The document is about a novel multi-class classification method called the voting-cross support vector machine (SVM) method. This method was proposed to improve the classification accuracy for multiple vehicle targets in wireless sensor networks. The method combines the advantages of the directed acyclic graph SVM (DAGSVM) method and the binary-tree SVM method. The study used datasets from a real-world experiment to validate the proposed method. The results showed that the cross-voting SVM method could effectively increase the classification accuracy for multiple vehicle targets, with a limited increase in the algorithm complexity, compared to the DAGSVM method. The method also decreased time consumption by approximately 70% compared to the DAGSVM method.

**Keywords**: multi-class classification; cross-voting SVM method; vehicle classification; wireless sensor networks (WSNs)
There are numerous new research studies on the first type of method, including the MC-SVMA model [8], the K-class support vector classification-regression (K-SVCR) method [9], the learning vector quantization (LVQ) method [10], the improving prototype selection k-nearest neighbor (PS-KNN) method [11], Bayesian network-based chain classifiers [12], the BPSO-AdaBoost-KNN ensemble learning algorithm [13], and the hyper-sphere multi-class SVM (HSMC-SVM) method [14]. Among these, the algorithm that improved over the SVM principle for classifying multiple targets is the most widely applied. In addition to the MC-SVMA method and HSMC-SVM method mentioned above, the second-order cone programming support vector machines (SCOPSVM) method [15] and the least squares twin K-class support vector classification (LST-KSVC) method [16], are two other multi-class classification methods based on the SVM method.

The multi-class classification algorithm based on the two-class SVM classifier is the most widely used in the second category of multi-class classification methods, which includes the DAGSVM method [17], the one-against-all SVM (OAA-SVM) method [18], the binary-tree SVM [19] method, the hierarchical binary-tree multi-class SVM (BTMSVM) method [20], and decision tree DAGs [21].

Of the algorithms mentioned above, a few can classify vehicle type in WSNs by using the acoustic signal, seismic signal, or magnetic signal of the target acquired by a single sensor node in WSNs. The first category of multi-classification methods is based on the improvement of the two-class classifier, and substantially increases the computational complexity of the algorithm. The limited storage space and computing capacity of the WSN node, hinder the implementation of the complex of classification algorithm on the node’s chip, and therefore the first type of algorithm, which is marginally more complicated, and is unable to apply for classifying vehicle types. For example, even when running on a computer, the HSMC-SVM algorithm takes 5–7 s to classify 1397 datapoints using trained classifiers [14], whilst LST-KSVC takes about 73 s to classify 5000 datapoints [16]. For WSN nodes, the complexity of the second type of methods are acceptable. For example, in this paper, the time taken by DAGSVM to classify averaging about 1060 16-dimensional vectors of target signals is 1.12 s, and the time consumed by the binary-tree to classify averaging about 1060 16-dimensional vectors is 0.28 s. The new algorithm aims to improve the accuracy of target classification, whilst rendering the increment of the computational complexity acceptable.

The rest of this paper is organized as follows. After introducing the feature extraction method, wavelet coefficient energy rate (WCER) method [22], and the dataset utilized for validating the proposed method, Section 2 compares the performances of commonly-used two-class classifiers. Section 3 compares the two available precursor algorithms of multi-class classification algorithms. Based on this comparison, Section 4 introduces the proposed voting-cross algorithm for vehicle classification in WSNs. The performance of the method is then discussed in Section 5, and Section 6 concludes the paper.

2. Comparison of Two-Class Classifiers

Prior to exploring the multi-target classification algorithm for WSN systems, we compare the two-class classification algorithms, which is the basis of multi-classification algorithms. Moreover, the wavelet coefficient energy ratio (WCER) feature extraction method was utilized to validate the classification algorithms, and the source datasets of targets were acquired from the third SensIT situational experiment [23].

2.1. Datasets and Feature Extraction Method

The datasets utilized in this study were gathered from a real-world experiment, the third SensIT situational experiment. In the experiment, 75 sensor nodes with acoustic, seismic, and infrared sensing capability were located at the Marine Corps Air Ground Combat Center. Testing runs were performed by driving different types of vehicles across the testing field. Four target vehicle-classes were used: Assault amphibian vehicle (AAV), Main battle tank, High-mobility multipurpose wheeled vehicle, and the Dragon wagon (DW). The acoustic and seismic data of each run were recorded, at a signal
sensing rate of 4096 Hz. There were three paths in the experimental area for the vehicles to pass, and a testing run was accomplished by driving the vehicle on the path. A series of numbers was utilized to indicate these runs, e.g., DW3, DW4, and DW5. The acoustic and seismic data of each run were recorded by the sensor nodes deployed at the side of the road. The details of the experiment are described in literature in Reference [23].

The WCER method, based on wavelet decomposition, was utilized for extracting the feature of the target signal after a specified length of signal sequence was acquired. Assuming that the feature of the target signal sequence was extracted at every K points, the process of the feature extraction method based on WCER is shown in Figure 1.

Figure 1. Process of feature extraction using wavelet coefficient energy ratio (WCER).

![Figure 1](image)

2.2. Comparison of Performance of Different Two-Class Classifiers

The commonly used two-class classification methods are the k-nearest neighbor (KNN) method [24], the DT method, the NB method, the adaptive boosting (AdaBoost) [25] method, and the SVM method. We used the seismic feature data of AAV3 and AAV4 to train the classifiers and classify
the seismic feature data of the other 15 runs (AAV5–AAV11 and DW5–DW12). Then, the average classification accuracy and time required were recorded to compare the performance of these methods, in terms of vehicle classification.

The classification accuracy is denoted as $Ac$. We evaluated the accuracy as follows:

$$ Ac = \frac{\#\text{correctly classified data}}{\#\text{total testing data}} \times 100\% $$ (1)

The classification accuracy of the $i$th dataset is denoted as $Ac(i)$ ($i = 1, 2, \ldots, N$), and the average classification accuracy, $Aca$, is

$$ Aca = \frac{1}{N} \sum_{i=1}^{N} Ac(i) $$ (2)

where $N$ is the number of datasets (acquired from different targets) that have been classified.

The classification results of the classification methods in Table 1, reveals that the classification accuracy of KNN is the highest in the two-class classification; however, its time consumption is excessive. The time consumptions of NB and DT for target classification were the lowest; however, the classification accuracy was low. Compared to the other classifiers, the SVM classifiers and AdaBoost exhibited reasonable performance; the classification was more accurate and required less time for subsequent classifications. The SVM classifier was more suitable for classifying the detected vehicle signals in the WSNs.

In practical applications, multiple types of targets would appear in the WSN system. Therefore, it is necessary to study multi-target classification method based on two-class SVM classifiers.

