Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 8672305, 18 pages
http://dx.doi.org/10.1155/2016/8672305

Research Article

Distributed Classification of Localization Attacks in Sensor Networks Using Exchange-Based Feature Extraction and Classifier

Su-Zhe Wang, Yong Li, and Wei Cheng

School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China

Correspondence should be addressed to Su-Zhe Wang; wangsuzhe@gmail.com

Received 5 August 2016; Revised 31 October 2016; Accepted 15 November 2016

Academic Editor: Alexandru Serbanati

Secure localization under different forms of attack has become an essential task in wireless sensor networks. Despite the significant research efforts in detecting the malicious nodes, the problem of localization attack type recognition has not yet been well addressed. Motivated by this concern, we propose a novel exchange-based attack classification algorithm. This is achieved by a distributed expectation maximization extractor integrated with the PECPR-MKSVM classifier. First, the mixed distribution features based on the probabilistic modeling are extracted using a distributed expectation maximization algorithm. After feature extraction, by introducing the theory from support vector machine, an extensive contractive Peaceman-Rachford splitting method is derived to build the distributed classifier that diffuses the iteration calculation among neighbor sensors. To verify the efficiency of the distributed recognition scheme, four groups of experiments were carried out under various conditions. The average success rate of the proposed classification algorithm obtained in the presented experiments for external attacks is excellent and has achieved about 93.9% in some cases. These testing results demonstrate that the proposed algorithm can produce much greater recognition rate, and it can be also more robust and efficient even in the presence of excessive malicious scenario.

1. Introduction

The location information of the sensor node performs a critical role for numerous applications in wireless sensor networks (WSNs) such as environment monitoring, target tracking, and automatic surveillance. It also helps some fundamental techniques in sensor networks (e.g., geographical routing protocol and topology control) to be aware of where the messages are located. Driven by those demands, earlier research efforts have resulted in many localization schemes, with most assuming that the sensors are deployed in a benign scenario. But when the sensor nodes are deployed in malicious environments, it is prone to different forms of threats and risks. A simple malicious attack can disturb the accurate position estimating and even make the entire network functioning improperly [1]. The existing attacks in localization can generally be divided into internal and external attack. Internal attackers usually are compromised nodes whose encryption key has been extracted, which can be prevented by using advanced cryptography techniques. The external attack is launched by one or more malicious nodes to distort the information without system's authorization, which means that traditional security mechanism like cryptography is limited to defend against this type of attack. In this paper, we will mainly analyze the recognition of the external attacks on the localization procedure.

In recent years, designing secure localization schemes that provide valid location information resistant to externals attacks has received much research attention [2–8]. Most of these secure location mechanisms can be broadly divided into two categories: cheating node detection and robust localization algorithms. The former such as [3, 5, 7] are characterized by verifying some location-related parameters like distance or time during positioning process to detect the inconsistency and then eliminating abnormal nodes, while the latter [2, 4, 6, 8] depend on designing robust localization schemes to tolerate attacks rather than detecting them.

Most existing works for WSN localization security focused on either achieving high detection ratio under different types of attacks or developing robust positioning

methods. Unfortunately, none of these techniques can explicitly differentiate those attacks. This may make the network defense fall into the passive situation and have a negative effect in preventing future repeated attacks. If the network only detects localization attacks without type classification and analyzing, the possible consequence can be implied as follows. One of the main results is that it is not convenient for network to restore location-related information. The other is that it could make the network difficult to provide more information services and evidence in security event processing. Only after alert information is collected and analyzed can we determine the dangerous region where attack frequently takes place and then design targeted localization scheme according to certain threat. Therefore, attack classification in localization is not only the premise and foundation of threat analysis, but also a crucial component in network security situation awareness. And attack recognition algorithm should be executed as second line of protection against attacks before the location information can be used by other applications.

In this work, we proposed a localization attack classification method based on the distributed expectation maximization algorithm followed by support vector machines called PECPR-MKSVM. The classification mechanism consists of two phases: the feature extraction phase and the classification phase. The techniques developed in our solutions offer the advantage of classifying various kinds of attacks. More specially, our approach possesses the following contributions.

(1) To extract more efficient attack features, an Exponential-Gaussian (EG) mixture distribution is firstly modeled by investigating the common properties of initial features based on their probability distribution. The initial features are composed of distance and topology-related measurements.

(2) A distributed version of expectation maximization (EM) algorithm which exchanges information with neighbor sensors is implemented for density estimation and feature extraction, where one term for time dependent information averaging is combined with another term for iterative information propagation.

(3) In order to recognize multiple attacks more accurately and adapt to the distributed characteristics of sensor networks, we design an exchange-based classifier called proximal extension contractive Peaceman-Rachford splitting-multiple kernel support vector machines (PECPR-MKSVM).

(4) To identify the effectiveness of our distributed recognition approach, comprehensive designed experiments are conducted by testing the attacks dataset under different conditions. Compared with other similar schemes, we find that the results obtained in these comparisons clearly show that the distributed classification algorithm achieves better recognition performance and has stronger robustness, with very competitive runtime.

The remainder of the paper is structured as follows. Some related works on secure localization and recognition algorithms are reviewed in the next section (Section 2). In Section 3, we describe the attack assumptions and model the initial features with a joint Exponential-Gaussian distribution, while Section 4 presents the distributed EM algorithm based feature extraction method by employing the distributed averaging approach. In Section 5, by improving the contractive Peaceman-Rachford splitting method algorithm, a novel distributed classifier PECPR-MKSVM is presented. In Section 6 we verify the performance of the classification algorithm by means of extensive experiments. Finally, some conclusions are devoted to Section 7.

2. Related Work

To investigate the scheme for classifying localization attack in WSN, a necessary literature survey on secure localization mechanism is firstly provided. Moreover, we provide a succinct summary of research on two essential components of the proposed method, that is to say, the EM algorithm for feature extraction and support vector machines for classifier.

2.1. Secure Localization Mechanism. In the prior work about the secure localization, one theme is able to discover and eliminate the suspicious nodes. In [9], the authors proposed a beacon-based securing localization method, which is also used by a Minimum Mean Square approach to filter out suspicious nodes. This work was implemented by observing the inconsistency in location references between the malicious beacon nodes and the benign ones [10]. Similarly, Du et al. [11] created a general scheme by using network deployment knowledge to detect localization anomalies if the level of inconsistency in expectations of the derived positions exceeds the certain threshold. Recently, another detection-based secure localization algorithm by Han et al. was proposed, which has two steps. The anchor nodes first identified the suspicious node if it sends abnormal reference information. Then a mesh generation method was developed to separate suspicious nodes [12].

The other theme is an error-intolerant localization when there exist malicious adversaries and great measurement inaccuracy. Li et al. in [13] employ an improved LM approach to achieve the goal of securing localization in a scenario where the fraction of the malicious nodes is less than 50%. Based on candidate locations identifying, a similar method called random sample consensus (RANSAC) algorithm was proposed in [14]. This method used picked subsets of sensors to detect and choose the value which minimizes the median of the remains as its solution. Alternatively, by using the Taylor-series least squares scheme with different weighting, Yu et al. developed two-stage secure localization method which applied beta distribution function to tolerate the presence of malicious beacons [15]. Some other approaches try to realize the secure location estimation by expressing it as a global optimization problem. For instance, by taking advantage of improved least median squares, a robust statistical method was developed to make positioning attack tolerant. In [16], Doherty et al. designed a feasible secure localization methodology using convex

optimization based on pairwise angles and connectivity between nodes. Bao et al. extended the work from static to mobile scenario with the help of a game-based strategy [17].

According to our current knowledge, the problem of localization attacks recognition for sensors network, which is our focus here, has not been well studied.

2.2. EM Algorithm for Feature Extraction. Unsupervised feature selection/extraction techniques are generally classified into three categories as wrappers, filters, and integrated-learning approaches. Several integrated-learning feature extraction algorithms like EM have been developed in various fields. In [18], the features were extracted from the continuous-valued dataset by using a primary integrated-learning strategy. In another algorithm of feature extraction, the feature saliency is firstly regarded as relevant features, and the pruning behavior is then outlined by using EM optimization. Moreover, a double-loop EM algorithm was applied in medical detection such as epileptic seizure so that the supervised learning could fit well with the mixture of experts network structure [19]. In [20], EM algorithm was applied in image feature extraction to identify parameters for generalized Gaussian mixture model. Subsequently, a Kullback divergence-based similarity measure was presented and analyzed. However, the fact that class of texture distribution is under the influence of its neighborhood is neglected. To address the issues of information loss, a shuffled frog-leaping method is added to the EM algorithm to enhance the performance of crack image segmentation [21]. According to an evaluation threshold φ, neighborhood of each pixel was classified into three types, respectively. Because the value threshold φ is selected by the experience, it may lead to inaccurate segmentation.

2.3. Support Vector Machines (SVM) for Classifier. SVM, the most popular branch of machine learning theory to address classification and regression problems, was firstly presented from research in statistical learning theory. Then the introduction of kernel skill breeds a new group of techniques for nonlinear program with high-dimensional or small-sample data [22, 23]. Based on MK-SVM, Yeh et al. proposed a new composite multiple kernel in a form of a linear weighted combination. They combine multiple kernel with SVMs to design a counterfeit banknotes detection system [24]. Although all of these centralized learning approaches have been well preformed in various scenarios, they also increased memory and computational resource consumption, especially for low energy constrained WSN. Therefore, some new algorithms on the topic of distributed SVM have recently been presented. In [25], Forero et al. proposed a distributed SVM scheme that combines alternating direction method of multipliers (ADMM) with consensus-based SVM to reduce the training time cost. This algorithm enhanced the prediction performance with the help of ADMM optimization. However, the collaborative pattern may face risk from shortage of local processors with the increment of the multiple kernel number.

3. Network Assumptions and Statistic Based Feature Model

It is considered that there exist three classes of nodes distributed randomly in the sensing area: sensors, anchors, and malicious nodes. The random network topology is modeled as Erdös-Rényi (ER) random graph denoted by $G = (V, E)$, where V symbolizes node set and E indicates edge set. The node set V consists of N sensors and C anchors, in which the sensor is expressed by $\{S_1, \ldots, S_n,\ 1 \le n \le N\}$ and the anchor is indicated by $\{A_1, \ldots, A_c,\ 1 \le c \le C\}$, respectively. The malicious node, labelled by $\{M_1, \ldots, M_h,\ 1 \le h \le H\}$, is included in sensor set. We define the distance between sensors i and j as ds_{ij} and the distance from sensor i to anchor j as da_{ij}. The total number of links through sensor i, which is calculated by the shortest path, is equal to $N + C$. The sensors are in charge of data-gathering and are not aware of their own coordinates. The anchor is a node that knows and broadcasts its location reference in advance, equipped with localization hardware such as GPS. The malicious nodes exist singly or in pairs to launch various attacks. We assume that the sensors' communication range is assumed to be a circle with the same value R while the malicious nodes' communication range is unlimited. The distance between two sensors is estimated by the received power strength, whose background noises are Gaussian distributed. Likewise, the distance between the sensor and the anchor can be provided by the measurement from anchor. We also assume that each sensor has its own ID and can broadcast it with distances between its neighbors, passively collects adjacent sensors' broadcast, and then makes one list of ID and position which is also called the sensor's neighborhood observation. When all the sensors receive such multiple kinds of packets from neighbors, they transport the information to nearest sensor node with the most energy in a multihop fashion, and the node is engaged in the calculation of feature extraction and recognition and so on in WSNs.

