Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 634748, 14 pages
http://dx.doi.org/10.1155/2014/634748

*Research Article*

# Applying an Ontology to a Patrol Intrusion Detection System for Wireless Sensor Networks

**Chia-Fen Hsieh, Rung-Ching Chen, and Yung-Fa Huang**

*Chaoyang University of Technology, 168 Jifeng E. Road, Wufeng District, Taichung 41349, Taiwan*

Correspondence should be addressed to Rung-Ching Chen; crching@cyut.edu.tw

Received 9 July 2013; Revised 4 November 2013; Accepted 8 November 2013; Published 6 January 2014

Academic Editor: Hung-Yu Chien



With the increasing application of wireless sensor networks (WSN), the security requirements for wireless sensor network communications have become critical. However, the detection mechanisms of such systems impact the effectiveness of the entire network. In this paper, we propose a lightweight ontology-based wireless intrusion detection system (OWIDS). The system applies an ontology to a patrol intrusion detection system (PIDS). A PIDS is used to detect anomalies via detection knowledge. The system constructs the relationship of the sensor nodes in an ontology to enhance PIDS robustness. The sensor nodes preload comparison methods without detection knowledge. The system transfers a portion of the detection knowledge to detect anomalies. The memory requirement of a PIDS is lower than that of other methods which preload entire IDS. Finally, the isolation tables prevent repeated detection of an anomaly. The system adjusts detection knowledge until it converges. The experimental results show that OWIDS can reduce IDS (intrusion detection system) energy consumption.

## 1. Introduction

Recently, wireless security issues have drawn the attention of wireless network and wireless sensor network (WSN) researchers. WSN is a novel technology that involves the deployment of low-cost microhardware and resource-limited sensor nodes. Applications of WSN include battlefield supervision, disaster response, and health care [1, 2]. After sensor nodes are deployed, they self-organize and establish routes automatically and transmit their information on their surroundings to a base station (BS). Since each sensor node has a limited and irreplaceable energy resource, energy conservation is the most important performance consideration in a WSN.

A WSN has two major defenses: cryptography and an intrusion detection system (IDS). Cryptography protects information via encryption, decryption, and authentication of each node. Cryptography is the first line of protection in WSN security. An IDS protects information by anomaly detection. An IDS detects each node by its behavior. If a sensor node is misbehaving, the IDS will alert its managers. This is the second line of defense in WSN security.

A Sybil attack is a common method that attackers use to gather information from the WSN. Intruders pretend to be sensor nodes, routes, and/or base stations. They use these roles to request and collect data. When they have received data, they copy it and return it to the real and victim nodes to establish their bona fides. The attackers thus obtain the information they need to finish their preparations. This attack type merely copies information without altering it. It is difficult for the system to detect it and to redeploy against further intrusion [3]. If the intrusion detection system prevents Sybil attacks, it can reduce the severity of attacks. To counter this attack, each node must be identified and authenticated correctly. The authentication method requires a simplified algorithm to reduce energy consumption. However, if it is too easy to decrypt, it will lose efficacy. Thus, the system combines IDS and an ontology to construct the relationship between each node, providing a novel way to detect Sybil attacks. We proposed an ontology IDS method which can detect Sybil attacks to prevent further attacks by intruders.

Soft computing has been widely used for wired security due to its high knowledge extraction capabilities. However, little research has been done on using soft computing in

WSN IDS. Soft computing consumes resources while the model is being trained and tested with various machine learning tools such as SVM [4], rough set [5], and ANN [6]. Unfortunately, WSN IDS resources are limited. Thus, lightweight soft computing applications for IDS are critical [7–9]. The IDS uses well-trained features to reduce the features of the system. If the training and testing of IDS WSN information use soft computing processing in the base station, it can be lightweight. In this paper, we will implement ontology-based lightweight method technologies to improve the effectiveness of WSN IDS.

An ontology is a knowledge representation method. The main aim of the ontology is to classify independent knowledge into concepts and to determine the relationship between them. The classification knowledge of the ontology is used to infer new knowledge. In our research, the nodes of the WSN are constructed in the ontology in their entirety. The relationships of the sensor nodes may then be applied to detect malicious nodes [10].

After deployment of the WSN is completed, the base station (BS) will gather position information. During the preparation stage, the IDS establishes the conceptual relationships of each sensor node in the ontology. The transmission of each node will depend on its relationship to the ontology. The attacker cannot then pretend that malicious nodes are valid nodes. Thus, the major contribution of this paper is to propose a lightweight intrusion detection method based on the domain knowledge.

The method is divided into four steps. (1) Construct the relationship of the sensor nodes in the ontology. The patrol nodes will use the relationships of ontology to enhance system robustness. (2) Choose detection knowledge depending on the monitoring environment. The patrol node loads detection knowledge to perform a circuit of anomaly detection. (3) Record the error information in an isolation table. And (4) repeat these steps until the detection knowledge has converged. In fact, we rename the ranger node the patrol node, since it is more appropriate for the node attributes. Thus, in this paper we use PIDS (patrol intrusion detection system) instead of RIDS (ranger intrusion detection system) as in our previous paper [11].

The rest of this paper is organized as follows. Section 2 presents the literature review. Section 3 introduces our methodology. Section 4 shows the experimental results and evaluations. In Section 5, conclusions and suggestions are given for future research.

## 2. Related Work

*2.1. Intrusion Detection Systems in Wireless Sensor Networks.* Intrusion detection systems detect intruders based on their attack behaviors. In cryptography, each node authenticates other nodes using their encryption method, which is known as a symmetric key. Authentication methods only protect against outsider attacks as the first line of defense. If attackers penetrate this defense, they can gather cryptographic information. Thus, the IDS is the second line of defense, which detects user misbehaviour and alerts managers immediately [12].

Wireless sensor networks broadcast data among nodes. A potential intruder gathers such data until it is able to decrypt authentication. After an intruder obtains the associated keys, it can connect to neighboring nodes and attack them freely. In some cases, intruders can gather sufficient information to crash the entire network.

However, intruder behavior is different from that of normal nodes. System managers use intrusion detection systems to detect anomalous behaviors. An intrusion detection system has two detection methods: misuse detection and anomaly detection [13]. The misuse detection system stores behaviors of known attacks in an attribute database. The system compares user behaviors with the attribute database to find intrusions. The anomaly detection system stores normal behaviors of common users in the rule database. The system compares user behavior with normal behavior to find intruders.

In a misuse detection system, the attack rules are composed of known attack behaviors. This type of system is similar to antivirus software in which scanned data are compared to known virus codes. If the behavior is found in the attribute database, the system deletes the affected files. The misuse detection system stores the known attack behaviors in an attribute database. If the attack behavior is similar to the rules in the database, an attack is detected and the system defends itself. The main drawback of misuse detection is that the detection is dependent on information already in the attribute database, so it is difficult to identify new attack behaviors.

Anomaly detection is different from misuse detection. In anomaly detection, the system constructs a user model based on the behavior of normal users. When user behavior is abnormal, the system notifies managers that there is a potential intruder. Since intruder attack methods rapidly evolve, the anomaly detection system collects normal user behaviors and detects intruder behavior by comparing it with normal behavior. The anomaly detection system must clearly define correct user behaviors. Otherwise, the system will have a high false detection rate. Thus, the drawback of an anomaly detection system is greater likelihood of false alarms.

WSN intrusion detection approaches are divided into four types: continuous, event-driven, observation-driven, and combination [14].

(1) Continuous: the IDS records alarms but does not transmit data to the administrator immediately. When the detection process has finished its duty cycle, the IDS will return alarms to the BS.

(2) Event-driven: this method has no duty cycle. When attack misbehavior is detected, the IDS will transmit the information to the administrator immediately. The administrator then must decide how to process the anomaly information.

(3) Observation-driven: the anomaly information is processed when the system detects an attack. If the intrusion is serious, the administrator can initiate the isolation method to limit the damage.