### Table 1. Average classification accuracy and time required for different classification methods using acoustic features. KNN—k-nearest neighbor; DT—decision tree; NB—naïve Bayes; AdaBoost—adaptive boosting; SVM—support vector machine.

| Method    | Average classification accuracy | Average time required for classification |
|-----------|---------------------------------|------------------------------------------|
| KNN       | 72.22%                          | 1.4783 s                                 |
| DT        | 52.82%                          | 0.0024 s                                 |
| NB        | 65.71%                          | 0.0102 s                                 |
| AdaBoost  | 69.83%                          | 0.0743 s                                 |
| SVM       | 71.03%                          | 0.1440 s                                 |

3. Comparison of Multi-Classes SVM Classifier

As summarized in Section 1, the available SVM algorithms for multi-target classification are divided into two categories. The first category improves on the principle of two-class classifier classification, for application to multi-target classification. Another multi-target classification category utilized multiple two-class classifiers to classify multiple targets.

3.1. M-RLP SVM Method

As shown in Figure 2a, under ideal conditions, the two-class SVM method is aimed at searching for the optimal parameter that provides the largest distance between the margins of the datasets of two targets, according to the sample dataset. This is described mathematically as follows:

$$ \min \left( \frac{1}{2} w^T w \right) $$ (3)

subject to

$$ y_i \left( w^T x_i + c \right) \geq 1, \quad i = 1, \ldots, N $$ (4)

where $w$ is the parameter of the classification hyperplane, $x_i$ is the classification sample point on the boundary, and $y_i$ is the label, and its value is either $-1$ or $+1$. $T$ is the matrix transpose symbol and the constant $c$ is the distance from the origin to the hyperplane.
Considering the mixed situation between the two types of sample datasets, the two-class SVM for dataset classification is aimed at solving the following optimization problem in practical applications:

$$\min \left( \frac{1}{2} \mathbf{w}^T \mathbf{w} + R \sum_{i=1}^{N_t} \epsilon_i \right)$$

(5)

under the following constraints:

$$1 - \epsilon_i - y_i \left( \mathbf{w}^T \mathbf{x}_i + c \right) \leq 0, \epsilon_i \geq 0, i = 1, \cdots, N_t$$

(6)

where \( R \geq 0 \) is the penalty factor, indicating the importance attached to the outliers, and \( \epsilon_i \) is the slack variable introduced when the \( i \)th sample data is linearly inseparable.

Figure 2 shows a multi-class SVM (MC-SVM) approach named Multi-class Robust Linear Programming (M-RLP) \([26]\), which was proposed based on the two-class SVM classifier. It is different from the univariate quadratic optimization problem in the case of the two-class classification using SVM classifier, and is aimed at solving the following multivariate quadratic optimal programming problems, when using the MC-SVM method to classify multiple types of targets:

$$\min \left( \frac{1}{2} \sum_{i=1}^{K} \sum_{j=1}^{i-1} \left\| \mathbf{w}_i - \mathbf{w}_j \right\|^2 + \frac{1}{2} \sum_{i=1}^{K} \left\| \mathbf{w}_i \right\|^2 \right)$$

(7)

subject to

$$( \mathbf{w}_i - \mathbf{w}_j )^T \cdot \mathbf{x}_i - (b_i - b_j) - 1 \geq 0 \quad i, j = 1, 2, \cdots, K, i \neq j$$

(8)

where \( \mathbf{w}_i \) and \( \mathbf{w}_j \) are the parameters of the piecewise linear margin. \( K \) is the number of data categories to be classified, and the specific derivation process is available in the literature in Reference \([26]\). This method utilizes a piecewise nonlinear classification function constructed by a single quadratic program to classify a multi-class dataset.

The MC-SVM method improved from principle, rapidly increases the algorithm complexity. For example, the computational complexity of the two-class SVM classifier is \( O(n) \), whereas that of the multi-class SVM classifier is \( O(n^M) \), where \( M \) is the number of categories of targets to be classified. For WSN nodes with insufficient computing capacity, a multi-target classification algorithm based on multiple two-class classifiers is more appropriate.
3.2. DAGSVM and Binary-Tree SVM

The more representative multi-classification algorithms based on two-class classifiers are DAGSVM and Binary-Tree SVM. These two methods use multiple two-class classifiers cooperatively, to classify multiple types of targets.

The DAGSVM method constructs a directed acyclic graph that resembles a normal binary tree. The one-against-one classifier, trained at all the nodes shown in Figure 3, was used to classify the unclassified dataset, and finally determine the category of the dataset to be classified by cycle comparison.

Considering the multi-target classification of four types of targets as an example, and denoting the four types of targets as A, B, C, and D, the classification process of the DAGSVM algorithm is illustrated in Figure 3.

![Figure 3. Architecture of directed acyclic graph SVM (DAGSVM) method.](image-url)

The number of one-against-one classifiers required by the DAGSVM multi-classification algorithm is $C^2_N = N \times (N - 1)/2$. The parameter N is the number of target types. The classification accuracy of the DAG-SVM algorithm is significantly affected by the classification accuracy of the one-against-one classifier, at the key node in the directed acyclic graph. The classification accuracy of A is denoted as $P_{ABCD}(A)$, which is also the probability of accurately determining A from A, B, C, and D. Similarly, the classification accuracy of B, C, and D are denoted as $P_{ABCD}(A)$, $P_{ABCD}(B)$, and $P_{ABCD}(D)$, respectively. The formulas for calculating these values are as follows:

$$P_{ABCD}(A) = P_{AD}(A) \cdot P_{AC}(A) \cdot P_{AB}(A)$$  \hspace{1cm} (9)

$$P_{ABCD}(B) = P_{AD}(A|B) \cdot [P_{AC}(A|B) \cdot P_{AB}(B) + (1 - P_{AC}(A|B)) \cdot P_{BD}(B) \cdot P_{BC}(B)]$$  \hspace{1cm} (10)

$$P_{ABCD}(C) = P_{AD}(D|C) \cdot [P_{BD}(D|C) \cdot P_{CD}(C) + (1 - P_{BD}(D|C)) \cdot P_{BC}(C)]$$  \hspace{1cm} (11)

$$P_{ABCD}(D) = P_{AD}(D) \cdot P_{BD}(D) \cdot P_{CD}(D)$$  \hspace{1cm} (12)

where $P_{AD}(A)$ is the classification accuracy of A in the one-against-one classifier (A/D), and $P_{AC}(A)$ is the classification accuracy of A in the one-against-one classifier (A/C). $P_{AD}(A|B)$ represents the probability that A is classified as B when using the one-against-one classifier (A/D), and $P_{BD}(D|C)$ represents the probability that C is classified as D when using the one-against-one classifier (B/D). The classification accuracy calculation Formulas (9)–(12), reveal that the target classification accuracy of the DAGSVM algorithm depends on the category number N, and the classification accuracy of the two-class classifiers at the node of the directed acyclic graph. Top-down error accumulation affects the classification performance of this method. For example, the classification accuracy of A ($P_{ABCD}(A)$) is determined by the classification accuracy of three two-class classifiers ($P_{AD}(A)$, $P_{AC}(A)$, and $P_{AB}(A)$), and the classification error of each two-class classifiers will reduce the final classification accuracy $P_{ABCD}(A)$. The most important factor affecting the classification accuracy of the two-class classifier...
is the geometric spacing between the training features of the two types of targets. The larger the geometric spacing is, the smaller the upper bound of the classification error rate is, i.e., the higher the classification accuracy that can be achieved by optimizing the parameter method.