The WSN is assumed to be deployed in an adversarial attacking environment. The adversary launches only external attacks to disrupt the localization procedure, which means it implements malicious behaviors without right cryptographic key. Moreover, the presence of malicious nodes is a small number compared with the benign number in local area. The attack type of the malicious node is divided into three categories including wormhole, replay, and interference attack [26]. Wormhole attack can eavesdrop on the packets of location reference at one position and then create a tunnel and send to other sensors that are far apart, thus causing inaccurate location estimating [27]. As illustrated in Figure 1, sensor S_1 could only capture the beacon signal of anchor A_1 in normal conditions. When a wormhole attack is launched, the malicious node M_2 copies the message from anchor A_2 and sensor S_2 and then tunnels it through a bidirectional link and replays it at the location M_1. Eventually, the node S_1 will determine its location based the positions of A_1 and A_2; it may consider sensor S_2 as neighbor at the same time. In interference attack, the hostile sensor may be an obstacle between signal sender and receiver to distort the signal measurement or time of arrival for ranging. For example, if a signal strength based localization process suffers range enlargement

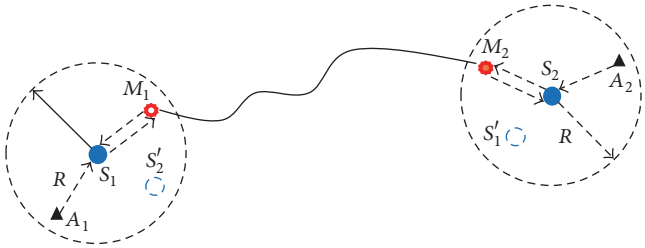


Figure 1: Sensor localization under a wormhole attack.

attack, attackers may attenuate the node's transmission power. Replay attack is another common type of attack which is more likely to appear under the circumstance that energy and computing resources are limited to the adversary. The location message will be captured by the malicious nodes from one anchor. Then an incorrect location reference will be retransmitted to the receiving sensor later. The position calculation in sensors can be frequently affected by the invalid information. In addition to the above characteristics, the adversarial node of wormhole or replay attack in practical environment also has the same ability of data communicating and processing as other normal sensors, which means that this malicious node can be capable of overhearing other types of packet and then modifies and broadcasts it [28, 29]. Again, in order to acquire more accurate attack related information, it is inappropriate to use encryption techniques to eliminate all the adversaries in advance. Furthermore, these attacks might be launched with irregular schedule during the whole classification process. But, in this study, the problem of recognition for localization attacks is more concerned, and, thus, it is required that the proportion and extent of modification in other types of packets do not exceed those in distance related information.

Because there is no single variable to directly characterize the external localization attack, it is necessary to build the original feature set. From the above-mentioned description of external attacks, we find that it might interact directly or indirectly with distance between nodes. The value of distance *ds* and *da*, which is closely associated with whether the node suffers from attacking, could be considered as one main initial feature for recognition. And thus the distance feature vector VD for sensor i is gained as $\mathrm{VD}_i = [ds_{i1}, \ldots, ds_{iN}, da_{i1}, \ldots, da_{iC}]$. While the distance value could describe some information about external attacks, it is still insufficient to classify those external attacks by this single feature. To handle this problem, the complex network theory is introduced to express feature information more comprehensively. Because WSN is comprised of large amounts of sensors, it belongs to complicated network structure. Furthermore, the topological properties will vary with the fluctuation of sensor's location and distance. It implies that these properties can reveal the impact of localization attack from a complex network perspective. Up to now, a number of indexes have been developed to measure the behavior in complex network such as degree and clustering coefficient, which also supply a framework that reflects various features of network. In this work, the indexes considered are degree, clustering coefficient, betweenness centrality (normalized), and coreness. The topological feature based vector VT for sensor i is defined as $\mathrm{VT}_i = [td_i, tc_i, tb_i, tco_i]$, where td_i, tc_i, tb_i, and tco_i represent degree, clustering coefficient, betweenness centrality (normalized), and coreness, respectively.

It seemingly makes sense that the value of the original feature will vary when sensors are under attacks. However, we found that the difference of change in original features between some types of localization attack is not significant, which cannot be classified by a threshold. Furthermore, this change will be expanded under multiple attacks. Thus an effective feature extraction method needs to be explored. We note that the above-mentioned original features can be described by statistics modeling. If a distribution model is constructed to represent the original features obtained by each sensor, the different attack type can be described more accurately by the model parameters extracted.

For each single element such as ds_{ij} in feature vector VD_i, the probability of it can be modeled into the Gaussian distribution with mean μ and variance Σ according to [30], which is analyzed from the point of error measurement:

$$d\left(ds_{ij} \mid \mu, \Sigma\right) = (2\pi \cdot |\Sigma|)^{-1/2} \exp\left\{-\frac{1}{2}\left(ds_{ij} - \mu\right)^T |\Sigma|^{-1}\left(ds_{ij} - \mu\right)\right\}. \quad (1)$$

Moreover, the feature vector VD_i is constituted by the shortest path length, which also possesses the property of complex network. In [31, 32], the length of shortest path is investigated as a negative exponential distributed variable with rate parameter *lambdad*:

$$d\left(ds_{ij} \mid lambdad\right) = \left(\frac{1}{lambdad}\right) e^{-(1/lambdad)ds_{ij}}, \quad 0 < lambdad < \infty. \quad (2)$$

Thus, in order to obtain more detailed properties, the distance vector VD_i is modeled as a mixed distribution as

$$d\left(ds_{ij}\right) = d\left(ds_{ij} \mid \mu, \Sigma\right) + d\left(ds_{ij} \mid lambdad\right). \quad (3)$$

For the topology-related feature, the probability distribution is further investigated to model the irregular deviation along with the normal and attack scenario. Because of limited space, two representative parameters that form the mixture distribution were chosen to analyze the impact of external attacks.

The first parameter analyzed is the node degree. It is expressed by the total amount of neighbors connected to a picked sensor. Degree distribution is defined by a probability $P(k)$, which is the proportion of the sensors with the same amount of k connections. A graph of Erdös-Rényi random WSN has a vertex degree following the Poisson distribution as [33]

$$d(k) \sim \frac{lambdag^k e^{-lambdadg}}{k!}, \quad k \geq 1. \quad (4)$$

In this formula, *lambdag* indicates the expectation value of number of sensors with degree k. Meanwhile, we observe

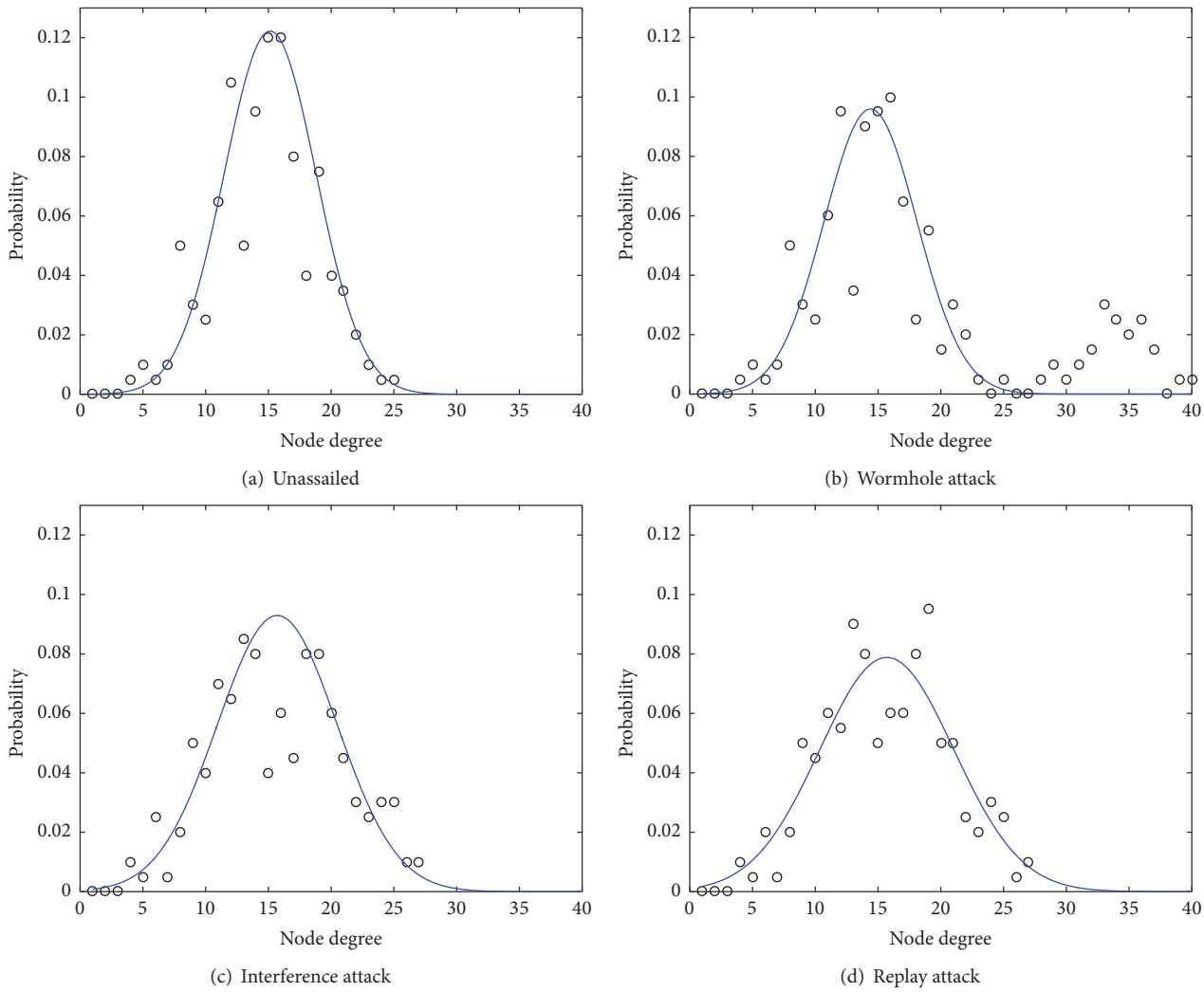


FIGURE 2: Degree distribution of sensors network and the probability density function of its Gaussian distribution approximation for different external attacks scenario. The parameters in Gaussian distribution are estimated as (a) $\mu = 15.14$ and $\Sigma = 26.96$, (b) $\mu = 14.16$ and $\Sigma = 27.43$, (c) $\mu = 15.71$ and $\Sigma = 45.32$, and (d) $\mu = 15.77$ and $\Sigma = 57.05$.

that the mixture distribution of the distance feature is known as a continuous probability distribution while Poisson distribution in degree feature is discrete. It is formidable to construct a unified model by using these two diverse variables. Moreover, if only employing the single parameter *lambdag* in Poisson distribution, it may be hard to distinguish between multiple attacks. It has been found that, according to the limit form when *lambdag* $\rightarrow \infty$ using central limit theorem [34], the Poisson probability density function tends to be excellently achieved by a Gaussian distribution with a high mean value in Poisson distribution. The mean *lambdag* of node degree in Poisson distribution under normal condition, calculated by maximum likelihood estimation, is equal to 14.72, which is not fairly satisfactory. But, for the sake of reducing computational complexity and realization of a feasible mixture model, the Gaussian probability density function is still applied in approximating Poisson distribution.

Figure 2 exhibits the degree distribution of the WSN and its variation under different external attacks. The curves of the probability density function (pdf) with Gaussian distribution approximation are also added. As seen in Figure 2(a), we notice that, for unassailed scene, the degree distribution approximately agrees with a Gaussian distribution. However, the measured distances by RSSI in real-world are affected by the multipath fading or data modification by attackers. Therefore, the value of probability for each degree varies with a fluctuation. Figures 2(b)–2(d) compare the variation of degree distribution and the probability density function of their Gaussian distribution approximation under three types of external attack, respectively. As depicted by Figure 2(b), a peak of probability emerges around degree 14, while another weak intensity probability peak appears around the degree of 35. And the mean μ in Gaussian approximation function decreases to 14.16. The reason for this change is that wormhole tunnel makes some nodes far away be identified incorrectly

as neighbors. When under interference attack, the maximum probability in the degree distribution is lower than that in Figure 2(a). The proportion of sensors with low degree value increases. And under the approximation of Gaussian distribution with $\mu = 15.71$ and $\Sigma = 45.32$, the spread of shape in probability density function looks wider than that in Figure 2(a). As shown in Figure 2(d), the degree distribution has similar variation as Figure 2(c). We also note that the variance Σ in the approximated Gaussian distribution has the highest value, which may correspond to the fact that relayed packets from malicious node increase the nonexistent connections. The above results demonstrated that the parameters in Gaussian distribution approximating help to differentiate these external attacks. Then, using the similar analysis as mentioned above, another feature clustering coefficient is also fitted by Gaussian distribution.

