Automated Excavator Activity Diagnosis Via a Topology and Statistical Information Based Classifier

Mushu Wang
School of Information Science and Electrical Engineering
Shandong Jiaotong University, China
wangmushu@126.com

YuKui Qin
School of Information Science and Electrical Engineering
Shandong Jiaotong University, China
18754035903@163.com

Tiepeng Bi
Baosteel Supervision Department
Shanghai BaoSteel Engineering Consulting Company
Limited, China
bi826.student@sina.com

Weigang Pan
School of Rail Transportation
Shandong Jiaotong University
China
panweigang1980@163.com

Abstract: *For automated diagnosis of excavator activities, a Topology and Statistical Information based Classifier (TSIC) is put forward, and it employs topology and statistical information of excavator activity samples. Specifically, a small sensor network is built on the excavator for its activity data acquisition. Distance metric learning is improved to explore sample features, making the same-class samples closer and different-class ones further. An improved mountain function is employed to construct covers for topology feature extraction, and then a weighted linear classifier is designed to diagnose (or classify) excavator activities. Generalization performance of TSIC is discussed. Experiments on excavator construction datasets and public datasets demonstrate competitive performance of TSIC.*

Keywords: *Metric learning, covering algorithm, topology and statistical information, excavator activity classify.*

Received October 1, 2024; accepted May 4, 2025.
https://doi.org/10.34028/iajit/22/4/8

1. Introduction

An excavator is an important mechanical equipment used in mining and civil engineering fields. Poor activities of the excavator increase the fuel and maintenance costs, and reduce work efficiency. And the worse, they can damage the excavator or bring significant safety hazards to relevant personnel. So many researchers have focused on construction site safety monitoring by using existing classification methods, e.g., Support Vector Machine (SVM), Decision Tree (DT), K-Nearest Neighbor (KNN), and Artificial Neural Network (ANN), etc., [2, 14, 22, 29, 31, 34]. Although excavators are important machinery in civil construction and have caused many accidents or human casualties [23, 32], existing literatures hardly focused on excavator activity diagnosis. Additionally, operation of measuring and analyzing the excavator is time-consuming and prone to errors, because supervisors must manually observe and record the operation process of excavators. Therefore, a new automatic method is urgently needed to accurately diagnose excavator activities. Moreover, existing diagnostic methods were not transparent to human beings. It is difficult to provide guidance on classifier structures or activity diagnosis. Thus, in this paper, we try to design a geometric-interpretable classifier, and reveal how and why its structure works from a geometric perspective.

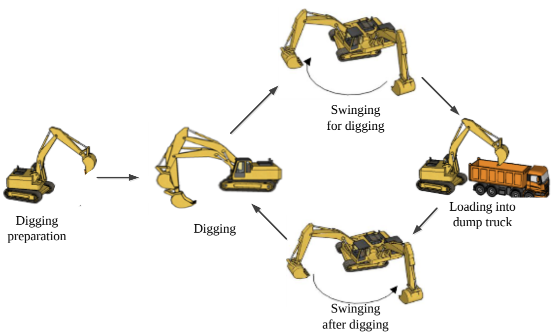


Figure 1. Excavator working process.

An excavator often works in collaboration with other machinery or workers, and its activities consists of five parts, i.e., digging preparation, digging, swinging for (after) digging, and loading into dump truck (see Figure 1). In the part of digging preparation, it is required that the ground is flat [23]. If not, it will put the excavator and driver in overturning danger. And the excavator has so many construction activities, and works in so complex working environment (mixed construction of humans and machinery). High speed motions of each part, e.g., sharp swing, intense actions of arms, boom and buckets, are important danger sources. So it is necessary to distinguish whether the excavator activity is normal or abnormal.

Existing methods of activity detection were composed of two classes, i.e., a vision-based method and a non-vision-based method. Considering the non-vision-based method, many sensors, e.g., the Inertial Measurement Unit (IMU), the Global Positioning System (GPS), the Radio Frequency Identification (RFID), etc., were used to capture excavator motion and position information, and were used to describe activities of the excavator [28, 30]. In the paper [28], IMU sensors were used to obtain engine vibration signals, and the frequency, extracted from the signals, was adopted to determine the working cycle of the excavator. Similarly, in the papers [29, 31], IMUs were used to obtain excavator motion information, many supervised classification methods, e.g., SVM, KNN, ANN), and DT, were adopted to classify excavator activities. Joystick signals of an excavator were captured, and the dynamic time warping algorithm was used to determine excavator activities [4]. 3D orientation sensors were attached on a loader boom, and its angle change was used to recognize activity of the boom [1]. An Ultra Wide Band (UWB) sensor was employed to estimate activities of an exactor and a truck by diving construction site into four regions [36]. Some researchers adopted audio signals to recognize machine activity [6, 8]. In the paper [6], microphones were installed on an excavator, and the features of audio signals were used to describe its activity, and recognized by using SVM. However, the audio method was not effective in some sites because it had poor anti-noise performance.