3.3. Binary-Tree SVM

The binary-tree SVM classifier is also composed of multiple two-class SVM classifiers. Different from the DAGSVM method, the binary-tree SVM classifier first classifies the several types of targets into two major categories, and then continues to classify each of the major categories into two classes, until each class contains only one type of target. Considering the multi-target classification of four types of targets as an example, and denoting the four types of targets as A, B, C, and D, the classification process of binary-tree SVM is shown in Figure 4.

As shown in Figure 4, the first two-class classifier \((A \cup C) / (B \cup D)\) is trained using the combined features, after the features of the targets A and C are combined to obtain the combination features \((A \cup C)\), and the features of B and D are combined to obtain the combination features \((B \cup D)\). The target features to be classified are first sent to the two-class classifier \((A \cup C) / (B \cup D)\) for classification. After the classification result of the first classifier is obtained, assuming it to be \((A \cup C)\), the result is further sent to the two-class classifier \((A)\) for the next level of classification.

The number of one-against-one classifiers required by the binary-tree SVM algorithm is \((N - 1)\), where \(N\) is the number of target types. In this study, we classified four targets, and therefore, three two-class classifiers are required here. The calculation formulas for all the types of targets are as follows:

\[
P_{ABCD}(A) = P_{AUC,BUD}(A \cup C|A) \cdot P_{AC}(A)
\]

\[
P_{ABCD}(B) = P_{AUC,BUD}(B \cup D|B) \cdot P_{BD}(B)
\]

\[
P_{ABCD}(C) = P_{AUC,BUD}(A \cup C|C) \cdot P_{AC}(C)
\]

\[
P_{ABCD}(D) = P_{AUC,BUD}(B \cup D|D) \cdot P_{BD}(D)
\]

where \(P_{AUC,BUD}(A \cup C)\) represents the probability that target A or C is classified as \(A \cup C\), when using the two-class classifier \((A \cup C) / (B \cup D)\). \(P_{AC}(A)\) denotes the classification accuracy of target A, when using the two-class classifier \((A)\).

The classification accuracy of the binary tree SVM method is as shown in Equations (13)–(16). As illustrated by the equations, combining multiple types of data into two major categories and sending it into the two-class SVM classifier significantly influences the final classification accuracy. If the classification result of the two-class SVM classifier \((A \cup C) / (B \cup D)\) at the first layer is incorrect, it is not feasible to correct it in the subsequent classification process.

**Figure 4.** Architecture of binary-tree SVM method.
4. Voting Comparison Method

A total of $N \times (N - 1)/2$ independent two-class classifiers are utilized to classify multiple targets, when using the DAG-SVM method. Compared to the other methods, the drawback of DAG-SVM is its excessive time consumption, both in the classifier-training step and in the step where trained classifiers are used to classify classification targets. The advantage of DAG-SVM is its high classification accuracy, which results from the one-to-one comparison in the algorithm. The binary-tree SVM method uses $N - 1$ two-class classifiers to classify multiple targets and using fewer two-class classifiers, results in less time consumption. The classification accuracy of the binary-tree SVM algorithm is significantly affected by the classification accuracy of the classifier at the first layer.

Combining the advantages of these two methods, we proposed a comparison voting method, named the cross-voting method.

Figure 5 shows the flow of multi-target classification using the cross-voting algorithm, which can classify the signals of the four types of targets using three SVM classifiers. At first, the pairwise combination of the sample datasets of the four types of targets ($A, B, C,$ and $D$) are treated as the new class dataset ($A \cup B, C \cup D, A \cup C, B \cup D, A \cup D,$ and $B \cup C$), and these new datasets are utilized for training the three two-class classifiers ($A \cup B/C \cup D, A \cup C/B \cup D,$ and $A \cup D/B \cup C$).

![Flowchart of the cross-voting algorithm.](image)

The trained classifiers are subsequently used to classify the new data, and class voting is performed according to the classification result of each classifier. For example, if the classifier ($A \cup B/C \cup D$) classifies the new data as $A \cup B$, the candidate results of classes $A$ and $B$ obtain a vote. Here, $r_{AB}$ and $r_{CD}$ are the results of the classifier ($A \cup B/C \cup D$). If the classification result of the new dataset is $A \cup B$, $r_{AB}$ is equal to 1, and $r_{CD}$ is equal to 0. $P_{AB}$ is the voting weight. At the end of the algorithm, the votes of all the candidate results are counted, and the class with the highest ticket is the final classification result of the data to be classified.

If the voting weights of the results in the voting method are identical, there may be cases wherein the votes of several candidate classes are equal when the votes of all the candidates are counted.

The pairwise combinations of the sample datasets of the targets ($A, B, C,$ and $D$) are treated as the new class dataset, and they train the three two-class classifiers ($A \cup B/C \cup D, A \cup C/B \cup D$, and $A \cup D/B \cup C$). It is assumed that the voting weights of $P_{AB} = P_{CD} = P_{AC} = P_{BD} = P_{AD} = P_{BC} = 1$.

The classification result of the target $A$ should preferably resemble the first data column shown in Figure 6a. A vote is provided for classes $A$ and $B$, according to the classification result ($A \cup B$), a vote is provided for class $A$ and category $C$, according to the classification result ($A \cup C$) of the second classifier, and a vote is provided for classes $A$ and $D$ according to the classification result ($A \cup D$) of the third classifier. After counting the votes of the respective candidate results of $A, B, C,$ and $D$;
the candidate class A obtains three votes, and the number of votes for the remaining classes (B, C, and D) is 1. Therefore, A is considered as the classification result.

In the process of using the cross-voting SVM classification algorithm to classify the target, there are likely to be cases wherein the votes of several candidate classes are identical when the distance between the different sample datasets is not large, that is, the difference between the target feature vectors is not large enough. In the cases shown in Figure 6b, the voting weights of the results in the voting method are identical, and the incorrect classification results of the two-class classifiers yield identical voting results for A, B, C, and D. In this case, the algorithm cannot determine the final classification result.