The second property analyzed is normalized betweenness centrality. The betweenness centrality [35], denoted by B_i, is used to examine the potential of a sensor on the connection control with other sensors and evaluate ratio summation of shortest paths passing through sensor i. Therefore, the betweenness centrality B_i of sensor i is formulated as

$$B_i = \sum_{s \neq d \neq i} \left[\frac{N_{sd}(i)}{N_{sd}} \right], \tag{5}$$

where N_{sd} represents the entire quantity of shortest paths from sensor V_s to sensor V_d and $N_{sd}(i)$ represent the shortest paths quantity from V_s to V_d including sensor i. For convenience, the normalization form of B_i is obtained by

$$NB_i = \frac{B_i - \min(B)}{\max(B) - \min(B)}. \tag{6}$$

Figure 3 plots the normalized betweenness centrality distribution and its probability density function of exponential distribution approximation with the same scenarios as node degree. It is noticed that the normalized betweenness centrality distribution in all scenarios are peaked at initial part and then decrease monotonically. Previous works found that normalized betweenness centrality tends to obey a power-law distribution [36]. However, the descending speed of the normalized betweenness centrality distribution for each scenario does not appear so sharply. Furthermore, in order to build the mixture model, the remainder of features should be also presented as a continuous function. Based on these considerations, the distribution of the normalized betweenness centrality is alternatively approximated by a negative exponential distribution which is of the form

$$d(NB_i \mid lambdab) = \left(\frac{1}{lambdab} \right) e^{-(1/lambdab)NB_i}, \quad 0 < lambdab < \infty, \tag{7}$$

where *lambdab* is a rate parameter. In addition, it can be observed in Figure 3(b) that the proportion of *NB* with high probability value is increased by a small amount because the malicious nodes indirectly enhance the communication capability of their neighbors. And under the approximation of negative exponential distribution with *lambdab* = 0.007114, the decay of pdf looks more rapid than that of Figure 3(a). Comparing to Figure 3(b), the variation of distribution for the interference attack case (Figure 3(d)) is analogous to that for wormhole case, but it has milder decreasing, whose rate parameter *lambdab* in exponential distribution approximation reaches the value of 0.007020. Referring to Figure 3(c), the distribution varies slightly, which leads to the change of approximation distribution parameter *lambdab* being not significant. In general, the introduction of the new parameter *lambdab* will contribute to distinguishing certain attacks and improving the performance of classification too. The last topological feature coreness has similar distribution characteristics with normalized betweenness centrality. Accordingly its distribution is also approximated by negative exponential function.

Table 1: Mean square error (MSE) of the distribution approximation curve for all topological features under different external attacks.

Feature category	Unassailed	Wormhole	Interference	Replay
Degree	0.0552	0.0376	0.0475	0.0437
Clustering coefficient	0.0195	0.0285	0.0124	0.0122
Normalized betweenness centrality	0.0153	0.0150	0.0146	0.0143
Coreness	0.0246	0.0232	0.0280	0.0273

For demonstrating the capability of the distribution approximation, the mean square error (MSE) of the approximation curve related to the probability of observed data for all topological features is calculated by setting different attack scenarios and the result is listed in Table 1. In general, the MSE basically maintains at the same magnitude even under attacks, except for some values in the wormhole attack. Secondly, it can be seen that the feature of normalized betweenness centrality yields the smallest MSE value compared with other topological features, which suggests that the exponential distribution is the best approximation. However, from the point of fitting accuracy, it is not easy to say that these approximation distributions precisely fit the feature data, even for normalized betweenness centrality. The reason is that, in the simulated localization of WSN, the distance related message and other data are influenced by some other factors such as channel fading, internode interference, and packet modification by malicious sensors. These elements will bias the true measurement and further increase the approximation error, which will also affect the recognition performance. Therefore, it is indispensable to integrate other approaches like classifier to strengthen the recognition ability in later processing.

For a set of N features collected by one sensor, with distribution of Gaussian and negative exponential, respectively, the probability density function for the mixed features l may be divided into two parts. One part is associated with node degree and clustering coefficient denoted by $dg(l \mid \varphi_1)$, and the other part is associated with normalized betweenness centrality and coreness, denoted by $de(l \mid \varphi_2)$. Combining (1)

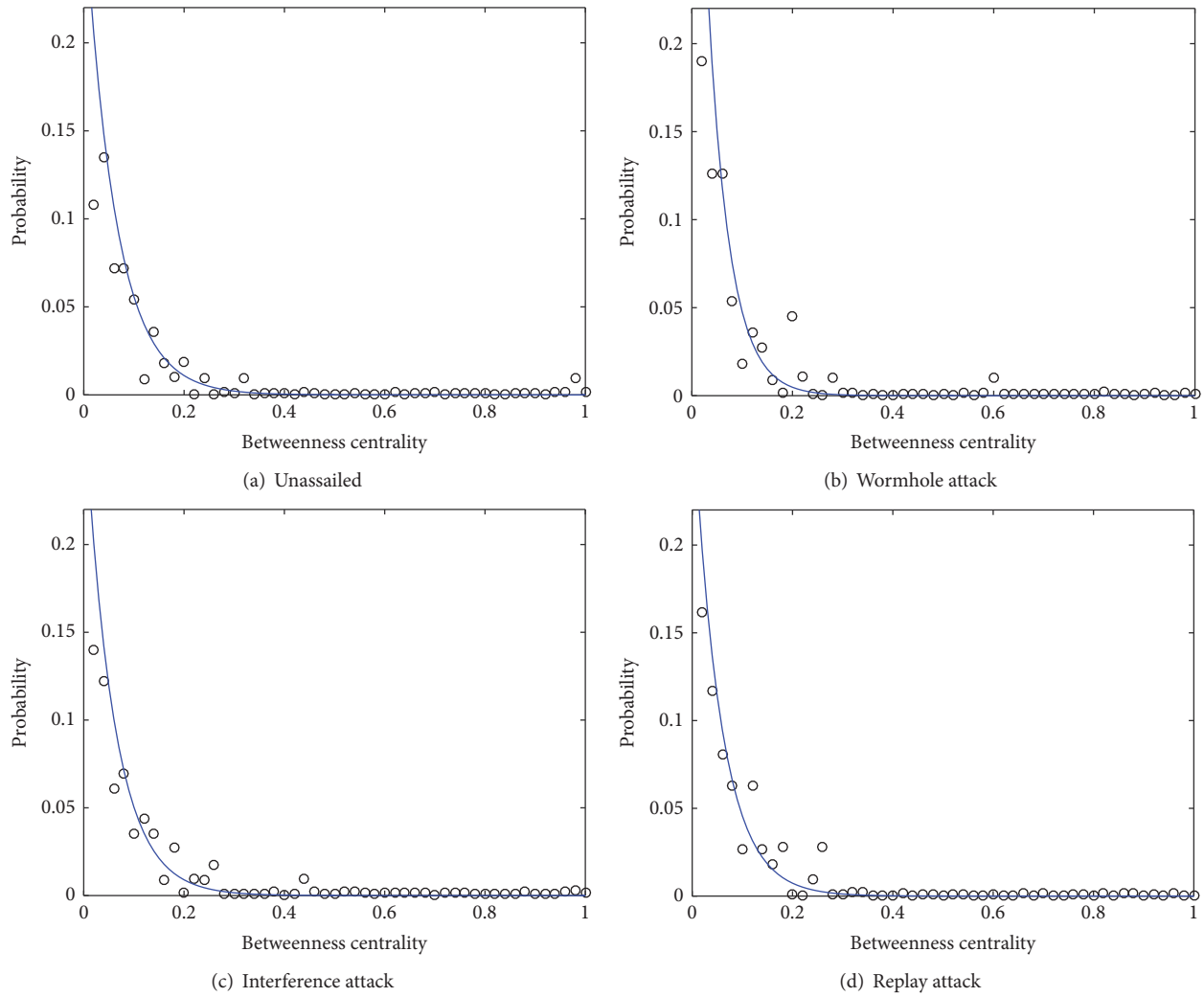


FIGURE 3: Normalized betweenness centrality distribution of sensors network and the probability density function of its exponential distribution approximation for different external attacks scenario. The parameter in exponential distribution is estimated as (a) *lambdab* = 0.006963, (b) *lambdab* = 0.007114, (c) *lambdab* = 0.006892, and (d) *lambdab* = 0.007020.

with (7), the probability density of features vector observation l is modeled in the following manner [37]:

$$d(l \mid \varphi) = p_1 dg(l \mid \varphi_1) + p_2 de(l \mid \varphi_2) \quad (0 < p_1 < 1,\ 0 < p_2 < 1),$$
$$dg(l \mid \varphi_1) = (2\pi \cdot |\Sigma|)^{-1/2} \exp\left\{-\frac{1}{2}(l-\mu)^T |\Sigma|^{-1} (l-\mu)\right\}, \quad (8)$$
$$de(l \mid \varphi_2) = \lambda e^{-\lambda l},$$

where $d(a \mid \varphi)$ represents the mixture probability density of a features vector. The vector of distribution parameters to be estimated is φ. The mixing weights are represented by p_1 and p_2, which satisfies that $p_1 + p_2 = 1$. The means and variances of the Gaussians are represented by μ and Σ, respectively, corresponding to $dg(l \mid \varphi_1)$. λ is the rate parameter corresponding to $de(l \mid \varphi_2)$.

4. Distributed Feature Extractor Design

In order to explore the statistical properties embedded in the mixture density function and to describe the behavior of attack more completely, the EM algorithm can be adopted for calculating unknown model parameters [38]. However, in face of hostile environments, it is unable to confirm whether the sensors for computing and recognition are malicious or not. The data may be ruined or viciously modified by adversary itself if only the centralized computation is used, and the correctness of feature extracted and classifier recognition will be further decreased. Consequently, the issues of security in computation should be taken into consideration. The great success that has been made currently on the research of distributed computing attracts our attention [39]. The primary benefit of this technique is that the multiple consistent intermediated variables updated at each incremental step can be conveyed to several adjacent nodes; the records kept on these nodes will help to detect and separate the attacker.

Depending on this merit, a distributed scheme of feature extraction based on information exchange is presented, and then a verification policy is added.

The exchange-based distributed EM method we proposed is to calculate and update the parameters in classic EM method by using the neighbors' information, which is based on the idea of distributed averaging approach in [40]. We use nc_i to denote set of nodes that communicate with sensor i; that is, there exists an edge $\{x, y\}$ between i and any sensor or anchor j. A distributed linear iteration problem for value x among sensor i and j could be described as

$$x_i(t+1) = U_{ii}x_i(t) + \sum_{j\in N_i} Ux_{ij}(t), \quad i = 1, \ldots, n, \tag{9}$$

where t is a time variable and U denotes a weight matrix where $U \in R^{N\times N}$ and $U_{ij} \neq 0$ only if $j \in nc_i$ and $i \in nc_i$. In order to solve problem (10) asymptotically by average consensus, U should be assumed symmetric and ensure the necessary and sufficient constraints as follows [40]:

$$\begin{aligned} 1^T U &= 1^T, \\ U1 &= 1, \\ \rho\left(U - \frac{11^T}{nc}\right) &< 1, \end{aligned} \tag{10}$$

where $1 \in R^N$ represents one vector whose elements are all equal to 1. $\rho(\cdot)$ is known as the spectral radius for the given matrix. $11^T/N$ denotes the averaging matrix.