(4) Combination: this method combines two or more of the above methods. In a normal system, sensor nodes report detection data periodically. If attack behavior is detected, the IDS will alert the administrator.

*2.2. Ontology Methods.* An ontology represents the relationships between the concepts of domain knowledge. In an ontology, the system often assumes correct relationships between each concept. A defined ontology for intrusion detection systems was constructed by Undercoffer et al. [15]. According to their definitions, three different components made up the construction of ontologies for intrusion detection systems. The first category reflects the network class, including the network layers of the protocol stack, such as TCP/IP. The second category is the system class, representing the operating system of the host. The last process class defines attributes representing particular processes that are to be monitored. They use the ontology specification language DAML (DARPA Agent Markup Language) and OIL (Ontology Inference Layer) to implement their ontology and to distribute information. The ontology and the inference engine were used as an event aggregation language to confirm the existence of an attack on a wired network.

Cuppens-Boulahia et al. [16] presented an approach to react to network attacks using an ontology to store policy information and to generate new security models. The policy information models can be combined to prove the instantiation of the policies. They used the ontology specification languages OWL (Web Ontology Language) and SWRL (Semantic Web Rule Language). They offer the advantage of existing generic tools for parsing and reasoning. OWL allows merging distributed ontologies. However, the expressivity of SWRL is limited, since it does not permit several logic operators such as OR or NOT. Further, the SWRL rules may not be used to alter or delete information in the ontology. This fact reduces the potential to revoke their method. Most of the literature on WSN focuses on constructing the rules for the IDS on WSN. This literature lacks information on how to construct the relationships of nodes in the WSN [17, 18].

## 3. Ontology-Based Wireless Intrusion Detection System (OWIDS)

When sensor nodes are taken over by an intruder, the normal sensor nodes become malicious nodes. The malicious nodes gather data from neighboring nodes in the preparation stage. They alter information and broadcast wrong information to normal nodes. Moreover, they rapidly exhaust WSN resources. In wired networks, the manager detects intruders using a well-trained intrusion detection system working with robust resources. However, the resources of a WSN are limited and the protocols of a WSN differ from those of wired networks. Managers cannot construct wired IDS on a WSN. In this paper, we propose a lightweight intrusion detection system that minimizes energy consumption in intrusion detection for wireless sensor networks.

The Sybil attack is a common attack method that is used to gather information from a WSN. It is hard to detect but enables an intruder to attack the WSN more easily. Our method uses an ontology to construct relationships between sensor nodes. The constructed relationship enables the IDS to detect a Sybil attack.

In this paper, the wireless sensor network is a hierarchical network that has four roles, namely, base station, cluster head, patrol nodes, and sensor nodes. They are defined as follows.

(1) Base station (BS): the BS is controlled by the administrator and has robust energy and computing power as well as a high degree of security. The base station receives environmental information and sensor node data from the cluster head. The base station uses a detection module to analyze the data. The patrol nodes will patrol the network and monitor it. When the patrol cycle is completed, the base station integrates the new collection of the network data. The integrated data is simplified into the database until the attacking pattern knowledge has converged. The administrator analyzes information from each node and constructs the relationship of the WSN in the ontology. The relationship adjustment depends on the WSN environment.

(2) Cluster head (CH): the CH is responsible for connectivity between the wireless sensor network and base station. The CH controls the work of the patrol nodes. The patrol nodes return information to CH. The CH will balance the duty cycle of the patrol nodes and then assign patrol nodes for monitoring and data integration. The CH transmits detection knowledge and ontology relationships to the patrol nodes.

(3) Patrol nodes (PN): a patrol node is a sensor node which carries knowledge of how to detect intrusion. They share the work of the CH. Since the WSN environment includes general events and emergencies, the patrol nodes collect information on their sensor nodes. The patrol nodes use the detection knowledge to monitor sensor nodes, integrate information, and transmit it back to a CH. The node relationships carried by the patrol node are used to detect attacks. Patrol nodes will record information into an isolation table. In normal cases, the patrol nodes send the isolation table to the CH regularly. However, if an unexpected situation occurs, the isolation table will be transmitted to the CH immediately.

(4) Sensor nodes (SN): they are responsible for the overall network and sense environmental data. They transmit the data to patrol nodes for regular integration. The sensor nodes have no ontology information at all.

The workflow of our system is shown in Figure 1. First, the system gathers WSN packages and attack packages to build an intrusion features database to enable evaluation of anomalous transmission packages. The system then applies the ontology to construct the relationship between the wireless sensor nodes. The manager sets the threshold value of the ontology relationship to detect attacks. The patrol intrusion detection system (PIDS) is a lightweight system that uses the detection knowledge to monitor the nodes

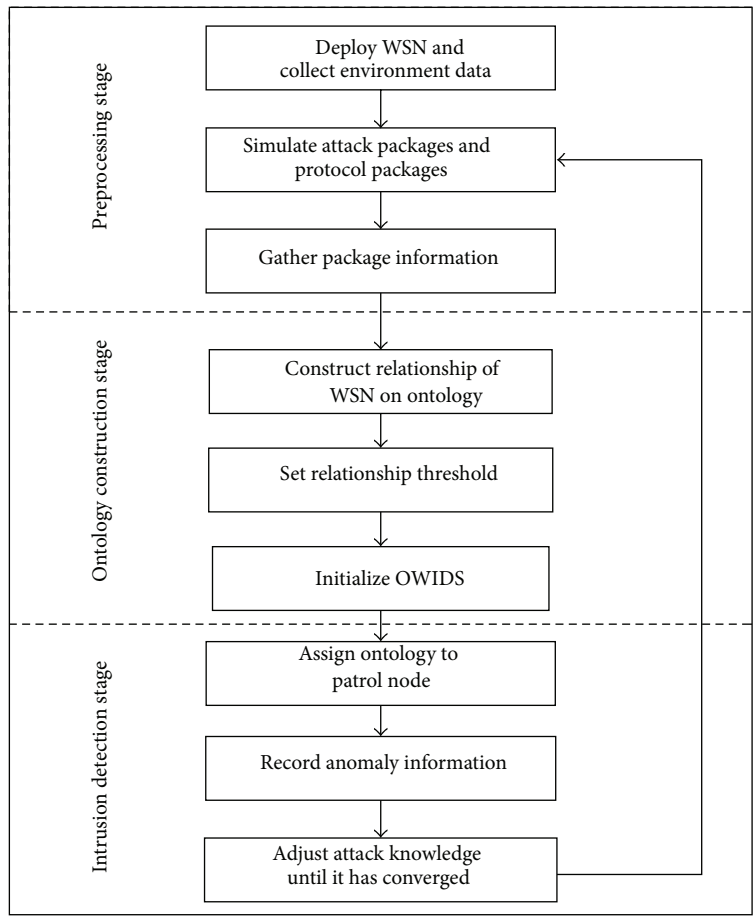


Figure 1: The workflow of our system.

in the wireless network. The patrol node carries different types of knowledge of how to detect intrusion depending on the environment. The ontology-based wireless intrusion detection system (OWIDS) is divided into three stages: the preprocessing stage, ontology construction stage, and intrusion detection stage.

The preprocessing stage shows that each node must translate its data to base station and find the best translation route for such data. The ontology construction stage is divided into the definition relationship and ontology construction phase. Each node calculates the membership value and defines the relationships between patrol nodes and sensor nodes. The intrusion detection stage compares the detection data to the ontology pattern to find intrusions. Finally, the system records anomaly information in an isolation table.

