Detecting Energy Depriving Malicious Nodes by Unsupervised Learning in Energy Harvesting Cooperative Wireless Sensor Networks

Boqi Gao[1,a)] Daichi Amagata[1,b)] Takuya Maekawa[1,c)] Takahiro Hara[1,d)]

Abstract: This paper presents an unsupervised learning-based method for detecting energy depriving malicious nodes in an energy harvesting cooperative wireless sensor network (EHC-WSN). In EHC-WSNs, nodes wirelessly transfer a portion of their energy to their neighboring nodes if their neighboring nodes lacks energy. An energy depriving malicious node may forge that it has little energy, thus it can deprive energy from its neighboring nodes. For detecting such malicious nodes, we utilize clustering method. In our method, each node first observes energy of neighboring nodes, then it utilizes this information to obtain data points for clustering. After the clusters are formed, each node decides the cluster of data points from malicious nodes and makes malicious node decision. We investigate the performance of our method and confirm that our method outperforms the baseline method in terms of detection accuracy and false detection rate.

1. Introduction

With the rapid development of IoT and edge computing technologies [35], wireless sensor networks (WSNs) have been receiving more and more interests due to their real-time monitoring and computing capability. The recent breakthrough in energy harvesting cooperation (EHC) technology [27] relieves the bottleneck of energy limitation in WSNs. Therefore, the concept of energy harvesting cooperative wireless sensor networks (EHC-WSNs) has come up and attracts attentions increasingly [10]. An EHC-WSN is a WSN where nodes can harvest energy from ambient environments (e.g., harvesting from solar energy [40] and vibration [5]) and transfer energy to other nodes.

Much effort has been devoted to extend the lifetime of EHC-WSNs. Most studies focus on optimizing the energy transferring [10] and designing energy-aware routing protocols [1] to enable a WSN to have longer lifetime. Meanwhile, few works focused on security issues of EHC-WSNs. For example, in the energy cooperative architecture, a node can obtain energy from its neighboring nodes if it lacks energy. A malicious node hence can claim that it lacks energy, even if it has enough energy. In this case, it can deprive energy from its neighboring nodes, and they lose their energy. This kind of energy depriving attack may destruct the network reliability and functionality, which triggers event losses. Even worse, such an attack may lead to severe risks, particularly for real-time and safety-critical applications, such as extreme weather monitoring [19], water quality monitoring [7], and forest fire alarming [34]. Therefore, a valuable insight should be offered into attacks and security issues in EHC-WSNs. In this paper, we focus on detecting energy depriving malicious nodes in EHC-WSNs.

1.1 Motivation

Numerous studies have figured out various attack models for malicious node and proposed classifier, rule, and encryption-based methods for detecting malicious nodes in a WSN environment [30], [33], [26]. However, these techniques cannot provide security with EHC-WSNs because they do not consider malicious nodes that harm energy harvesting and energy cooperation.

When developing a countermeasure for the energy depriving attack, we face two challenges. (i) The information about energy of a node is private data, and it cannot be known by other nodes. This is an inherent problem because a malicious node can easily claim that it has little energy without any risk. Some energy-aware routing protocols demand nodes to report their current status of the energy storage periodically or add the status of energy storage to header [17], [39]. However, malicious nodes can still ignore these settings and pretend to have little energy. We may be able to design a rule-based method that decides nodes that always claim to have low energy as malicious nodes. However, this is impractical because of the second challenge that (ii) the energy harvesting efficiency of each node in EHC-WSN is different. For example, in forest fire alarming EHC-WSN where nodes harvest solar energy, the movements of the sun and clouds will result in shadows over some nodes. Their harvesting efficiencies would be low and demand energy. A rule-based method that simply decides

[1] Department of Multimedia Engineering Graduate School of Information Science and Technology Osaka University 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
[a)] gao.boqi@ist.osaka-u.ac.jp
[b)] amagata.daichi@ist.osaka-u.ac.jp
[c)] maekawa@ist.osaka-u.ac.jp
[d)] hara@ist.osaka-u.ac.jp

nodes with low energy as malicious nodes thereby cannot deal well with this problem. Moreover, any classifier-based methods are not suitable for this situation because there is no labeled data for training to build a classifier. Consequently, a well-designed energy depriving nodes detection method in EHC-WSNs is required.

1.2 Contribution

In this paper, we design a malicious node detection method based on unsupervised learning for EHC-WSNs, where energy depriving malicious nodes exist. Specifically, we make the following contributions.

- We tackle the problem of energy depriving nodes detection in EHC-WSNs online. To the best of our knowledge, we are the first to address this problem.
- We propose a deep neural network-based clustering method to detect energy depriving malicious nodes in EHC-WSNs.
- We propose a method to obtain data for the clustering, and propose inherent features to represent the energy depriving attack model.
- We conduct extensive experiments to investigate the performance of our method. Our experimental results demonstrate that our method outperforms the baseline method.

This paper is organized in the following way. Section 2 gives a brief overview of related works. Section 3 introduces our assumption in this paper. Section 4 presents our proposed method and experimental results are illustrated in Section 5. Finally, this paper is concluded in Section 6.