Then, based on the probability density in Section 3, the mixture distribution for N features is

$$d(l_i \mid \varphi) = \sum_{i=1}^{N} \left[p_1 dg(l \mid \varphi_1) + p_2 de(l \mid \varphi_2)\right]. \tag{11}$$

Then the log-likelihood for the features vector satisfies

$$L(\varphi \mid y) = \log \sum_{i=1}^{N} \left[p_1 dg(l \mid \varphi_1) + p_2 de(l \mid \varphi_2)\right]. \tag{12}$$

After initializing $p_1^0, p_2^0, \mu^0, \Sigma^0, \lambda^0$, the distributed EM algorithm can be written as follows.

(A) Expectation Process. Let h be the binary hidden variable vectors of having observed the jth density given l_i. For one feature l_i, we calculate the a posteriori probabilities of h_j using Bayes rule and the previous values of parameters $p_1^{k-1}, p_2^{k-1}, \mu^{k-1}, \Sigma^{k-1}, \lambda^{k-1}$:

$$\tau_{ij} = P\left(h_j \mid l_i, \varphi\right) = \frac{p_j d\left(l_i \mid \varphi_j\right)}{p_1 dg(l_i \mid \varphi_1) + p_2 de(l_i \mid \varphi_2)}, \quad j = 1, 2. \tag{13}$$

Then the condition expectation with respect to the actual observed feature l is defined by $L(\varphi \mid y)$:

$$\begin{aligned} Q\left(\varphi, \varphi^{(k-1)}\right) &= E\left[\log d(l, h \mid \varphi) \mid l, \varphi^{k-1}\right] \\ &= \sum_{i=1}^{N} E\left[\log d(l_i, h \mid \varphi) \mid x_i, \varphi^{k-1}\right] \\ &= \sum_{i=1}^{N} \left(\tau_{i1} \log\left(p_1 dg(l_i \mid \varphi_1)\right)\right. \\ &\quad \left. + \tau_{i2} \log\left(p_2 de(l_i \mid \varphi_2)\right)\right). \end{aligned} \tag{14}$$

(B) Maximization Process. In the maximization process, the model parameters are updated by maximizing $Q(\varphi, \varphi^{(\text{ste}-1)})$, which compute the intermediate variables along with iterative step $k = 1, \ldots, K$:

$$\begin{aligned} Q_i(k) &= \gamma_k Q_i^P(k) + (1-\gamma_k) Q_i^W(k) \\ &= \gamma_k \left[\delta_M Q_i^W(k) + \sum_{j=1}^{M-1} \delta_j Q_j(k - M + j)\right] \\ &\quad + (1-\gamma_k)\left[U_{ii} Q_i(k-1) + \sum_{j\in N_i} U_{ij} Q_j(k-1)\right]. \end{aligned} \tag{15}$$

Note that the calculation of current intermediate state $Q_i(\text{ste})$ at the ith sensor at its ste time exchanges information with its neighbors by application of averaging matrix U, where $Q_i(k)$ is nonzero for $j \in nc_i$. It became a weighted combination of a prediction and the value derived from neighborhood averaging. By this mean, the local information of $Q_i(k)$ gradually spread over the network. Thus, each sensor can update its prediction values p_1^k, p_2^k, μ^k, Σ^k, and λ^k using the intermediate variable $Q_i(k)$ until all nodes reach a fixed point on their values. The mixing parameter γ_k in (15) determines the influence of information transmitted across the network, whereas the predictor parameter δ_k associated with the convergence rate. These two step-size coefficients are predecided real constants. It is necessary to investigate the choice for the value of γ_k and δ_k for our scenario. The time-related mixing parameters γ_k and δ_k are defined as [41]

$$\begin{aligned} \gamma_k &= \frac{1}{k}, \\ \delta_k &= \frac{1}{k^\tau}, \quad 0 < \tau < 1, \end{aligned} \tag{16}$$

where τ is a growth rate parameter. And, based on the analysis in [42], the convergence rate of distributed averaging algorithm is going to speed with a large value of τ when the random network is well connected. Therefore, a tentative experimental test on the time cost of the proposed algorithm is conducted with the change on the τ value, in which the variation interval is 0.05. Figure 4 approximately shows the varying trend of τ and result of its corresponding time cost. From the figure we can observe that the computational

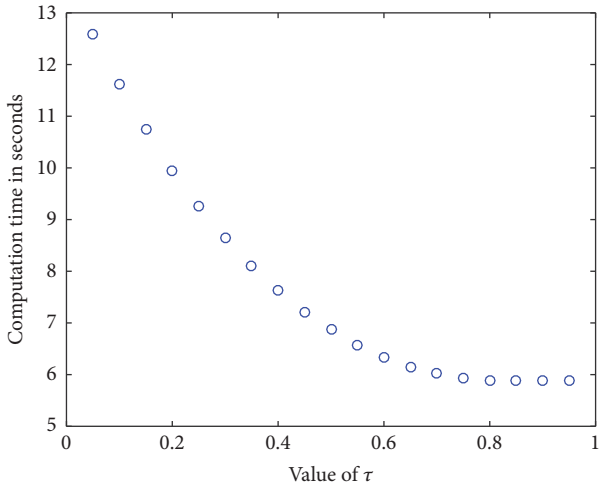


Figure 4: Time cost of distributed EM versus variation of τ.

time decays gradually when τ is increased from 0.05 to 0.95, which means the convergence rate of distributed EM method increases. Moreover, the decline of the time cost nearly stops improving when the τ value achieves 0.8, which demonstrated that the effect of τ will be not necessary after 0.8. And thus the value of rate parameter τ for distributed EM is chosen as 0.8. According to available convergence analysis not included due to space limitation, it is claimed that the proposed method converges to a fixed point of the centralized EM solution when it holds the assumption in (10).

Consequently, the estimation of parameters can be updated as follows: p_1^k, p_2^k, μ^k, Σ^k, and λ^k:

$$
\begin{aligned}
p_1^k &= \frac{\sum_{i=1}^{N} v_{i1}^{k-1}}{\sum_{i=1}^{N} \left(v_{i1}^{k-1} + v_{i2}^{k-1}\right)}, \\
p_2^k &= 1 - p_1^k, \\
\mu^k &= \frac{\sum_{i=1}^{N} v_{i1}^{k-1} l_i}{\sum_{i=1}^{N} v_{i1}^{k-1}}, \\
\Sigma^k &= \frac{\sum_{i=1}^{N} v_{i1}^{k-1} \left(l_i - \mu^{k-1}\right)^2}{\sum_{i=1}^{N} v_{i1}^{k-1}}, \\
\lambda^k &= \frac{\sum_{i=1}^{N} v_{i2}^{k-1} l_i}{\sum_{i=1}^{N} v_{i2}^{k-1}}.
\end{aligned}
\tag{17}
$$

Iterate processes (A) and (B) until a suitable stopping criterion is reached. After each update for condition expectation $Q_i(k)$ and mixture model parameters, the neighbor sensors organized into computing store their local values into memory. And before the end of operation, every node begins to compare the calculated results with at least two neighbors' record after fixed time interval. If it is found a discrepancy in the information, the node with inconsistent values will be considered as an adversary attack and discarded. Then the remainder of sensors will rerun distributed EM algorithm.

When feature extraction using distributed EM algorithm is finished, five new features are acquired for each node, which is defined as $\mathrm{VE}_i = [p_{1i}^k, p_{2i}^k, \mu_i^k, \Sigma_i^k, \lambda_i^k]$. These new features are used to provide more statistical feature information for attack classification. As a result, the sum of feature vector for sensor i can be expressed by $V_i = [\mathrm{VD}_i, \mathrm{VT}_i, \mathrm{VE}_i]$, in which the dimension of V_i is equal to $N + C + 9$. And then V_i will be entirely used as an input into the classifier at the next stage of recognition.

5. Distributed Classifier Design

After these features have been selected and further extracted, we plan to perform classification to recognize the external attacks. A classification process with excellent generalization properties and minimal test error is sufficient to compensate for deficiency in the feature dimensions. As described in the last section, the EG mixture modeling and distributed EM feature extraction all belong to the generative model, which could establish more distinct features from the variation of distance and topological parameters by exploiting their probability density. The generative model possesses excellent ability of modeling and flexibility for the nonnormalized data. However, the optimization capability of generative scheme in recognition phase is always weaker than its discriminative counterpart, especially when the labelled data is sufficient [43]. Another class of technique for recognition is discriminative method. It maps the posterior probability directly as a class label, which avoids the rigid hypotheses for background posterior probability estimation. It generally obtains lower asymptotic error than generative approach in recognition task [44]. However, this manner cannot capture the intrinsic relationship between the feature distribution and the observed feature. In order to make the best of advantages from discriminative approach and generative approach, it is better to couple the generative features with a discriminative classifier to get higher recognition accuracy.

Here, it is noticed that, besides the MK-SVM algorithm, logistic regression (LR) is another prominent and competitive methodology among the discriminative classifiers, which has been used for an extensive range of recognition tasks [45]. Although the computational time of LR is fast and it can often achieve higher accuracy than support vector machine, especially for huge dataset case. Localization attack data classification by using LR has a potential challenge. LR is known as a linear classifier, which means that it can make the best performance on the linear separable features. For the distance and topological features, the variation trend of their approximated distribution and parameters has close relationship with each other. But there still exists certain difference between them for some attacks, which will bring nonlinearity into the features extracted from the unified mixed distribution. In addition, the uncertain data modification by the malicious node will also enhance the nonlinearity. The accumulative nonlinearity may degrade the accuracy of LR in attack recognition. Comparatively, the MK-SVM utilized a kernel function to transform the feature into higher-dimensional space nonlinearly, which is more appropriate to make the feature distinguishable. Therefore, the MK-SVM is

chosen as the classifier for attack classification. Furthermore, in order to adapt to distributed sensor networks, the PECPR-MKSVM algorithm is devised to fully exploit the strengths of machine learning, and a two-stage data verification policy is added finally.

5.1. Extension for CPRSM. For the multiclass problem, we can equal it to a linear equality-constrained optimization problem which consists of multiple separable objective functions:

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} \vartheta_i (v_i), \\ & \sum_{i=1}^{m} g_i v_i = e \\ & v_i \in V_i, \\ & i = 1, \ldots, m, \end{aligned} \tag{18}$$

where ϑ_i is closed proper convex function, v_i is the feature vector, g_i are given matrices, and e is a designated vector. Although the objective function in (18) is convex and linearly constrained, it is not suited for a classic centralized optimization to solve due to a lack of safety and time efficiency. On one hand, after feature extraction, a new feature vector is generated and the potential adversaries are discarded. Although the period of attack launching is uncertain, there is still a small possibility for the undetected adversaries to launch a data modifying assault, which could pose severe damage to recognition performance. On the other hand, the input feature dataset is always large and high dimensional, which is not easy to fulfill the classifier training and testing task for a common optimization scheme. So it becomes important to continue exploiting a parallelizing optimization method to prevent the malicious sensors and process the large feature dataset.

