Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9974279, 18 pages
https://doi.org/10.1155/2021/9974279

Research Article

Multiagent Minimum Risk Path Intrusion Strategy with Computational Geometry

Jianguo Sun, Zining Yan, and Sizhao Li

College of Computer Science, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Sizhao Li; sizhao.li@hrbeu.edu.cn

Received 19 March 2021; Revised 26 May 2021; Accepted 21 June 2021; Published 9 July 2021

Academic Editor: Ding Wang

In wireless sensor networks (WSNs), inefficient coverage does affect the quality of service (QoS), which the minimum exposure path (MEP) is traditionally used to handle. But intelligent mobile devices are generally of limited computation capability, local storage, and energy. Present methods cannot meet the demand of multiple target intrusion, lacking the consideration of energy consumption. Based on the Voronoi diagram in computational geometry, this paper proposed an invasion strategy of minimum risk path (MRP) to such a question. MRP is the path considered both the exposure of the moving target and energy consumption. Federated learning is introduced to figure out how to find the MRP, expressed as $C(t_i, t_j) = f(E, e)$. The value of $C(t_i, t_j)$ can measure the success of an invasion. At the time when a single smart mobile device invades, horizontal federated learning is taken to partition the path feature, and a single target feature federated (SPF) algorithm is for calculating the MRP. Moreover, for multi smart mobile device invasion, it has imported the time variable. Vertical federated learning can partition the feature of multipath data, and the multi-target feature federated (MFF) algorithm is for solving the multipath MRP dynamically. The experimental results show that the SPF and MFF have the dominant advantage over traditional computational performance and time. It primarily applies the complex conditions of a massive amount of sensor nodes.

1. Introduction

Wireless sensor networks (WSNs) composed of many sensors are widely used in many fields, such as building monitoring, intelligent transportation system (ITS), and enemy status report [1, 2]. What mainly affects network quality is coverage, especially the WSN barrier coverage problem [3]. According to the coverage classification, barrier coverage is one of the three coverage strategies to detect unauthorized intruding behaviour in monitored areas. Moreover, its quality measurement factors are mainly breaching, supporting, or exposure of the penetration path [4]. To reduce the risks of being found, invaders can make effective path planning and minimize the exposure [5] through intelligent algorithm.

Minimum exposure path (MEP) [6] is a significant way that assesses the coverage effect of WSN to optimize the management of WSN. By finding out the MEP, the defender can estimate where the sensor network coverage is weak because the invader crossed the sensing field along this path is the most difficult to detect [7]. Therefore, this paper will study how to find an optimal invasion path from the perspective of intruders.

Meguerdichian et al. [8] put forward the Voronoi diagram numerical solution combined with WSN and solved the MEP problem by the shortest path algorithm. Chechik et al. [9] defined the exposure by the number of nodes adjacent to the moving target, considering single-path connectivity to solve the MEP problem.

In this paper, considering the perspective from invaders, the monitoring area is modelled by the Voronoi diagram that divided the field into undirected graphs that are a set of segmented linear components to limit the optimal searching space.

Previous studies have shown that for intruders, the threat can be viewed as the probability of being detected by the sensor nodes when passing through the WSN, namely, the exposure. Exposure is the expected average ability of intrusion targets moved in the monitored area. Mathematically, the invader's exposure time and sensor field intensity should be taken into account when calculating the exposure degree.

Exposure can be formulated as a path integral when the sensor field intensity is accumulated along a path from the source point to the destination point during a time interval.

According to the Cannikin Law [10], the reliability of the penetration path depends not only on its total exposure degree but also on the exposure of the discrete edges that make up the path. And when the invader passes through the monitoring area in some circumstances, it will be limited by time, distance, exposure, and energy consumption. Therefore, it is not accurate to find the path with the minimum total exposure degree. It is necessary to find an approach that is more in line with actual needs. The minimum risk path (MRP) is called to sum up the above paths in this paper.

However, most of the researchers consider only one single intruder when solving the intrusion path. In actual situations, such as the Underwater Listening Network, a system collects underwater sounds and tracks and monitors the Navy's passing ships and submarines. Underwater listeners are installed on the seafloor and act as sensor nodes. The moving target is an autonomous underwater vehicle (AUV) that needs to pass through the monitoring area. Multiple AUV intruders cross the sensing field simultaneously and cooperate with each other, as shown in Figure 1. Invader 1 and invader 2 need to walk along two different paths to collect more information as far as possible, which will add to the complexity of path planning.

By contrast with the traditional method, federated learning allows multiple data owners to establish a shared model [11] with the protection of local data. It is conducive for updating the global data model dynamically to path planning. Yang et al. [12] introduced a federated learning framework of comprehension and security in 2019. Zhu and Jin [13] optimized the structure of neural network models by multiobjective evolutionary algorithms while minimizing communication costs and global model testing errors. Sodhro et al. [14] proposed a dynamic method of forward-center by adjusting the running time of the sensing and transmission process in the Internet of Things devices, which allocates resources with effectiveness and fairness.

The horizontal federated learning applies to that the data features of participants overlap more, and the sample IDs are less. For a single intrusive target, the edges of the Voronoi diagram generated of WSN can be seen as a data set. The evaluation indicators of exposure and energy consumption can be regarded as a feature space. The intersection of the feature space is massive between the data sets. Hence, a data set can be divided into horizontal sections. It aggregates different data in the same feature space to train a model for the ideal path of the moving target. The above single intrusion target pathfinding algorithm combined with federated learning is called SPF.

The vertical federated learning is suitable to the case of the ID of the training samples of participants overlap more and the data features less. Moving intrusion targets can be considered as data sets of this time. The feature space is the exposure and the energy consumption, which is the current position of the moving target location at the moment. Consequently, the different features of the common samples of multiple participants (intrusion targets) can be trained by vertical federated learning, which increased the feature dimensions of samples to build a multiobjective path planning model. The dynamic pathfinding algorithm for multiple intrusion targets based on federated learning is called MFF.

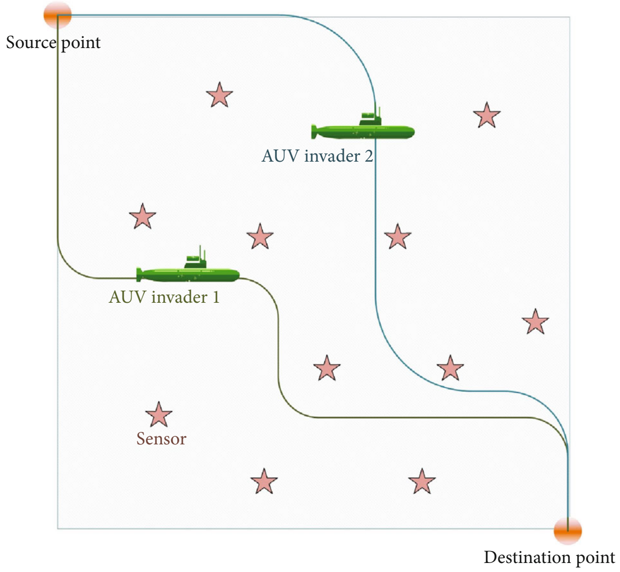


Figure 1: Multitargets: AUVs.

In summary, the main contributions of this paper are as follows.

(1) Considering the influence of various factors on the path, propose the concept of a minimum risk path

(2) Innovatively combine federated learning with path planning to build an intrusion algorithm and find out the MRP

(3) The optimal path is worked out through a dynamically segmented solution of the generated model. Experimental results show that the SPF algorithm can reduce computational complexity and improve performance

(4) Based on the vertical federated learning, it established a dynamic path planning model and MFF algorithm for multiple intrusion targets that make full use of multiobjective programming characteristics. Augmenting additional requirements can work out diverse path combinations

The rest part of this paper is as follows. The second part is the model building process. The detail of the proposed algorithm is described in the third section. In the fourth section, the experimental results are discussed. We formulate the evaluation indexes of the intrusion path planning algorithm in the fifth section. The final part is the summary of the whole paper.

2. Model Aggregation Based on Feature-Partitioned

2.1. Preliminaries. The sensing model determines the scope, and monitoring capability can be a usage of abstracting the sensing range of the same class nodes. In a bounded region F, presumed n active sensor nodes $s_1, s_2, \cdots, s_n$ would be deployed in predetermined locations. They can detect the target T appearing in any point within the sensing field. The detected sensing signal attenuates with the increasing Euclidean distance between the sensor node $s_i (i = 1, \cdots, n)$ and the target T. The specific *sensing model* for detecting the target T signal can determine the exposure of the target which moves along the path.

With the increasing transmission distance, the practically environmental noise interference and signal strength will attenuate. The detection probability of the node tapers as the distance grows between the moving target and the sensor node [15]. The detection capability of sensor nodes shows the uncertainty that attenuated disk perception model reflected.

On this basis, Rai and Daruwala [16] studied the coverage problem under the attenuated disk perception model. The detection probability $p(u, s_i)$ of s_i against the point u in region F can be expressed as in equation (1). In this paper, we selected the order type attenuated disk perception model.

$$p(u, s_i) = \lambda \cdot d(u, s_i)^{-K}. \quad (1)$$

Euclidean distance between u and s_i is $d(s_i, u)$. λ and K are the positive parameters relevant to the sensing capability. λ depicts the energy factor transmitted or reflected by the target, and K represents the path attenuation exponent. In standard conditions, the value of K is in a range of constant between 2 and 5.

In region F, u is at any point, and s_m is the sensor node with the shortest Euclidean distance to point u. The maximum-sensor intensity function $I_C(F, u)$ indicates the effective sensing measurement of s_m to u. Function I_C value of point P can be expressed as in (2) while the freely moving target P runs to the situation of u.

$$\begin{aligned} s_{\min} &= s_m \in S \mid d(s_m, u) \le d(s, u) \forall s \in S, \\ I_C(F, u) &= p(u, s_{\min}). \end{aligned} \quad (2)$$

t is the time variable. When the target P is moving at a constant speed v, (2) can be converted into

$$E(t_i, t_j) = v \cdot \int_{t_i}^{t_f} I(F, u) dt. \quad (3)$$

Exposure is a quantitative expression of network coverage performance, and the definition varies with the changing of application environments. Binh et al. [17] reckoned that exposure is the ability of WSN to detect objects passing through the sensing region. Aravinth et al. [18] defined exposure as the probability of detecting a target in the sensing region.