2. Related works

Energy harvesting and energy cooperation are promising methods for extending lifetime of WSNs. In [32], Raghunathan *et al.* firstly designed a solar energy harvesting wireless system. They proposed that wireless nodes can harvest from solar energy to relieve energy constraints. Inspired from their work, many studies suggested that nodes in WSNs can harvest from various ambient environment [5], [37]. Later on, with the rapid development of wireless transfer technology, the lifetime of WSN has been further extended. In [15], Huang *et al.* firstly enabled the wireless power transfer (WPT) in cellular networks. They designed the architecture, model, and deployment for WPT in cellular networks. As an application of wireless energy transfer in WSNs, Shi *et al.* [36] studied a scenario where nodes in a WSN charge their energy from a mobile charging vehicle wirelessly. Their experimental results proved that the lifetime of WSN can be extensively extended by wireless energy transfer. Then, Gurakan *et al.* [10] proposed a method that combines the energy harvesting and wireless energy transfer to create wireless energy harvesting cooperation (EHC) systems (EHC-WSNs). The work of Minasian *et al.* [27] further improved EHC systems by optimizing energy allocation. These studies proved that EHC-WSNs are promising; however, no work addressed security issues of EHC-WSNs.

For WSNs, a large number of studies have investigated various security issues. For a single kind of attack detection, Li *et al.* [23] developed a method that detects jamming attacks by enabling normal nodes to respond correctly to the jammers. For replica node attacks, Ho *et al.* [12] utilized a sequential analysis to identify abnormal nodes in a WSN. A few years later, considering security issues in wireless rechargeable sensor networks, Lin *et al.* [24] proposed a novel attack model, which is called cooperative denial of charging attack, to demonstrate that security in wireless rechargeable sensor networks needs to be further emphasized. However, they do not concern about the energy depriving attack.

Besides the countermeasure for a single kind of attack, many studies also aimed to provide a secure environment with WSNs. In [25], Liu *et al.* firstly proposed an application-friendly method to detect insider attackers in WSNs by monitoring many aspects of sensor networking behaviors. Hu *et al.* [14] presented an attack-tolerant time-synchronization for secure data aggregation in WSNs. However, these studies consider only a WSN that is unaware of energy issues. When energy issues are concerned, their methods become inapplicable. Hence, an energy depriving node detection method in EHC-WSN needs to be devised.

3. Assumption

3.1 Network model

In this paper, we assume an energy harvesting cooperative wireless sensor network (EHC-WSN) consisting of o wireless nodes with unique identifiers. These nodes can directly communicate with other nodes if they are within the communication range. We assume that all nodes have the same communication range, and if a given node is within the communication range of other nodes, it is a neighboring node of them.

As a routing protocol, AODV, which is a standard routing protocol in WSNs, is employed. That is, when a node `s` wants to send a data packet to another node `d` and it does not know the route, `s` broadcasts a route request (RReq) to create a packet transmission route, and this message is transmitted by some intermediate nodes. When node `d` receives the RReq, it sends a route reply (RRep) toward `s`, then `s` sends the data packet and it is transmitted through the route (see [31] for detail). Note that, nodes have to add their current status of energy storage in the headers of all the packets they send for energy cooperation.

3.2 Energy harvesting and energy cooperation

We assume that all nodes are capable of harvesting energy from ambient environments. Due to the random nature of ambient sources (e.g., shadows over solar energy harvesting panel), we assume each node harvests energy with different efficiency, which is the same as existing studies [6], [8]. All nodes have the same maximum energy storage, and cannot harvest more energy if the current energy storages reach the maximum energy storages.

Following the groundbreaking work of Zhang *et al.* [43], we assume nodes transfer information and energy simultaneously when they send packets. The amount of transferred energy is based on the water-filling algorithm [10]. That is, when a node `s` transfers energy to a node `d`, it aims to balance the energy storage of them. We also assume that when energy is transferred between nodes, a particular amount of the energy is lost because of the power loss [29]. Let E_s and E_d denote the status of energy storage of node `s` and node `d`, respectively. Assume that E_s is larger

than E_d, and let E_{tr} and E_{re} denote the energy transfered from node s and received by node d, respectively. Let λ denote the energy transferring efficiency:

$$E_{re} = \lambda \cdot E_{tr}. \quad (1)$$

Consequently, in order to keep the balance of status of energy storage after energy transferring, E_{tr} is calculated as:

$$E_{tr} = \frac{E_s - E_d}{1 + \lambda}. \quad (2)$$

3.3 Attack model

In the energy depriving attack in EHC-WSN, a malicious node pretends to have less energy level than its real energy level before it sends a packet. Recall that the status of energy storage should be included in the header.

4. Proposed method

In this section, we describe our proposed method for detecting energy depriving nodes in EHC-WSNs. This method, which is based on unsupervised learning techniques, utilizes a clustering method to detect malicious neighboring nodes.

A large number of studies, e.g., [3], [16], [41], have demonstrated that, compared with other clustering methods, deep neural network-based clustering methods have better performance due to the theoretical function approximation properties [13] and their feature learning capabilities [2]. Therefore, we utilize a deep neural network-based clustering method. Note that, in general, the task of clustering is to divide a set of data points into some clusters. In our method, each normal node, playing the roll as an observing node, first prepares data points for clustering by observing its neighboring nodes. Then, observing nodes utilize the data points to form clusters. After the clustering, observing nodes utilize the clustering results to decide malicious nodes.

4.1 Preparation of data points for clustering

In this section, we describe how normal nodes prepare data points for clustering. We assume that a node clusters its neighboring nodes at time slot T.