Referring to the literature [46], the contractive Peaceman-Rachford splitting method (CPRSM) has been developed for the linearly constrained convex optimization problem that has been split into two parts. The augmented Lagrangian iterations are given by

$$\begin{aligned} v_1^{k+1} &= \arg\min \left\{ \vartheta_1 (v_1) - \left(\zeta^k\right)^T \left(G_1 v_1 + G_2 v_2^k - e\right) \right. \\ &\quad \left. + \frac{\beta}{2} \left\| G_1 v_1 + G_2 v_2^k - e \right\|^2 \mid v_1 \in V_1 \right\} \\ \zeta^{k+1/2} &= \zeta^k - \alpha\beta \left(G_1 v_1^{k+1} + G_2 v_2^k - e\right) \\ v_2^{k+1} &= \arg\min \left\{ \vartheta_2 (v_1) \right. \\ &\quad - \left(\zeta^{k+1/2}\right)^T \left(G_1 v_1^{k+1} + G_2 v_2 - e\right) \\ &\quad \left. + \frac{\beta}{2} \left\| G_1 v_1^{k+1} + G_2 v_2 - e \right\|^2 \mid v_2 \in V_2 \right\} \\ \zeta^{k+1} &= \zeta^{k+1/2} - \alpha\beta \left(G_1 v_1^{k+1} + G_2 v_2^{k+1} - e\right), \end{aligned} \tag{19}$$

where ϑ_1 and ϑ_2 are closed convex functions, $v_1 \subset R^{n_1}$ and $v_2 \subset R^{n_2}$ are primal variables, $G_1 \in R^{m \times n_1}$ and $G_2 \in R^{m \times n_2}$ are given matrices, and e is a designated vector. ζ^k and $\zeta^{k+1/2}$ are the intermediate updated Lagrange multiplier corresponding to the linear constraints and $\beta > 0$ is a penalty scalar; here the value of relaxing factor $\alpha \in (0, 1)$ is not determined to ensure the sequence derived by (19) under strictly contractive condition. For convenience, it is assumed that α is chosen close to 1. Inspired by effectiveness of CPRSM, a natural idea for solving (18) is to extend the CPRSM scheme from the special situation to the general situation, so the straightforward extension of CPRSM results is the following scheme:

$$\begin{aligned} v_1^{k+1} &= \arg\min \left\{ \vartheta_1 (v_1) \right. \\ &\quad - \left(\zeta^k\right)^T \left(G_1 v_1 + \sum_{j=2}^{m} G_1 v_j^k - e \right) \\ &\quad \left. + \frac{\beta}{2} \left\| G_1 v_1 + \sum_{j=2}^{m} G_j v_j^k - e \right\|^2 \mid v_1 \in V_1 \right\} \\ \zeta^{k+1/m} &= \zeta^k - \alpha\beta \left(G_1 v_1 + \sum_{j=2}^{m} G_1 v_j^k - e \right), \\ v_2^{k+1} &= \arg\min \left\{ \vartheta_2 (v_2) \right. \\ &\quad - \left(\zeta^{k+1/m}\right)^T \left(G_1 v_1^{k+1} + G_2 v_2 + \sum_{j=3}^{m} G_j v_j^k - e \right) \\ &\quad \left. + \frac{\beta}{2} \left\| G_1 v_1^{k+1} + G_2 v_2 + \sum_{j=3}^{m} G_j v_j^k - e \right\|^2 \mid v_2 \in V_2 \right\}, \\ \zeta^{k+2/m} &= \zeta^{k+1/m} - \alpha\beta \left(G_1 v_1^{k+1} + G_2 v_2 + \sum_{j=3}^{m} G_j v_j^k \right. \\ &\quad \left. - e \right) \\ &\quad \vdots \\ v_m^{k+1} &= \arg\min \left\{ \vartheta_m (v_m) \right. \\ &\quad - \left(\zeta^{k+(m-1)/m}\right)^T \left(\sum_{j=1}^{m-1} G_j v_j^{k+1} + G_m v_m - e \right) \\ &\quad \left. + \frac{\beta}{2} \left\| \sum_{j=1}^{m-1} G_j v_j^{k+1} + G_m v_m - e \right\|^2 \mid v_m \in V_m \right\} \\ \zeta^{k+1} &= \zeta^{k+(m-1)/m} - \alpha\beta \left(\sum_{j=1}^{m-1} G_j v_j^{k+1} + G_m v_m - e \right), \end{aligned} \tag{20}$$

where ϑ_i, $i = 1, \ldots, m$, is closed convex function, $v_i \subset R^{n_i}$, $i = 1, \ldots, m$, is primal variable, $G_i \in R^{m \times n_i}$, $i = 1, \ldots, m$, is given matrices, and e is a designated vector. Noting here that $\zeta^{k+j/m}$ is an intermediate variable, its value is updated between iterations of v_i, $i = 1, \ldots, m$. $\beta > 0$ is a penalty scalar and $\alpha \in (0, 1)$ is a relaxing factor. In order to reduce (19) to improve calculation efficiency, $\zeta^{k+j/m}$ can be further rewritten as

$$
\begin{aligned}
&\zeta^{k+1/m} = \zeta^k - \alpha\left(\zeta^k - \zeta^{k+1/m}\right),\\
&v_2^{k+1} = \arg\min \left\{ \vartheta_2(v) - \left[\zeta^k - \alpha\left(\zeta^k - \zeta^{k+1/m}\right)\right]^T G_2 v_2 \right.\\
&\quad \left. + \frac{\beta}{2}\left\| G_1 v_1^{k+1} + G_2 v_2 + \sum_{j=3}^{m} G_j v_j^k - e \right\|^2 \mid v_2 \in V_2 \right\},\\
&\zeta^{k+2/m} = \zeta^k - \left[2\alpha\left(\zeta^k - \zeta^{k+1/3}\right) - \alpha\beta G_2\left(v_2^k - v_2^{k+1}\right)\right]\\
&v_3^{k+3/m} = \arg\min \left\{ \vartheta_3(v_3) - \left(\zeta^{k+2/m}\right)^T \right.\\
&\quad \cdot \left(\sum_{j=1}^{2} G_j v_j^{k+1} + G_3 v_3 + \sum_{j=4}^{m} G_j v_j^k - e \right)\\
&\quad \left. + \frac{\beta}{2}\left\| \sum_{j=1}^{2} G_j v_j^{k+1} + G_3 v_3 + \sum_{j=4}^{m} G_j v_j^k - e \right\|^2 \mid v_3 \in V_3 \right\}\\
&\quad = \arg\min \left\{ \vartheta_3(v_3) \right.\\
&\quad - \left(\zeta^k - \left[2\alpha\left(\zeta^k - \zeta^{k+1/m}\right) - \alpha\beta G_2\left(v_2^k - v_2^{k+1}\right)\right]\right)^T\\
&\quad \cdot G_3 v_3 + \frac{\beta}{2}\left\| \sum_{j=1}^{2} G_j v_j^{k+1} + G_3 v_3 + \sum_{j=4}^{m} G_j v_j^k - e \right\|^2 \mid v_3\\
&\quad \left. \in V_3 \right\}\\
&\zeta^{k+3/m} = \zeta^{k+2/m} - \alpha\beta\left(\sum_{j=1}^{2} G_j v_j^{k+1} + G_3 v_3 + \sum_{j=4}^{m} G_j v_j^k \right.\\
&\quad \left. - e \right) = \zeta^k - \left[3\alpha\left(\zeta^k - \zeta^{k+1/m}\right) - 2\alpha\beta G_2\left(v_2^k\right.\right.\\
&\quad \left.\left. - v_2^{k+1}\right) - \beta G_3\left(v_3^k - v_3^{k+1}\right)\right].
\end{aligned}
\tag{21}
$$

Substituting (21) into (20) and applying scaled dual form, problem (19) can be simplified to

$$
\begin{aligned}
v_1^{k+1} &= \operatorname{argmin} \left\{ \vartheta_1 v_1(v_1) \in +V_1 \frac{\beta}{2}\left\| G_1 v_1 + \sum_{j=2}^{m} G_j v_j^k \right.\right.\\
&\quad \left.\left. - e - \frac{1}{\beta}\zeta^k \right\|^2 \right\},\\
v_2^{k+1} &= \operatorname{argmin} \left\{ \vartheta_2 v_2(v_2) \in +V_2 \frac{\beta}{2}\left\| G_1 v_1^{k+1} + G_2 v_2 \right.\right.\\
&\quad + \sum_{j=3}^{m} G_j v_j^k - e - \frac{1}{\beta}\left[\zeta^k - \alpha\left(\zeta^k - \zeta^{k+(1/m)}\right)\right]\left.\left\|^2 \right.\right\}\\
&\quad \vdots\\
v_m^{k+1} &= \operatorname{argmin} \left\{ \vartheta_m v_m(v_m) \in +V_m \frac{\beta}{2}\left\| \sum_{j=1}^{m-1} G_j v_j^{k+1} \right.\right.\\
&\quad + G_m v_m - e - \frac{1}{\beta}\left[\zeta^k \right.\\
&\quad - (m-1)\alpha\left(\zeta^k - \zeta^{k+\left(\frac{1}{m}\right)}\right)\\
&\quad - (m-2)\alpha\beta G_2\left(v_2^k - v_2^{k+1}\right)\\
&\quad - (m-3)\alpha\beta G_3\left(v_3^k - v_3^{k+1}\right) - \cdots\\
&\quad - 2\alpha\beta G_{m-2}\left(v_{m-2}^k - v_{m-2}^{k+1}\right)\\
&\quad \left.\left. - \beta G_{m-1}\left(v_{m-1}^k - v_{m-1}^{k+1}\right)\right]\right\|^2 \right\}.\\
\zeta^{k+1} &= \zeta^k - m\alpha\left(\zeta^k - \zeta^{k+(1/m)}\right) - (m-1)\alpha\beta G_2\left(v_2^k\right.\\
&\quad \left. - v_2^{k+1}\right) - (m-2)\alpha\beta G_3\left(v_3^k - v_3^{k+1}\right) - \cdots\\
&\quad - 2\alpha\beta G_{m-1}\left(v_{m-1}^k - v_{m-1}^{k+1}\right) - \beta G_m\left(v_m^k - v_m^{k+1}\right).
\end{aligned}
\tag{22}
$$

So the minimization problem with more than three convex functions can be obtained via splitting the $(k+1)$ subproblem of (18) alternatingly. We name (22) as the extending contractive Peaceman-Rachford splitting method (ECPR).

5.2. Proximal MK-SVM with ECPR Substitution. Considering a labelled training set TS $= \{(v_1, b_1), \ldots, (v_n, b_n)\}$, where feature vector $v_i \in \mathfrak{R}^m$ and $b_i \in \{+1, -1\}$, MK-SVM places a

separating hyperplane between the two categories in feature space. So the minimization optimal problem of the MK-SVM utilizing the unweighted kernels combination is given as follows [47]:

$$\begin{aligned} \min \quad & \left(\frac{1}{2}\right) \cdot \sum_{m=1}^{M} \left\| w_m \right\|^2 \\ \text{s.t.} \quad & b_i \left(\sum_{m=1}^{M} w_m \kappa_m \left(v_i \right) + \text{bia} \right) \geq 1 \\ & (i = 1, 2, \ldots, n), \end{aligned} \tag{23}$$

where w_m denotes the mth element of weight vector. bia represents a bias term corresponding to the hyperplane. κ_m is a basic kernel function. The objective of this formulation is to optimize the variable of w and bia, which will also find the maximum margin and the minimum empirical error.