The algorithm of the preprocessing stage is shown in Algorithm 1, the algorithm of the ontology construction stage is shown in Algorithm 2, and the algorithm of the intrusion detection stage is shown in Algorithm 3. The symbols of the algorithms are as follows: $I$ is information including "SensorNodes ID, energy $E$, 4 hops, sensed data type ST"; arr[] is the type of resource of the sensor nodes; Patrolcandidate[] represents the candidate list of patrol nodes; CH is the cluster head; $s_i$ and $s_j$ represent different sensor nodes in the WSN; resource ($s_i$) represents the resources of $s_i$, such as energy, hops, and sensor data type; hop($s_i$, $s_j$) is the number of hops between $s_i$ and $s_j$; patrolnode($s_i$) represents the patrol node assigned to $s_i$; $pn_k$ represents the ID of the patrol nodes; *ontology*[] means the ontology constructed by our method; and $A$[] represents the isolation table.

*3.1. Preprocessing Stage.* The system attempts to configure wireless sensor network nodes through random distribution to simulate the wireless option connection. When the distribution of every node is completed, the packet will be collected by the base station (BS). In addition, the BS can use a routing protocol package to analyze intrusion behaviors. To define several attacking behaviors, the manager can use attack thresholds or attack features. The transmission package can be captured by the intrusion detection system. The data then need to be normalized in preprocessing, followed by the construction of a network intrusion detection module. The algorithm of the preprocessing stage is shown in Algorithm 1.

After the WSN has been deployed, the sensor nodes broadcast to each other and construct route information. The BS gathers data from the sensor nodes that includes the sensor node identification (number), energy (joules), hop distance (hops), and sensing data type (ST). The system uses the received information to construct the ontology.

**Algorithm** OWIDS
***Input***: The packages of wireless sensor networks $p$
***Output***: $A$ list of anomaly nodes $A[]$
**BEGIN**
/* *Pre-processing Stage** /
/* *broadcasting routing** /
**for** each sensor node $s$ connected to $x$ **do**
**for** each sensor node $y$ in the network **do**
**if** $D_s[y] + s(x, s) < D_x[y]$ **then**
/* a better route from $x$ to $y$ through $s$ has been found* /
$D_x[y] \leftarrow D_s[y] + s(x, s)$.
$C_x[y] \leftarrow (x, s)$
/* *integrating information** /
**for** each sensor node $s$ transmitted to base station $bs$ **do**
send data of itself $I$ to base station $bs$
**return** $I$

Algorithm 1: Algorithm of preprocessing stage of OWIDS.

*3.2. Construct Ontology Stage.* Due to the nature of a Sybil attack, it is hard to use an IDS to detect it. In the system, the relationships of each sensor node in the ontology are constructed first. Then, the patrol node compares the relationships of the ontology conceptual nodes to the tested wireless networks to determine whether the system has been attacked. The method for constructing the ontology is given below. The algorithm of the ontology construction stage is shown in Algorithm 2.

After the entire WSN has been deployed, the information on each sensor node is translated to the base station for the system to construct the ontology. The relationships of the sensor nodes are constructed in the ontology. Each sensor node transmits its connection information to the base station. This method thus is a top-down design for ontology construction in which construction proceeds from the base station to the sensor nodes. The system sorts sensor nodes depending on the energy of the sensor nodes and selects the sensor node that has the best energy to be the cluster head. The candidates for patrol nodes are the top 20% of sensor nodes. The patrol nodes are chosen by the candidate of patrol nodes that are one hop from the cluster head. After the system picks the patrol nodes up, it begins to construct the ontology.

First, the system constructs patrol nodes in the ontology. The system calculates the similarity between the patrol nodes using formula (1). Formula (1) is calculated between patrol nodes and patrol nodes. $pn_i \cap pn_l$ means that the system compares the data of $pn_i$ with the data of $pn_l$ to gather the minimum number (the data is energy(), hop(), ST(), etc.). The $pn_i \cup pn_l$ means that the system computes the data of $pn_i$ with the data of $pn_l$ to gather the minimum number. Consider

$$Similarity\left(pn_i, pn_l\right) = \frac{\sum_{l=1}^{n}\left(\left|pn_i \cap pn_l\right| / \left|pn_i \cup pn_l\right|\right)}{n}. \quad (1)$$

The construction of the patrol nodes in the ontology depends on formula (1), which indicates the relationship of the patrol nodes. Similar patrol nodes exchange information with each other. The relationship is used to construct sensor nodes in the ontology. There are two concepts of sensor node: equal concept and sibling concept as follows.

*Definition 1* (equal concept). Consider

$$Equal\left(s_i^1, s_i^2\right) := \left\{\left(s_i^1, s_i^2\right) \mid hop\left(s_i^1, s_i^2\right) = 1 \wedge \left(SensorNode\left(s_i^1\right) \neq SensorNode\left(s_i^2\right) \wedge resource\left(s_i^1\right) \approx resource\left(s_i^2\right)\right)\right\}. \quad (2)$$

The $s_i^1$ and $s_i^2$ are two different sensor nodes in the WSN. The distance between $s_i^1$ and $s_i^2$ is equal to one hop. $s_i^n$ indicates the sensor node $n$ is managed by *patrolnode* $i$. $SensorNode(s_i^n)$ and $resource(s_i^n)$ indicate the class name of the *SensorNode* and the resources of *SensorNode* $s$, respectively. When $resource(s_i^1) \approx resource(s_i^2)$, it means the resources (energy, hops, sense type) of $s_i^1$ are similar to the resources of $s_i^2$. Definition 1 defines the concepts of sensor nodes having equivalent resources and being close to each other. For example, if a pair of concepts "$s^1$" and "$s^2$" has overlapped broadcasting range and possesses equivalent resources, the equal concept definition is satisfied and the system will construct the relationship between them. The rest of concepts will be used to determine the hierarchical relations in the ontology concepts.

*Definition 2* (sibling concept). Consider

$$Sibling\left(s_i^1, s_i^2\right) := \left\{\left(s_i^1, s_i^2\right) \mid s_i^1 \neq s_i^2 \wedge \left(SiblingTerm\left(s_i^1, s_i^2\right)\right) \wedge 2\right.$$

```
/* Construct Ontology Stage */
        /* sort sensor nodes */
        for each sensor node s in the network do
            for (i = n − 1; i > 0; i − −) do
                for (j = 0; j < i; j++) do
                    if energy(s_j) > energy(s_i) then
                        buffer = arr[j]
                        arr[j] = arr[j + 1]
                        arr[j + 1] = buffer
            return arr[] /* return sort information */
        set CH = arr[1] /* the system selects up best sensor node to be cluster head */
        /*pick up patrol nodes*/
          for (p = 0; p < n * 0.2; p++) do
          Patrolcandidate[p] = arr[p] /* pick up top 20% of sensor nodes */
          for each sensor node s in the Patrolcadidate[] do
          if hop(s_i, CH) = 1 then
              /*a patrol node has found*/
              set patrolnode[] = s_i
        /*Construct patrol nodes in Ontology*/
          for each patrol node in the network do
              for (l = 1; l≤ n; l++) do      /*l is the index of patrol node */
                  t_l = intersection (pn_i, pn_l) / union (pn_i, pn_l) /*t_l is similarity of patrol node l*/
                  t = t + t_k /*t is the similarity */
          return t
         similiarity(pn_i, pn_l) = t/k
         pn_l[] ← pn_i
        Ontology[] ← pn_l[]
        /* definition sensor nodes relationship */
        for each sensor node s in the network do
          if s_i <> s_j and resource(s_i) resource(s_j) and hop(s_i, s_j) = 1 then
                  /* a equal sensor has found */
                  set s_i and, s_j are equal sensor nodes
        for each sensor node s in the network do
          if s_i <> s_j and resource(s_i) resource(s_j) and 2 ≦ hop(s_i, s_j) ≦ 3 then
                  /*a sibling sensor has found*/
                  set s_i and, s_j are sibling sensor nodes
          else if patrolnode(s_i) ∩ patrolnode(s_j) <> then
                /* a sibling sensor has found */
                set s_i and, s_j are sibling sensor nodes
        /* Construct Ontology */
        for each sensor node in the network do
  for (k = 1; k≤ n; k++) do      /*k is the index of patrol node */
   t_k = intersection (pn_k, s_i) / union (pn_k, s_i) /*t_k is similarity of patrol node k*/
   t = t + t_k   /*t is the similarity */
  return t
similiarity(pn_k, s_i) = t/k
pn_k[] ← s_i
Ontology[] ← pn_k[]
```