We add voting weights in the voting process to solve this problem. The higher the credibility of the classification result, the larger the weight assigned, and if the classification of the two-class classifier is lower, a smaller weight is assigned. After the weight is introduced, notwithstanding whether the classification result shown in Figure 6b appears, the classification result can still be obtained from the final (different) number of votes.

![Figure 6](image.png)

**Figure 6.** Two instances of multi-classification using the cross-voting algorithm: (a) in an ideal condition; (b) in the condition where it is impossible to classify.

### 4.1. Class Distance and Classification Accuracy of Two-Class SVM Classifiers

When using the SVM classifier for target classification, the two-class classifier is first trained using the sample dataset. Then, the trained classifier is used to classify the dataset to be classified. Considering the target classification using the SVM method as an example, the seismic signal datasets of AAV3, DW3, AAV4, and DW4 are used as the training datasets, for training a two-class classifier. Then, the trained classifier is utilized to classify datasets of other runs. After the classifier is trained, the accuracy of the classification is determined mainly by the classifier and the data to be classified. The performance of the classifier is largely affected by the sample datasets used for training. We utilized the dataset introduced in Section 2.1, to study the performance of the two-class SVM classifier when using different training datasets. For studying the relationship between the class distance of the sample data used for training and the classification accuracy of the two-class classifier, we used different sample data to train the SVM classifier. The trained classifier was utilized to classify the remaining dataset of runs; and its average classification accuracy was then calculated.

The method for calculating the distance between the sample data utilized for training the two-class classifier is described in detail in Section 4.2.

Figure 7a shows the distance between the training datasets and the average classification accuracy of the other datasets of runs using the two-class SVM classifiers, when utilizing the following seismic signal datasets to train the SVM classifier: (AAV3, DW3); (AAV3, DW5); (AAV3, DW7); (AAV3, DW9); (AAV3, DW11); (AAV5, DW3); (AAV5, DW5); (AAV5, DW7); (AAV5, DW9); and (AAV5, DW11). The data used for training is the feature matrix of the target signal extracted using the WCER
method, and the method described in Section 4.2 was used to calculate the distance between two feature matrices. Figure 7b shows the average classification accuracy of other datasets of runs using the two-class SVM classifiers, when utilizing the following seismic signal datasets to train the SVM classifier: (AAV4, DW3); (AAV6, DW3); (AAV8, DW3); (AAV10, DW3); and (AAV4, DW5); (AAV6, DW5); (AAV8, DW5); (AAV10, DW5).

Figure 7 shows that the performance of the classifier is positively related to the distance of the sample dataset used when training the classifier. The larger the geometric distance, the higher the classification accuracy that can be achieved; the smaller the distance, the lower the classification accuracy of the classifier.

\[
\begin{align*}
A &= \left[ \begin{array}{cccc}
\mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_M \\
\mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_M \\
\mathbf{a}_2 & \mathbf{a}_2 & \cdots & \mathbf{a}_M \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{a}_N & \mathbf{a}_N & \cdots & \mathbf{a}_M \\
\end{array} \right] \\
B &= \left[ \begin{array}{cccc}
\mathbf{w}_b & \mathbf{w}_2 & \cdots & \mathbf{w}_M \\
\mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_M \\
\mathbf{b}_2 & \mathbf{b}_2 & \cdots & \mathbf{b}_M \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{b}_N & \mathbf{b}_N & \cdots & \mathbf{b}_M \\
\end{array} \right]
\end{align*}
\]

As illustrated by Equation (17), A and B, which are \( N \times M \) matrices, represent the feature matrices of the acoustic signals of two targets. The number of rows in the matrix N, denotes the number of segments N, that the target signal corresponding to A was divided into. When using the WCER method to extract features from a signal sequence, segmentation is performed every 1024 points to extract the features. M is the number of variables of each line in the matrix, which is also twice the wavelet decomposition depth number in the WCER feature extraction method. The parameter \( \mathbf{w}_{ai} \) represents...
where \( \alpha \) is the number of segments of the target signal corresponding to the feature matrix \( B \). The different columns in the matrix react to the characteristics of the target signal in different dimensions. Therefore, when calculating the distance between two feature matrices, the distance between the feature variables in the same column of the two feature matrices is first calculated using Equation (18).

\[
d(w_{ai}, w_{bi}) = \sqrt{(w_{ai} - w_{bi})(w_{ai} - w_{bi})^T} = \sqrt{\sum_{j=1}^{N} (a_{ij} - b_{ij})^2}
\]  

(18)

where \( a_{ij} \) is the \( j \)th variable in the \( i \)th eigenvector of the feature matrix \( A \), and \( b_{ij} \) is the \( j \)th variable in the \( i \)th eigenvector of the feature matrix \( B \). There are \( M \) eigenvectors \((w_1, \ldots, w_M)\) in the feature matrix of the target signal. After calculating the distance between the eigenvectors of the feature matrix, it is also necessary to quantify the distance between the feature matrices, which is the distance between the target signals. The distance between the matrices is calculated by the formula below:

\[
d(A, B) = \frac{1}{M} \sum_{i=1}^{M} d(w_{ai}, w_{bi}) = \frac{1}{M} \sum_{i=1}^{M} \sqrt{\sum_{j=1}^{N} (a_{ij} - b_{ij})^2}
\]  

(19)

In the cross-voting method, the merged sample datasets of the multiple targets are utilized to train the two-class SVM classifier. Therefore, based on the distance between the feature matrices of a single target, it is necessary to further calculate the distance between the merged datasets of the multiple targets. We used the Lance–Williams dissimilarity update formula [27], to calculate the distance between the feature matrices, which is the distance between the target signals. The distance between the single target dataset \( A \) and the union dataset of multiple targets \( B \cup C \):

\[
d(A, (B \cup C)) = \alpha_{B,C} \cdot d(A, B) + \alpha_{C,B} \cdot d(A, C) + \beta d(B, C) + \gamma d(A, B) - d(A, C)
\]  

(20)

using the following equation to set the parameters:

\[
\alpha_{B,C} = \frac{|N_B|}{|N_B| + |N_C|}, \alpha_{C,B} = \frac{|N_C|}{|N_B| + |N_C|}, \beta = 0, \gamma = 0
\]  

(21)

where \( N_B \) is the number of rows in feature matrix \( B \), and \( N_C \) is the number of rows in the feature matrix \( C \), which is also the number of segments of the target signal corresponding to the feature matrix \( C \). Using Equation (20) for further derivation, we can obtain the distance between the union dataset \((A \cup D)\) and the merged dataset \((B \cup C)\):

\[
d((A \cup D), (B \cup C)) = \alpha_{A,D} \cdot d(A, (B \cup C)) + \alpha_{D,A} \cdot d(D, (B \cup C))
\]

\[
= \alpha_{A,D} (\alpha_{B,C} \cdot d(A, B) + \alpha_{C,B} \cdot d(A, C)) + \alpha_{D,A} (\alpha_{B,C} d(B, D) + \alpha_{C,B} d(C, D))
\]

(22)

where \( \alpha_{A,D} \) and \( \alpha_{D,A} \) are the adjustment coefficients and can be calculated by Equation (21).