This paper endues two factors to explicate exposure: target moving time and induction intensity. It is noticeable that time accumulation affects induction intensity as the target passing through the sensor area. In the sensor region F, targets moving along the path over a while, its exposure degree can be described as (4).

$$E(t_i, t_j) = \int_{t_i}^{t_j} I(F, u) \left| \frac{du}{dt} \right| dt. \quad (4)$$

The threat model in this paper is a quantitative description of the optimal invasion path for intruders. The model can analyze the situation of the WSN, quantify the capability of the sensor nodes, and plan the invasion path of the adversary to the whole WSN reasonably. Specifically speaking, we firstly simulate the distribution of sensor network nodes to generate the network topology, and then, the intrusion path is planned according to the quantized threat data. A definition for the threat model is determined.

Definition 1. Given the overwhelming probability that an intruder can successfully penetrate the entire network if the value of $E(t_i, t_j)$ is as small as possible.

The concept of the Voronoi diagram is derived from computational geometry. It divides a plane into multiple regions by a collection of points called generators [19]. Considering a convex Euclidean domain F and a set Q of points $q_0, q_1, \cdots, q_n$ in F, Voronoi regions related to set Q are defined as (5) where $d(p, q)$ is the Euclidean distance between p and q.

Equation (5) means that a point $p \in F$ belongs to the Voronoi region $V(q_i)$, if the distance between p and q_i is the shortest distance between p and any other point $q_j \in Q$. The point $q_j \in Q$ is generating points or generators of the Voronoi partition.

$$V(q_i) = \bigcap_{j \neq i} \left\{ p \in F \mid d(p, q_i) < d\left(p, q_j\right), \forall q_j \in Q \right\}, \forall q_i \in Q. \quad (5)$$

In this paper, specific partitions (Voronoi diagram) can be produced according to the position of generating points (sensor nodes) in mission space (WSN) which separates sensor region F into multiple convex polygons. Each one contains a sensor node that its edge is called the *Voronoi edge*, and the intersection of *Voronoi edge* is the *Voronoi vertex*.

The characteristics of the Voronoi diagram played a key role. As a result, the minimum exposure path is right on the *Voronoi edge* precisely, under the maximum-sensor intensity. The Voronoi diagram or Voronoi partition that was generated via WSN consists of all the Voronoi cells of sensors.

Two types of methods can generate the Voronoi diagram. One is direct method that Voronoi diagram is generated from a set of points, such as half plane method, incremental construction method, divide and conquer method, plane scanning line method. Another is indirect which it takes the relationship between Voronoi diagram and the dual diagram of Delaunay triangle network. The point set is firstly subdivided to generate Delaunay triangle network and then to construct the Voronoi diagram.

If the corresponding Voronoi cells of sensor q_i and q_j have a same boundary, they become Delaunay neighbours and all the ones gathered to be an edge set of the Delaunay graph. Delaunay graph is dual to the Voronoi diagram, and consequently, one graph can be drawn from its dual counterpart. According to Table 1, triangulation generation method

Table 1: Comparison of Delaunay generation algorithms.

Algorithm	General condition	Worst condition
Lewis and Robinson	$O(n \log n)$	$O(n^2)$
Lee and Schachter	$O(n \log n)$	$O(n \log n)$
Dwyer	$O(n\text{loglog}n)$	$O(n \log n)$
Chew	$O(n \log n)$	$O(n \log n)$
Lawson	$O(n^{4/3})$	$O(n^2)$
Lee and Schachter	$O(n^{3/2})$	$O(n^2)$
Bowyer	$O(n^{3/2})$	$O(n^2)$
Watson	$O(n^{3/2})$	$O(n^2)$
Sloan	$O(n^{5/4})$	$O(n^2)$
Green and Sibson	$O(n^{3/2})$	$O(n^2)$
Brassel and Reif	$O(n^{3/2})$	$O(n^2)$
McCullagh and Ross	$O(n^{3/2})$	$O(n^2)$
Improved point-by-point insertion method (IPI)	$O(n \log n)$	$O(n \log n)$

is at the lowest time efficiency, point-by-point insertion method the middle, and divide-and-conquer algorithm the highest. The point-by-point insertion algorithm that owned high time efficiency can implement simply, occupying less space in operation.

The first part of the table from top to bottom is divide-and-conquer algorithm, the second part is point-by-point insertion algorithm, and the third part is triangulation generation method. This paper considering time and space efficiency, with the improved point-by-point insertion method (IPI), constructs Delaunay triangulation. The Voronoi diagram is structured upon the Delaunay triangle. By contrast to the point-by-point insertion algorithm, it can diminish the number of relevant point judgment and the computational complexity by arranging points first x and then y.

The specific construction process is shown in Algorithm 1.

V(WSN) is specific Voronoi diagram generated upon sensor network node distribution. Vertex corresponding of *V(WSN)* has a one-to-one relationship to each *Voronoi vertex* of the general Voronoi diagram *V(A)*. Vertexes in *V(WSN)* include the vertexes of *V(A)* and other points intersected the edges of *V(A)* and boundary of region *F*, as shown in Figure 2. The process of constructing *V(WSN)* is shown in Algorithm 2.

2.2. Parameter Calculation. The network flow is a specific flow solution that is closely related to linear programming [20]. This paper presents a new idea that edges have been defined as *Voronoi edges* and the flow as the moving target passed by this edge. The capacity can be considered as the maximum number of moving targets that each *Voronoi edge* can hold. Each node (not a source point or a sink point) is like an analogy to a *Voronoi vertex*, while the capacity is not limited.

Definition 2. The capacity network $D = (V, A)$ (V is the vertex of an undirected graph and A is the arc), and each arc (V_i, V_j) endues the transmission cost per unit moving $b_{ij} \geq 0$, denoted as

$$D = (V, A, B), b_{ij} \in B. \quad (6)$$

The maximum flow from source to sink can be found in the cost network D, and the total transfer cost of the stream is at a minimum.

Definition 3. If the moving target velocity was a constant v, the time increment dt would have a significant linear correlation between the unit length ds defined as

$$ds = v \cdot dt. \quad (7)$$

The energy consumption is defined below.

Definition 4. Supposing q is the energy consumption of unit length while target moving, the time variable is t and the time interval from V_i to V_j is $[t_i, t_j]$. In the sensor region F, the energy consumption of the target which moves along the path $r(t)$ at a constant speed v, is defined as

$$e(t_i, t_j) = \int_{t_i}^{t_j} vq dt. \quad (8)$$

The analytic discrete method can work out this problem, using a variational method to obtain the analytical solution to the risk optimization problem. Such a method simplifies original question to a set of differential equations about the optimal trajectory coordinates. How the trajectory is defined will make the complexity of the system.

```
Input: Vertex list (Vertices)
Output: A list of determined triangles (Triangles)
Initialise the Vertices
Create an index list indices
Sort the indices bases on the x-coordinates of Vertices
Get the super triangle (A super triangle means that the triangle contains all the points in the point set)
Save the duper triangle to the temp triangles list
Push the super triangles into Triangles list
for Each sample point in the Vertices list sorted by indices do
  Initialize the edge buffer
  for Each triangle in temp triangles do
      Calculate the center and radius of the triangle
      if The point is on the right side of the circumcircle then
          This triangle is a Delaunay triangle, save it in Triangles
          Remove the triangle from temp triangles list
          Skip
      else if The point is outside the circumcircle then
          This triangle is uncertain
          Skip
      else The point lies on the inside of the circumcircle
          This triangle is not a Delaunay triangle, add the three triangle edges to edge buffer
          Remove the triangle from temp triangles list
      end if
  end for
  Delete all repetitive edges in edge buffer
  Combine the edges in the edge buffer with the current point, and save these triangles into temp triangles
end for
Combine the temp triangles and Triangles
Remove triangles associated with super triangles vertices from Triangles list
```

Algorithm 1: IPI algorithm.

Presuming the coordinate of *Voronoi vertex* V_k is (x_{V_k}, y_{V_k}), and $t(p)$ is the path crossing time. The length of the edge $\langle V_i, V_j \rangle$ is expressed as $l_{V_iV_j}$.

The travel time along the edge $\langle V_i, V_j \rangle$ of the moving target is indicated by $t_{V_iV_j}$. $I_{V_iV_j}$ is the induction intensity of the nearest sensor nodes $s_1(s_2)$ when the target moves along the edge. $|d_{s_i,V_k}|$ is the distance from the sensor node s_1 (s_2) to the point V_k, and θ_{s_i,V_iV_j} is the angle of $\angle V_i s_1 V_j$. The coordinates of s_1 is (x_1, y_1); hence, $t_{V_iV_j}$ can be expressed as

$$t_{V_iV_j} = \frac{l_{V_iV_j}}{v} = \frac{\sqrt{\left(x_{V_j} - x_{V_i}\right)^2 + \left(y_{V_j} - y_{V_i}\right)^2}}{v}. \tag{9}$$

Analyzing the cost solution, suppose a sensor of Voronoi cells located in the coordinate system origin point, using the polar coordinate system. The relationship between Cartesian coordinate x, y, polar radius ρ, and polar angle Ψ is as in (10).

$$\begin{aligned} x(s) &= \rho(s) \cos \Psi(s), \\ y(s) &= \rho(s) \sin \Psi(s). \end{aligned} \tag{10}$$

For points V_i and V_j, polar radius ρ and angle θ can be expressed as in (11) and (12)