Considering the vision-based method, the activity of an excavator was captured by using vision sensors [13], and the histogram of optical flow and the Histogram of Oriented Gradients (HOG) were used to obtain features from videos, and the excavator activity was recognized by using a Bayesian network. Similarly, the 3D HOG was used to extract features, and based on which the activity of an excavator was recognized by using SVM [12]. Many researchers employed the deep CNN, Long Short-Term Memory (LSTM), YOWO [5, 10, 11, 19, 20] to recognize excavator activities. However, it is sensitive to dusty and weather conditions, and it is highly susceptible to the influence of excavator movement speed. Moreover, it needs too much computing power.

It is obvious that SVM, ANN, DT, etc. were widely used for construction equipment activity recognition [2, 14, 22, 29, 31, 34]. Considering a simple and famous ANN, the Extreme Learning Machine (ELM), its number of hidden layer nodes was either prefixed or determined based on training errors [16, 25]. Regarding the coefficient matrix between input and hidden layers, some methods of designing an input weight were introduced [26, 27], and the weight was used to extract discriminative information or remove noise of input. In the paper [38], a metric learning method was embed into ELM to design input weights, and it was carried out by using iterative two-step training. But the training method converged to a local optimum and had high computational complexity. Another simple classifier, SVM, was also suitable for construction site safety monitoring [1, 4, 6, 8, 12, 13, 36]. Except for standard SVMs, a weighted Least Squares SVM (LSSVM) was selected as a classifier of excavator activities [15, 35], its weights of slack variables were given based on an error variable and its distribution estimation or information gain. Considering large-sample computation infeasibility, a new SVM was proposed by using the truncated Huber loss function [39], but further research was needed on use of nonlinear kernel functions. In order to detect network anomaly, changes of the network-data characteristics were described by using a concept drift method, then SVM was given combined with K-Means clustering [17]. Recently, TSVM had also attracted a lot of attentions [9, 33]. To sum up, considering ELM, SVM or other ANN, it is difficult to determine the number of the ANN hidden nodes or the kernel function of SVM, and answer the question “why the function is selected”. It is one of the motivations behind this article.

In order to accurately diagnose (or classify) excavator activities, Topology and Statistical Information based Classifier (TSIC) is put forward based on distance metric learning and the cover algorithm. The main idea of TSIC is as follows. In order to collect data of excavator activities, a small sensor network is built on an excavator. Then, a method of distance metric learning is improved through adding regularization terms to explore discriminative features from excavator activity samples. Compared with Large Margin Nearest Neighbor classification (LMNN) [43], it not only focuses on the similarity and distinction of the samples, but also emphasizes its generalization performance. Considering the covering algorithm [41], cover centers are optimized by using an improved mountain function, and they are beneficial for better cognition of the feature distribution [42]. A cover is considered a class anchor located in the feature space. For each feather within or near the cover, it can be intuitively inferred that the feather has the same class attribute of the cover. Based on the optimal covers, the topology feature of an excavator activity sample is extracted. By using a statistical learning method, a weighted linear classifier is built, and it is different from SVM [15, 35], its sample weights are obtained from topology rather than statistical information. At last, generalization performance of TSIC is discussed, and it is related to the parameters of the TSIC. Experiments on excavator construction datasets and public datasets demonstrate competitive performance of TSIC.

The major contributions of the work are as follows.

1) Activity data of the excavator is captured by using a small sensor network built on the excavator, and a method of extracting topology features of the data is proposed based on a mountain function and a covering algorithm.
2) A weighted linear classifier is put forward based on the topology features, and it fully utilizes statistical and topological information of the activity data. Moreover, generalization performance of TSIC is discussed. And it is revealed that the relationship between generalization performance and the parameters of TSIC.

The rest of the paper is organized as follows. In section 2, a research framework is introduced. Section 3 introduces TSIC, and its generalization performance is discussed. Section 4 verifies the performance of TSIC by using comparative experiments on public datasets and an excavator construction dataset. Section 5 presents conclusions.