In order to convert the inequality constraints to an equality z constraint, a slack variable z^2 is introduced into optimization problem:

$$\begin{aligned} \min \quad & \left(\frac{1}{2}\right) \sum_{m=1}^{M} \left\| w_m \right\|^2 \\ \text{s.t.} \quad & b_i \left(\sum_{m=1}^{M} w_m \cdot \kappa_m \left(v_i \right) + \text{bia} \right) = 1 + z_i^2 \\ & (i = 1, 2, \ldots, n). \end{aligned} \tag{24}$$

Then optimization procedures can be divided into two parts, with w, bia optimization as a group and z^2 as another. The first part is to solve the minimization problem with respect to parameters w, bia by using ECPR when z^2 is fixed. For solving the w, bia optimization, the augment Lagrangian function of (24) can be expressed as

$$\begin{aligned} L\left(w_m, \text{bia}, \zeta_i, \beta, z_i^2\right) &= \frac{1}{2} \sum_{m=1}^{M} \left\| w_m \right\|^2 \\ &+ \sum_{i=1}^{n} \zeta_i \left(b_i \left(\sum_{m=1}^{M} w_m \kappa_m \left(v_i \right) + \text{bia} \right) - 1 - z_i^2 \right) \\ &+ \sum_{i=1}^{n} \frac{\beta}{2} \left\| b_i \left(\sum_{m=1}^{M} w_m \kappa_m \left(v_i \right) + \text{bia} \right) - 1 - z_i^2 \right\|^2 = \frac{1}{2} \\ &\cdot \sum_{m=1}^{M} \left\| w_m \right\|^2 + \sum_{i=1}^{n} \frac{\beta}{2} \left\| b_i \left(\sum_{m=1}^{M} w_m \kappa_m \left(v_i \right) + b \right) - 1 \right. \\ &\left. - z_i^2 - \frac{1}{\beta} \zeta_i \right\|^2, \end{aligned} \tag{25}$$

where v_i denotes the feature vector and w_m denotes the mth element of weight vector. bia represents a bias term corresponding to the hyperplane. κ_m is a basic kernel function, ζ_i is a Lagrange multiplier, and β is a positive scalar. By applying ECPR to the augmented Lagrangian function, the distributed iterative form of problem (25) is obtained. To reduce the calculation of derivative, we then use a linearized proximal method that was proposed by Xu and Wu [48]:

$$\begin{aligned} w_1^{k+1} &= \arg\min \left\{ \frac{1}{2} \left\| w_1 \right\|^2 - \sum_{i=1}^{n} b_i \kappa_1 \left(v_i \right) \left(w_1 - w_1^k \right) \sum_{i=1}^{n} \left(b_i \left(\sum_{m=1}^{M} w_m^k \kappa_m \left(v_i \right) + \text{bia} \right) - 1 - z_i^2 - \frac{1}{\beta} \zeta_i \right) + \frac{r_1}{2} \left\| w_1 - w_1^k \right\|^2 \right\}. \\ w_2^{k+1} &= \arg\min \left\{ \frac{1}{2} \left\| w_2 \right\|^2 \right. \\ &\left. - \sum_{i=1}^{n} b_i \kappa_2 \left(v_i \right) \left(w_2 - w_2^k \right) \sum_{i=1}^{n} \left(b_i \left(w_1^{k+1} \kappa_1 \left(v_i \right) + \text{bia} \right) + b_i \left(\sum_{m=2}^{M} w_m^k \kappa_m \left(x_i \right) + \text{bia} \right) - 1 - z_i^2 - \frac{1}{\beta} \zeta_i^{k+1/m} \right) + \frac{r_2}{2} \left\| w_2 - w_2^k \right\|^2 \right\} \\ &\vdots \\ w_M^{k+1} &= \arg\min \left\{ \frac{1}{2} \left\| w_M \right\|^2 \right. \\ &\left. - \sum_{i=1}^{n} y_i \kappa_M \left(v_i \right) \left(w_M - w_M^k \right) \sum_{i=1}^{n} \left(b_i \left(\sum_{m=1}^{M-1} w_m^{k+1} \kappa_m \left(v_i \right) + \text{bia} \right) + b_i \left(w_M^k \kappa_M \left(x_i \right) + \text{bia} \right) - 1 - z_i^2 - \frac{1}{\beta} \zeta_i^{k+(M-1)/M} \right) + \frac{r_M}{2} \left\| w_M - w_M^k \right\|^2 \right\} \\ \sum_{i=1}^{n} \zeta_i^{k+1} &= \sum_{i=1}^{n} \left\{ \zeta_i^k - M \left(\zeta_i^k - \zeta_i^{k+1/m} \right) - (M-1) \alpha \beta b_i \kappa_2 \left(v_i \right) \left(w_2^k - w_2^{k+1} \right) - (M-2) \alpha \beta b_i \kappa_3 \left(w_3^k - w_3^{k+1} \right) - \cdots - 2\alpha \beta b_i \kappa_{M-1} \left(v_i \right) \left(w_{M-1}^k \right. \right. \\ &\left. \left. - w_{M-1}^{k+1} \right) - \beta y_i \kappa_M \left(v_i \right) \left(w_M^k - w_M^{k+1} \right) \right\}. \end{aligned} \tag{26}$$

After taking the differentiation of (26), the primal variables can be calculated as

$$\begin{aligned}
&\frac{\partial L}{\partial w_1} = w_1 - \sum_{i=1}^{n} b_i \kappa_1(v_i) \sum_{i=1}^{n} \left(b_i \left(\sum_{m=1}^{M} w_m^k \kappa_m(v_i) + \text{bia} \right) - 1 - z_i^2 - \frac{1}{\beta}\zeta_i \right) + r w_1 - r_1 \left\| w_1^k \right\| = 0 \\
&w_1 = \frac{1}{1+r} \left(\sum_{i=1}^{n} b_i \kappa_1(v_i) \sum_{i=1}^{n} \left(b_i \left(\sum_{m=1}^{M} w_m^k \kappa_m(v_i) + \text{bia} \right) - 1 - z_i^2 - \frac{1}{\beta}\zeta_i \right) - r_1 \left\| w_1^k \right\| \right) = U_1 \\
&\frac{\partial L}{\partial \text{bia}} = \sum_{i=1}^{n} b_i \sum_{i=1}^{n} b_i \kappa_1(v_i) \left(w_1 - w_1^k \right) = 0 \\
&\qquad\qquad \vdots \\
&w_2 = \frac{1}{1+r} \left(\sum_{i=1}^{n} b_i \kappa_2(v_i) \sum_{i=1}^{n} \left(b_i \left(w_2^{k+1} \kappa_2(v_i) + \text{bia} \right) + b_i \left(\sum_{m=2}^{M} w_m^k \kappa_m(v_i) + b \right) - 1 - z_i^2 - \frac{1}{\beta}\zeta_i^{k+1/m} \right) - r_2 \left\| w_2^k \right\| \right) = U_2 \\
&\frac{\partial L}{\partial \text{bia}} = \sum_{i=1}^{n} y_i \sum_{i=1}^{n} y_i \kappa_2(v_i) \left(w_2 - w_2^k \right) = 0 \\
&w_M = \frac{1}{1+r} \left(b_i \kappa_M(v_i) \left(b_i \left(\sum_{m=1}^{M-1} w_m^{k+1} \kappa_m(v_i) + \text{bia} \right) + b_i \left(w_M^k \kappa_M(v_i) + \text{bia} \right) - 1 - z_i^2 - \frac{1}{\beta} \sum_{i=1}^{n} \zeta_i^{k+1/m} \right) - r_M \left\| w_M^k \right\| \right) \\
&\quad = U_M \\
&\frac{\partial L}{\partial \text{bia}} = b_i^2 \kappa_M(v_i) \left(w_M - w_M^k \right) = 0.
\end{aligned} \tag{27}$$

Substitute equations in (27) into primal problem $L(w_m, \text{bia}, \zeta_i, \beta)$, and the primal problem of minimization is converted to a dual function:

$$\begin{aligned}
&w_1^{k+1} = \arg\max \left\{ \frac{1}{2} \left\| U_1 \right\|^2 - \sum_{i=1}^{n} b_i \kappa_1(v_i) \left(U_1 - w_1^k \right) \sum_{i=1}^{n} \left(b_i \sum_{m=1}^{M} w_m^k \kappa_m(v_i) - 1 - z_i^2 - \frac{1}{\beta}\zeta_i \right) + \frac{r_1}{2} \left\| U_1 - w_1^k \right\|^2 \right\} \\
&w_2^{k+1} = \arg\max \left\{ \frac{1}{2} \left\| U_2 \right\|^2 - \sum_{i=1}^{n} b_i \kappa_2(v_i) \left(U_2 - w_2^k \right) \sum_{i=1}^{n} \left(b_i w_1^{k+1} \kappa_1(v_i) + b_i \sum_{m=2}^{M} w_m^k \kappa_m(v_i) - 1 - z_i^2 - \frac{1}{\beta}\zeta_i^{k+1/m} \right) \right. \\
&\quad \left. + \frac{r_2}{2} \left\| U_2 - w_2^k \right\|^2 \right\} \\
&\qquad\qquad \vdots \\
&w_M^{k+1} = \arg\max \left\{ \frac{1}{2} \left\| U_M \right\|^2 \right. \\
&\quad \left. - \sum_{i=1}^{n} b_i \kappa_M(v_i) \left(U_M - w_M^k \right) \sum_{i=1}^{n} \left(b_i \sum_{m=1}^{M-1} w_m^{k+1} \kappa_m(v_i) + b_i w_M^k \kappa_M(v_i) - 1 - z_i^2 - \frac{1}{\beta}\zeta_i^{k+(M-1)/M} \right) + \frac{r_M}{2} \left\| U_M - w_M^k \right\|^2 \right\} \\
&\sum_{i=1}^{n} \zeta_i^{k+1} = \sum_{i=1}^{n} \left\{ \zeta_i^k - M \left(\zeta_i^k - \zeta_i^{k+1/m} \right) - (M-1)\alpha\beta b_i \kappa_2(v_i) \left(w_2^k - w_2^{k+1} \right) - (M-2)\alpha\beta b_i \kappa_3 \left(w_3^k - w_3^{k+1} \right) - \cdots \right. \\
&\quad \left. - 2\alpha\beta b_i \kappa_{M-1}(v_i) \left(w_{M-1}^k - w_{M-1}^{k+1} \right) - \beta b_i \kappa_M(v_i) \left(w_M^k - w_M^{k+1} \right) \right\}.
\end{aligned} \tag{28}$$

In the following, we consider the z^2 optimization by fixing w, bia which can be easily solved via gradient method. Setting derivatives of (25) with respect to z^2 equal to 0 yields the following results:

$$\frac{\partial L}{\partial z_i^2} = -z_i^2 \left[b_i \left(\sum_{k=1}^{M} w_m^k \kappa(v_i) + \text{bia} \right) - 1 - z_i^2 - \frac{1}{\beta}\zeta^k \right] = 0, \quad (29)$$

$$z_i^2 = b_i \left(\sum_{k=1}^{M} w_m^k \kappa(v_i) + \text{bia} \right) - 1 - \frac{1}{\beta}\zeta^k \text{ or } z_i^2 = 0.$$

Repeat the above iterations until convergence. Therefore, we name (28) and (29) as the proximal extension contractive Peaceman-Rachford splitting-multiple kernel support vector machines (PECPR-MKSVM).

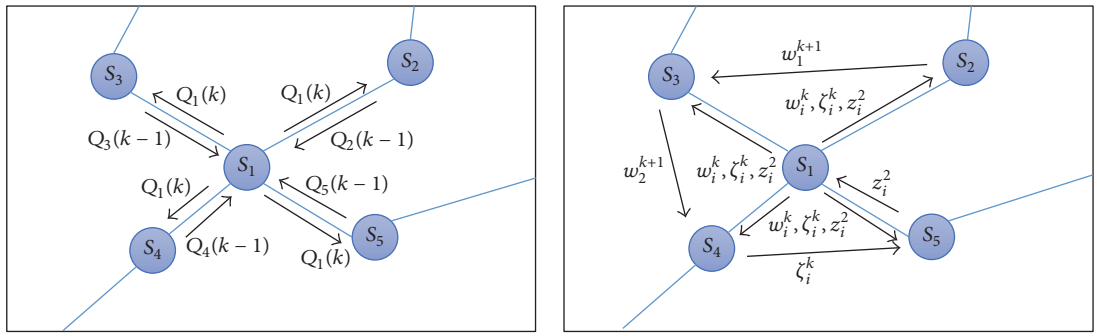
While the PECPR-MKSVM algorithm runs effectively under the condition of normal, it ignores the attack scenarios that the intermediate variables may be negatively disrupted by the modified data from attacker. To address these inadequacies, a simple two-stage calculation verification policy is supplemented to avoid the adversary and improper output. It requires the following steps to carry out during the PECPR-MKSVM training. (1) *Neighbor Node Verification Stage*. As the variable updated at each neighbor node is not the same, the sequence of data forwarding for all neighbor nodes should be rearranged such as forward or backward one position after the first fixed time interval. Then the set of renewing parameter information produced in the next interval is checked with the record maintained in the same sensor. Finally, the first sensor with divergence is determined to be attacker. (2) *Host Node Verification Stage*. Although the validation on neighbor node can eliminate the malicious node, it does not exclude the potential risk on host node itself. Therefore, it is necessary to conduct the algorithm repeatedly on the neighbor node that has been authenticated, which ensures the attacks prevention and calculation precision.