Algorithm 2: Algorithm of the construct ontology stage of OWIDS.

$$\begin{aligned} &\leq hop\left(s_i^1, s_i^2\right) \\ &\leq 3 \vee \left(PatrolNode\left(s_i^1, 1\right)\right. \\ &\qquad \left.\cap PatrolNode\left(s_i^2, 1\right) \neq \emptyset\right)\Big\} \\ &resource\left(s_i^1\right) \approx resource\left(s_i^2\right). \end{aligned} \tag{3}$$

The $SiblingTerm(s_i^1, s_i^2)$ indicates that $s_i^1$ has a sister term $s_i^2$. In other words, the sensor nodes have an identical patrol node in the WSN, and $PatrolNode(s_i^1, 1)$ means that the distance of $s_i^1$ from the patrol node is one hop. When the distance between $s_i^1$ and $s_i^2$ is less than or equal to 3, it will be adjusted depending on the scale of the WSN. Definition 2 has two situations: either a pair of concepts has sister-term relations in the WSN, or a pair of concepts has a common

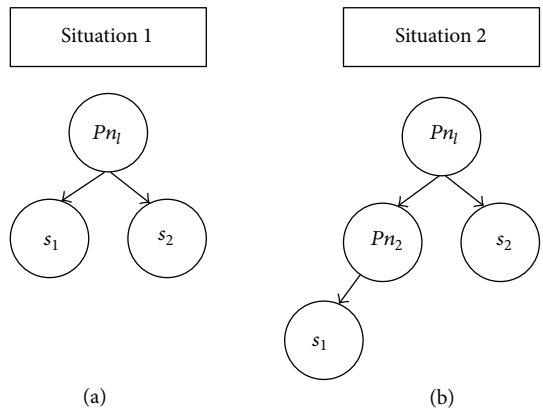


Figure 2: The sibling relationship.

patrol node over more than one level. The difference between them is shown in Figures 2(a) and 2(b). Situation 1 and situation 2 are the relationships of *SensorNode* $s_i^1$ and *SensorNode* $s_i^2$ in the WSN structure. Situation 1 indicates that $s_i^1$ and $s_i^2$ have a sister-term relationship in the WSN; situation 2 indicates $s_i^1$ and $s_i^2$ are not sister terms. However, they satisfy the sibling concept because they have a common patrol node on the upper level. In this case, the distance between $s_i^1$ and $s_i^2$ is 3 hops, as shown in Figure 2(b).

An ontology has various elements including concepts, attributes, operators, instances, relations, and axioms. Our ontology has membership values between the patrol node ($pn$) and the sensor node ($s$). The set of values between the patrol nodes and sensor nodes is called the subclass. For example, the sensor node contains attributes such as sense information, remaining resources, and route information. The membership values between patrol nodes and sensor nodes can be calculated by formula (4) [19]. Such values might include, for example, $n_1${energy(0.8), hop (0.2), ST(0.9)}, $pn_2$(energy(0.3), hop(0.8), ST(0.8)}, and $s_1$(energy(0.75), hop(0.2), ST(0.7)}. The subclass is $(s_1, pn_1)$ = (0.75 + 0.2 + 0.7) / (0.8 + 0.2 + 0.9) = 1.65/1.9 = 0.87. The subclass is $(s_1, pn_2)$ = (0.3 + 0.2 + 0.7) / (0.3 + 0.8 + 0.8) = 1.2/2.1 = 0.57. The membership value consists of sensor nodes, such as $pn_1$ (0.87) and $pn_2$ (0.57); the numbers represent the membership values in each patrol node. The membership value between a patrol node and a sensor node may then be found. The $s_1$ is the subclass of $pn_1$. Consider

$$subclass\,(s, pn) = \frac{|s \cap pn|}{|pn|}. \tag{4}$$

The relations between sensor nodes can be defined in a number of ways, such as Belong-to and Consist-of, while the subclass defines the values of the relationships between the sensor nodes. After constructing the concept layer, the membership values can be simply and directly joined to the ontology. Each concept has the membership values for each relevant patrol node. The system can then build up the concept hierarchy. The ontology information includes sensor node identify (number), patrol node identify (number), energy (joule), hop distance (hops), and sensing data type (ST), as shown in Figure 3.

The related value of the concept should define a suitable threshold. Let $pn_k$ be the $k$th patrol node of WSN. $s_i$ indicates the sensor node $i$ in the WSN, $pn_m$ is the patrol node that belongs to $pn_k$, and $n$ is the number of comparison nodes under the patrol node $pn_k$. The similarity formula is listed in formula (5). Formula (5) is similar to formula (1). Formula (1) is the similarity between patrol node and patrol node. Formula (5) is the similarity between patrol node and sensor node:

$$similarity\,(pn_k, s_i) = \frac{\sum_{m=k}^{n} \left(|pn_m \cap s_i| \,/\, |pn_m \cup s_i|\right)}{n}. \tag{5}$$

For example, the system uses formula (5) to calculate the similarity of formal concepts. Each patrol node has many sensor nodes. Figure 4 shows an example of calculation between $pn_k$ and $s_i$. It is not an ontology figure. An example of the ontology is shown in Figure 5. The similarity between $pn_k$ and $s_i$ is calculated as follows:

$$\begin{aligned} &similarity\,(pn_1, s_1) \\ &\quad = \left(\frac{0.75 + 0.2 + 0.7}{0.8 + 0.2 + 0.9} + \frac{0.2 + 0.7}{0.75 + 0.2 + 0.75} \right. \\ &\qquad \left. + \frac{0.75 + 0.6}{0.75 + 0.2 + 0.6}\right) \times (3)^{-1} = 0.68, \\ &similarity\,(pn_2, s_1) \\ &\quad = \left(\frac{0.3 + 0.2 + 0.7}{0.3 + 0.8 + 0.8} + \frac{0.3 + 0.7}{0.75 + 0.2 + 0.7}\right) \times (2)^{-1} \\ &\quad = 0.47. \end{aligned} \tag{6}$$

We select the maximum similarity value of the conceptual pairs between $pn_k$ and $s_i$ to determine the sonsor node $s_i$ belongs to which *patrol nodes pn*. In this case, $s_1$ is more similar to $pn_1$. The IDS calculates the relationship between $np_k$ and $s_i$ to define the relationship threshold. The IDS calculation threshold depends on the similarity formula in different environments. A higher threshold makes the entire WSN more secure. However, a higher threshold for the IDS means that attacks are easily misidentified.

An example of an ontology is shown in Figure 5. There are 10 sensor nodes in the example. For practical applications, our method supports 50 sensor nodes in the ontology. The ontology has a domain layer, a category layer, a patrol node layer, and a sensor node layer. The domain layer represents the ontology domain issue that we focus on in constructing an IDS for a WSN. Next, the category layer represents different jobs on the WSN, such as sensing humidity, temperature, and brightness. One sensor node will take on one or more tasks. The patrol node layer contains the information for each patrol node in the WSN. Each patrol node has an ID, energy (joule (j)), sensing data type (ST), the distance to the cluster head (Hop), and membership value between each relevant object, for example, $pn_{\text{id}}$(Energy(j), hop(1/hop), ST(1/number of sense types)}. The sensor node layer contains each sensor node ID, energy (joule (j)), sense data type

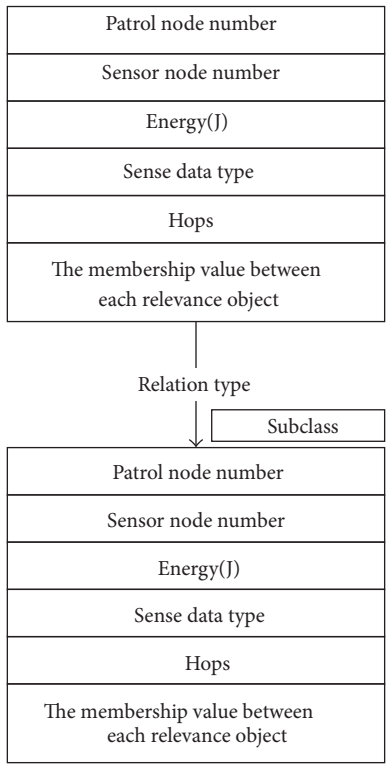


Figure 3: The information of the concept of nodes in ontology.