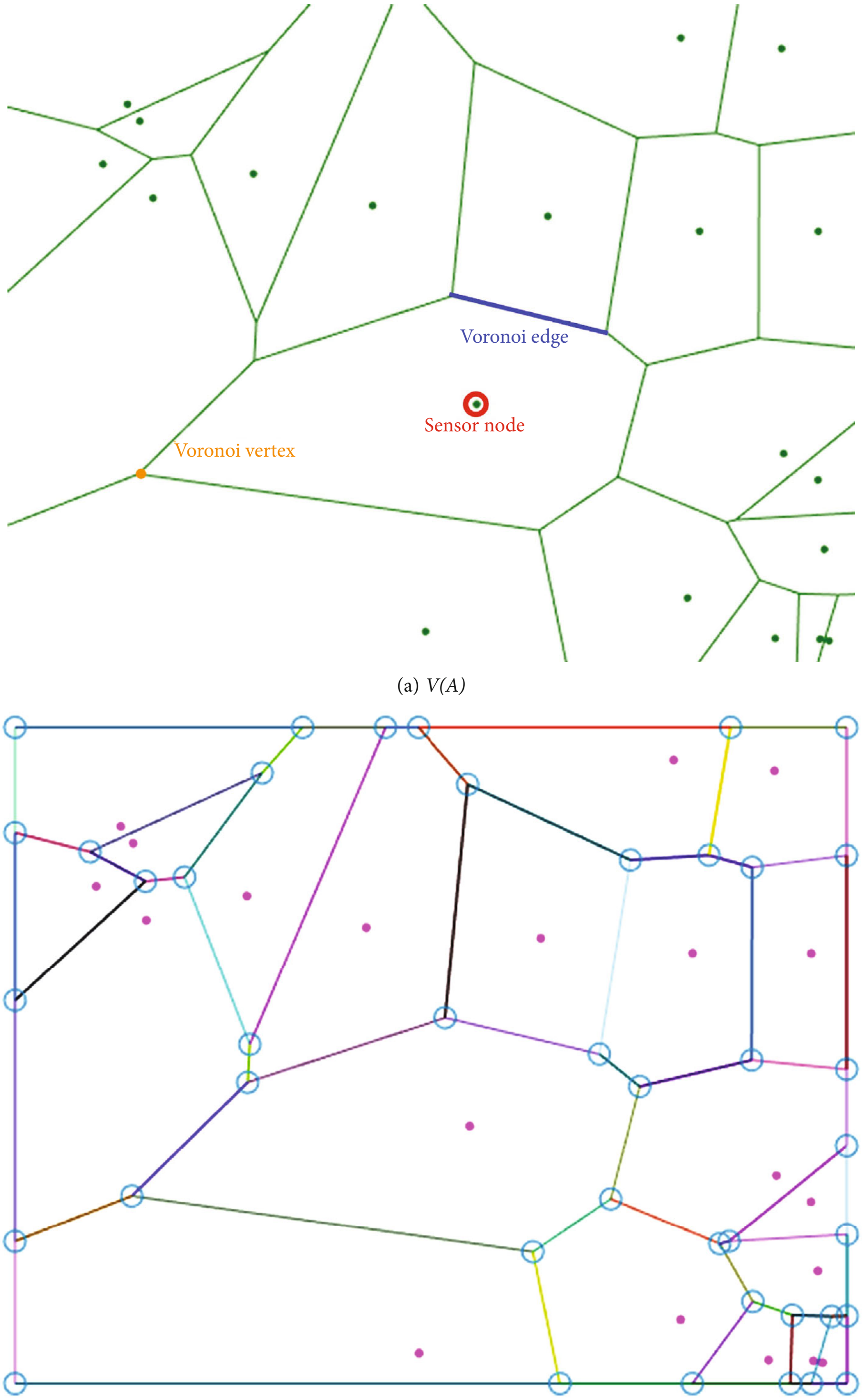
$$\begin{aligned} \rho_i &= \sqrt{x_i{}^2 + y_i{}^2}, \\ \rho_j &= \sqrt{x_j{}^2 + y_j{}^2}, \end{aligned} \tag{11}$$

$$\theta = \arccos \left(\frac{x_i x_j + y_i y_j}{\rho_i \rho_j} \right). \tag{12}$$

To derive the formula for the exposure index, it calculates the cumulative exposure of a moving target affected by s_1 from V_i to V_j. λ is the induction parameter determined by the sensor node hardware, and K is the distance impact factor with its value in common between 2 and 5. It has presumed $\lambda = 1, K = 2$ and induction intensity expressed as in (13)

$$\begin{aligned} v \cdot \int_{t_{V_a}}^{t_{V_b}} d^{-2} \left| \frac{dp}{dt} \right| dt = \int_{V_a}^{V_b} d^{-2} dp = \int_{x_{V_a}}^{x_{V_b}} \frac{\sqrt{1 + \left(y'_x\right)}}{(x - a_i)^2 + (y - b_i)^2} \, dx \\ = \frac{\left(\arctan\left(y_{V_b} - b_i / x_{V_b} - a_i\right) - \arctan\left(y_{V_a} - b_i / x_{V_a} - a_i\right)\right) \cdot l_{V_aV_b}}{(x_{V_a} - a_i)\left(y_{V_b} - b_i\right) - (x_{V_b} - a_i)\left(y_{V_a} - b_i\right)}. \end{aligned} \tag{13}$$

Vector $(x_{V_k} - x_1, y_{V_k} - y_1)$ can be indicated by d_{s,V_k}, and the angle between vector d_{s,V_i} and d_{s,V_j} can be indicated by θ_{s,V_iV_j} that meets the condition $0 \le \theta_{s,V_iV_j} \le \pi$. So, it can be simplified to (14)



(a) *V(A)*

(b) *V(WSN)*

Figure 2: Voronoi diagram of WSN.

$$v\cdot\int_{t_{V_i}}^{t_{V_j}} d^{-2}\left|\frac{ds}{dt}\right|dt = \frac{\theta_{s,V_iV_j}}{\sin\theta_{s,V_iV_j}}\cdot\frac{l_{V_iV_j}}{\left|d_{i,V_i}\right|\cdot\left|d_{i,V_j}\right|}. \tag{14}$$

The induction intensity is transformed into (15). The exposure can be calculated by (16), and the target moving total exposure $E(p)$ is (17).

$$I_{s,V_iV_j} = \frac{\theta_{s,V_iV_j}}{\sin\theta_{s,V_iV_j}}\cdot\left|d_{s,V_i}\right|^{-1}\left|d_{s,V_j}\right|^{-1}, \tag{15}$$

```
Input: Vertex list (Vertices), Region F
Output: V(WSN)
Initialize the Vertices
Create an edge list Voronoi edge
Use IPI algorithm to construct Delaunay triangular network based on Vertices
Number the discrete points and formed them into triangles. Then record three discrete points composed of each triangle
Calculate the center of each circumcircle of triangle and record it
for The triangle list do Find adjacent triangles TriA, TriB, and TriC that share side with the current triangle tempTri
  if Three adjacent triangles are found then
     Connect the outer center of TriA, TriB, and TriC to the outer center of tempTri. Put the connected edge into Voronoi edge list
  elseThree adjacent triangles are not found
     Find the outermost midperpendicular ray and put it into Voronoi edge list
  end if
end for
Calculate the point of intersection between Voronoi edge and region F
Connect intersection point according to F (acquire line of intersection)
V(WSN) is constructed by Voronoi edge and line of intersection
```

Algorithm 2: Construction of *V(WSN)*.

$$E_{s,V_iV_j} = I_{s,V_iV_j} \cdot l_{V_iV_j}, \tag{16}$$

$$E(p) = \sum_{i=c}^{j=f} E_{V_iV_j}. \tag{17}$$

3. Find the MRP Using Dynamic Programming and Federated Leaning

For the intrusion path selected, the primary goal is to avoid being detected by WSN. It is essential of thinking about how to select a path according to topology of WSN that meets the demands of the actual path seeking process. Therefore, the network topology data preprocessing is a very critical step. We can learn from federated learning, divide, and conquer.

The main idea of federated learning is to build machine-learning models based on data sets distributed over multiple devices [21]. Hao et al. [22] proposed an efficient scheme to solve sensitive data-driven industrial scenarios, and Lu et al. [23] formulate the data-sharing problem into a machine-learning problem via incorporating privacy-preserved federated learning. WSN can generate a significant amount of data [24]. According to the federated learning framework, we can perform data preprocessing on the WSN network topology. It should be noted that the research object of this paper is the static network, only using federated learning methodology, and there is no actual learning process.

Compared to the traditional method, federated learning is beneficial for dynamically updating the global data model on path planning. Thus, federated learning can deal with the MRP problem.

3.1. Single Mobile Device

3.1.1. The Calculation Model Established by Feature-Partitioned. Let matrix D_i denote data onto each *Voronoi edge*. Each row of the matrix represents a sample, and every column marks a feature. The feature space X includes exposure (E) and energy consumption (e), defining I the sample ID space (*Voronoi edge* number). The feature FE and sample Ids SI constitute the entire training data set (SI, FE).

Horizontal federated learning, also sample-based federated learning, was introduced in that data sets shared the same space in feature but distinct space in samples [25]. Horizontal federated learning is summarized as

$$FE_i = FE_j, SI_i \neq SI_j, \forall D_i, D_j, i \neq j. \tag{18}$$

The intersection of feature space is enormous between the data set of *Voronoi edges*. Then, the data set can be horizontally segmented, and varied data onto the same feature space can be aggregated. Consequently, it trains the model to solve a selection of the optimal path while the target is moving.

Step 1. Participants compute exposure locally with encryption and send results to the server.

Step 2. The server performs secure aggregation without learning information about any participant and calculates the average value of exposure.

Step 3. The server sends back the aggregated results to participants.

Step 4. Participants update their respective model with the decrypted results.

The primary goal of intrusion path is avoiding detection by WSN and then considering over the influence of energy consumption. In Step 2, the method of calculating the average exposure value of the entire network is as shown in (19), and l_{all} is the length of path.

$$E' = \frac{E_1 + E_2 + \cdots + E_n}{l_{\text{all}}}. \tag{19}$$

Therefore, the predicted exposure value $E_{\text{pre}}(V_i, V_j)$ from point V_i to point V_j can be expressed as (20)

$$E_{\text{pre}}\left(V_iV_j\right) = d_{V_iV_j} \cdot E'. \quad (20)$$

Because the length is the fundamental dimension of quantity, we can describe them by length L. According to (15) and (16), E is dimensionless parameter because the dimension of I is $1/L$ and the dimension of l_{V_i,V_j} is L. So, the dimension of E_{pre} is L. The dimension of e can be computed according to (8) in the same way. L is the dimension of e, too.