As mentioned, nodes have to add their current status of energy storage in the headers. Each normal node thus can observe the status of energy storage of its neighboring nodes by overhearing their packets. At each time slot, normal nodes create features from observed energy. These features are hereinafter called *energy features*. At each time slot, an energy feature vector of each neighboring node is created. Therefore, at time slot T, a normal node has T energy feature vectors for each of its neighboring nodes.

Fig. 1 illustrates the energy feature vector set extraction procedure of neighboring nodes by node a. Hereinafter, we use the term *original feature vector set* to denote the set of T vectors of a neighboring node. It is important to note that the energy feature vector set of each neighboring node is time-series data because it is obtained along with time.

Recall that our approach is to cluster the neighboring nodes of each normal node. We have already obtained the original feature vector set of each neighboring node by the above procedure.

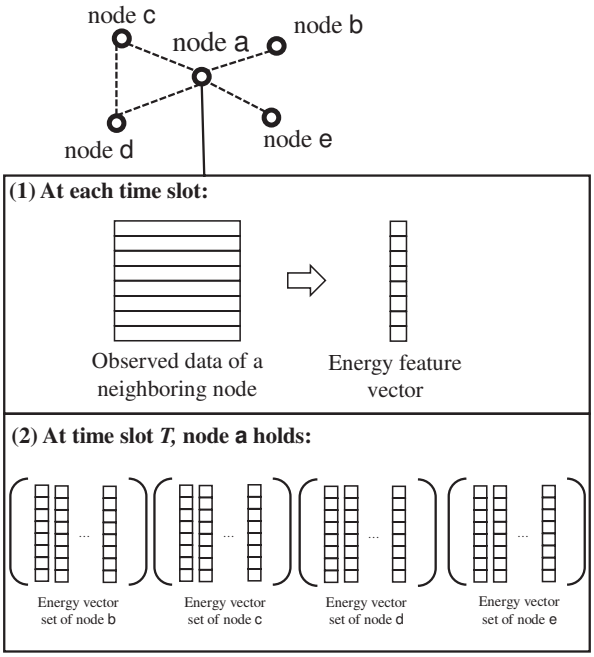

Fig. 1 An example of energy feature vector set creation. (1) At each time slot, node a creates an energy feature vector of each neighboring node. (2) At time T, for each neighboring node, node a holds an energy vector set which contains T energy feature vectors.

However, because we utilize a deep neural network-based clustering method, it is impractical to treat an original feature vector set as a data point for clustering. The reason is that the number of energy feature vector sets is usually small (the number of energy feature vector sets is equal to the number of neighboring nodes). It is clear that less information makes machine learning approaches not functional and easy to overfit [4]. We hence need more data points to enable the clustering method functional.

To deal with this problem, we propose a method that creates more data points for clustering. Instead of using an original feature vector set as a data point for clustering, we use subsets of it. Let k denote the size of a subset. Our method extracts vectors between $sk + 1$-*th* vector to $(s + 1)k$-*th* vector from each vector set to form the s-*th* subsets. In each subset, the vectors are still time-series data. Our method thus maintains the properties of time-series in each subset. Fig. 2 shows an example of our method. In this example, we create 2 subsets with size 5 from the original feature vector set with size 10. Then, we treat these subsets as data points for clustering. That is, a subset is a data point for future clustering.

4.2 Clustering method

As mentioned above, we utilize a deep neural network-based clustering method to cluster the above-mentioned subsets. In particular, we re-organize the method proposed in [41], which is a standard deep neural network-based clustering method, called deep embedding clustering (DEC), to obtain our clustering method.

We first make a brief introduction of DEC. DEC is a method that simultaneously learns feature representations and cluster assignments using deep neural networks. DEC learns a mapping

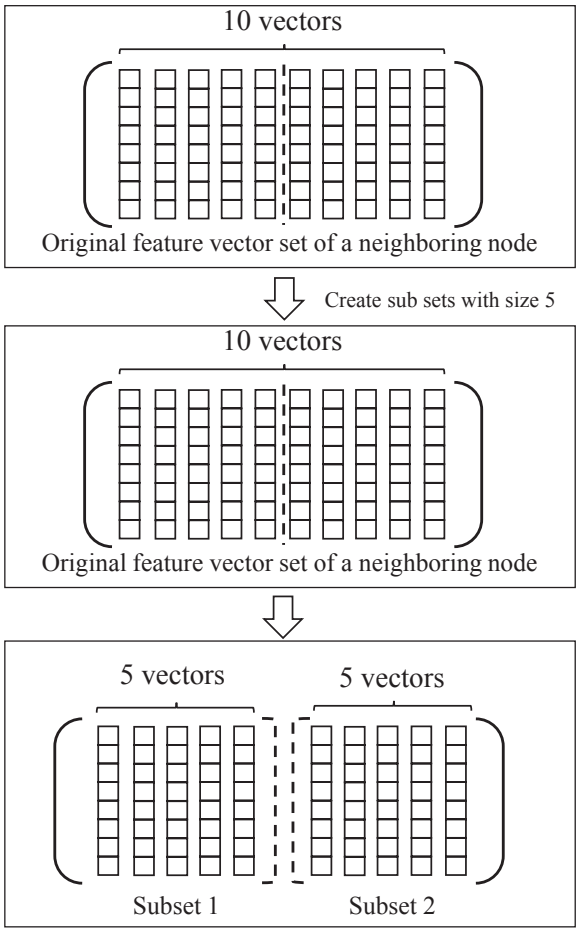

Fig. 2 An example of creating subsets. Two subsets with size 5 are created from the original feature vector set with size 10.

from the data space to a lower-dimensional feature space in which it iteratively optimizes a clustering objective. In general, the task of clustering is to cluster a set X of x points into y clusters. Instead of clustering directly in the *data space* X, DEC aims to transform the data with a non-linear mapping $f_\theta : X \rightarrow Y$, where θ is a learn-able parameter, which is parametrized by deep neural networks, and Y is the latent cluster space. Finally, DEC can learn the parameters of the deep neural networks, and the deep neural networks represent a clustering model of $X \rightarrow Y$ (see more details in [41]).