After calculating the distance between the union dataset and the union dataset yielded by using Equation (22), the distance between the aggregation datasets is mapped to the voting weight range of \([0.7, 1.3]\); this voting weight range is selected to ensure that the sum of two classifier result weights are larger than the largest classifier result weight. The distance between the aggregation datasets in the three SVM classifiers in the voting method are denoted as \( \text{Dist}(1) \), \( \text{Dist}(2) \), and \( \text{Dist}(3) \). We used Equation (23) to calculate the voting weight of the result of each classifier.

\[
P(i) = \frac{2}{\sqrt{2\pi}} e^{-\frac{2}{2}\text{Nor}D^2(i)} + 0.5
\]  

(23)

\[
\text{Nor}D(i) = \frac{\text{Dist}(i) - \text{Max}(\text{Dist}(i))}{\text{Max}(\text{Dist}(i)) - \text{Min}(\text{Dist}(i))} + \sigma
\]  

(24)
where $NorD(i)$ is the normalized distance between the aggregation datasets, and $P(i)$ is the voting weight of the result of the $i$th SVM classifier. To ensure that the value of the voting weight $P(i)$ is within the range of $[0.7, 1.3]$, then $|NorD(i)|$, which is the absolute value of $Dist(i)$ after normalization, should have a value range of $[0.2, 0.8]$. Since the value range of $\frac{Dist(i) - Max(Dist(i))}{Max(Dist(i)) - Min(Dist(i))}$ is $[-1, 0]$, to make the value of $|NorD(i)|$ satisfy the condition, the preferable range of the adjustment coefficient $\delta$ is $[0.2, 0.8]$, and here we set $\delta$ to 0.3.

### 4.3. Average Segmentation of Training Data

The algorithm classifies the new input features using the pre-trained two-class SVM classifiers. The classification result is then voted out. Partial training datasets with large differences from the mean values of the sample data, are likely to cause incorrect assessments when using the trained SVM classifier to classify the input new features, which is likely to result in low performance of the cross-voting algorithm. Additionally, using the same training dataset to train the three two-class SVM classifiers cannot maximize the performance of the result comparison of the different classifiers in the algorithm, which is the main characteristic of the algorithm. Therefore, the segmentation averaging method shown in Figure 8 is utilized to process the training dataset. Through this approach, the training datasets of the three SVM classifiers maintain a certain degree of difference, and the influence of the wild points on the classification performance of the classifier in the training data are eliminated as much as possible.

![Figure 8. Average segmentation of the training feature matrix.](image)

**Figure 8.** Average segmentation of the training feature matrix.

### 4.4. Cross-Voting SVM Classification Algorithm

The algorithm flow of the cross-voting comparison method when there are four types of targets is illustrated in Algorithm 1. When the number of targets is five or six (in general, for WSN application system, the number of vehicle types to be classified does not exceed six), it is only necessary to change the aggregation datasets in the two-class SVM classifiers.
Algorithm 1 Cross-Voting Algorithm for 4 Targets Classification

---

Input: Training dataset: Feature matrices \{AFT, BFT, CFT, DFT\} of 4 target signals (A, B, C, D). Feature vector X of unknown target.

Output: Vehicle type of feature vector X.

1. Using the method shown in Figure 8 to segmentally average the training data to obtain a different target signal features matrix for training classifier: \{(AFT1, AFT2, AFT3); (BFT1, BFT2, BFT3); (CFT1, CFT2, CFT3); (DFT1, DFT2, DFT3)\};

2. Using the obtained feature matrix to construct different training datasets and train three classifiers separately:
   - using ((AFT1 ∪ BFT1), (CFT1 ∪ DFT1)) to train the first two-class classifier 1;
   - using ((AFT2 ∪ CFT2), (BFT2 ∪ DFT2)) to train the second two-class classifier 2;
   - using ((AFT3 ∪ DFT3), (BFT3 ∪ CFT3)) to train the third two-class classifier 3;

3. Calculating the distance between the training datasets for the three classifiers using Equation (22):
   - Dist (1): D((AFT1 ∪ BFT1), (CFT1 ∪ DFT1));
   - Dist (2): D((AFT2 ∪ CFT2), (BFT2 ∪ DFT2));
   - Dist (3): D((AFT3 ∪ DFT3), (BFT3 ∪ CFT3));

4. Calculating the voting weight of the three classifier results \(P(i)\) (i = 1, 2, 3), using Equations (23) and (24);

5. Classifying the feature vector X using the trained three classifiers, i.e., classifier 1, classifier 2, and classifier 3, and obtain the classification result: \((r_{AB}, r_{CD}); (r_{AC}, r_{BD}); (r_{AD}, r_{BC});\)

6. Counting vote results of 4 different targets using the classification results of classifier 1, classifier 2, and classifier 3.
   - \(Vr(A) = r_{AB}·P(1) + r_{AC}·P(2) + r_{AD}·P(3);\)
   - \(Vr(B) = r_{AB}·P(1) + r_{BD}·P(2) + r_{BC}·P(3);\)
   - \(Vr(C) = r_{CD}·P(1) + r_{AC}·P(2) + r_{BC}·P(3);\)
   - \(Vr(D) = r_{CD}·P(1) + r_{BD}·P(2) + r_{AD}·P(3);\)

6. Finding Max{\(Vr(A), Vr(B), Vr(C), Vr(D)\)} and the corresponding vehicle type is the vehicle type of feature vector X.

5. Performance of Cross-Voting SVM Algorithm

The datasets provided by the SensIT experiment consisted of acoustic signals, seismic signals, and infrared signals for two types of vehicles (AAV and DW). The infrared signals of the AAV vehicle and DW vehicle were pulse signals, and the frequency distribution of the signals was almost the same. The features of the infrared signals extracted by the WCER method had no recognizability, which made them unable to be utilized as different class signals. To study the performance of the cross-voting SVM method for multi-vehicle classification, we utilized the acoustic and seismic signals of AAV to represent two vehicle signals, and the acoustic and seismic signals of DW were utilized to represent two other vehicle signals.

5.1. Classification Accuracy of Cross-Voting SVM Algorithm

The features extracted from runs AAV3, AAV4, DW3, and DW4 were utilized for training the SVM classifier. Subsequently, the trained classifier was used to classify the target using the features extracted from the dataset of the other 15 runs (AAV5–AAV11 and DW5–DW12). The Radial Basis Function (RBF) function was used as the kernel function of the SVM classifier, and the gamma value in the kernel function was set to 0.5.