The time complexity of Voronoi graph generation was $O(n \log n)$ [26]. With Voronoi diagram $(3n-6)$ edges, accordingly it takes no more than $O(n)$ to calculate the exposure of all *Voronoi edges*. Based on dimensional analysis, the judgment function $C(t_i, t_j)$ can be defined as shown in (21). The value of $C(t_i, t_j)$ can measure the successfulness of an invasion.

$$C\left(t_i, t_j\right) = \left(\int_{t_i}^{t_j} E' \left|\frac{dp}{dt}\right| dt\right) + \left(\int_{t_i}^{t_j} q \left|\frac{dp}{dt}\right| dt\right). \quad (21)$$

3.1.2. The Flow of Algorithm and the Procedure of Realization. In actual path planning, combinations of edges that are possibly feasible flows can be various. Hence, it has to dynamically update the priority to the selected edges keeping the searching direction under control in planning an optimal path. Furthermore, the cost is optimal for each extended path. The valuation function of moving target from the starting node s via current node $X(x, y)$ to the end node f can be defined.

Definition 5. $G(X)$ is the total exposure value from the start node to the current node, which can be obtained by adding up the weights of any neighboring edge of the path that traveled. $H(X)$ is the heuristic estimated cost function (included exposure and energy consume) which is only as a judgment, not as a part of cost calculation. The total cost value $F(X)$ can be expressed as

$$F(X) = G(X) + H(X). \quad (22)$$

Considering an ideal condition that the target in motion is supposed to be increasingly closer towards the destination, according to the feature-partitioned of federated learning, $H(X)$ could be accordingly expressed as

$$H(X) = E' \cdot d_{Y,f} + e(t_X, t_Y). \quad (23)$$

Y is the antecedent node of X. $d_{Y,f}$ is the Euclidean distance from Y to the end node s, and $E' * d_{Y,f}$ expressed the exposure estimation from Y to f.

Define an empty table named $NotVis$ and another $Been$ Vis: nodes that have not been visited (need to be examined) are stored in the $NotVis$ table, and nodes that have been visited are stored in the $BeenVis$ table. The single intrusion target pathfinding algorithm combined with federated learning (SPF algorithm) is as Algorithm 3.

The SPF algorithm can output an optimal path by inputting the *V(WSN)* generated by the Voronoi diagram that contains the connected edges and the position coordinates of each node and the point identifier from the source point s to final point f.

3.2. Multiple Mobile Devices

3.2.1. The Calculation Model Established by Feature-Partitioned. It is sometimes a practical demand to transfer multiple moving targets over a time interval. The vertical federated learning is suitable for the ID of the training samples of participants overlap more and the data features less. The moving target is deemed to be a sample set SI. The feature space FE is formed into the moving target position, the moment of the current position, and exposure degree. Vertically federated learning is a process aggregating different features above, in which a model can be built with data from moving targets in cooperation. Under such assumption, (24) can be got.

$$FE_i \neq FE_j, SI_i = SI_j, \forall D_i, D_j, i \neq j. \quad (24)$$

Training the model can solve the multioptimal path of the moving target. Specific steps are as follows.

Step 1. Participants locally compute exposure and time and send masked results to the server.

Step 2. The server performs secure aggregation without learning information about any participant.

Step 3. The server sends back the aggregated results to participants.

Step 4. Participants update their respective model with the decrypted results.

Whereas varied positions of moving target A as time running would decide the direction of moving target B, time variable was imported over multiobjective path judgment. The probability of being detected by the sensor would be increasing (like a traffic problem [27]) as two moving targets on the same *Voronoi edge*. The presumption is that when target A is at *Voronoi edge* V_iV_j, if target B passing by edge V_iV_j, the heuristic estimated cost $H(B)$ of target B would be $H'(B)$.

$$H'_B = k \cdot H_B(k > 1). \quad (25)$$

The Underwater Sensor Network is a system that collects underwater sounds and tracks and monitors the Navy's passing ships and submarines. The underwater acoustic detectors (UADs) are installed on the seafloor and act as sensor nodes. The moving target is an autonomous underwater vehicle (AUV) that needs to pass through the monitoring area. For more information gathering and security reasons, it is a method to keep the AUV from going the same path as much as possible by adjusting the k value in (25).

3.2.2. The Algorithm Flow and the Procedure of Realization. When multiple intrusion targets need to be transmitted, a

```
Input: Graph V(WSN); s; f
Output: optimal path
NotVis = [s]; BeenVis=[];
while The NotVis table is not empty do
  Select $V_i$ with the lowest total proxy value $F$ from NotVis. Then delete $V_i$ from NotVis and insert $V_i$ into BeenVis.
  if $V_i$ is the end node $f$ then
    Return path
  else
    Extend node $V_i$
    for Each adjacent node $V_j$ of $V_i$ do
      Calculate the exposure $E_{V_iV_j}$.
      if $V_j$ is not in neither the NotVis or BeenVis table then
        Calculate $F(V_j)$
        Insert $V_j$ into the NotVis table. And add it a pointer variable pointing to node $V_i$.
      else if $V_j$ is in the NotVis table then
        if $F(V_j)$ is less that the estimated value $F'(V_j)$ in NotVis table then
          Update $F'(V_j)(F'(V_j) = F(V_j))$.
          Change the node pointer in the NotVis table to point to the current node $V_i$
        else
          if $F(V_j)$ is less than the estimates value $F'(V_j)$ in BeenVis table then
            Update $F'(V_j)(F'(V_j) = F(V_j) )$.
            Delete $V_j$ from BeenVis table and insert $V_j$ into NotVis table
          end if
        end if
      end if
      Insert $V_i$ into BeenVis table
    end for
  end if
end while
From $f$ backtrack to $s$. Return path.
```

Algorithm 3: SPF algorithm.

time-variable is introduced in this paper to conduct dynamic pathfinding of multiple intrusion targets based on federated learning feature (MFF). Assuming that multiple moving targets $M_1, M_2, \cdots, M_n$ start from s every time interval T, the $H(X)$ value is updated according to the formula (25) and the location of the moving target at different times. Based on the SPF algorithm, we need to create some new data structures to store variables in the MFF algorithm, as shown in Table 2.

The specific algorithm flow is as Algorithm 4. By contrast to the SPF algorithm for a single intrusion target, the MFF algorithm sets a time variable to record the positions of distinct targets moving every period, which position relationship can update data onto the model.

4. Experiment

In this section, the performance of the SPF and MFF algorithm is validated through simulation experiments. It analyzed the condition that finding MRP was influenced by distinct node quantities and various distributed modes in the sensing area and whether the algorithms are effective against different network topologies.

It is assumed that the network topology application scenario is an Underwater Sensor Network in this paper. The experimental environment is in an ideal state, as shown in Figure 3. The UADs are deployed on the seabed to form a monitoring area, and the AUVs move at a uniform speed in the same horizontal plane. From the starting point, the AUVs pass through the monitoring area to the endpoint to perform the underwater penetration mission of the submarine.

We plan to use underwater acoustic communication technology to solve information transmission and sharing among unmanned underwater clusters. In a multiagent case, clock synchronization between two agents is generally calculated by multiround communication to calculate the deviation and offset rate between clocks. According to the research of Liu et al. [28], several particular nodes named anchors are selected among all moving targets in this paper to aid the localization process.

The anchors are clock synchronized that the moving target achieves clock synchronization by communicating with anchors and estimating the message propagation time. The targeted target broadcasts the location demand, and the anchor reveals the location message containing the sending time and location information after receiving the location request. When the targeted target receives the location message, the transmission delay of the message can be calculated based on the local clock and the sending time in the news. Then, the distance to the anchor can be estimated [29].

Table 2: Comparison of Delaunay generation algorithms.

DetCost_Lines	The exposure value of each *Voronoi edge*
DetTime_Lines	The time required for the moving target to move along each *Voronoi edge* $(V_i V_j)$
Path_Cost	The respective cost and total cost of all departing moving targets at a given time
Path_FCost	The respective projected cost values of all departing moving targets at a given time
Path_Record _CurrentPoint	The name of the current point
Path_Record _PreviouPoint	The previous node name

```
Input: Graph V(WSN)
Output: optimal path
1: NotVis=[s]; BeenVis=[]
2: while The NotVis table is not empty do
3:   Select the path with the lowest total proxy value F from NotVis. And record the corresponding moving target serial number $i$.
4:   if All moving targets are the end node f then
5:     Return path
6:   end if
7:   Update DetCost_line according to (25) and path
8:   Sort Path_Time_CurrentPoint in ascending order, and record the moving targer serial number respectively. (Determine the moving target $M_j$ that reaches the next Voronoi vertex first)
9:   for All moving targets do
10:    if $M_i$ reaches the end point then
11:      Calculate path of $M_j$
12:    elseBreak;
13:    end if
14:  end for
15:  Make path prediction for $M_i$ which has minimum value of Path_Time_CurrentPoint by SPF algorithm
16:  Update Path_Record_CurrentPoint
17:  Update Path_Time_CurrentPoint
18:  Goto line 7
19: end while
```

Algorithm 4: MFF algorithm.