The original model of DEC only employs fully connected dense layers, which are not good at processing time-series data [42]. Recall that our subsets for clustering are time-series data. We thus re-organize the structure of deep neural networks of DEC by adding 1D-convolutional layers and pooling layers to process time-series data better, which is inspired by existing studies [21], [42]. To avoid the over fitting, we discard some fully connected layers of the original DEC structure. Fig. 3 shows our re-organization to original DEC.

Note that our clustering method can set the number of clusters as a hyper-parameter. We thus set the number of clusters as two because we assume two categories of nodes (normal and malicious nodes).

4.3 Energy features

Here we describe the energy features, which are used in malicious node detection. We assume that each node observes the status of energy storage of its neighboring node. Before we describe the energy features, we present the information about energy related to neighboring nodes observed by each node (Table 1). This is used to compute the energy features. To enable normal nodes to have better understandings of neighboring nodes, we design short-term and long-term features. Short-term features enable normal nodes to recognize instant status of energy storage of themselves and their neighboring nodes, and long-term features let normal nodes understand the historical behaviors about the energy of their neighboring nodes. In particular, we directly employ the current status of energy storages of the observing node and the neighboring node as two short-term features. They are utilized as energy features because (i) they can directly represent the status of energy storage at that time slot, and (ii) they both have the maximum and minimum values, which are proper as inputs of the deep neural networks, because they can be normalized smoothly.

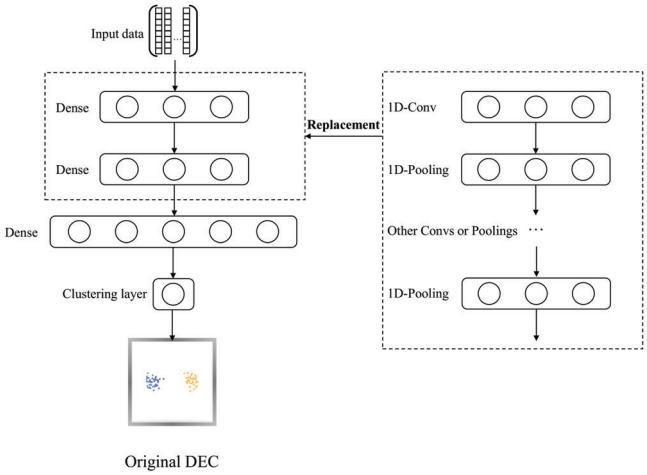

Fig. 3 Re-organization of the deep neural network structure of DEC by replacing some dense layers with convolutional and pooling layers of dense layers to processing time-series data

Table 2 shows the long-term energy features used in our method, and they are designed based on the inherence of the energy depriving attack. These long-term energy features are all ratios calculated from the information observed by nodes. Such ratios are more robust against the difference in node density than numbers, because it is clear that the number of neighboring nodes can easily influence features based solely on numbers.

Here, we take TransferRatio (energy transferred ratio) and StorageRatio (energy storage ratio) as two examples to illustrate the reason why our energy features can describe the inherence of energy depriving attack. Assume that a normal node a observes its neighboring node b. (i) TransferRatio: this feature of b can help node a to measure how much energy is transferred to node b compared with all its transferred energy. If node b is a malicious node, a transfers comparably more energy to node b than the other neighboring nodes because it claims to have lower energy. (ii) StorageRatio: this feature measures the ratio of the current status of energy storage of nodes b and a. If node b is a malicious node, this feature value would be comparably lower than the feature values of the other neighboring nodes, because

Table 1 Information about energy observed by each node

Information	Definition
NTra	Energy transferred to one neighbor
NRec	Energy received from one neighbor
TTra	Total energy transfered
TRec	Total energy received
SS	Current status of energy storage of observing node
SN	Current status of energy storage of one neighbor
TH	Total energy harvested

Table 2 Long-term energy features

Behavioral features	Definition
TransferRatio	NTra / TTra
ReceiveRatio	NRec / TRec
CompensateRatio	NTra / TRec
IncomeRatio	NRec / TTra
GiveRatio	NTra / TH
VoluntaryRatio	TTra / TH
StorageRatio	SS / SN
DeliveryRatio	TTra / TH

node b forges its status of energy storage to a lower value. Therefore, a can utilize these two features to cluster b as a malicious node. Other long-term energy features are also designed based on the same intuition.

It is important to note that the energy features are obtained from messages sent by neighboring nodes. In other words, each normal node can obtain the energy features by overhearing messages, so our method incurs no additional communication cost.

4.4 Malicious node detection

In this section, we describe our method for determining the categories of the two clusters and deciding malicious nodes. After two clusters are formed, an observing node needs to determine the categories of them, i.e., which one contains the subsets of malicious nodes. Then, based on the result, an observing node decides malicious nodes.

