magnetic field that can be detected by a hall device.

An acoustic sensor is another type of sensor that can be used in underwater target detection. In

<p>| Table 3 |</p>
<table>
<thead>
<tr>
<th>Radiation microsensors [14]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical characteristics</td>
</tr>
<tr>
<td>Available wavelengths (μm)</td>
</tr>
<tr>
<td>Sensitivity</td>
</tr>
<tr>
<td>Cost</td>
</tr>
<tr>
<td>Physical size</td>
</tr>
</tbody>
</table>

<p>| Table 4 |
| Microswitches [15] |</p>
<table>
<thead>
<tr>
<th>Mechanical life</th>
<th>Temperature range</th>
<th>Operating limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;10⁷ operations</td>
<td>0 °C and 70 °C</td>
<td>Operating force (max.) Release force (min.)</td>
</tr>
<tr>
<td>4.2 N</td>
<td>1.7 N</td>
<td></td>
</tr>
</tbody>
</table>
acoustic detection we can use a SAW device given in Table 6. A SAW device has a polished surface, which produces an electrical wave when an acoustic wave hits its surface.

5.2. Decision tree for target classification

One of the important tasks in data mining is classification of data which can be carried out by using techniques such as naïve bayes, neural networks and decision trees. We prefer decision trees because of their fast execution time, ease in the interpretation of the rules, and scalability for large multi-dimensional data sets [18–21]. To utilize collaborative computing we apply classification based data mining, and use a decision tree to detect and classify targets.

Well-known, top-down, greedy search algorithms for building decision trees are ID3 (Induction of Decision Tree) [22] and C4.5 [23] for inducing Decision Trees from data. These algorithms use entropy and information gain metrics to induce a decision tree. Entropy and information gain are calculated for all attributes and the attribute with the greatest information gain is selected for the branch. Our scheme uses J48 algorithm [24], which is an implementation of the C4.5 scheme [23]. The algorithm reduces the size of a decision tree by pruning.

J48 algorithm needs training data to induce decision tree. For our scheme some assumptions are made by using target specifications explained above and sensor selection criteria in Table 2. We study each target type in order to define the possible values that a sensor measures when the target is within the sensing range of a node, and than prepare the training data set for our classification mining based detection algorithm. The differences that the target types in our domain cause in the ambient magnetic and acoustic conditions of their vicinity are shown in Table 7, which is our training data set.
In our training data set we have 26 tuples to classify an object into one of five different classes. When we apply this training set to J48 decision tree algorithm in WEKA (Waikato Environment for Knowledge Analysis) package [24] we get the pruned decision tree shown in Fig. 5.

J48 [24] calculates the information gain metric for all attributes and chooses the highest one for the node. It continues recursively to the end of the tree. Please note that the first node after detection by a mechanical or radiation sensor is acoustic. For our training data set acoustic has the highest information gain value so the tree for the classification starts with the acoustic attribute. A mine can be correctly classified by using only an acoustic microsensor after the detection of proximity with a radiation or a mechanical microsensor. If we have proximity report and the acoustic value is 'none', we can say that it is a ‘mine’.

J48 algorithm gives the confusion matrix in Table 8 for our case. As shown in Table 8 all targets except for SDVs are correctly classified. 2 of 6 SDVs are classified as submarine because the changes in the magnetic and acoustic conditions made by an SDV or submarine is almost the same in short distance. As you can see in Table 7 tuple 3 for SDV and tuple 4 for submarine is the same. So there can be misclassification.

We have two metrics for classification. Our first classification metric is the classification probability $P_{ik}$, where the target type $i$ is classified as the target type $k$. In Table 9 we show the classification probabilities for all possible target type pairs in our target domain. For example the probability that an SDV is classified as an SDV is 66%, and an SDV is classified as a submarine is 34%. These probabilities are derived by WEKA [24] based on our training data and confusion matrix.