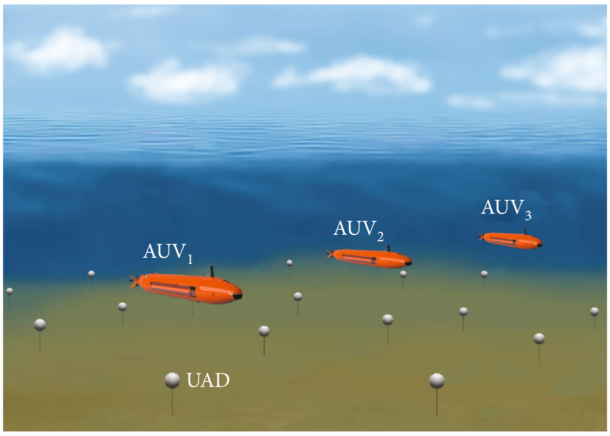


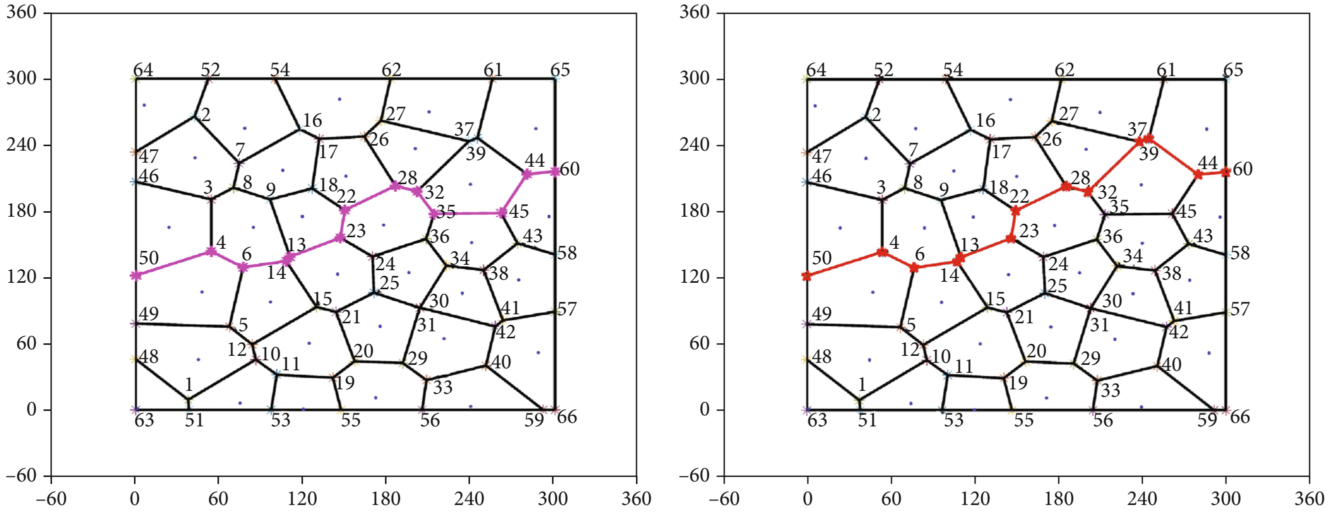
Figure 3: Underwater Sensor Network.

Table 3: Experimental parameters (1).

Parameter	Value
Monitoring area	300 ∗ 300
λ	1
K	2
q	0
k	2
n	32
Starting point	(0,120)
Ending point	(300,195)

4.1. Single Mobile Device. To verify the accuracy of the model and algorithm, we can figure out the minimum exposure path via the topology in [6]. The data manifested in Table 3 are the parameters in the experiment. And the results can be found in Figure 4. The number of sensor nodes is expressed by variable n.

Thirty-two sensors are deployed on a 300 ∗ 300 monitoring field. We used the HGA-NFE method [6] and SPF algorithm in this paper for path planning, respectively, as shown in Figure 4. Figure 4(a) reveals the MEP found based on the HGA-NFE method in [6]. The Voronoi nodes of the



(a) MEP based on [6]

(b) MRP based on SPF

Figure 4: Minimum exposure path.

path passed are [50,4,6,14,13,23,22,28,32,35,45,44,60], and the exposure value of this path is 104.1807. Figure 4(b) displays the MRP found based on the SPF algorithm. The Voronoi nodes of the path passed are [50,4,6,14,13,23,22,28,32,37,39,44,60], and the exposure value of this path is 104.1459. The lengths of MRP and MEP are both 379.07, but MRP has a smaller exposure value than the MEP, which is computed by the HGA-NFE method. It indicates that the SPF algorithm is more accurate than the HGA-NFE method [6].

Network topologies can be divided into three different network types according to the deployment of sensors. They are the uniform distribution method, Gaussian distribution method, and exponential distribution method [17]. We limited the distance relationship between nodes in the experiment to achieve all kinds of distribution. For studying the efficiency of the SPF, topologies were randomly generated according to the three network types mentioned. The calculation time of SPF under different topologies is shown in Figure 5.

The total computation time of MRP falls into two parts. One is the time to construct the Voronoi diagram, and the other is the time to find the path. We can find that the total computation time of MRP increases as the number of nodes increases because the time complexity of constructing the Voronoi diagram is related to the number of nodes. The total computation time comparison of the SPF algorithm under the three topologies is shown in Figure 5(d). It is no doubt that the SPF algorithm is more efficient under the topology of uniform distribution.

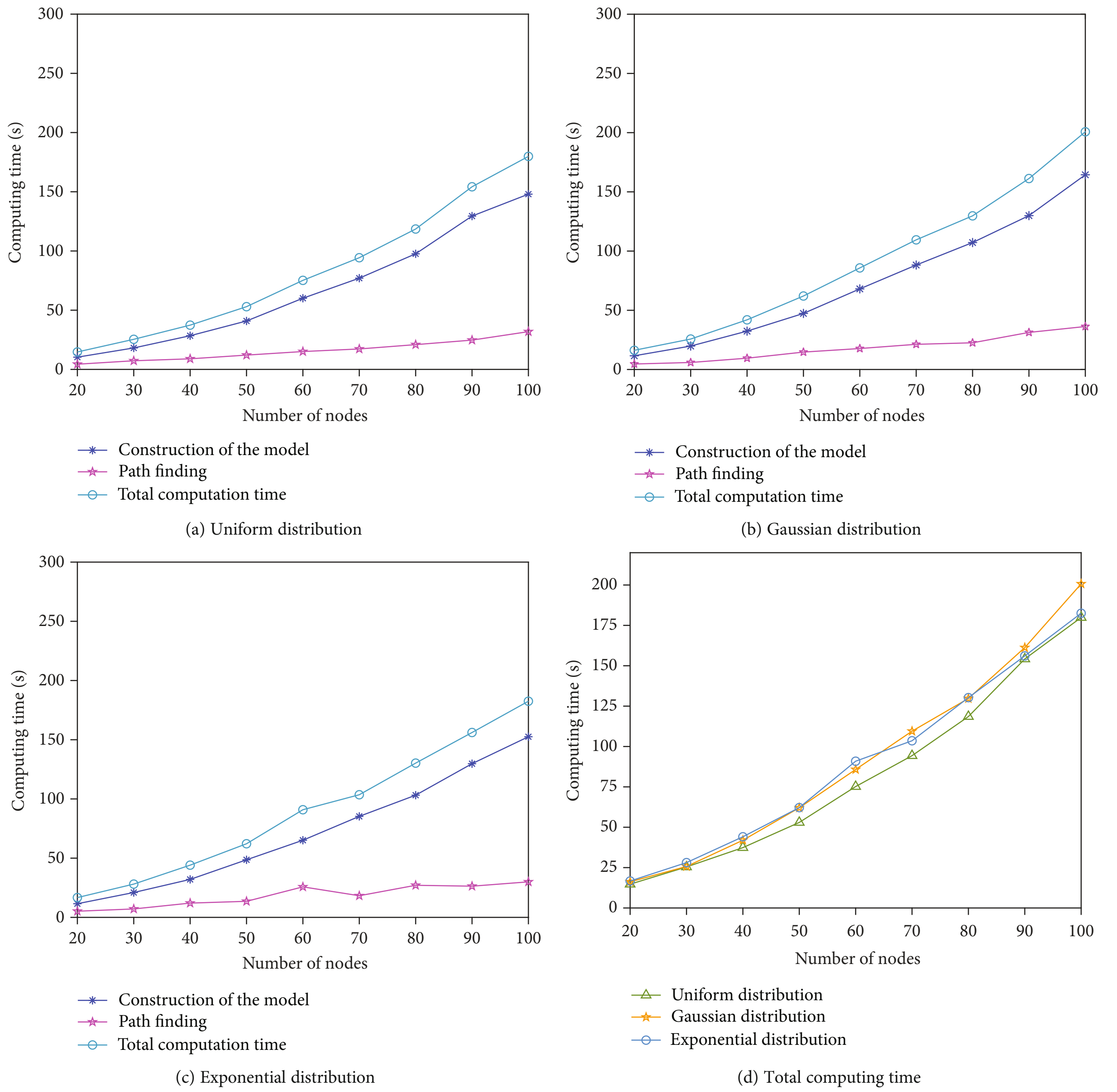
Binh et al. [17] proposed the GA-MEP algorithm for intrusion path planning. It is difficult to compare the time complexity of the SPF algorithm with the genetic algorithm GA-MEP [17] because the complexity of GA-MEP is not related to the number of sensor nodes. Hence, to test the efficiency of SPF, the calculation time of SPF and GA-MEP was compared. Figure 6 displays the results.

The computing time of the SPF algorithm grows slowly with the increase of sensor nodes in this area, which is better than the GA-MEP algorithm. It indicates that the SPF algorithm is more suitable for the condition of a large number of sensors. The calculation time of SPF is shorter than that of GA-MEP, and the SPF algorithm is more stable for different network topologies. In the case of the same intrusion path exposure, the efficiency of the algorithm proposed in this paper is far better than the GA-MEP algorithm [17].

To explore path-finding law limited by various constraints, experiments have been done on the uniform distribution network topology where $n = 20$ and $n = 40$, respectively. The selected parameters are put in Table 4. The results are shown in Figure 7. To explore path-finding law limited by various constraints, experiments have been done on the uniform distribution network topology where $n = 20$ and $n = 40$, respectively.

Figure 7 illustrates three simulation scenarios. The first one is only considering the impact of exposure on the path (as explicated in Figures 7(a) and 7(d)), which is transformed into find the minimum exposure path. The second solely thinks about the impact of the moving target energy consumption on the path (as shown in Figures 7(b) and 7(e)). Thus, it is equivalent to solving the minimal path problem. Thirdly, we consider both the exposure and energy consumption of the moving target (as revealed in Figures 7(c) and 7(f)).