Table 2 illustrates that the three multi-target classification methods based on the two-class SVM classifier can effectively classify multiple vehicles detected in the WSN. For revealing the performance of these three methods more clearly, we calculated the overall average classification accuracy and time consumption of the vehicle classification using these methods. The overall average classification accuracy was calculated based on the average classification accuracy of a single target class. The classification accuracy of the ith type of target was denoted as \(Aca(i)\) (i = 1, 2, … , C), and the overall average classification accuracy, \(Oca\), is...
Table 2. Classification accuracy of four types of target signals using the cross-voting SVM method, DAG SVM method, and binary-tree SVM method.

| Run Name | Cross-Voting SVM | DAG SVM | Binary-Tree SVM |
|----------|------------------|---------|----------------|
| AAV5     | 99.77%           | 41.14%  | 99.32%         |
| AAV6     | 98.44%           | 47.92%  | 93.07%         |
| AAV7     | 93.67%           | 45.01%  | 89.90%         |
| AAV8     | 93.50%           | 42.37%  | 92.34%         |
| AAV9     | 97.64%           | 45.80%  | 99.78%         |
| AAV10    | 82.33%           | 54.17%  | 90.77%         |
| AAV11    | 89.75%           | 38.10%  | 90.03%         |
| Average  | 93.59%           | 44.93%  | 93.60%         |
| DW5      | 68.58%           | 86.13%  | 80.97%         |
| DW6      | 86.31%           | 89.93%  | 85.94%         |
| DW7      | 73.71%           | 88.40%  | 25.37%         |
| DW8      | 69.81%           | 75.72%  | 31.02%         |
| DW9      | 72.21%           | 74.88%  | 7.26%          |
| DW10     | 67.17%           | 92.38%  | 33.69%         |
| DW11     | 56.28%           | 98.52%  | 47.11%         |
| DW12     | 61.53%           | 91.36%  | 40.96%         |
| Average  | 70.58%           | 87.16%  | 44.03%         |

\[
O_{aca} = \frac{1}{P} \sum_{i=1}^{P} Aca(i)
\]  
(25)

where \( P \) is the number of target categories for all the targets that have been classified. Here, there were four vehicles categories.

Table 3 presents the overall average classification accuracy and average time consumption of the various multi-classification methods based on the SVM classifier. The results demonstrated that the average classification accuracy using the cross-voting SVM method was the highest, and that the time consumption of the method was satisfactory. The time required for the DAGSVM method was the highest of the three, and the average classification accuracy of this method was lower than that of the cross-voting SVM. Compared to the cross-voting method and DAGSVM method, the binary-tree SVM method exhibited the lowest classification accuracy and the lowest time consumption. In summary, the cross-voting SVM method significantly improved the classification accuracy of multi-target classification, in terms of time consumption. Compared with the DAGSVM method and binary-tree SVM method, the classification accuracy improved by approximately 7% when the cross-voting SVM method was used.

Table 3. Comparison of average classification accuracy and time consumption using the cross-voting SVM method, DAG SVM method, and binary-tree SVM method.

|               | Cross-Voting SVM | DAG SVM | Binary-Tree SVM |
|---------------|------------------|---------|----------------|
| Overall average classification accuracy | 74.52%    | 67.39%  | 67.15%         |
| Time Consumption of Classification       | 0.3255 s   | 1.1207 s | 0.2796 s       |

The main advantage of the cross-voting SVM method was the increased acoustic signal classification accuracy of DW7–DW12, compared to other methods. The main reason for the low acoustic signal classification accuracy of DAGSVM and Binary-tree SVM for DW7–DW12, was that the acquired acoustic signals were very similar to the acoustic signal generated by AAV, and a single two-class SVM classifier is easy to misclassify. Since the DAGSVM and binary-tree SVM methods used the same training set to train each classifier, there was no diversity of training samples and no further cross-validation steps after classification, resulting in low classification accuracy when classifying the
acoustic signals of DW7–DW12. The classification accuracy of acoustic signals of DW9, obtained by these three methods (the DAGSVM method, binary-tree SVM method, and cross-voting SVM method) were quite different, so we utilized the acoustic signal of DW9 as an example to compare the differences in the classification processes of the three methods.

Figure 9a,b indicate the classification accuracy for the acoustic signal of DW9 at each step, which is also the classification accuracy of each two-class classifier, when using the DAGSVM method and binary-tree SVM method, respectively. In the DAGSVM classification process, the probability that the classifier ((DW_S/AAV_A) classified the acoustic signal of DW9 as non-DW_S was 98.63%. However, the classification accuracy of the next layer classifier (AAV_A/DW_A) was only 24.91%, which made the overall classification accuracy unsatisfied. In the binary-tree SVM classification process, the first-level classifier (AAV_A ∪ AAV_S/DW_A ∪ DW_S) had a probability that the acoustic signal of DW9 was classified to be the correct class (DW_A ∪ DW_S) of 22.96%. The low classification accuracy of the first-level classifier, determined that the overall classification accuracy would not exceed 22.96%, even though the classification accuracy of the next level classifier reached 100%.

Figure 9. Classification accuracy of each step when using DAGSVM method: (a) and binary-tree SVM method; (b) to classify acoustic signal of DW9.

Table 4 shows that the classification accuracy of each step when using the cross-voting SVM method classified the acoustic signal of DW9. The cross-voting method utilized multiple classifiers and verified the classification results of these classifiers, to reduce the impact of low classification accuracy of a single classifier on the overall classification accuracy. In the cross-voting SVM classification process, the classification accuracy of the classifier (AAV_A ∪ DW_S/AAV_S ∪ DW_A) in the three classifiers was only 19.75%, but the classification accuracies of the other two classifiers, (AAV_A ∪ DW_A/AAV_S ∪ DW_S) and AAV_A ∪ DW_S/AAV_S ∪ DW_A) were 73.71% and 98.16%, respectively. The overall classification accuracy rate reached 72.21% after the voting link of the algorithm, which effectively prevented the influence of the third classifier’s bad classification performance on the overall classification results.

Table 4. Classification accuracy of each step when using the cross-voting SVM method to classify the acoustic signal of DW9.

| Classifier | Classification Accuracy | Overall Classification Accuracy |
|------------|-------------------------|--------------------------------|
| Classifier 1 | (AAV_A ∪ AAV_S/DW_A ∪ DW_S) | 73.71% | (DW_A ∪ DW_S) |
| Classifier 2 | (AAV_A ∪ DW_A/AAV_S ∪ DW_S) | 98.16% | (AAV_A ∪ DW_A) |
| Classifier 3 | (AAV_A ∪ DW_S/AAV_S ∪ DW_A) | 19.75% | (AAV_S ∪ DW_A) |
5.2. Cross-Voting Method Using Different Two-Class Classifier

In addition to the SVM method, other commonly used two-class classifiers include the NB, TreeBag method, adaboosting method, and KNN method. The two-class SVM classifier in the cross-voting SVM algorithm can be replaced with these two-class classifiers. We replaced the two-class SVM classifier in the cross-voting SVM algorithm with the NB, TreeBag, and adaboosting classifiers. The vehicle classification results are presented in Table 4.