(ST), the distance to the cluster head (hop), and membership value between each relevant object.

*3.3. Intrusion Detection Stage.* The attackers that intrude wireless sensor network can be divided into two stages: preparatory phase, the attack and destruction phase. The attack behaviours of attack and destruction phase include sinkhole attack, blackhole attack, and hello flooding. In order to obtain the required information of attack and destruction phase, the attackers apply Sybil attack to disguise various roles. The behaviours of attack phase used different transmission technology to burn entire network, even cause WSN unusable. All the above behaviours are called anomaly behaviours. The assignment system will complete the information for the integration of the environmental analysis. The ontology contains the entire relationship of the WSN. The system can preselect the knowledge based on those environmental attacks. The ontology can thus detect illegal nodes in the WSN. The rules are shown in Algorithm 3.

The WSN is usually deployed unequally, causing it to rapidly consume energy. We have proposed a PIDS using detection knowledge to detect anomalies [11]. The PIDS is transferred between patrol nodes to detect whether neighboring nodes exhibit anomalies. The PIDS will choose different detection knowledge in different environments. The PIDS merely requires a portion of the detection knowledge to detect an anomaly, which reduces WSN energy consumption. The architecture of the OWIDS combines the PIDS with the ontology. The OWIDS is transferred between the patrol nodes to detect whether neighbouring nodes are anomalous. Similar to the PIDS, the OWIDS will choose different detection knowledge in different environments. It requires merely a portion of the detection knowledge to detect anomalies, reducing WSN energy consumption. The system will match the database and the attack pattern. If there appears to be nonselected detection knowledge, the detection knowledge will be added. The intrusion detection mechanisms were computed on the base station. The classification method of detection knowledge is described in the following paragraphs.

The detection knowledge is classification by support vector machine (SVM) [21]. The proposed method provided a strong argument for the improvement of WSN intrusion detection systems. The SVM uses a high dimension space to find a hyperplane to perform binary classification to find minimal error rate. Notably, the SVM is able to handle the problem of linear inseparability. The SVM uses a portion of the data to train the system, finding several support vectors which represent the training data. These support vectors will be formed into a model by the SVM, representing a category. According to this model, the SVM will classify a given unknown document. A basic input data format and an output data domain are given as follows:

$$(x_i, y_i), \ldots, (x_n, y_n), \quad x \in R^m, \; y \in \{+1, -1\}, \tag{7}$$

where $(x_i, y_i), \ldots, (x_n, y_n)$ are training data, $n$ is the number of samples, $m$ is the input vector, and $y$ belongs to category of $+1$ or $-1$.

Regarding linear problems, a hyperplane can be divided into two categories. The hyperplane formula is

$$(w \cdot x) + b = 0. \tag{8}$$

The category formula is

$$\begin{aligned} (w \cdot x) + b &\geq 0, \quad \text{if } y_i = +1, \\ (w \cdot x) + b &\leq 0, \quad \text{if } y_i = -1. \end{aligned} \tag{9}$$

However, it is not easy to find hyperplanes with which to classify the data. The SVM has several kernel functions which users can apply to solve different problems. As such, selecting the appropriate kernel function can solve the problem of linear inseparability. Also, internal product operations affect the classification function and a suitable inner product function $K(Xi \cdot Xj)$ can solve certain linear inseparable problems without increasing the complexity of the calculation. Original data includes 31 features tagged with numbers from 0 to 30. The system will convert the contents of the 31 features into numeric values. For instance, the first element is "Event" and the parameter is "$s$" which is mapped to "0 : 0.1." An example of SVM data is shown in Figure 9. The OWIDS combines ontology with detection knowledge. The patrol nodes carry part of ontology to detect Sybil attack and the detection knowledge is used to detect sinkhole, blackhole, and hello flooding.

The ontology is divided into several parts depending on the region of the patrol node. The system sends detection knowledge and the ontology to the cluster head. The patrol

```
/* Intrusion Detection Stage */
        for each neighbor of patrol node s do
        receive (s_id, pn_id, s_INFO, E_i)    /* Receive s_id, pn_id, s_INFO and the remaining energy */
        if s_id <> Ontology [] then      /* Check whether the sensor node is constructed in the Ontology */
           then A[] = (s_id, pn_id, A_n)  /* Record s_id, pn_id and anomaly information */
        if Pattern (s) ! = Pattern (AM)
        /* Check whether the receive information is different from attack knowledge */
           then A[] = (s_id, pn_id, A_n)  /* Record s_id, pn_id and anomaly information */
           else if broadcast A[] to C_id    /* Broadcast isolation table to CH to back up */
           end for
```

Algorithm 3: Algorithm of the intrusion detection stage of OWIDS.

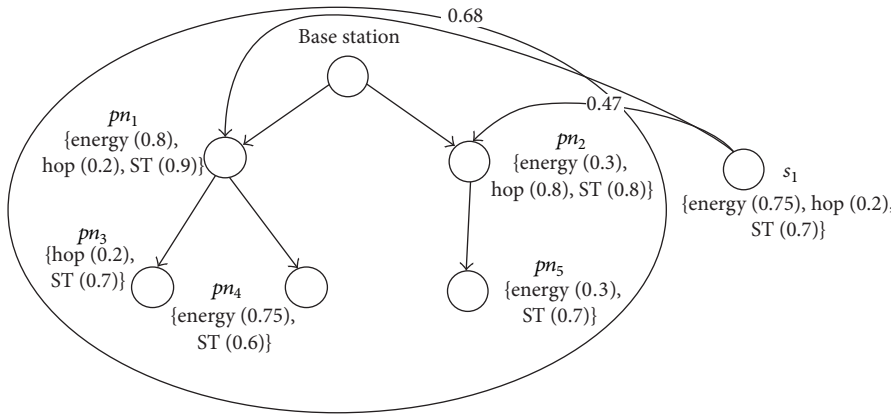


Figure 4: An example of membership calculation.

node is a sensor node to do detection, but its energy is limited. The OWIDS does not train detection mechanism on patrol node and only the cluster head transmits detection rules to patrol nodes. The patrol nodes detect attacks by rotation to reduce the overall WSN energy consumption and extend the lifetime of the WSN. When the duty of the patrol nodes is over, the CH will collect the WSN information and transmit it to the base station. The BS analyzes the information, trains the detection data, and constructs the ontology again. Thus, the detection knowledge can be adjusted according to the WSN environment. To improve the accuracy of the intrusion detection, the system repeats the detection knowledge training to remove less frequently used data until the knowledge converges. This reduces the features of the detection knowledge and makes the detection knowledge more lightweight.

The WSN is a self-organizing network, meaning that the routing table will be changed. The clustering system of the WSN will change cluster heads periodically. The WSN as a whole thus cannot keep isolating anomalous nodes. Redetection of anomalous nodes consumes energy. This paper thus proposes an isolation table, recorded in the base station [20]. If new cluster heads are assigned or the entire WSN is changed, the patrol maintains isolation of anomaly nodes using the isolation table. Intruders thus cannot attack the WSN through isolated anomaly nodes, reducing redetection energy consumption.