We calculated the path exposure value and path length under three simulation conditions. The experimental results in Table 5 show that when only considering the effect of exposure on the path, the total exposure value of the path is the lowest. However, the length of path is the longest, which means that the invader has the highest energy consumption. When only considering the effect of energy consumption on the path, the length is the lowest, which means energy consumption is the lowest. However, the exposure degree increases accordingly. When considering both exposure and



(a) Uniform distribution

(b) Gaussian distribution

(c) Exponential distribution

(d) Total computing time

Figure 5: Computing time of MRP in 3 topologies.

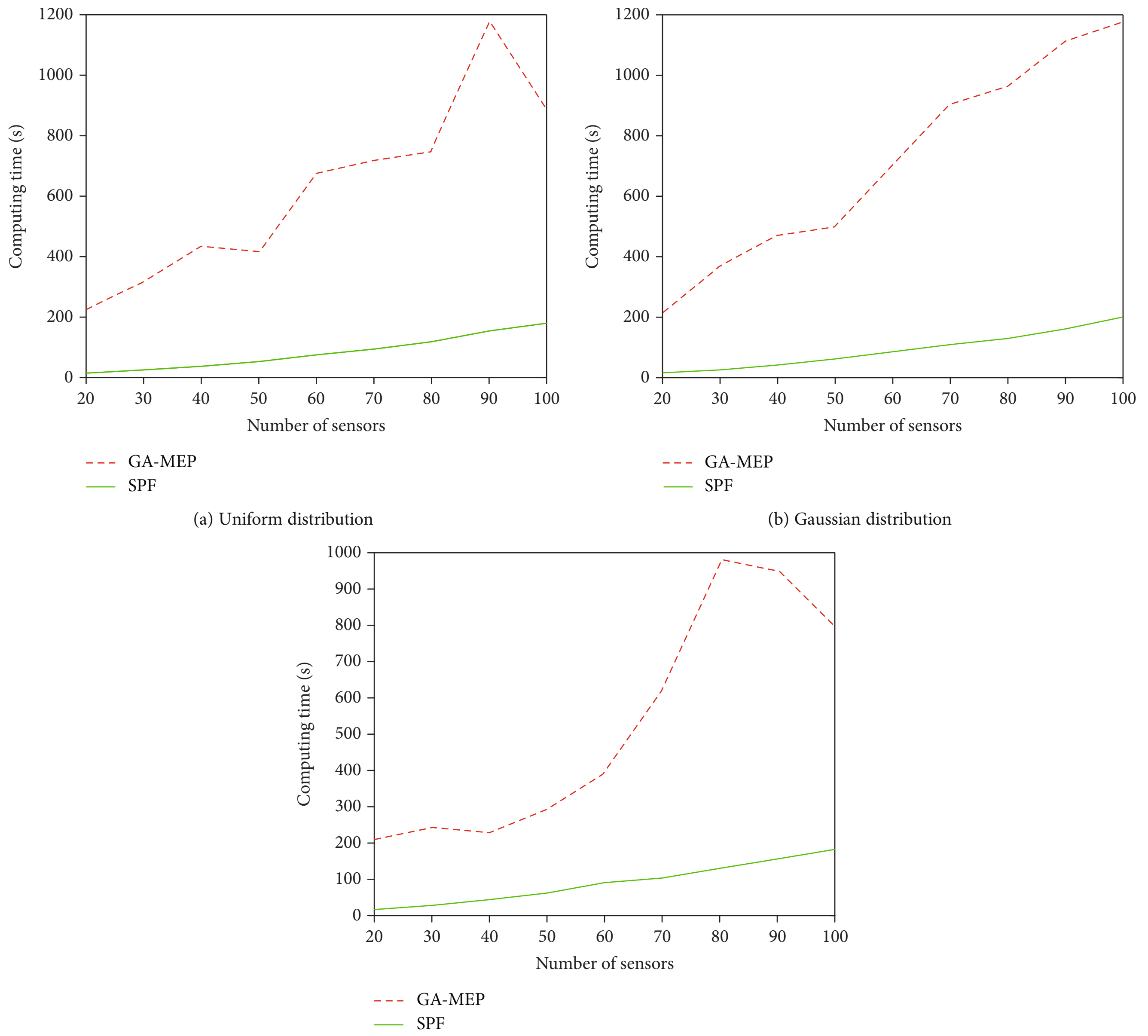
energy consumption, the relevant parameters of the path in the data model generated by federated learning are between the above two results. It conforms to the hypothesis, proving that the SPF algorithm proposed can calculate the path according to different requirements.

The quality of MRP relates to the path's exposure, and it was affected by the amount of sum and various parts composed. Thus, we compared the maximum exposure value of parts $E_{\max}$ of the path, and the results can be found in Table 6.

It can be seen in the column that the maximum exposure value of part of MRP obtained by the SPF algorithm is the same as that of MEP. But it takes less time and energy when an invader moves along MRP. To test the universality of this circumstance, we also calculated MRP with different node numbers. Relevant experimental data is in Table 7.

According to Tables 6 and 7, we can get a conclusion. When calculating MRP by SPF algorithm, there is a 42.86% probability of getting a path with the same maximum exposure value of MEP but shorter travel time and less energy consumption. The simulation test was carried out under the parameter settings displayed in Table 4.

Figure 8 describes the relationship between MRP exposure (travel time) and the number of sensor nodes with different q values. With the augmentation of sensor nodes,



(a) Uniform distribution

(b) Gaussian distribution

(c) Exponential distribution

Figure 6: Comparison between SPF and GA-MEP.

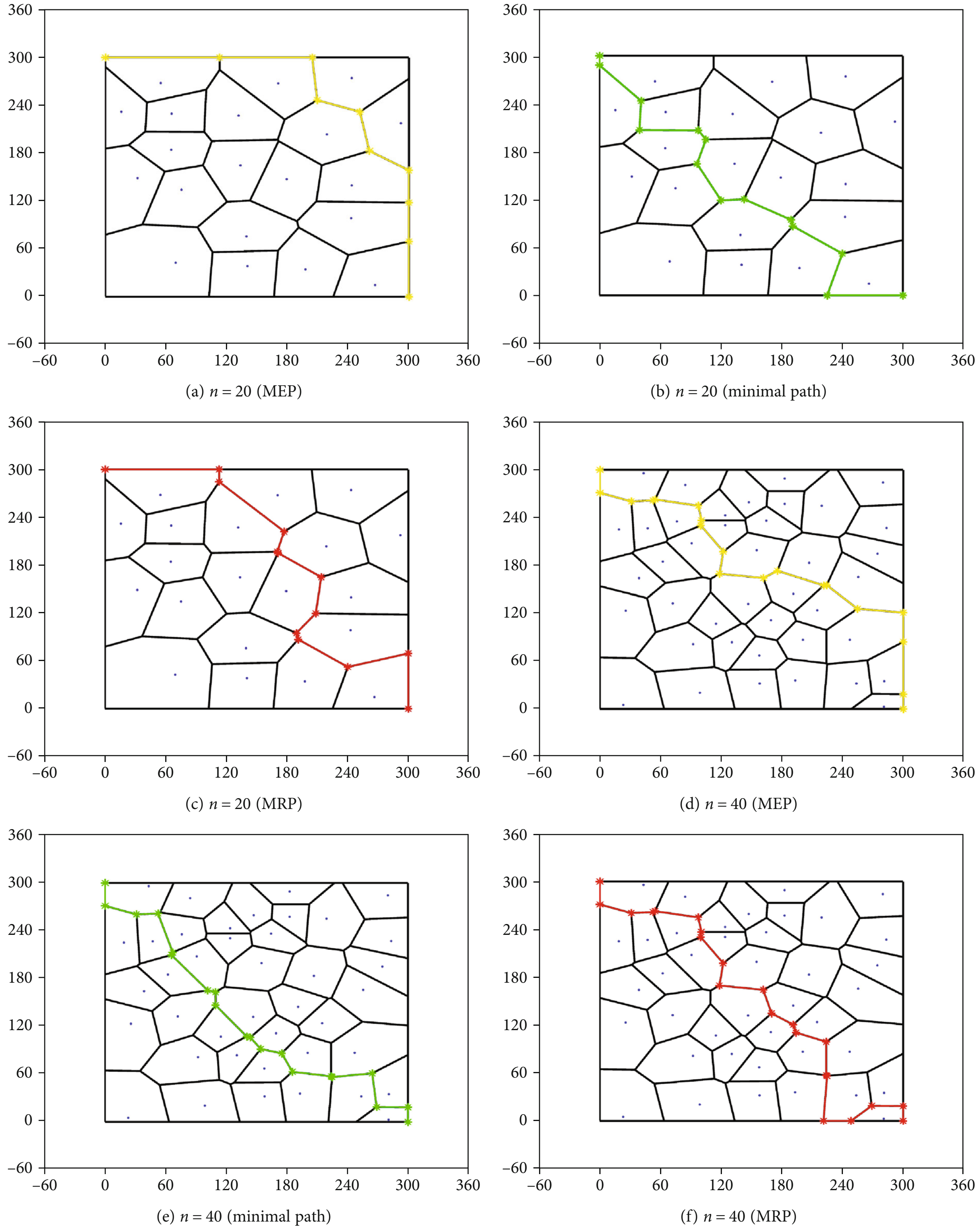
Table 4: Experimental parameters (2).

Parameter	Value
Monitoring area	300 ∗ 300
λ	1
K	2
q	0,50
k	2
Number of sensor nodes	20,40
Starting point	(0,300)
Ending point	(300,0)

the exposure degree keeps an upward trend and the intrusion time has a specific change.

In the identical topologies, we can adjust the model created by federated learning to get the path (MRP) with less exposure and shorter intrusion time. It is conductive to save the energy consumption and reduce the probability of being detected by the sensor nodes.

4.2. Multi Mobile Devices. In terms of multi-objective, a simulation was made for the verification of MFF performance. The value k represents the change of the edge when two moving targets are on the same edge. In the tangible situation, k can be adjusted according to different requirements for obtaining different path combinations.