2. Research Framework

In order to collect data of excavator activities, a small sensor network is built on an excavator. And it consists of Beidou/GPS positioning sensors, IMUs, laser range sensors and an industrial computer (see Figure 2). The 485 bus is used for communication between the sensors and the computer. From Figure 2, positions and yaw angles of the excavator are captured by using Beidou/GPS positioning sensors, and its body attitude is obtained from IMUs. Actuator information of boom, arm and bucket is captured by using laser range sensors rather than IMUs. The main reason is that the length of the actuator is original information of the boom, arm, and bucket actions.

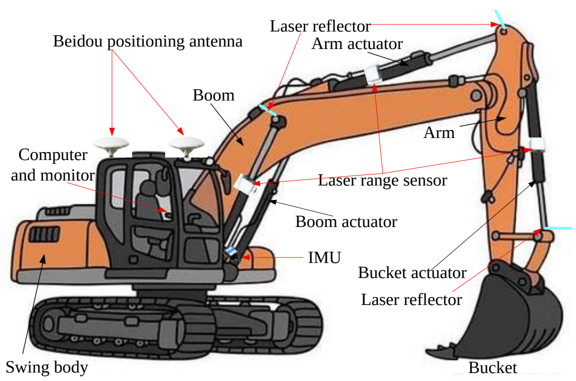


Figure 2. The sensor network on the excavator.

Data captured is used to build a 10-D activity vector $\hat{x}$. It is composed of body attitude angles, swing speed, lengths of boom, arm and bucket actuators and their change speeds. The vector is shown in Table 1. Based on the set of the vectors, excavator activities are recognized, and results are shown in a monitor in the excavator cab. If there is any anomalous activity, an alarm will be triggered.

Table 1. Elements of activity vectors.

$\hat{x}_1$ - $\hat{x}_3$	$\hat{x}_4$	$\hat{x}_5$ - $\hat{x}_7$	$\hat{x}_8$ - $\hat{x}_{10}$
Body attitude angles	Swing speed	The actuator lengths	Length change speeds of the actuators

In the vector, $\hat{x}_1 \sim \hat{x}_3$ are used to describe the attitude of the body, $\hat{x}_4$ is the swing speed of the body, $\hat{x}_5$ - $\hat{x}_{10}$ present the actuator lengths of the boom, arm and bucket, and the change speeds of the lengths.

Based on captured activities data, construction procedures of TSIC are introduced, and shown in Figure 3.

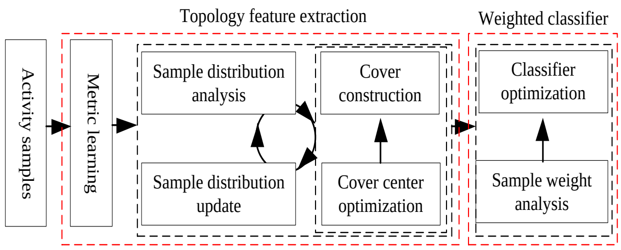


Figure 3. The block diagram of TSIC.

As shown in Figure 3, steps of building TSIC are introduced as follows. The distance metric learning is improved for feature extraction [43]. Then an improved mountain function is given and used to describe sample distribution. For the sample near its class border, its mountain value is lower, and it is not a constant value, will be updated with the cover construction. Based on sample distribution analysis, the center of a cover is initialized and optimized, and then it is used to extract topology features of excavator activities. At last, a weighted linear classifier is designed to classify excavator activities, and the weight is obtained based on the distance between a sample and its corresponding cover boundary.

3. Topology and Statistical Information Based Classifier

3.1. Distance Metric Learning

Considering many classifiers, e.g., K-means, nearest-neighbor classifiers and SVM, their performance depends critically on a sample distance metric over input space. And the distance metric assigns a smaller distance between the same-class samples, and enlarges the distance between different-class samples. Thus, we need to learn a linear or nonlinear distance metric [43]. In this paper, an improved linear distance metric is adopted, i.e., Equation (1).

$$\bar{x} = L\hat{x} \tag{1}$$

where $\hat{x}$ is an input sample, L is a real matrix and row full rank. From LMNN [43], an improved distance metric learning method is proposed, and the matrix L is obtained by solving the problem in Equation (2).