5.3. The Process Overview of the Proposed Algorithm and Calculation Verification Policy. Based on the above design, the message transmission through neighboring nodes in the distributed recognition method can be summarized as follows and is clarified with Figure 5. As shown in Figure 5(a), in the feature extraction phase, when every node obtains its own original feature set, then it conveys to the sensor with the most energy (named S_1) and its one-hop neighbors to compute statistical attack feature set and verifies its local states $Q_i(k)$ by exchanging record with neighbors. As shown in Figure 5(b), in the PECPR-MKSVM training phase, the authenticated sensor S_1 with the most residual energy sets an initial value to $w_1^0, \ldots, w_m^0, \sum_{i=1}^{n} \lambda_i^0$, z_i^0 and then computes w_1^{k+1} via (23); next, node sends its newly updated w_1^{k+1} to one of its one-hop neighboring sensors. After receiving w_1^{k+1}, iteration resumes when another node updates w_2^{k+1} with the features set included in itself. According to the forwarding rule, all the intermediate variables will be transmitted along the path in order of S_1's direct neighbor sensor one by one. Eventually, z^2 was sent to S_1 to start a new circulation. And the final global minimum of the associated cost function can be got by iterative update on the distributed classifier.

6. Experimental Setup and Results

6.1. Simulation Setup. To assess the effective aspects of our mechanism, we presented four groups of experiments that were carried out under different localization attacks. In our simulation, 600 sensors including 48 anchors are randomly distributed over an 300 m × 300 m area. We set the value of communication distance regarding sensors and anchors all to 35 m. Moreover, three types of external localization attacks (wormhole, replay, and interference) exist in network simultaneously. The fraction of malicious sensors is 20%, where each kind of external localization attack has one-third number of the total. If the sensor responsible for performing computation happens to be the adversary, there will be a possibility and range of data modification which is lower than 30%. At first all the sensors in the network begin to collect the information of original feature and then convey the feature data to the sensor with the most energy in the network and its one-hop neighbors. And these sensors will conduct distributed EM scheme to compute the new statistical features. At last, we chose one of the authenticated sensors with new feature dataset to run PECPR-MKSVM for training and classification. The experiments are then repeated for 5 times. The features of first four times were adopted as training sets whereas the ones of the last time were used as testing sets. Our classifier for PECPR-MKSVM uses RBF+Poly kernels and one-versus-all approach.

6.2. Attack Classification Performance with the Proposed Algorithm. For the effectiveness evaluation of combining distributed feature extraction and classifier scheme, the recognition performances on two kinds of feature datasets are compared first between the proposed classifier and four similar classifiers, such as a distributed SVM (MoM-DSVM), a multiple kernel SVM (SimpleMKL), a typical SVM (C-SVM), and a logistic regression (LR) classifier. Table 2 shows the average recognition accuracy obtained by these algorithms under different external localization attacks. In general, as depicted in Table 2, the average success classification rate for each kind of attack using feature extraction technology significantly rose 9.4% compared to the one recognized only by classifier. Furthermore, it is worth mentioning that the proposed classifier obtains relatively higher accuracy than the rest of other classifier schemes. For example, for the replay attack, the proposed classifier with the features extracted offers the highest classification accuracy of 93.28%. For the same case, SimpleMKL and C-SVM only result in recognition accuracy of 84.56% and 68.72%, respectively. Although the MoM-DSVM classifier achieves satisfactory classification performance using a consensus-based support vector for replay attack, it is still not sufficiently to recognize the wormhole and interference attacks. The recognition performance of LR is improved obviously by the extracted features, whose performance is superior for the interference attacks compared to the MoM-DSVM and Simple-MKL.



(a) Iteration visualization of distributed EM algorithm in feature extraction phase

(b) Iteration visualization of PECPR-MKSVM classifier in training phase

Figure 5: Overview of distributed calculation among sensor nodes in the proposed recognition algorithm.

Table 2: Comparison of recognition rate in percentage (%) on the attack feature sets with different classifiers.

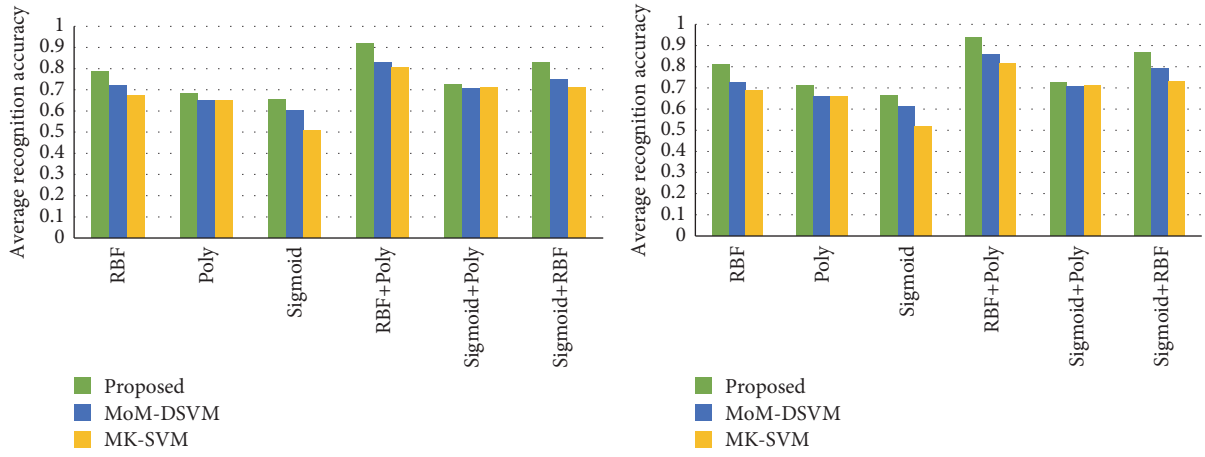
Related works	Average recognition rate per category under localization attacks (without feature extraction)				Average recognition rate per category under localization attacks (with distributed feature extraction)			
	Unassailed	Wormhole	Interference	Replay	Unassailed	Wormhole	Interference	Replay
PEPMK-SVM	80.27	73.78	79.03	78.49	91.56	91.97	90.85	93.28
MK-SVM	72.99	64.29	67.47	70.59	80.47	76.98	79.31	84.56
C-SVM	59.14	56.49	38.63	61.17	62.54	66.15	49.25	68.72
MoM-DSVM	73.42	68.96	73.85	75.21	82.39	83.76	80.02	86.54
LR	71.84	59.31	71.75	66.84	79.27	70.38	82.59	82.81

However, it is still not comparable to the PEPMK-SVM due to the limitation of safety and the nonlinear features. These comparisons show that the combination of distributed feature extraction and the proposed classifier is able to achieve higher recognition accuracy than any other recognition methods.

6.3. Classification Robustness of the PECPR-MKSVM with Different Kernel Function. We further explore classification robustness of the PECPR-MKSVM classifier with different kernel function. Figure 6 shows the average recognition accuracies for varied numbers of multikernel classifiers by combining different kernels, such as RBF kernel, sigmoid kernel, and polynomial kernel. In Figure 6(a), the average recognition accuracy using the proposed classifier is 4%–7% higher than the MoM-DSVM and MK-SVM method. Moreover, the kernel combination of RBF and polynomial kernel achieves higher recognition accuracy than the others; on the contrary, a single kernel fails to offer good recognition accuracy. For classification error existing in the result, it can often be attributed to the lack of sufficient training samples for classifier. Next, to show the robustness of the proposed classifier, Figure 6(b) compares the recognition performance under a higher malicious sensors ratio. When the ratio of the malicious sensors exaggerates, the average attack recognition rates have a certain improvement for all classifiers, which means that the additional data of the malicious sensors provides more sample to the classifier and affects the classification hyperplanes. Particularly, the average recognition rate of the proposed classifier for RBF+Poly kernel is increased from 91.9% to 93.9%. Thus, the proposed algorithm is more robust to recognize localization attacks even under a severe scenario.

6.4. Convergence Performance of PECPR-MKSVM with Different Positive Scalar β and Relaxation Factor α. In order to assess the impact of positive scalar β and relaxation factor α for the proposed classifier, each node trains a local PECPR-MKSVM and its convergence of test error is compared with the one obtained via MoM-DSVM. We first fix α and choose two different values of $\beta = \{1, 10\}$ with respect to PECPR-MKSVM classifier. Then the evolutions of iteration are plotted for each choice of β. For comparison purposes, we also plot the convergence performance of MoM-DSVM with $\beta = 1$ and $\beta = 10$. As illustrated in Figure 7, we see that the test error of PECPR-MKSVM reduces very rapidly with a fewer steps of iterations and soon approaches the minimum value, which outperforms MoM-DSVM based method. Moreover, plot in Figure 7(a) also reveals that a very large value of β may lead to dispersion and hinder the convergence rate. These results further reflect the importance of choosing β when constructing the EPRSM classifier. Last, plot in Figure 7(b) illustrates that, for each of the test scalar β, larger relaxation factor α tends to accelerate the convergence rate of the proposed classifiers, thus shortening the runtime.

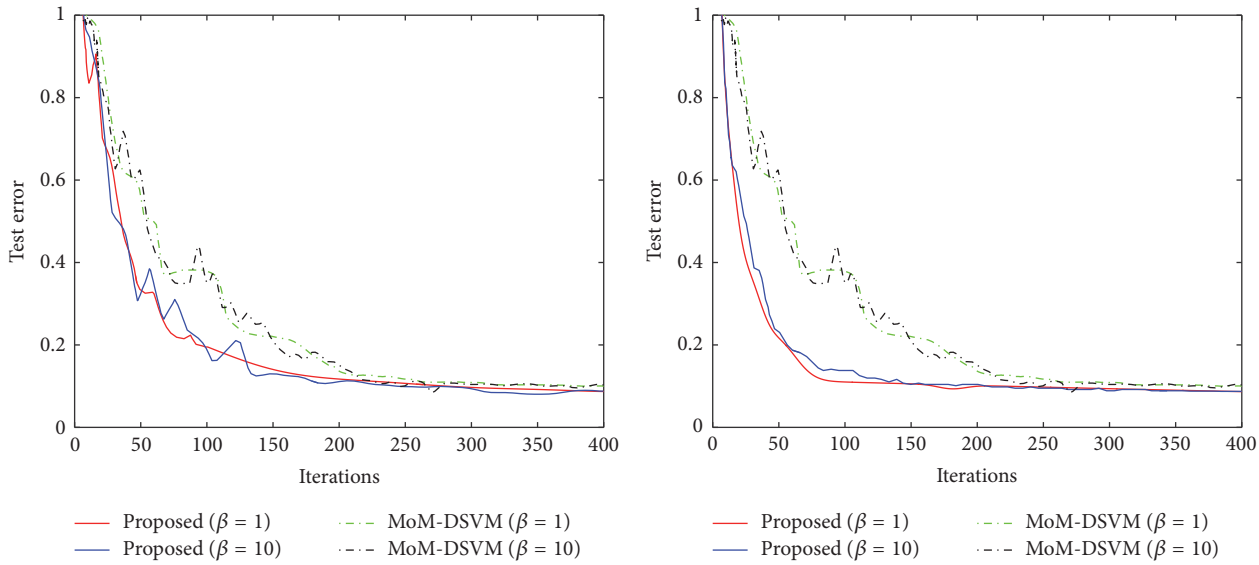
6.5. Time Cost with the Proposed Algorithm. Additionally, to assess different algorithms in saving the time cost of the classification, we further perform experiment under the situation that the number of sensors varies from 200 to 1000



(a) When 20% sensors are malicious

(b) When 30% sensors are malicious

Figure 6: Average recognition accuracy comparison of the proposed classifiers with MoM-DSVM and MK-SVM under different kernel function.