## 4. Experimental Results

The system performance was evaluated by network simulation (NS-2). The experimental hardware environments consisted of an Intel Core 2 i5-2410M CPU @ 2.30 GHz Notebook with 8 GB RAM and was implemented under the Windows 7 operating system. The area for the simulation of the WSN was within 10,000 square meters. The field is static and deploys 300 sensor nodes randomly, each with a broadcast radius of 50 m. The OWIDS uses a protégée to construct the ontology. The behaviors of the attacker were generated by a Java program. The attacks were randomly generated by different kinds of attack patterns. For example, Sybil attacks were generated every 20 seconds, but the attackers will masquerade different kinds of roles randomly, such as cluster head, sensor node. The OWIDS detects Sybil attacks using the relationship of ontology. In addition, the experiment was meant to simulate transmission packages in wireless sensor networks. The communications of packets can be captured by OWIDS. The data was collected from the WSN packets of NS-2 simulator. The data set can be divided into two protocols: the training data and the testing data. The packets are randomly selected for training and testing.

Before using SVM classification, the system perform a data scaling operation to increases the accuracy while reducing complexity. The system kernel is RBF $K(x, y) = \ell^{-g\|x-y\|^2}$. The system will classify the attributes of the features

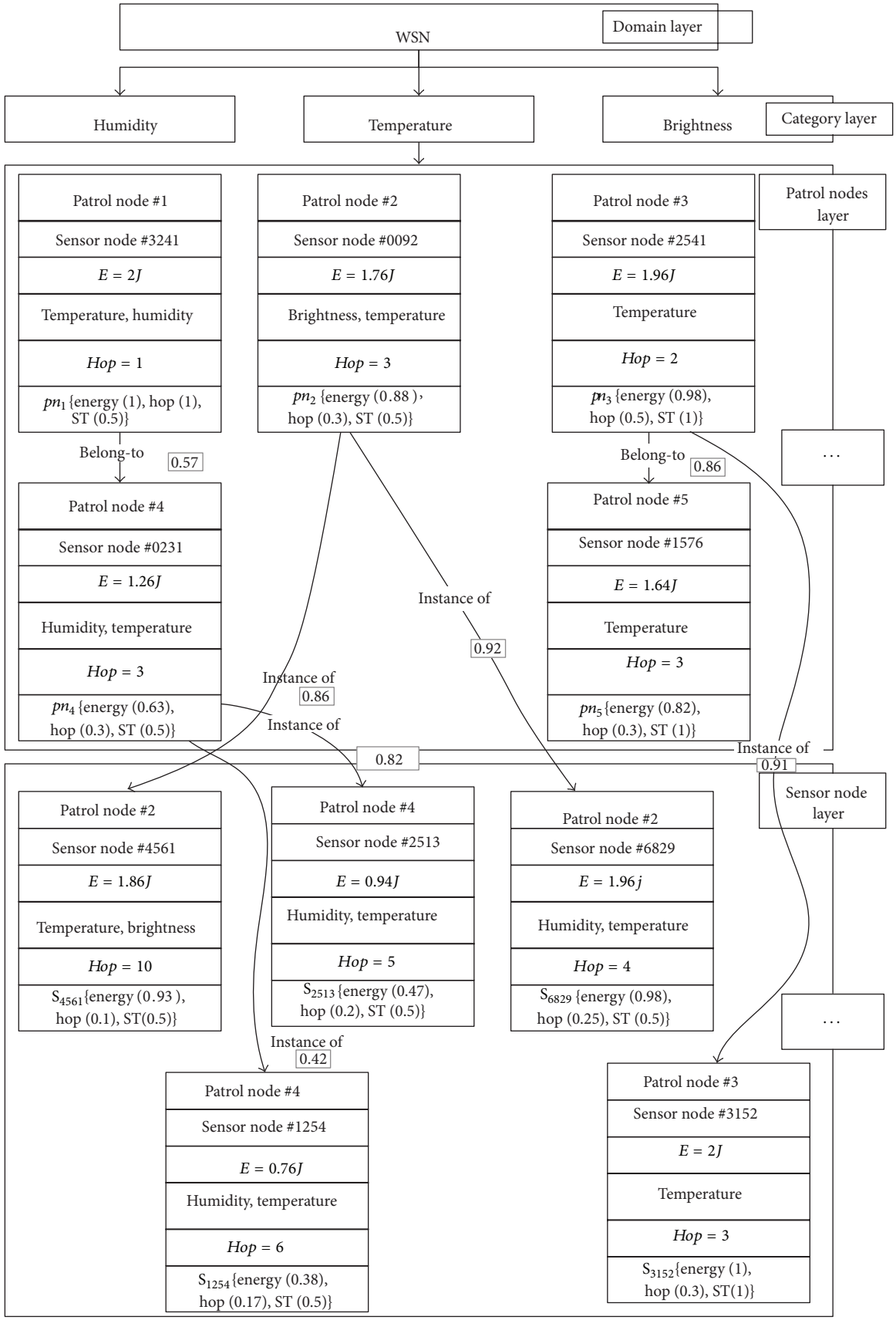


Figure 5: An example of ontology construction.

prior to preprocessing and use the SVM to train and test processes. The output of SVM is 1 or −1. If the output is 1, there is an intrusion behavior on the model. If the output is −1, it is normal. The distribution of training data and testing data are shown in Table 1. Specifically, the user can employ this model to evaluate the IDS. 27 feature values are used to train three SVM models and to compare the accuracy of the three models [21].

This experiment compared OWIDS with the PIDS based on energy consumption, transmission accuracy, and performance. The attack behaviour is camouflaged as different roles in the WSN. The implementation environment is listed in Table 2. The total simulation time was 3600 sec. The attacks were randomly executed every 20 sec with attacks beginning at six hundred sec. The experiment assumes that the preprocessing stage is secure for attacks. The maximum connection, meaning the maximum number of connections (end-to-end connection) used in the simulation, was 80. The experimental IDS was intended to simulate transmission packages in wireless sensor networks. The IDS integrates communications of packages and analyzes them.

To estimate the performance of the OWIDS system, three important formulas and an indicator method are used to evaluate system accuracy: attack detection rate (ADR), false positive rate (FPR), system accuracy (SA), and live nodes (indicator). The ADR represents the number of attacks that the IDS has detected out of the total number of attacks. The FPR represents the number of normal processes that the IDS has misclassified. The accuracy rate is the total number of processes the IDS has classified correctly. The live nodes represent the number of useful nodes in the entire WSN, which tests the loading of our methods and its performance:

$$\text{Attack Detection Rate} = \frac{\text{Total number of detected attacks}}{\text{Total number of attacks}} \times 100\%,$$

$$\text{False Positive Rate} = \frac{\text{Total number of misclassified processes}}{\text{Total number of normal processes}} \times 100\%,$$

$$\text{Accuracy Rate} = \frac{\text{Total number of correct classified processes}}{\text{Total number of processes}} \times 100\%. \quad (10)$$

In this section, we present three experimental results for the OWIDS. The first experimental method focuses on the percentage of patrol nodes. The patrol node is a kind of sensor node. The BS chooses patrol nodes to execute the IDS. The percentage of sensor nodes which should be patrol nodes is a key issue. Second, the energy consumption and remaining resources are present in the live nodes. The third experimental method focuses on the ADR, FPR, and SA of the OWIDS. Thus, two types of experiments were implemented: comparison of the number of live nodes across the non-IDS, PIDS, and OWIDS and comparison of the transmission accuracy with PIDS.