Cluster of malicious subsets decision. We assume that a normal node a in a given EHC-WSN forms clusters of subsets of its neighboring nodes. After two clusters are formed, we calculate the average of feature SN (see Table 1) for both the clusters:

$$SN_{ave} = \frac{\sum_{i=0}^{N}\sum_{j=0}^{k} SN}{N \cdot k}, \tag{3}$$

where N denotes the number of subsets from a cluster.

Then, the cluster with lower SN_{ave} is decided as a cluster that contains subsets from malicious nodes. This is because of the inherence of the energy depriving malicious nodes, i.e., they forge to have little energy.

Malicious node decision. After the categories of two clusters are obtained, node a decides the malicious neighboring nodes. Assume that node a holds n subsets of each neighboring node, and m subsets of a particular neighboring node are in the cluster with malicious subsets. Let $\hat{y}$ denote the result of malicious node decision determined by node a for this neighboring node. $\hat{y}$ is obtained as follows:

$$\hat{y} = \begin{cases} normal & (m < \frac{n}{2}) \\ malicious & (otherwise) \end{cases}$$

We set $\frac{n}{2}$ as a decision threshold because during the clustering procedure, all the subsets are obtained from the same node (node a). As a result, in the malicious node decision procedure, there is no difference of the weights of each subset. When $m = \frac{n}{2}$, this neighboring node is identified as a malicious node because detecting malicious nodes is considered as more important than misdetecting normal nodes. Note that our detection method detects malicious nodes online. Therefore, the malicious node detection procedure does not require labeled data from a certain network environment.

5. Experiment

This section summarizes our experiments that evaluate the performance of the proposed method.

5.1 Setting

We used the Qualnet 7.4 network simulator[*1], and we set our experiments similarly to previous studies [24], [6], [29], [10].

We randomly deployed 500 nodes in a $100m \times 100m$ square field, and 100 nodes were malicious nodes. Each node transmitted messages and data packets with a payload sizes of 256 bytes, using an IEEE 802.11b device. The communication range of each node was adjusted to roughly 3 meters, and the network bandwidth was 11Mbps. The maximum energy of each node was $100mJ$. We decided a time slot as 4 seconds, and a node harvested energy per time slot randomly from $0.01mJ$ to $1mJ$ [6]. The energy consumption of transmitting and receiving a packet were $0.1mJ$ and $0.08mJ$, respectively. We considered a random destination node scenario [20] because the development of edge computing enables each sensor node to process data [35]. We randomly chose a pair of source node and destination node every 1 seconds. If the source node has an active route to the destination node, the source node sends a data packet to the destination node directly. Otherwise the source node broadcasts an RReq to find a route to the destination node. When a node forwards a data packet through a neighboring node, if the status of energy storage of this neighboring node is less than the this node, this node transfers energy to this neighboring node simultaneously (see Section 3.2). The simulation time was 4000 second, which means a simulation consisted of 1000 time slots, and the subset size k is 10. Similarly to [29], we set λ as 0.29.

Malicious nodes. In our assumption, malicious nodes forged to have little energy (Section 3.3). However, if malicious nodes always forged its status of energy storage to an extreme low value (e.g., 0), it is clear that these malicious nodes can be detected easily. We thus added noises to their forged amount of energies. In particular, let E_{real} and E_{forged} denote the real status of energy storage of a malicious node and the forged status of energy storage, respectively. To obtain E_{forged}, this malicious node deducts E_{real} by a Gaussian white noise [22] denoted by E_{gwn} as:

$$E_{forged} = E_{real} - E_{gwn}. \tag{4}$$

Then, the malicious node will add E_{forged} into the header when it sends a packet. The mean of E_{gwn} is 0.1, which is same as energy consumption of transmitting a packet, and the standard deviation is 0.05.

[*1] http://web.scalable-networks.com/qualnet-network-simulator-software

Evaluation methods. To investigate the effectiveness of the proposed method, we prepared the following methods.

- *K-means*: This method clusters the original energy vector set into two clusters by K-means [11], which is a standard method for clustering. After two clusters are formed, each node decides the cluster with lower average SN as a cluster that contains malicious energy vector set. Then, the neighboring nodes holds the original energy vector set of the malicious cluster are decided as malicious nodes. We prepare this method to investigate the effectiveness of creating subsets.
- *WSK-means*: This method clusters subsets into two clusters by K-means. We prepare this method to investigate the effectiveness of our deep neural network-based clustering method.
- *Proposed*: This is the proposed method in this paper.
- *WSDEC*: This method cluster subsets into clusters by the original DEC [41]. Therefore, this method does not have convolutional and pooling layers in the deep neural network model. We prepare this method to investigate the effectiveness of our method to handle time-series data.

All data obtained during the simulation time were used to compute energy features. We assumed that nodes executed a neighboring node detection procedure every 100 time slots.

Implementation. We implemented our deep neural network set on Keras 2.2.4 [*2] with TensorFlow [*3] as backend. Determining hyper-parameters by cross-validation on a validation set is not an option in unsupervised clustering because we do not have labeled data. Thus we use commonly used parameters for deep neural networks. In particular, inspired by [38], we set network dimensions of $WSDEC$ to d-25-25-100-10, where d is the original data-space dimension determined by the subset size k and the number of time slots. All layers are densely (fully) connected. For our proposed method, we set network dimensions to d-Conv1D(25, 10)-Conv1D(25, 10)-MaxPooling1D(4)-Conv1D(50, 10)-Conv1D(50,10)-GlobalAveragePooling()-100-10, where Conv1D denotes a 1-dimension convolutional layer. The number of clusters are set as 2, because we have two categories of nodes.