$$\begin{aligned}\min_{L} J(L) &= (1-\rho)\sum_{i,j}(\hat{x}_i-\hat{x}_j)^T M(\hat{x}_i-\hat{x}_j) \\ &\quad +\rho\sum_{i,j,l}(1-y_{il})\xi_{ijl}+\lambda_M tr(M) \\ s.t.& \\ &(\hat{x}_i-\hat{x}_l)^T M(\hat{x}_i-\hat{x}_l) \\ &\quad -(\hat{x}_i-\hat{x}_j)^T M(\hat{x}_i-\hat{x}_j)\geq 1-\xi_{ijl} \\ &\xi_{ijl}\geq 0 \\ &M \geq 0\end{aligned} \tag{2}$$

where $M=L^TL$, $tr(M)$ is the trace of M, $\hat{x}_j$ is called as a target neighbor of $\hat{x}_i$, $\hat{x}_l$ is an impostor with label $y_l \neq y_i$, ξ_{ijl} is a slack variable, $\rho\in[0,1]$, $\lambda_M>0$. It is obvious that $tr(M)$ is a penalty term. And if the distance metric method is considered a classifier, it not only focuses on the similarity and distinction of the samples, but also emphasizes its generalization performance.

Then, by using Equation (3), the obtained feature $\bar{x}_i$ is mapped onto a hypersphere denoted by S_R, whose radius is R_S,

$$x_i=\tau(\bar{x}_i)=\left[\bar{x}_i^T, \sqrt{R_s^2-\|\bar{x}_i\|^2}\right]^T \tag{3}$$

where $R_s=\max_i\{\|\bar{x}_i\|\}$.

3.2. Topology Feature Extraction

Based on the samples obtained from Equation (3), this section introduces topology feature extraction, it consists of cover construction and cover-based topology feature extraction.

3.2.1. Cover Construction

A key step of topology feature extraction is to construct covers for a class. The definition of a cover is introduced firstly, and it is as follows.

- **Definition 1 [41]**: Suppose $_j$ is j^{th} class of all samples. For any sample $x_i\in\Omega_i$ and any real number $r_i\geq 0$, there exists a suprasphere denoted by (x_i, r_i) where x_i and r_i are the center and the radius, respectively. If every element in (x_i, r_i) belongs to Ω_i only, $\delta(x_i, r_i)$ is called a *cover* of Ω_i.

The sketch map of a cover is shown in Figure 4. As shown in Figure 4, suppose that black dots belong to the i^{th} class Ω_i, its j^{th} cover is denoted by (c_{ij}, r_{ij}), and abbreviated as δ_{ij}. $\bar{r}_{ij}$ is the minimum distance from the cover center to different-class samples, $\underline{r}_{ij}$ is the maximum distance from the cover center to the same-class samples. $r_{ij}=0.5(\bar{r}_{ij}+\underline{r}_{ij})$ is a cover radius. A set of samples covered by (c_{ij}, r_{ij}) is denoted by $C\Omega_{ij}$. Let $m_{ij}=\bar{r}_{ij}-\underline{r}_{ij}$, and it is named as a margin width of the δ_{ij}.

The procedure of a cover construction refers to paper [41], and its key is to determine cover centers. Samples with the same class label (black dots in Figure 4) are covered by several covers (the blue circles in Figure 4). In other words, distribution of samples with the same class label is described and cognized by using the union of its covers. The class boundary can be presented by using the union border of the covers, which is more in line with human cognitive processes.

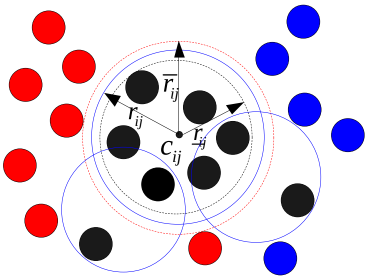


Figure 4. The sketch map of a cover.

In order to initialize cover centers, sample distribution is described by using an improved mountain function. The mountain value D_i of the sample x_i is obtained by using Equation (4),

$$D_i=\sum_r e^{-\|x_i-x_r\|^2/\upsilon_k^2}-\sum_j e^{-\|x_i-x_j\|^2/\upsilon_k^2} \tag{4}$$

where x_i, $x_r\in\Omega_k$, $x_j\notin\Omega_k$, υ_k is a neighbor radius for k^{th} class in Equation (5),

$$\upsilon_k=0.25\max_{i,}\left\{\min_j\left\{\|x_i-x_j\|\right\}\right\} \tag{5}$$

In the mountain Equation (4), the neighbor radius is individually set for each class compared with the mountain function proposed by Kim *et al.* [18]. It is obvious that for a sample, if there is no different-class sample around the sample, the mountain value is higher, otherwise, the value is lower. In other words, for the sample near its class border, its mountain value is lower. Thus, the mountain function can describe distribution of samples, and it reveals the relationship between the sample and samples around it.