(a) $n = 20$ (MEP)

(b) $n = 20$ (minimal path)

(c) $n = 20$ (MRP)

(d) $n = 40$ (MEP)

(e) $n = 40$ (minimal path)

(f) $n = 40$ (MRP)

Figure 7: Different paths when $n = 20$ and $n = 40$.

Assume that there are four intruders with the same starting point (0,300) and the same ending point (300,0). Each moving target sets off every 0.6 seconds and $k = 2$. The results are shown in Figure 9, which show the four paths found when 40 nodes distributed evenly in F. The four intruders took a total of 12.32 seconds to travel from starting point to the ending point, with a total exposure degree of 678.58. And the exposure of each path is 162.13, 176.27, 162.13, and 178.05. It can be found that the trajectory of the third target moving coincides with

Table 5: Experimental results.

	Exposure	Length of path
$n = 20$ (MEP)	91.42	557.31
$n = 20$ (minimal path)	156.70	536.49
$n = 20$ (MRP)	94.72	551.91
$n = 40$(MEP)	183.93	616.32
$n = 40$ (minimal path)	244.53	498.75
$n = 40$ (MRP)	200.96	535.20

Table 6: Experimental results.

	$E_{\max}$(exposure)	Length of path
$n = 20$	19.90	21.62
$n = 40$	36.93	36.93

Table 7: Exposure and length of path.

	$E_{\max}$(exposure)	$E_{\max}$(both)	l(exposure)	l(both)
$n = 60$	37.56	37.56	505.6	500.55
$n = 80$	31.07	43.10	524.55	510.60
$n = 100$	45.08	45.08	534.84	523.86
$n = 120$	43.27	47.76	563.91	539.76

the first as time going by. The result is identical when n equals 60.

What can be explained is that the first moving target starts first and moves farther during the same period without affecting the path planning of subsequent moving targets. Therefore, when $k = 2$, the first moving target trajectory is likely to be the same as the trajectory of the third moving target.

The experiment is carried out in this section when $n = 40$ so as to study the influence of departure time interval of moving targets on multiobjective path planning. Each intruder sets off at a time interval of 0.4, 0.6, 0.8, and 1.0, and the relevant experimental data are shown in Table 8. The maximum and minimum intrusion time of a single target is Time(max) and Time(min), the maximum exposure of a single target is Exp(max), and the minimum exposure of a target is Exp(min).

It can be seen from Table 8 that with the increase of the time interval of sending moving target, the total travel time increases continuously. But the total exposure decreases continuously until a minimum value and then increases accordingly. This is because the longer a target stays in the WSN region, the greater the risk probability detected by the sensor. When the time interval reaches a certain peak, cross-path interference factors are excluded, and the greater the total exposure. The minimum exposure and time consumption of a single target do not change with a time interval because the MRP of a single target is the path obtained by the SPF algorithm.

5. Performance Evaluation

Well-defined evaluation criteria and adversary models put the pathfinding schemes on common WSN topology, making it possible to assess the scenarios fairly and comprehensively [30–32]. The adversary model in this paper refers to different types of WSN topologies. Sensor nodes can detect moving targets but will not attack them. Invaders require finding one or more paths through the sensor area. Security, availability, efficiency, and deployability are important evaluation indexes for intrusion detection systems [33–36]. Similarly, we select some evaluation properties [34, 35] as the evaluation indexes of the intrusion path planning algorithm.

(A) Security

(i) *S1 Exposure*. For an intruder, its security is reflected in the exposure of the invasion path mostly. Lower exposure means a safer moving target.

(ii) *S2 Energy Consumption*. An ideal invasion path for an intruder requires low exposure and minimal energy consumption as the intruder moves along the trail. Otherwise, the moving target is likely to run out of energy on the way so that it cannot fulfill the task.

(B) Availability

(i) *A1*. It can be used in WSN with a Boolean disk perception model.

(ii) *A2*. It can be used in WSN, whose perception model is the probabilistic perception model.

(iii) *A3*. It can be used in WSN with an attenuated disk perception model.

(C) Deployability

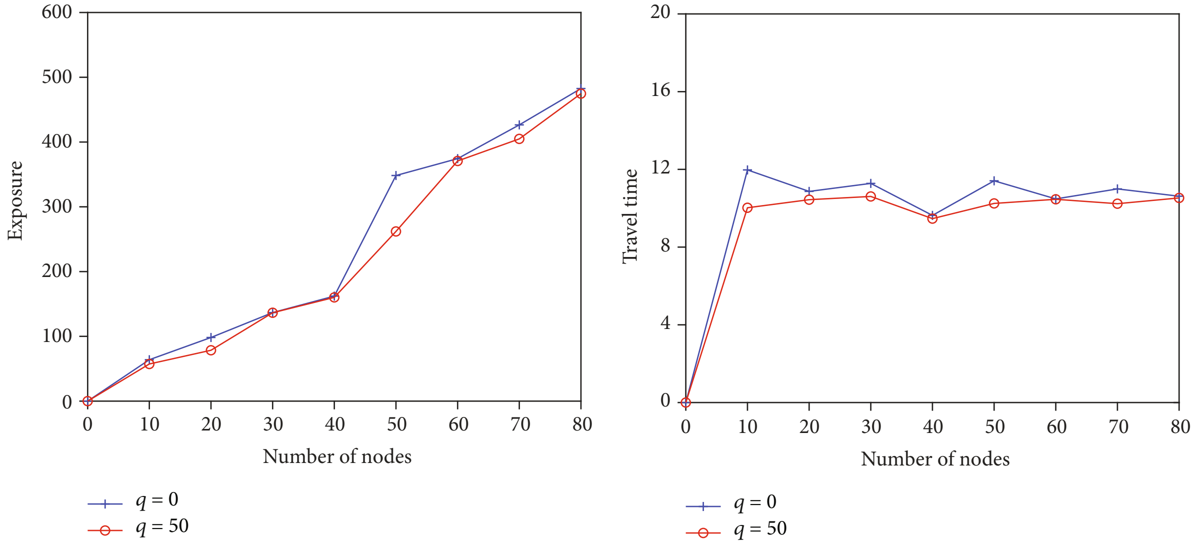
(i) *D1*. Whether it is suitable for large-scale networks. In actual application scenarios, the scale of WSN is often large, so if we want to expand the application algorithm, it is imperative to apply it to large-scale networks.

(ii) *D2*. Whether it considered multiple moving targets. The "moving target" is not only a single target. When the intrusion of a single target expands to multiple target intrusion, whether the algorithm is still effective also needs to be included in the evaluation.

(iii) *D3*. Whether multiple sensor network topologies are considered.

(iv) *D4*. Algorithmic efficiency, which is mainly compared from the time complexity and running time. It is divided into two levels, "excellent" and "good."

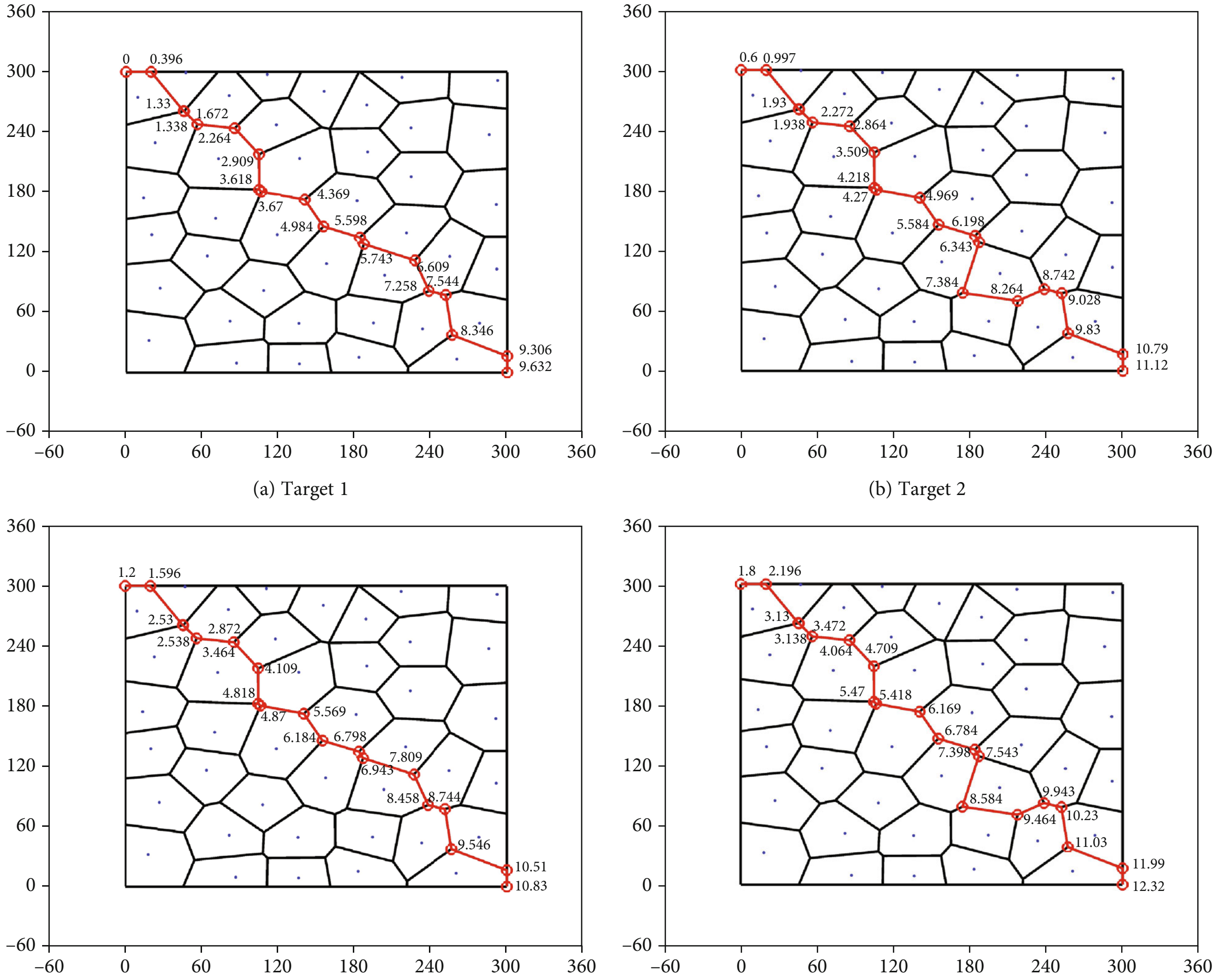
(v) According to the above criteria, we evaluate the algorithm proposed in this paper and the existing methods. The results are summarized in Table 9.