Criteria. As mentioned earlier, our method incurs no additional communication costs. We therefor focus on the following criteria to measure the performance of the above methods.

- Accuracy: This is represented by $\frac{T_{nor\to nor,mal\to mal}}{T}$, where $T_{nor\to nor,mal\to mal}$ and T are respectively the set of correctly decided neighboring nodes of all normal nodes and the set of all neighboring nodes of all normal nodes.
- Detection rate: This is represented by $\frac{T_{mal\to mal}}{T_{mal}}$, where $T_{mal\to mal}$ and T_{mal} are respectively the set of correctly decided malicious neighboring nodes of all normal nodes and the set of all malicious neighboring nodes of all normal nodes.
- Mis-Detection rate: This is represented by $\frac{T_{nor\to mal}}{T_{nor}}$, where $T_{nor\to mal}$ and T_{nor} are respectively the set of wrongly decided normal neighboring nodes of all normal nodes and the set of all normal neighboring nodes of all normal nodes.

[*2] https://keras.io/
[*3] https://www.tensorflow.org/

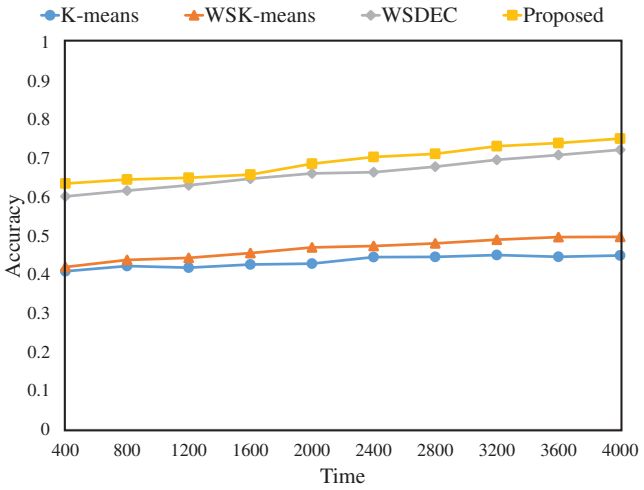

Fig. 4 Accuracies of four methods

5.2 Result

Comparison with K-means and WSK-means. Figs. 4–6 show the performances of our method (Proposed), and the performances of K-means and WSK-means. In particular, K-means is about averagely 25% worse than that of Proposed with regard to the three criteria. This is because (i) K-means uses the original vector set, (ii) compared with K-means, deep neural network-based clustering method has better feature learning capabilities, and (iii) our data points are not balanced (the data points from normal nodes is much more than those of malicious nodes), and K-means is not good to handle unbalanced data [18]. Compared with K-means, WSK-means is about averagely 5% better with regard to the three criteria. This is because WSK-means clusters subsets, and more data points provide a better performance.

Comparison with WSDEC. Figs. 4–6 show the performances of our method (Proposed), and the performances of WSDEC. We can see that Proposed is averagely 4% better than that of WSDEC with regard to the three criteria. This is because subsets are time-series data, and Proposed utilizes convolutional layers to better process time-series data.

Influence of simulation time. From Figs. 4–6, we can see that the performances of all the methods become better as the time spends. This is because longer time can generate larger amount of data points for clustering. It is clear that clustering methods can work better with more data points for obtaining a good result.

Information gain of energy features. Table 3 shows the information gain of energy features. The information gain is used to find distinguishing features of feature vectors. The more information gain of a feature increases, the better the feature classifies the vector. We obtained the information gain by adding ground truth label to each energy feature vector. For example, if a neighboring node is malicious, the labels of energy vectors observed from it are malicious. From this table, we can find that the information gain of our designed features are all over 0.009. Compared with other classifier based studies for detecting malicious nodes in WSNs [28], [9], our energy features have competitive or better information gain. We bold top-3 information gain. This result shows that our designed features are effective and can be employed for detecting energy depriving attack in EHC-WSNs.