Table 1: The training data and test data.

| | Original | Train | Test |
|---|---|---|---|
| Normal | 80,641 | 24,192 | 7,258 |
| Sinkhole | 3,568 | 1,070 | 321 |
| Blackhole | 5,368 | 1,610 | 483 |
| Hello flooding | 44,084 | 13,225 | 3,968 |
| Total | 133,661 | 40,098 | 12,029 |

Table 2: Implementation environment.

| Parameter | Values |
|---|---|
| Sensor nodes | 50, 150, 300 |
| Patrol nodes (20% of sensor nodes) | 10, 30, 60 |
| WSN size | $1000 * 1000$ ($m^2$) |
| Starting energy | 2 J |
| Transmission radius | 50 m |
| Transmission consumption | 0.036 w |
| Receive consumption | 0.024 w |

*4.1. Percentage of Patrol Nodes.* The patrol node is used to detect abnormal behaviours. OWIDS needs to balance the numbers of patrol nodes and the overall life cycle of wireless sensor networks. From the experiment results, twenty percentages of patrol nodes can provide better detection results. The percentage of patrol nodes was set between 10% and 40%, at intervals of 10%. The experiments are divided into 50, 150, and 300 sensor nodes. These numbers are used to calculate the suggested percentage of patrol nodes. The experimental results are shown in Tables 3, 4, and 5.

The experimental results show that 30% and 40% yield the highest accuracy. However, the lifecycle of entire WSN is shorter than that when 10% and 20% are used. The lifecycle of 10% is the longest but its accuracy is the lowest. When OWIDS sets the patrol node percentage to 10%, the loading is the lightest. However, for the OWIDS at 10%, the patrol nodes lack sufficient data to be compared with each other, meaning that the accuracy is the lowest. The accuracy of the case of 20% patrol nodes is close to that of the case of 30% or 40% patrol nodes. Hence, the system selected 20% of whole sensor nodes as patrol nodes to do detection to save energy.

*4.2. Live Nodes of OWIDS.* The first simulation is shown in Figures 6, 7, and 8. The number of live nodes is defined as the number of sensor nodes in the WSN that still work normally. The normal IDS trains detection features and translates the entire IDS onto the sensor nodes. This consumes the WSN energy more rapidly. The lightweight IDS trains the detection method on the base station and uses filtered features to detect intrusions. In this case, the IDS is more lightweight and nodes remain alive longer. Since a WSN may be used in many applications, our method is implemented in eight situations.

Table 3: Percentage of patrol nodes (50 sensor nodes).

| Percent of patrol nodes (%) | Patrol nodes | Lifecycle (sec) | Accuracy (%) |
|---|---|---|---|
| 10 | 5 | 9835 | 85 |
| 20 | 10 | 9573 | 93 |
| 30 | 15 | 8952 | 94 |
| 40 | 20 | 8703 | 94 |

Table 4: Percentage of patrol nodes (150 sensor nodes).

| Percent of patrol nodes (%) | Patrol nodes | Lifecycle (sec) | Accuracy (%) |
|---|---|---|---|
| 10 | 15 | 9605 | 81 |
| 20 | 30 | 9495 | 96 |
| 30 | 45 | 8864 | 96 |
| 40 | 60 | 8518 | 97 |

Table 5: Percentage of patrol nodes (300 sensor nodes).

| Percent of patrol nodes (%) | Patrol nodes | Lifecycle (sec) | Accuracy (%) |
|---|---|---|---|
| 10 | 30 | 9598 | 79 |
| 20 | 60 | 9468 | 96 |
| 30 | 90 | 8867 | 97 |
| 40 | 120 | 8361 | 97 |

The live nodes experiment has six situations: no IDS and no attack, no IDS but faces attacks, loads PIDS but no attacks, loads PIDS and faces attacks, loads OWIDS but no attacks, and loads OWIDS and faces attacks. The overhead for PCH monitoring is high if the notion of collaborative monitoring is absent. If the patrol nodes are too few, an intruder can easily infiltrate the network. The OWIDS combines PIDS and the ontology to detect anomalies. Though it consumes more energy than the PIDS, the energy expenditure is nearly identical. We simulate 50, 150, and 300 sensor nodes in the non-IDS, PIDS, and OWIDS. Results show that the lifetime of a sensor node is longest using OWIDS and that the sensor nodes die slowly. The MN sends data to the RN but not to any MN. The RN obtains information directly from the MN. The connections between the RN and the PCH are similar, meaning that the OWIDS consumes less energy in gathering data and monitoring the WSN. The OWIDS takes a part of the ontology from PIDS. The energy consumption of PIDS and OWIDS is similar in the no attack environment. In Figures 6–8, there are more sensor nodes in the WSN, meaning that the system has more patrol nodes to detect anomalies. Hence, the OWIDS detects misbehaviour more effectively. The results of the simulation show that the OWIDS can easily be loaded onto the WSN.

*4.3. ADR, FPR, and SA of OWIDS.* The second simulation addresses the attack detection rate, the false positive rate, and the accuracy rate in PIDS and OWIDS, as shown in Table 6. The total number of simulation packages is 1,065,474

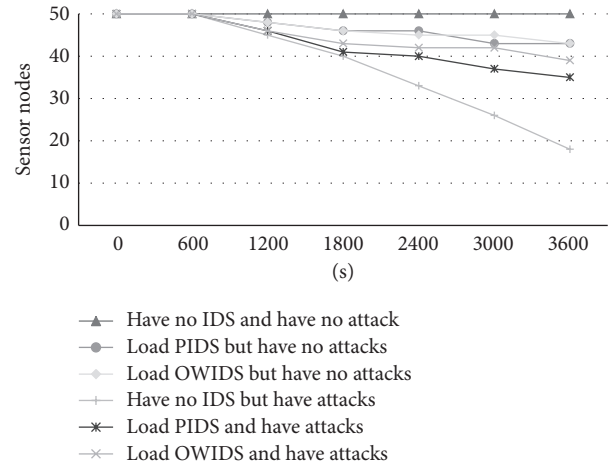


Figure 6: The case of 50 sensor nodes in alive nodes.

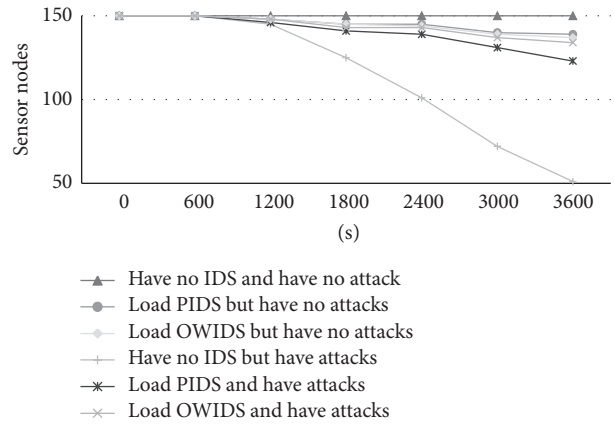


Figure 7: The case of 150 sensor nodes in alive nodes.