(a) Exposure

(b) Time

Figure 8: Exposure and travel time of different sensor nodes.



(a) Target 1

(b) Target 2

(c) Target 3

(d) Target 4

Figure 9: Paths for four intruders when $n = 40$ and $T = 0.6$.

Table 8: Exposure and length of path.

	$T = 0.4$	$T = 0.6$	$T = 0.8$	$T = 1.0$
Travel time	11.85	12.32	13.38	13.56
Total exposure	727.35	716.35	677.08	815.96
Time(max)	11.04	11.1	11.01	11.13
Time(min)	10.14	10.17	10.17	10.17
Exp(max)	206.26	232.68	214.01	217.18
Exp(min)	127.01	127.01	122.01	189.54

Table 9: Measuring security, availability, and deployability of algorithm.

Scheme		No.	Security		Availability			Deployability			
			S1	S2	A1	A2	A3	D1	D2	D3	D4
Meguerdichian et al.	2001	[8]	✓	✗	○	○	✓	✗	✗	✗	☆
Song et al.	2014	[37]	✓	✗	○	○	✓	✗	✗	✓	☆
Ye et al.	2016	[6]	✓	✗	○	✓	○	✓	✗	✗	☆
Binh et al.	2019	[17]	✓	✗	○	✓	○	✓	✗	✓	★
Aravinth et al.	2021	[18]	✓	✓	✓	○	○	✓	✗	✗	★
Our scheme			✓	✓	○	○	✓	✓	✓	✓	★

Note: ✓ means achieving the corresponding goal, while ✗ not. ○ does not mean anything. ☆ means "good," and ★ means "excellent.".

6. Conclusions

This paper built a data model of the dynamic-planning algorithm to keep the risk of multi-intrusion targets minimized, which utilizes computational geometry methods amalgamated with federated learning features. The experiment results show that the algorithm that introduced time variables can actualize the global optimization. By contrast with traditional planning ways, its computational performance prevails in complexity and time. $H(x)$ weights by $C(t_i, t_j)$ can be adjusted to meet various demands of path selection.

The MRP exposure of all the discrete paths distinctly declined in comparison with MEP. MRP can markedly lower the energy consumption without being detected by the sensor nodes as much as possible. Consequently, the algorithm accurately satisfies the real-life conditions, such as the battlefield crossing scenarios. The model has an inspired significance to the sensor network deployment.

The common application of multiple types of sensors in WSN becomes increasingly popular, which increases the path analysis complexity. Hence, it is necessary to analyze more actual models of WSN deployment to explore further. Besides, fulfilling federal learning in three-dimensional space to make path planning is a new direction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of Heilongjiang Province of China (JJ2019YX0922) and Basic Science Research Plan (JCKY2020208B045). We thank laboratory team members for assistance with the experiments.

References

[1] J. Pan, F. Fan, S. Chu, Z. du, and H. Zhao, "A node location method in wireless sensor networks based on a hybrid optimization algorithm," *Wireless Communications and Mobile Computing*, vol. 2020, 14 pages, 2020.

[2] W. Fang, W. Zhang, W. Chen, T. Pan, Y. Ni, and Y. Yang, "Trust-based attack and defense in wireless sensor networks: a survey," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 2643546, 20 pages, 2020.

[3] H. Mostafaei, M. U. Chowdhury, and M. S. Obaidat, "Border surveillance with WSN systems in a distributed manner," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3703–3712, 2018.

[4] S. Murali and A. Jamalipour, "A lightweight intrusion detection for Sybil attack under mobile RPL in the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 379–388, 2020.

[5] C. Ryan, F. Murphy, and M. Mullins, "Spatial risk modelling of behavioural hotspots: risk-aware path planning for autonomous vehicles," *Transportation Research Part A: Policy and Practice*, vol. 134, pp. 152–163, 2020.

[6] M. Ye, Y. Wang, C. Dai, and X. Wang, "A hybrid genetic algorithm for the minimum exposure path problem of wireless sensor networks based on a numerical functional extreme model," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8644–8657, 2015.

[7] L. Liu, X. Zhang, and H. Ma, "Minimal exposure path algorithms for directional sensor networks," *Wireless Communications and Mobile Computing*, vol. 14, no. 10, p. 994, 2014.

[8] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, “Exposure in wireless ad-hoc sensor networks,” in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 139–150, Rome, Italy, 2001.

[9] S. Chechik, M. Johnson, M. Parter, and D. Peleg, “Secluded connectivity problems,” *Algorithmica*, vol. 79, no. 3, pp. 708–741, 2017.

[10] T. Chang, D. Kong, N. Hao, K. Xu, and G. Yang, “Solving the dynamic weapon target assignment problem by an improved artificial bee colony algorithm with heuristic factor initialization,” *Applied Soft Computing*, vol. 70, pp. 845–863, 2018.

[11] P. Yu and Y. Liu, “Federated object detection: optimizing object detection model with federated learning,” in *ACM International Conference Proceeding Series*, pp. 1–6, Vancouver BC, Canada, August 2019.

[12] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: concept and applications,” *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[13] H. Zhu and Y. Jin, “Multi-objective evolutionary federated learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1310–1322, 2020.

[14] A. H. Sodhro, S. Pirbhulal, and V. H. C. De Albuquerque, “Artificial intelligence-driven mechanism for edge computing-based industrial applications,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4235–4243, 2019.

[15] J. Chen, J. Li, and T. H. Lai, “Trapping mobile targets in wireless sensor networks: an energy-efficient perspective,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3287–3300, 2013.

[16] N. Rai and R. Daruwala, “Node density optimisation using composite probabilistic sensing model in wireless sensor networks,” *IET Wireless Sensor Systems*, vol. 9, no. 4, pp. 181–190, 2019.

[17] H. T. T. Binh, N. T. M. Binh, N. H. Ngoc, D. T. H. Ly, and N. D. Nghia, “Efficient approximation approaches to minimal exposure path problem in probabilistic coverage model for wireless sensor networks,” *Applied Soft Computing*, vol. 76, pp. 726–743, 2019.

[18] S. S. Aravinth, J. Senthilkumar, V. Mohanraj, and Y. Suresh, “A hybrid swarm intelligence based optimization approach for solving minimum exposure problem in wireless sensor networks,” *Concurrency and Computation: Practice and Experience*, vol. 33, no. 3, article e5370, 2021.

[19] Y. J. Liu, D. Fan, C. X. Xu, and Y. He, “Constructing intrinsic Delaunay triangulations from the dual of geodesic Voronoi diagrams,” *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–15, 2017.

[20] F. Engmann, F. A. Katsriku, J. D. Abdulai, and K. S. Adu-Manu, “Reducing the energy budget in WSN using time series models,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8893064, 15 pages, 2020.

[21] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, “VerifyNet: secure and verifiable federated learning,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2019.

[22] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, “Efficient and privacy-enhanced federated learning for industrial artificial intelligence,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6532–6542, 2020.

[23] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, “Blockchain and federated learning for privacy-preserved data sharing in industrial IoT,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.

[24] S. Rani, S. H. Ahmed, R. Talwar, and J. Malhotra, “Can sensors collect big data? An energy-efficient big data gathering algorithm for a WSN,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1961–1968, 2017.

[25] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, “Federated learning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[26] A. Driemel, S. Har-Peled, and B. Raichel, “On the expected complexity of Voronoi diagrams on terrains,” *ACM Transactions on Algorithms*, vol. 12, no. 3, pp. 1–20, 2016.

[27] A. Hilmani, A. Maizate, and L. Hassouni, “Automated real-time intelligent traffic control system for smart cities using wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8841893, 28 pages, 2020.

[28] J. Liu, Z. Wang, M. Zuba, Z. Peng, J.-H. Cui, and S. Zhou, “DA-sync: a Doppler-assisted time-synchronization scheme for mobile underwater sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 582–595, 2013.

[29] T. Ojha, S. Misra, and S. Obaidat, “SEAL: self-adaptive AUV-based localization for sparsely deployed underwater sensor networks,” *Computer Communications*, vol. 154, no. 1, pp. 204–215, 2020.

[30] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K. K. R. Choo, “Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9390–9401, 2020.

[31] D. Wang and P. Wang, “Two birds with one stone: two-factor authentication with security beyond conventional bound,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2016.

[32] C. Wang, D. Wang, G. Xu, and D. He, “Efficient privacy-preserving user authentication scheme with forward secrecy for industry 4.0,” *Science China Information Sciences*, 2020.

[33] S. Huang and K. Lei, “IGAN-IDS: an imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks,” *Ad Hoc Networks*, vol. 105, article 102177, 2020.

[34] J. Bonneau, C. Herley, P. Oorschot, and F. Stajano, “The quest to replace passwords: a framework for comparative evaluation of web authentication schemes,” in *2012 IEEE Symposium on Security and Privacy*, pp. 553–567, San Francisco, CA, USA, May 2012.

[35] D. Wang, W. Li, and P. Wang, “Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.

[36] M. Eskandari, Z. Janjua, M. Vecchio, and F. Antonelli, “Passban IDS: an intelligent anomaly-based intrusion detection system for IoT edge devices,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, 2020.

[37] Y. Song, L. Liu, H. Ma, and A. Vasilakos, “A biology-based algorithm to minimal exposure problem of wireless sensor networks,” *IEEE Transactions on Network and Service Management*, vol. 11, no. 3, pp. 417–430, 2014.