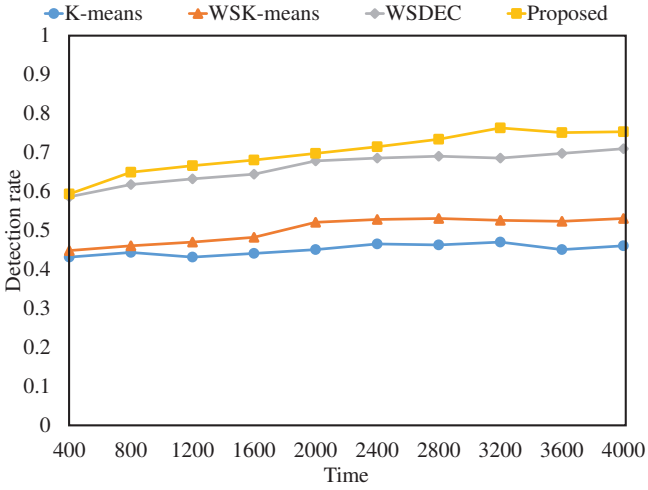

Fig. 5 Detection rates of four methods

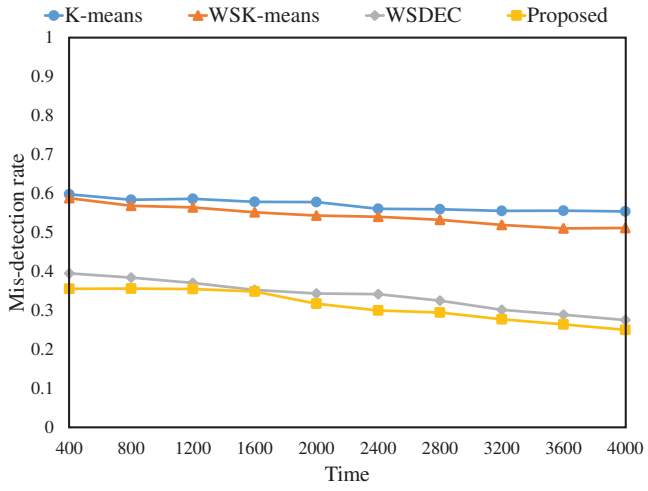

Fig. 6 Mis-detection rates of four methods

Table 3 Information gain of energy features

Energy features	Info. gain
SS	0.009
SN	0.012
TransferRatio	**0.134**
ReceiveRatio	**0.103**
IncomeRatio	0.026
GiveRatio	0.019
VoluntaryRatio	0.011
StorageRatio	0.023
DeliveryRatio	0.014
CompensateRatio	**0.091**

6. Conclusion

This paper presented an energy depriving malicious node detection method for EHC-WSNs. We designed inherent features of energy depriving attack. We proposed a method for obtaining more data points for clustering, as well as a deep neural network based clustering method. The experiments revealed that the proposed method outperformed the comparison methods.

References

[1] Anisi, M. H., Abdul-Salaam, G., Idris, M. Y. I., Wahab, A. W. A. and Ahmedy, I.: Energy harvesting and battery power based routing in wireless sensor networks, *Wireless Networks*, Vol. 23, No. 1, pp. 249–266 (2017).

[2] Bengio, Y., Courville, A. and Vincent, P.: Representation learning: A review and new perspectives, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 35, No. 8, pp. 1798–1828 (2013).

[3] Caron, M., Bojanowski, P., Joulin, A. and Douze, M.: Deep clustering for unsupervised learning of visual features, *ECCV*, pp. 132–149 (2018).

[4] Caruana, R., Lawrence, S. and Giles, C. L.: Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping, *NIPS*, pp. 402–408 (2001).

[5] Chamanian, S., Baghaee, S., Uluşan, H., Zorlu, Ö., Uysal-Biyikoglu, E. and Külah, H.: Implementation of Energy-Neutral Operation on Vibration Energy Harvesting WSN, *IEEE Sensors Journal* (2019).

[6] Chen, Q., Gao, H., Cai, Z., Cheng, L. and Li, J.: Energy-collision aware data aggregation scheduling for energy harvesting sensor networks, *IEEE INFOCOM*, pp. 117–125 (2018).

[7] Du, W., Xing, Z., Li, M., He, B., Chua, L. H. C. and Miao, H.: Sensor placement and measurement of wind for water quality studies in urban reservoirs, *ACM TOSN*, Vol. 11, No. 3, pp. 41:1–41:7 (2015).

[8] Elvin, N. and Erturk, A.: *Advances in energy harvesting methods*, Springer Science & Business Media (2013).

[9] Gao, B., Maekawa, T., Amagata, D. and Hara, T.: Environment-Adaptive Malicious Node Detection in MANETs with Ensemble Learning, *IEEE ICDCS*, pp. 556–566 (2018).

[10] Gurakan, B., Ozel, O., Yang, J. and Ulukus, S.: Energy cooperation in energy harvesting communications, *IEEE TCOM*, Vol. 61, No. 12, pp. 4884–4898 (2013).

[11] Hartigan, J. A. and Wong, M. A.: Algorithm AS 136: A k-means clustering algorithm, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 28, No. 1, pp. 100–108 (1979).

[12] Ho, J.-W., Wright, M. and Das, S. K.: Fast detection of replica node attacks in mobile sensor networks using sequential analysis, *IEEE INFOCOM*, pp. 1773–1781 (2009).

[13] Hornik, K.: Approximation capabilities of multilayer feedforward networks, *Neural networks*, Vol. 4, No. 2, pp. 251–257 (1991).

[14] Hu, X., Park, T. and Shin, K. G.: Attack-tolerant time-synchronization in wireless sensor networks, *IEEE INFOCOM*, pp. 41–45 (2008).

[15] Huang, K. and Lau, V. K.: Enabling wireless power transfer in cellular networks: Architecture, modeling and deployment, *IEEE TWC*, Vol. 13, No. 2, pp. 902–912 (2014).

[16] Huang, P., Huang, Y., Wang, W. and Wang, L.: Deep embedding network for clustering, *ICPR*, pp. 1532–1537 (2014).

[17] Jakobsen, M. K., Madsen, J. and Hansen, M. R.: DEHAR: A distributed energy harvesting aware routing algorithm for ad-hoc multi-hop wireless sensor networks, *IEEE WoWMoM*, pp. 1–9 (2010).

[18] Jianliang, M., Haikun, S. and Ling, B.: The application on intrusion detection based on k-means cluster algorithm, *IEEE International Forum on Information Technology and Applications*, Vol. 1, pp. 150–152 (2009).