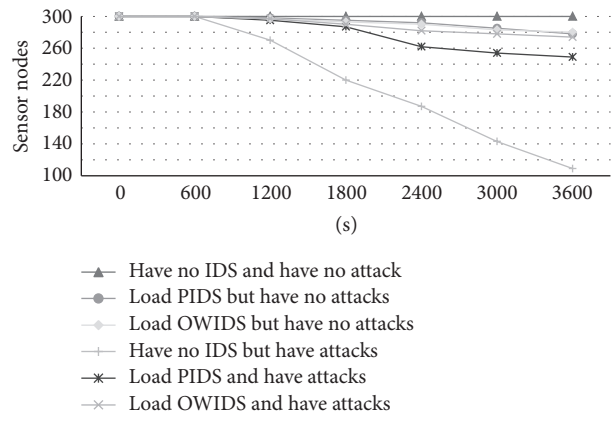


Figure 8: Case of 300 sensor nodes in live nodes.

which include 1,697 attack packages and 1,063,777 normal packages. In the PIDS, patrol nodes carry attack knowledge to detect anomalies. The false positive rate is higher and the attack detection rate is lower than that of OWIDS. The OWIDS appends the relationships of ontology, making it easier to detect illegal sensor nodes in the WSN. The system

Table 6: Accuracy, attack detection rate, and false positive rate of OWIDS.

| IDS method | Attack detection rate | | False positive rate | | Accuracy | |
|---|---|---|---|---|---|---|
| PIDS | (1,535/1,697) | 90.45% | (141,445/1,063,777) | 13.29% | (953,287/1,063,777) | 89.61% |
| OWIDS | (1,665/1,697) | 98.11% | (40,122/1,063,777) | 3.77% | (1,025,352/1,063,777) | 96.39% |

Table 7: The accuracy rate of detection classification.

| IDS method | Sybil attack | | Sinkhole | | Blackhole | | Hello flooding | |
|---|---|---|---|---|---|---|---|---|
| PIDS | (136/150) | 90.67% | (118/130) | 90.77% | (55/62) | 88.71% | (1,226/1,355) | 90.48% |
| OWIDS | (145/150) | 96.67% | (128/130) | 98.46% | (57/62) | 91.94% | (1,335/1,355) | 98.52% |

**Original Data**:
s -t 1.000000000 -Hs 8 -Hd -2 -Ni 8 -Nx 2032.00 -Ny 693.00 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw — -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 8.0 -Id 4.0 -It cbr -Il 1000 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
**SVM Input Data**:
0: 0.1 1: 1.000000000 2: 8 3: −2 4: 8 5: 2032.00 6: 693.00 7: 0.00 8: −1.000000 9: 1 10: 0 11: 0 12: 0 13: 0 14: 0 15: 8.0 16: 4.0 17: 1 18: 1000 19: 0 20: 0 21: 32 22: 1 23: 0 24: 0 25: 0 26: 0 27: 0 28: 0 29: 0 30: 0

Figure 9: The SVM input data.

accuracy of PIDS and OWIDS is higher than 89.61%, the tolerance value of the IDS in the WSN. The results of the second simulation show that the OWIDS detects attacks more effectively.

The attack packages include four types of attacks: Sybil attack, sinkhole attack, blackhole attack, and hello flooding; the detailed detection classification is shown in Table 7. The system simulates 150 Sybil attacks, 130 sinkhole attacks, 62 blackhole and 1355 hello flooding attacks. The results of accuracy rate are shown in Table 7. The accuracy rate of detection classification is better than 90% and OWIDS is better than PIDS. The results of the second simulation show that the OWIDS detects attacks more effectively.

## 5. Conclusions and Future Work

In this paper, a preliminary research of intrusion detection systems based on domain ontology is proposed and the relevance of a lightweight patrol intrusion detection system is explored. A major finding is that the effect of ontology can be observed in attack detection at all levels of the wireless sensor network. The results indicated that the constructed ontology relationship between the WSNs can detect attacks effectively. This implies that an ontology indicating each role and its membership in the WSN could be constructed for other attack types. This research leads to an ontology-based intrusion detection system which allows us to study the relationship mechanism involving patrol node status and sensor nodes. Such ontologies can also be applied to reduce the burden of lightweight intrusion detection systems on wireless sensor networks. In general, they will be useful in improving the lifecycle of wireless sensor networks and, particularly, the usability of intrusion detection systems for wireless sensor networks. This research can also serve to reinforce the use of soft computing technology for intrusion detection systems and to systematize preprocessing technology to reduce the features of the intrusion detection system. The attacker may intrude network in preprocessing stage. The preprocessing security issue will be solved in the future work. This ontology-based intrusion detection system remains experimental only and much work remains to be done. More must be known about constructing ontologies to detect different types of attacks in wireless sensor networks. There is a continuing need for an adequate theoretical basis for the practical application of ontology-based intrusion detection systems.

## Conflict of Interests

In this paper, the specifications and brand of CPU do not have any financial relationship with the commercial identities.

## Acknowledgments

This work is partially supported by the National Science Council (NSC), Taiwan, with Project no.: NSC-99-2221-E-324-021. The authors also express many thanks for the comments from the reviewers that improved the paper.

## References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] T. P. Hong and C. H. Wu, "An improved weighted clustering algorithm for determination of application nodes in heterogeneous sensor networks," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 2, pp. 173–184, 2011.

[3] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: analysis & defenses," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 259–268, April 2004.

[4] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," *Computers and Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005.

[5] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[6] A. N. Toosi and M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers," *Computer Communications*, vol. 30, no. 10, pp. 2201–2212, 2007.

[7] A. Patwardhan, J. Parker, M. Iorga, A. Joshi, T. Karygiannis, and Y. Yesha, "Threshold-based intrusion detection in ad hoc networks and secure AODV," *Ad Hoc Networks*, vol. 6, no. 4, pp. 578–599, 2008.

[8] R.-C. Chen and C.-H. Hsieh, "Web page classification based on a support vector machine using a weighted vote schema," *Expert Systems with Applications*, vol. 31, no. 2, pp. 427–435, 2006.

[9] G. Song, J. Zhang, and Z. Sun, "The research of dynamic change learning rate strategy in BP neural network and application in network intrusion detection," in *Proceedings of the 3rd International Conference on Innovative Computing Information and Control (ICICIC '08)*, pp. 514–517, June 2008.

[10] R. C. Chen, C. J. Yeh, and C. T. Bau, "Merging domain ontologies based on the WordNet system and Fuzzy Formal Concept Analysis techniques," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1908–1923, 2011.

[11] R. C. Chen, Y. F. Haung, and C. F. Hsieh, "Ranger intrusion detection system for wireless sensor networks with sybil attack based on ontology," in *Proceedings of the 10th WSEAS International Conference on Applied Informatics and Communications*, 2010.

[12] M. A. Simplício Jr., P. S. L. M. Barreto, C. B. Margi, and T. C. M. B. Carvalho, "A survey on key management mechanisms for distributed Wireless Sensor Networks," *Computer Networks*, vol. 54, no. 15, pp. 2591–2612, 2010.

[13] M. Depren, E. Topallar, and M. Kemal, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," *Expert Systems with Applications*, vol. 29, no. 4, pp. 713–722, 2005.

[14] S. Tilak, B. Abu-Ghazaleh, and W. A. Heinzelman, "Taxonomy of wireless micro-sensor network models," *Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28–36, 2002.

[15] J. Undercoffer, A. Joshi, and J. Pinkston, "Modeling computer attacks: an ontology for intrusion detection," *Computer Science*, vol. 2820, pp. 113–135, 2003.

[16] N. Cuppens-Boulahia, F. Cuppens, F. Autrel, and H. Debar, "An ontology-based approach to react to network attacks," *International Journal of Information and Computer Security*, vol. 3, no. 3-4, pp. 280–305, 2009.

[17] H.-C. Hsieh, J.-S. Leu, and W.-K. Shih, "Reaching consensus underlying an autonomous local wireless sensor network," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 4, pp. 1905–1914, 2010.

[18] W. K. Lai, C.-S. Fan, and C.-S. Shieh, "Optimal cluster size for balanced power consumption in wireless sensor networks," *ICIC Express Letters*, vol. 6, no. 2, pp. 1471–1476, 2012.

[19] T. T. Quan, S. C. Hui, A. C. M. Fong, and T. H. Cao, "Automatic Fuzzy ontology generation for semantic Web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 6, pp. 842–856, 2006.

[20] R.-C. Chen, C.-F. Hsieh, and Y.-F. Huang, "An isolation intrusion detection system for hierarchical wireless sensor networks," *Journal of Networks*, vol. 5, no. 3, pp. 335–342, 2010.

[21] C. F. -Hsieh, K. F. Cheng, Y. F. Huang, and R. C. Chen, "An intrusion detection system for Ad Hoc networks with multi-attacks based on a support vector machine and rough set theory," *Journal of Convergence Information Technology*, vol. 8, no. 2, pp. 767–776, 2013.