<table>
<thead>
<tr>
<th>Target</th>
<th>Target classified as</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diver</td>
<td>4</td>
</tr>
<tr>
<td>Mine</td>
<td>6</td>
</tr>
<tr>
<td>SDV</td>
<td>4</td>
</tr>
<tr>
<td>Submarine</td>
<td>6</td>
</tr>
<tr>
<td>Sea animal</td>
<td>4</td>
</tr>
</tbody>
</table>
Another metric is precision $e_i$, which is the probability that a target classified as type $i$ is of type $i$. Precision $e_i$ values for our application is shown in Table 10. Precision for a mine is 1. This means if a target is classified as mine, the probability that the target is a mine is 1. This is different from the values in Table 9 which shows the probability that a given type of target is classified as a certain type. For example the probability that an SDV is classified as a submarine is 0.34. However, when a target is classified as an SDV, we are sure that it is an SDV. Precision for a submarine is 0.75 because if an object is classified as a submarine, it is 75% (100/134) a submarine and 25% (34/134) an SDV. The precision for other target types is 1.

If the nodes are time synchronized and location aware, proximity reports can be used to derive the speed and the vectoral displacement of a target, which may increase the precision of the target classification. The vectoral displacement parameter indicates the variations in the speed and the movement direction of a tracked target. For this reason a node that reports a proximity attaches also the location and time tags to the proximity report. The measurements made by magnetic and acoustic sensors are used for data association, which is to associate the data reported by multiple sensors to a specific target. When we have a time series about the location of a tracked target, then its speed and vectoral displacement can easily be computed. Since we focus on the basic detection and classification scheme in this paper, we do not explain the details about our solution for speed and vectoral displacement calculations.

### 6. Performance evaluations

#### 6.1. Coverage efficiency

In our simulations varying number of sensor nodes are deployed randomly over a space of $100 \times 100 \times 60$ in size. The resolution distance parameter is 10. We assume that each sensor node has a radio module that has a transmission range of 20. After running our distributed algorithm sensor nodes select their depths between 0 and 60.

As our fundamental purpose is to maintain the maximum coverage, the coverage efficiency and the average distance between the nodes are the main criterion in our experiments. Please note that coverage efficiency ($\delta$) and average distance ($\theta$) parameters are explained in Section 3.1. We compare our algorithm with a random depth selection model where depths of each node are selected randomly according to uniform distribution. Since random numbers are determined by using the same seed, the random deployment model deploys the nodes almost evenly in the depth axis. Although the random deployment algorithm provides an even distribution of nodes, our algorithm outperforms it.

In Fig. 6 the coverage efficiency ($\delta$) of our algorithm and the random deployment are shown for varying number of nodes deployed in the sensor field. Our algorithm tries to cover all the uncovered cubes while maximizing the distance between the nodes. But since the nodes are randomly deployed to sea surface, the number of nodes in the same column may exceed the total number of layers. For that reason a node may not find an uncovered cube, and arranges its depth to the minimum covered cube, i.e., the cube that has the minimum number of nodes, in its column. Unless the number of the sensor nodes in the same column overruns the total number of layers, there will be no reduction in the coverage efficiency of our algorithm.

<table>
<thead>
<tr>
<th>Table 9</th>
<th>The classification probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>Target classified as</td>
</tr>
<tr>
<td></td>
<td>Diver Mine SDV Submarine Sea animal</td>
</tr>
<tr>
<td>Diver</td>
<td>100%</td>
</tr>
<tr>
<td>Mine</td>
<td>100%</td>
</tr>
<tr>
<td>SDV</td>
<td>66% 34%</td>
</tr>
<tr>
<td>Submarine</td>
<td>100%</td>
</tr>
<tr>
<td>Sea animal</td>
<td>100%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 10</th>
<th>Correct identification precision table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Precision</td>
</tr>
<tr>
<td>Diver</td>
<td>1</td>
</tr>
<tr>
<td>Mine</td>
<td>1</td>
</tr>
<tr>
<td>SDV</td>
<td>1</td>
</tr>
<tr>
<td>Submarine</td>
<td>0.75</td>
</tr>
<tr>
<td>Sea animal</td>
<td>1</td>
</tr>
</tbody>
</table>
In Fig. 7 the coverage efficiency ($\delta$) of our algorithm for varying coordination distances is depicted. When the coordination distance increases, the coordination space of nodes gets larger, and this results in an improvement in the coverage efficiency as it can be seen in Fig. 7. Since all the uncovered cubes in the coordination distance of a node will be covered after a certain number of nodes are deployed, our algorithm produces the same results for all the coordination distance values as expected.