[19] Kanagaraj, E., Kamarudin, L., Zakaria, A., Gunasagaran, R. and Shakaff, A.: Cloud-based remote environmental monitoring system with distributed WSN weather stations, *IEEE SENSORS*, pp. 1–4 (2015).

[20] Khan, M. I., Gansterer, W. N. and Haring, G.: Static vs. mobile sink: The influence of basic parameters on energy efficiency in wireless sensor networks, *Computer communications*, Vol. 36, No. 9, pp. 965–978 (2013).

[21] LeCun, Y., Bengio, Y. et al.: Convolutional networks for images, speech, and time series, *The handbook of brain theory and neural networks*, Vol. 3361, No. 10, p. 1995 (1995).

[22] Lepskii, O.: On a problem of adaptive estimation in Gaussian white noise, *Theory of Probability & Its Applications*, Vol. 35, No. 3, pp. 454–466 (1991).

[23] Li, M., Koutsopoulos, I. and Poovendran, R.: Optimal jamming attacks and network defense policies in wireless sensor networks, *IEEE INFOCOM*, pp. 1307–1315 (2007).

[24] Lin, C., Shang, Z., Du, W., Ren, J., Wang, L. and Wu, G.: CoDoC: A Novel Attack for Wireless Rechargeable Sensor Networks through Denial of Charge, *IEEE INFOCOM*, pp. 856–864 (2019).

[25] Liu, F., Cheng, X. and Chen, D.: Insider attacker detection in wireless sensor networks, *IEEE INFOCOM*, pp. 1937–1945 (2007).

[26] Mansouri, D., Mokdad, L., Ben-Othman, J. and Ioualalen, M.: Detecting DoS attacks in WSN based on clustering technique, *IEEE WCNC*, pp. 2214–2219 (2013).

[27] Minasian, A., ShahbazPanahi, S. and Adve, R. S.: Energy harvesting cooperative communication systems, *IEEE TWC*, Vol. 13, No. 11, pp. 6118–6131 (2014).

[28] Mitrokotsa, A. and Dimitrakakis, C.: Intrusion detection in MANET using classification algorithms: The effects of cost and model selection, *Ad Hoc Networks*, Vol. 11, No. 1, pp. 226–237 (2013).

[29] Park, C., Lee, S., Cho, G.-H. and Rim, C. T.: Innovative 5-m-

off-distance inductive power transfer systems with optimally shaped dipole coils, *IEEE TPE*, Vol. 30, No. 2, pp. 817–827 (2014).
[30] Pathan, A.-S. K.: *Security of self-organizing networks: MANET, WSN, WMN, VANET*, CRC press (2016).
[31] Perkins, C. E. and Royer, E. M.: Ad-hoc On-Demand Distance Vector Routing, *IEEE WMCSA*, pp. 90–100 (1999).
[32] Raghunathan, V., Kansal, A., Hsu, J., Friedman, J. and Srivastava, M.: Design considerations for solar energy harvesting wireless embedded systems, *ACM IPSN*, p. 64 (2005).
[33] Raymond, D. R. and Midkiff, S. F.: Denial-of-service in wireless sensor networks: Attacks and defenses, *IEEE PerCom*, No. 1, pp. 74–81 (2008).
[34] Saoudi, M., Bounceur, A., Euler, R. and Kechadi, T.: Data mining techniques applied to wireless sensor networks for early forest fire detection, *ACM CCIOT*, pp. 71:1–71:7 (2016).
[35] Shi, W., Cao, J., Zhang, Q., Li, Y. and Xu, L.: Edge computing: Vision and challenges, *IEEE Internet of Things Journal*, Vol. 3, No. 5, pp. 637–646 (2016).
[36] Shi, Y., Xie, L., Hou, Y. T. and Sherali, H. D.: On renewable sensor networks with wireless energy transfer, *IEEE INFOCOM*, pp. 1350–1358 (2011).
[37] Tan, Y. K. and Panda, S. K.: Energy harvesting from hybrid indoor ambient light and thermal energy sources for enhanced performance of wireless sensor nodes, *IEEE TIE*, Vol. 58, No. 9, pp. 4424–4435 (2010).
[38] Van Der Maaten, L.: Learning a parametric embedding by preserving local structure, *AiStats*, pp. 384–391 (2009).
[39] Veerayya, M., Sharma, V. and Karandikar, A.: SQ-AODV: A novel energy-aware stability-based routing protocol for enhanced QoS in wireless ad-hoc networks, *IEEE MILCOM*, pp. 1–7 (2008).
[40] Wang, C., Li, J., Yang, Y. and Ye, F.: A hybrid framework combining solar energy harvesting and wireless charging for wireless sensor networks, *IEEE INFOCOM*, pp. 1–9 (2016).
[41] Xie, J., Girshick, R. and Farhadi, A.: Unsupervised deep embedding for clustering analysis, *International conference on machine learning*, pp. 478–487 (2016).
[42] Yang, J., Nguyen, M. N., San, P. P., Li, X. L. and Krishnaswamy, S.: Deep convolutional neural networks on multichannel time series for human activity recognition, *IJCAI* (2015).
[43] Zhang, R. and Ho, C. K.: MIMO broadcasting for simultaneous wireless information and power transfer, *IEEE TWC*, Vol. 12, No. 5, pp. 1989–2001 (2013).