The average classification accuracy and average time consumption of the cross-voting method, when using the different two-class classifiers are presented in Table 5. The results demonstrated that the cross-voting method using the DT two-class classifier, adaboosting two-class classifier, or SVM classifier could effectively classify the targets in the WSN system. When using the TreeBag classifier as the basis two-class classifier in the cross-voting method to classify the vehicles in the WSN, the classification accuracy reached approximately 76%; however, the time consumption was excessive. Compared with the TreeBag two-class classifier, the time consumption decreased significantly when using the NB classifier as the basis two-class classifier in the cross-voting method, and the classification accuracy of multi-target classification simultaneously decreased by approximately half. The average classification accuracy obtained by using the SVM classifier as the basis two-class classifier in the cross-voting method was 1% lower than that obtained using the NB classifier, but the time consumption decreased by 90% when using the SVM two-class classifier. Compared to using the SVM classifier as the basis two-class classifier in the cross-voting method, the average classification accuracy obtained using the adaboosting classifier was lower, whereas the time consumption was higher. In summary, the SVM two-class classifier is the most suitable classifier amongst the studied classifiers and can be utilized in the cross-voting method for multi-target classification in a WSN system.

Table 5. Average classification accuracy and time consumption of the cross-voting method using different two-class classifier.

|                      | Cross-Voting NB | Cross-Voting DT | Cross-Voting Adaboosting | Cross-Voting SVM |
|----------------------|-----------------|-----------------|--------------------------|-----------------|
| Overall average      | 37.48%          | 75.64%          | 72.55%                   | 74.52%          |
| classification       | Time Consumption|                 |                          |                 |
| accuracy             | 0.0253 s        | 4.1748 s        | 1.0750 s                 | 0.3255 s        |

5.3. Comparison with Other Multi-Class Classification Method

In addition to the multi-class classification algorithm based on the two-class classifier, there are certain methods originally capable of multiple class classifications. Among these methods, the NB method and adaboosting method are widely used. We utilized these three methods to classify the four vehicle signals mentioned above, and the results were compared with the classification results of the cross-voting method.

As shown in Table 6, the average classification accuracy obtained using the cross-voting SVM method is the highest among the three methods, being approximately 10.7% higher than that obtained using the NB method when classifying the multiple vehicles in a WSN system. This is mainly because the prior probability model obtained by analyzing the training dataset in the NB method is inaccurate, when the sample dataset is not sufficient. The classification accuracy obtained using the adaboosting method to classify the multiple targets was the lowest among the three methods.
Table 6. Comparison of average classification accuracy and time consumption using different multi-class classification algorithms.

|                      | NB         | Adaboosting | Cross-Voting SVM |
|----------------------|------------|-------------|------------------|
| Overall average classification accuracy | 63.85% | 53.52% | 74.52% |
| Time Consumption of Classification | 0.0077 s | 0.3621 s | 0.3255 s |

We additionally observed that the time consumed by the proposed cross-voting SVM method to classify the vehicles detected by a WSN system was less than that consumed by the adaboosting method. Moreover, the time consumption of the NB method was significantly less than that of the other three methods.

Deep learning methods based on multilayer perception and neural networks, are state-of-the-art algorithms for multi-class classification. To classify the four vehicle signals mentioned above, we utilized a feedforward neural network (FFNN) [28], a deep belief network (DBN) [29], which was an alternative class of deep neural network (DNN), and the extreme learning machines (ELM) [30] algorithm, and the results were compared to the classification results of the cross-voting method. In addition to classification accuracy and average time consumption of classification, the training time before classification and the memory consumption of the algorithms were also recorded for comparison. Both the DBN method and FFNN methods utilize a dual hidden layer network to perform classification; the two hidden layers of the DBN network contain 300 hidden units and 100 hidden units, and the two hidden layers of the FFN network contain 300 hidden units and 200 hidden units. The number of neurons in the ELM network that we utilized was 300, and the sigmoid function was utilized in the hidden neurons of ELM.

As shown in Table 7, the average classification accuracy obtained using the DBN method and FFNN are higher than that obtained using the cross-voting SVM method, when classifying the multiple vehicles in a WSN system, and classification accuracy may be higher after optimizing the DBN method and FFNN network. The classification accuracy obtained when using the ELM method was the lowest among these methods, being approximately 2.6% lower than that obtained using the cross-voting SVM method. The results in Table 7 also indicate that, although the average classification accuracy obtained using the ELM method and cross-voting SVM method is lower than that obtained using the DBN and FFNN method, their training time and memory consumption are more satisfactory than those obtained using the DBN and FFNN method. Excessive memory requirements (9253.94 MB for DBN and 34,660.9 MB for FFNN), made these two methods impractical for vehicle classification in WSN systems with limited computing capacity and memory resources. Using a neural network-based approach on a computer may be a sensible choice when offline classifying a large amount of multiple vehicle signals, acquired by multiple sensor nodes in the WSN system. Although the training time and classification time consumption of the cross-voting SVM method were slightly higher than those of the ELM method, the higher accuracy and lower memory requirements made the cross-voting SVM method more suitable than the ELM method, for multiple vehicles classification in a WSN system.

Table 7. Comparison of average classification accuracy, time consumption, and memory consumption using different multi-class classification algorithms.

|                      | DBN         | FFNN        | ELM          | Cross-Voting SVM |
|----------------------|-------------|-------------|--------------|------------------|
| Overall average classification accuracy | 78.40% | 75.76% | 71.88% | 74.52% |
| Time Consumption of Training | 350.84 s | 452.36 s | 0.5581 s | 15.1698 s |
| Average Time Consumption of Classification | 0.0298 s | 0.0452 s | 0.1783 s | 0.3255 s |
| Memory Consumption  | 9253.94 M | 34,660.9 M | 369.02 M | 123.95 M |

6. Conclusions

In this paper, we proposed a novel multi-class target classification method, named the cross-voting SVM method, based on the SVM classifier for vehicle classification in WSNs. The SVM classifier was
selected as the basis two-class classifier for vehicle classification, after comparing the classification performances of several commonly-used classification methods. We then combined the advantages of the two available methods, the DAGSVM method and the binary-tree SVM method, for multi-class target classification based on the SVM classifier. We subsequently proposed the cross-voting SVM method. Finally, we studied the target classification accuracy and time consumption of the proposed target classification method. The experimental results demonstrated that the classification accuracy of the targets detected in the WSN, was effectively improved by using the cross-voting method to classify vehicle targets in the WSN system.