In Fig. 8 coverage efficiency ($\delta$) for various resolution distances is shown. As the resolution distance increases, the volume of a unit cube increases. For the higher resolution distance, the number of sensor nodes per a cube increases and the coverage efficiency decreases.

In Figs. 9 and 10 average distance ($\bar{h}$) for varying coordination distance is shown for various numbers of sensor nodes. When the coordination distance increases, the number of neighbors also increases. Nodes try to select an uncovered layer when the total number of neighbors of a node
exceeds the total number of layers, and ignore maximizing average distance. Therefore for the higher coordination distance there will be a decrease in the average distance until all the layers are covered. After all layers are covered, nodes try to cover an uncovered cube in their column. Therefore after a certain number of nodes deployed, the average distance is not sensitive to the changes in the coordination distance. The anomalies in the results between 100 and 350 nodes are due to the randomness in the deployment of nodes.

In Fig. 11 we compare the number control packets transmitted for our scheme per sensor node. We use the worst case distance between nodes while evaluating \( \text{ttl} \) value of control packets. When the total number of sensor nodes increases, the number of neighbor nodes of a sensor node also increases and this results in more control packets. The increase in the coordination distance results more neighbor nodes to exchange information. As we can see in Fig. 10 the number of control packets sent by a sensor node increases for the higher coordination distance.

In Fig. 12 the number of nodes deployed in one layer are shown for varying coordination distances and random deployment when 600 nodes are deployed in our sensor space partitioned into six layers. As shown in Fig. 12, our algorithm maintains a homogenous distribution of nodes among the layers.

### 6.2. Detection and classification efficiency

In this section we evaluate the performance of our classification mining based detection and classification (CMDC) scheme for various metrics. The first two of these metrics are correct classification rate (\( \beta \)) and precision (\( \varepsilon \)). Correct classification rate \( \beta \) is the ratio of the number of
correctly classified targets to the total number of targets.

We compare the performance of three mostly used learning algorithms in terms of classification rate $\beta$ and precision $\epsilon$. These algorithms are OneR, Naïve Bayes and J48.

Fig. 13 depicts the correct classification ratio for each target type separately. The aggregated correct classification ratio for all types of targets in our domain is shown in Fig. 14. The experiments are performed on the WEKA based on our training data set given in Table 7.

As shown in Fig. 13, J48 and Naïve Bayes algorithms correctly classify all of the targets except for SDVs. The correct classification rate of J48 for SDV is 66%. The probability that an SDV may be incorrectly classified as submarine by J48 algorithm is 0.34. OneR algorithm classifies all divers, mines and submarines correctly. Correct classification rate of OneR for SDV and sea animals is 0. All of SDVs and sea animals are incorrectly classified by OneR algorithm.

Aggregated correct classification rates of algorithms are shown in Fig. 14. J48 algorithm has 92%, OneR has 61% and Naïve Bayes has 84% correct classification rate ($\beta$). As shown in Fig. 13 and Fig. 14, J48 algorithm has the highest correct classification rate.

Precision is different from the correct classification rate. It gives the probability that a target is of the type that it is categorized in by our scheme. For example the precision for a mine is the probability that the target is a mine when it is classified as a mine. In Fig. 15 we show the precisions for J48, OneR, Naïve Bayes algorithms.