(a) $\alpha = 0.4$

(b) $\alpha = 0.8$

Figure 7: Evolution of test error with different positive scalar β and relaxation factor α for the proposed and MoM-DSVM method.

and plot the computational time in Figure 8. Here, we combine all the classifiers with the proposed feature extraction process. Generally, it is not hard for us to find that the proposed algorithm is the fastest among three schemes. More importantly, we can observe that the time for the proposed algorithm increased linearly with the number of sensors, but the growth rate is slower, even though the number of sensor increases to 1000. This is because the classification process is distributed computing, and the computational complexity is depending on the number of neighbor sensors. In contrast, although time cost of the consensus-based MoM-DSVM scheme is more efficient than the MK-SVM and LR algorithm, it still requires higher calculation amount in the training process. The performance of LR algorithm lies between the MK-SVM and the distributed SVM. The MK-SVM algorithm uses the centralized architecture to execute the classification, which increases the number of iterations and computational complexity. Thus, the proposed algorithm is more computationally efficient than the MoM-DSVM, LR, and the MK-SVM method.

7. Conclusion

This paper generalized a distributed classification scheme, which is used for external localization attack classification in WSN. A novel distributed version of EM feature extractor and MK-SVM classifier is also proposed. These new schemes help each sensor computing during feature extraction and

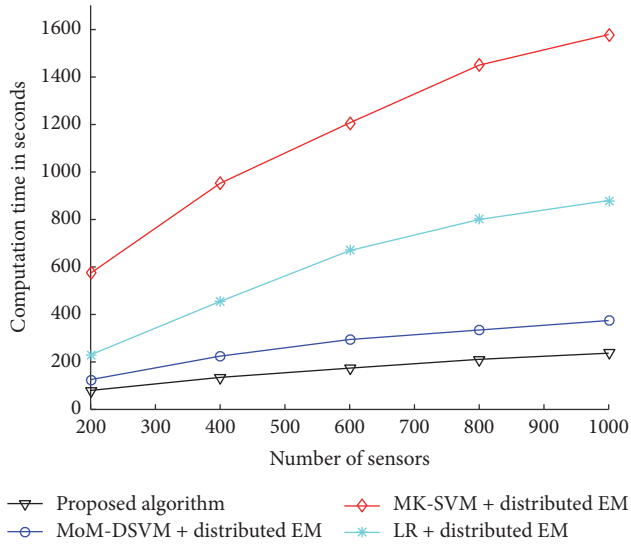


Figure 8: Time cost comparison by using different methods.

recognition across different neighbor sensors. The algorithm models the distance and topological based features into a mixed distribution at the first frame of the phase. Then the parameter features are extracted with a distributed EM scheme that fuses the time and neighbors' information, as it evolves over iteration. Eventually, a distributed classifier, which corporates MK-SVM with extension for CPRSM, is designed to classify localization attack datasets into multiclass. The experimental results have shown that using the distributed EM as feature extractor and PECPR-MKSVM as classifier can be able to achieve higher classification accuracy than other similar methods. Moreover, the attack recognition scheme presented in this paper is more robust to a wide range of attacks with competitive time efficiency.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61401360), the Fundamental Research Funds for the Central Universities of China (no. 3102014JCQ01055), and the Natural Science Basis Research Plan in Shaanxi Province of China (no. 2014JQ2-6033).

References

[1] J. Jiang, G. Han, C. Zhu, Y. Dong, and N. Zhang, "Secure localization in wireless sensor networks: a survey," *Journal of Communications*, vol. 6, no. 6, pp. 460–470, 2011.

[2] Y. Zhang, W. Liu, Y. Fang, and D. Wu, "Secure localization and authentication in ultra-wideband sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 4, pp. 829–835, 2006.

[3] Y. Chen, W. Trappe, and R. P. Martin, "Attack detection in wireless localization," in *Proceedings of the IEEE 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1964–1972, May 2007.

[4] A. Boukerche, H. H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Secure localization algorithms for wireless sensor networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 96–101, 2008.

[5] D. He, L. Cui, H. Huang, and M. Ma, "Design and verification of enhanced secure localization scheme in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 7, pp. 1050–1058, 2009.

[6] R. Garg, A. L. Varna, and M. Wu, "An efficient gradient descent approach to secure localization in resource constrained wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 717–730, 2012.

[7] Y. Wei and Y. Guan, "Lightweight location verification algorithms for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 938–950, 2013.

[8] H. Li, Y. He, X. Cheng, H. Zhu, and L. Sun, "Security and privacy in localization for underwater sensor networks," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 56–62, 2015.

[9] D. Liu, P. Ning, and W. Du, "Detecting malicious beacon nodes for secure location discovery in wireless sensor networks," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS '05)*, pp. 609–619, Columbus, Ohio, USA, June 2005.

[10] M. Jadliwala, S. Zhong, S. J. Upadhyaya, C. Qiao, and J.-P. Hubaux, "Secure distance-based localization in the presence of cheating beacon nodes," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 810–823, 2010.

[11] W. Du, L. Fang, and N. Peng, "LAD: localization anomaly detection for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 66, no. 7, pp. 874–886, 2006.

[12] G. Han, J. Jiang, L. Shu, M. Guizani, and S. Nishio, "A two-step secure localization for wireless sensor networks," *Computer Journal*, vol. 56, no. 10, pp. 1154–1166, 2013.

[13] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 91–98, Boise, Idaho, USA, April 2005.

[14] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 2010.

[15] N. Yu, L. Zhang, and Y. Ren, "BRS-based robust secure localization algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 3, Article ID 107024, 2013.

[16] L. Doherty, L. E. Ghaoui, and K. S. J. Pister, "Convex position estimation in wireless sensor network," in *Proceedings of the 8th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, Anchorage, Alaska, USA, April 2001.

[17] T. Bao, J. Wan, K. Yi, and Q. Zhang, "A game-based secure localization algorithm for mobile wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 9, Article ID 642107, 2015.

[18] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1154–1166, 2004.

[19] S. Boutemedjet, N. Bouguila, and D. Ziou, "A hybrid feature extraction selection approach for high-dimensional non-Gaussian data clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1429–1443, 2009.

[20] A. Subasi, "EEG signal classification using wavelet feature extraction and a mixture of expert model," *Expert Systems with Applications*, vol. 32, no. 4, pp. 1084–1093, 2007.

[21] H. Yuan and X.-P. Zhang, "Statistical modeling in the wavelet domain for compact feature extraction and similarity measure of images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 3, pp. 439–445, 2010.

[22] P. Bouboulis, S. Theodoridis, C. Mavroforakis, and L. Evaggelatou-Dalla, "Complex support vector machines for regression and quaternary classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1260–1274, 2015.

[23] J. Peng, Y. Zhou, and C. L. Philip Chen, "Region-kernel-based support vector machines for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 9, pp. 4810–4824, 2015.

[24] C.-Y. Yeh, W.-P. Su, and S.-J. Lee, "Employing multiple-kernel support vector machines for counterfeit banknote recognition," *Applied Soft Computing*, vol. 11, no. 1, pp. 1439–1447, 2011.

[25] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research (JMLR)*, vol. 11, pp. 1663–1707, 2010.

[26] A.-Y. Ye, J.-F. Ma, Q.-Q. Pei, and L. Xu, "Survey on secure node positioning in wireless sensor networks," *Journal on Communications*, vol. 30, no. 10, pp. 74–84, 2009.

[27] X. Lu, D. Dong, and X. Liao, "MDS-based wormhole detection using local topology in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 145702, 9 pages, 2012.

[28] Y. C. Hu, A. Perrig, and D. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, no. 2, pp. 1976–1986, IEEE, San Francisco, Calif, USA, 2003.

[29] L. Hung, S. Lee, Y.-K. Lee, and H. Lee, "SCODE: a secure coordination-based data dissemination tomobile sinks in sensor networks," *IEICE Transactions on Communications*, vol. E92-B, no. 1, pp. 131–142, 2009.

[30] H. A. Nguyen, H. Guo, and K.-S. Low, "Real-time estimation of sensor node's position using particle swarm optimization with log-barrier constraint," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3619–3628, 2011.

[31] V. G. Kulkarni, "Shortest paths in networks with exponentially distributed arc lengths," *Networks*, vol. 16, no. 3, pp. 255–274, 1986.

[32] A. El-Zaart and D. Ziou, "Statistical modelling of multimodal SAR images," *International Journal of Remote Sensing*, vol. 28, no. 10, pp. 2277–2294, 2007.

[33] J. Wu, C. K. Tse, F. C. M. Lau, and I. W. H. Ho, "Analysis of communication network performance from a complex network perspective," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 12, pp. 3303–3316, 2013.

[34] M. H. DeGroot and M. J. Schervish, *Probability and Statistics*, Pearson Education, 2010.

[35] T. Nie, Z. Guo, and K. Zhao, "The dynamic correlation between degree and betweenness of complex network under attack," *Physica A: Statistical Mechanics and Its Applications*, vol. 457, pp. 129–137, 2016.

[36] L. Tian, C.-P. Zhu, D.-N. Shi, Z.-M. Gu, and T. Zhou, "Universal scaling behavior of clustering coefficient induced by deactivation mechanism," *Physical Review E*, vol. 74, no. 4, Article ID 046103, pp. 1–7, 2006.

[37] X. Chen, L. Liang, G. Xu, and D. Liu, "Feature extraction of kernel regress reconstruction for fault diagnosis based on self-organizing manifold learning," *Chinese Journal of Mechanical Engineering*, vol. 26, no. 5, pp. 1041–1049, 2013.

[38] S. S. Ram, V. V. Veeravalli, and A. Nedic, "Distributed non-autonomous power control through distributed convex optimization," in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM '09)*, pp. 3001–3005, IEEE, Rio de Janeiro, Brazil, 2009.

[39] M. Zhu and S. Martinez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.

[40] T. C. Aysal, B. N. Oreshkin, and M. J. Coates, "Accelerated distributed average consensus via localized node state prediction," *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1563–1576, 2009.

[41] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[42] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, 2009.

[43] J. Liu and J. Yang, "Action recognition using spatiotemporal features and hybrid generative/discriminative models," *Journal of Electronic Imaging*, vol. 21, no. 2, Article ID 023010, 2012.

[44] H. Sun, C. Wang, and B. Wang, "Hybrid generative-discriminative human action recognition by combining spatiotemporal words with supervised topic models," *Optical Engineering*, vol. 50, no. 2, Article ID 027203, 2011.

[45] Ø. Birkenes, T. Matsui, K. Tanabe, S. M. Siniscalchi, T. A. Myrvoll, and M. H. Johnsen, "Penalized logistic regression with HMM log-likelihood regressors for speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1440–1454, 2010.

[46] B. He, H. Liu, Z. Wang, and X. Yuan, "A strictly contractive Peaceman-Rachford splitting method for convex programming," *SIAM Journal on Optimization*, vol. 24, no. 3, pp. 1011–1040, 2014.

[47] M. Kang, J. Kim, J.-M. Kim, A. C. C. Tan, E. Y. Kim, and B.-K. Choi, "Reliable fault diagnosis for low-speed bearings using individually trained support vector machines with kernel discriminative feature analysis," *IEEE Transactions on Power Electronics*, vol. 30, no. 5, pp. 2786–2797, 2015.

[48] M. H. Xu and T. Wu, "A class of linearized proximal alternating direction methods," *Journal of Optimization Theory and Applications*, vol. 151, no. 2, pp. 321–337, 2011.