**Author Contributions:** Conceptualization—H.Z. and Z.P.; software—H.Z.; supervision, Z.P.; validation—H.Z.; writing (original draft)—H.Z.; writing, review, and editing—Z.P.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Sohraby, K.; Minoli, D.; Znati, T. *Wireless Sensor Networks: Technology, Protocols, and Applications*, 1st ed.; John Wiley & Sons, INC.: Hoboken, NJ, USA, 2007.

2. Mishra, D.P.; Dorale, S.S. An application of wireless sensor network in intelligent transportation system. In Proceedings of the 2013 6th International Conference on Emerging Trends in Engineering and Technology, Nagpur, India, 16–18 December 2013; pp. 90–91.

3. Balid, W.; Tafish, H.; Refai, H.H. Development of portable wireless sensor network system for real-time traffic surveillance. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation System, Las Palmas, Spain, 15–18 September 2015; pp. 1630–1637.

4. Myles, A.J.; Feudale, R.N.; Liu, Y.; Woody, N.A.; Brown, S.D. An introduction to decision tree modeling. *J. Chemom.* 2004, 18, 275–285. [CrossRef]

5. Langseth, H.; Nielsen, T.D. Classification using Hierarchical Naive Bayes models. *Mach. Learn.* 2006, 63, 135–159. [CrossRef]

6. Galán, M.; Fernández, A.; Barrenechea, E.; Bustince, H.; Herrera, F. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognit.* 2011, 44, 1761–1776. [CrossRef]

7. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2011, 2. [CrossRef]

8. Guo, H.S.; Wang, W.J. An active learning-based SVM multi-class classification model. *Pattern Recognit.* 2015, 48, 1577–1597. [CrossRef]

9. Angulo, C.; Parra, X.; Catala, A. K-SVCR. A support vector machine for multi-class classification. *Neurocomputing* 2003, 55, 57–77. [CrossRef]

10. Melin, P.; Amezcua, J.; Valdez, F.; Castillo, O.A. New neural network model based on the LVQ algorithm for multi-class classification of arrhythmias. *Inf. Sci.* 2014, 279, 483–497. [CrossRef]

11. Calvo-Zaragoza, J.; Valero-Mas, J.J.; Rico-Juan, J.R. Improving kNN multi-label classification in Prototype Selection scenarios using class proposals. *Pattern Recognit.* 2015, 48, 1608–1622. [CrossRef]

12. Sucar, L.E.; Bielza, C.; Morales, E.F.; Hernandez-Leal, P.; Zaragoza, J.H.; Larranaga, P. Multi-label classification with Bayesian network-based chain classifiers. *Pattern Recognit. Lett.* 2014, 41, 14–22. [CrossRef]

13. Guo, H.X.; Li, Y.J.; Li, Y.A.; Liu, X.; Li, J.L. BPSO-AdaBoost-KNN ensemble learning algorithm for multi-class imbalanced data classification. *Eng. Appl. Artif. Intell.* 2016, 49, 176–193. [CrossRef]

14. Xu, T.A. New sphere-structured multi-class classifier. In Proceedings of the 2009 Pacific-Asia Conference on Circuits, Communications and Systems, Chengdu, China, 16–17 May 2009; pp. 520–525.

15. Lopez, J.; Maldonado, S. Multi-class second-order cone programming support vector machines. *Inf. Sci.* 2016, 330, 328–341. [CrossRef]

16. Tomar, D.; Agarwal, S.A. Comparison on multi-class classification methods based on least squares twin support vector machine. *Knowl.-Based Syst.* 2015, 81, 131–147. [CrossRef]

17. Hsu, C.W.; Lin, C.J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* 2002, 13, 415–425. [CrossRef] [PubMed]
18. Manikandan, J.; Venkataramani, B. Design of a modified one-against-all SVM classifier. In Proceedings of the 2009 IEEE International Conference on Systems Man and Cybernetics, San Antonio, TX, USA, 11–14 October 2009; pp. 1869–1874. [CrossRef]

19. Fei, B.; Liu, J.B. Binary tree of SVM: A new fast multiclass training and classification algorithm. *IEEE Trans. Neural Netw.* 2006, 17, 696–704. [CrossRef] [PubMed]

20. Cheng, L.L.; Zhang, J.P.; Yang, J.; Ma, J. An improved hierarchical multi-class support vector machine with binary tree architecture. In Proceedings of the 2008 International Conference on Internet Computing in Science and Engineering, Harbin, China, 28–29 January 2008.

21. Vers, C.; Struyf, J.; Schietgat, L.; Dzeroski, S.; Blockeel, H. Decision trees for hierarchical multi-label classification. *Mach. Learn.* 2008, 73, 185–214. [CrossRef]

22. Zhang, H.; Pan, Z.; Zhang, W. Acoustic-seismic mixed feature extraction based on wavelet transform for vehicle classification in wireless sensor networks. *Sensors* 2018, 18, 1862. [CrossRef] [PubMed]

23. Duarte, M.F.; Hu, Y.H. Vehicle classification in distributed sensor networks. *J. Parallel Distrib. Comput.* 2004, 64, 826–838. [CrossRef]

24. Guo, G.D.; Wang, H.; Bell, D.; Bi, Y.X.; Greer, K. KNN model-based approach in classification. In Proceedings of the OTM Confederated International Conference CoopIS, DOA and ODBASE, Catania, Italy, 3–7 November 2003; pp. 986–996.

25. Wen, X.Z.; Shao, L.; Xue, Y.; Fang, W. A rapid learning algorithm for vehicle classification. *Inf. Sci.* 2015, 295, 395–406. [CrossRef]

26. Erin, J.B.; Kristin, P.B. Multicategory classification by support vector machines. *Comput. Optim.* 1999, 53–79. [CrossRef]

27. Murtagh, F.; Legendre, P. Ward’s hierarchical agglomerative clustering method: Which algorithms implement Ward’s criterion? *J. Classif.* 2014, 31, 274–295. [CrossRef]

28. Nauman, A.; Junaid, Q.; Nasir, A. Neural networks in wireless networks: Techniques, applications and guidelines. *J. Netw. Comput. Appl.* 2017, 68, 1–27. [CrossRef]

29. Sun, N.; Han, G.; Du, K.; Liu, J.X.; Li, X.F. Person/Vehicle Classification based on Deep Belief Networks. In Proceedings of the 10th International Conference on Natural Computation, Xiamen, China, 19–21 August 2014; pp. 113–117.

30. Zhang, W.Y.; Zhang, Z.J. Belief function based decision fusion for decentralized target classification in wireless sensor networks. *Sensors* 2015, 15, 20524–20540. [CrossRef] [PubMed]

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).