The precision for mine is 1 in all three algorithms. If an object is classified as a mine that means the probability that it is a mine is 1. The precision of diver for OneR is 0.33 and precision of diver for Naïve Bayes is 0.66. These two algorithms always correctly classify the divers but may classify also the other objects as a diver. Therefore we prefer J48 algorithm.
7. Conclusions

In this paper, we introduce a new distributed 3D space coverage scheme for tactical underwater sensor networks where sensor nodes transmit their data packets through the antenna in surface buoys. Although sensor nodes are randomly deployed and their $x$ and $y$ coordinates cannot be arranged, the sensors of these nodes can be lowered at any depth. Our scheme finds out an appropriate depth for each node such that the maximum 3D coverage of the field is maintained. Although our scheme is distributed and adaptive, it maintains a high coverage of the sensor space in the expense of acceptable control traffic overhead.

We also introduce a classification mining based detection and classification scheme for tactical underwater sensor networks where mechanical, radiation, magnetic and acoustic microsensors are used. Our scheme first detects a target in the vicinity based on the readings of radiation and mechanical sensors. Then the detected target is classified into one of the following target types based on the data coming from acoustic and magnetic microsensors: a diver, an SDV, a submarine or a mine.

References


Erdal Cayirci graduated from Turkish Army Academy in 1986, and from Royal Military Academy Sandhurst in 1989. He received his MS degree from Middle East Technical University, and the PhD degree from Bogazici University in computer engineering in 1995 and 2000, respectively. He was a visiting researcher with Broadband and Wireless Networking Laboratory and a visiting lecturer with the School of Electrical and Computer Engineering at Georgia Institute of Technology in 2001. He is the director of Combat Models Operations Department in Turkish War Colleges Wargaming and Simulation Center, and the director of Sensor and Actuator Networks Laboratory in Istanbul Technical University. His research interests include sensor networks, mobile communications, tactical communications, and military constructive simulation. He is an editor for IEEE Transactions on Mobile Computing, AdHoc Networks (Elsevier Science), ACM/Kluwer Wireless Networks, ASP Sensor Letters. He received 2002 IEEE Communications Society Best Tutorial Paper Award for his paper titled “A Survey on Sensor Networks” published in the IEEE Communications Magazine in August 2002.

Hakan Tezcan graduated from Turkish Naval Academy in 1997. He received his MS degree in Computer Science from Turkish Naval Academy Naval Science and Engineering Institute in 2004. He is the system support and management officer of Combat Models Operations Department in Turkish War Colleges Wargaming and Simulation Center. His research interests include wireless communications and sensor networks.

Yasar Dogan was born in Cankiri in 1973. He was graduated from Turkish Army Academy in 1995. He received his M.Sc. degree in Computer Science from Turkish Naval Academy, Istanbul. He is currently Network Administrator of Turkish Ranger School, Egirdir. His current research areas include computer security and sensor networks.

Vedat Coskun was born in Istanbul in 1962. He was graduated from Turkish Naval Academy in 1984. He received his M.Sc. degree in Computer Science from Naval Post Graduate School, California and his Ph.D. degree in Computer Security from Yildiz Technical University, Istanbul. He managed the software development group in Naval Wargaming Center. He is currently Chairman of Computer Engineering Department, Turkish Naval Academy. His current research areas include software engineering, sensor networks, computer security and cryptography.

Vedat Coskun was born in Istanbul in 1962. He was graduated from Turkish Naval Academy in 1984. He received his M.Sc. degree in Computer Science from Naval Post Graduate School, California and his Ph.D. degree in Computer Security from Yildiz Technical University, Istanbul. He managed the software development group in Naval Wargaming Center. He is currently Chairman of Computer Engineering Department, Turkish Naval Academy. His current research areas include software engineering, sensor networks, computer security and cryptography.