If samples in Ω_k are deleted, the mountain value of an uncovered sample is updated by using Equation (6),

$$D_i=D_i-\sum_l e^{-\|x_i-x_l\|^2/\upsilon_k^2} \tag{6}$$

where $x_l\in\Omega_k$ is a deleted sample.

A sample with the maximum mountain value is considered a temporary cover center c_{ki} of the cover δ_{ki}. Here, a cover center is initialized. After initialization of the cover center, the next step is to find an optimal cover center based on an optimization indicator in Equation (7),

$$\begin{aligned}\min_{c_{ki}} J(c_{ki}) &= \frac{(1-\lambda_c)}{2N_c}\sum_l\|c_{ki}-x_l\|^2 \\ &\quad -\frac{\lambda_c}{2N_b}\sum_t\|c_{ki}-x_t\|^2\end{aligned} \tag{7}$$

where N_c, N_b are the number of covered samples and that of the nearest different-class samples, respectively. $x_l\in\Omega_k$ is covered by δ_{ki}, $x_t\notin\Omega_k$, $\lambda_c\in(0,1)$. A gradient descent method is used to solve Equation (7). And the gradient w.r.t. c_{ki} is presented by Equation (8), i.e.

$$\nabla_{c_{ki}} = \frac{(1-\lambda_c)}{N_c}\sum_l (c_{ki} - x_l) - \frac{\lambda_c}{N_b}\sum_t (c_{ki} - x_t) \tag{8}$$

A learning rule of a cover center c_{ki} is given by Equation (9),

$$c_{ki}^{(n)} = c_{ki}^{(n-1)} + \alpha^n (\alpha_0 \min_k(r_k))\nabla_{c_{ki}} \Big/ \left\|\nabla_{c_{ki}}\right\| \tag{9}$$

where $\alpha\in(0, 1)$, $\alpha_0\in(0, 0, 1)$. Then the center $c_{ki}^{(n)}$ is mapped onto the S_R in Equation (10),

$$c_{ki}^{(n)} = \frac{c_{ki}^{(n)}}{\left\|c_{ki}^{(n)}\right\|} R_s \tag{10}$$

Samples covered by constructed covers are deleted. And mountain values of the left ones are updated by using Equation (6). The procedure of constructing covers is formulated in Algorithm (1).

Algorithm 1: Covers Construction Procedure.

Input: all train samples on the S_R
Output: cover centers and radii
Procedures:
1:*Initialization:* $\lambda_c = 0.4$, $N_b = 3$, D_{ki} , $n = 1$, $N_m = 100$, $k = 1$, $i = 1$, $\alpha = 0.995$, $\alpha_0 = 0.05$
2:*while 1*
3: $C\Omega_{ki}^{(0)} \leftarrow \Phi$.
4: *Select uncovered sample of the maximum mountain value* x *of the* k^{th} *class, let* $c_{ki}^{(1)} \leftarrow x$
5: *Get a temporary radius* $r_{ki}^{(n)}$.
6: $C\Omega_{ki}^{(n-1)} \leftarrow C\Omega_{ki}^{(n)}, n \geq 1$, *get a set of covered samples* $C\Omega_{ki}^{(n)}$.
7: $c_{ki}^{(n-1)} \leftarrow c_{ki}^{(n)}, n \geq 2$, *update the center* $c_{ki}^{(n)}$ *by using equations (9-10).*
8: *if* $C\Omega_{ki}^{(n-1)} \subseteq C\Omega_{ki}^{(n)} \;\&\; n \leq N_m$ *the then*
9: $n \leftarrow n+1$
10: *Goto 5*
11: *else*
12: *Save* $(c_{ki}^{(n-1)}, r_{ki}^{(n-1)})$
13: *Remove the covered samples.*
14: *Update sample mountain value by using equation (6).*
15: $i \leftarrow i+1$
16: *end if*
17: *if all samples of the* k^{th} *class are covered then*
18: $k \leftarrow k+1$
19: *end if*
20: *if all samples are covered then*
21: *while end.*
22: *end if*
23:*end while*
24:*return* $\{(c_{ki}, r_{ki})\}$

3.2.2. Topology Feature Extraction

Suppose that all covers are obtained, denoted by $\{\delta_k=(c_k, r_k)\}$. Based on the covers, the topology feature is extracted for each x_i and presented in Equation (11).

$$T = [t_1, t_2, \cdots, t_k, \cdots]^T \tag{11}$$

where $t(\delta_k, x_i) = \frac{1}{1+e^{-\gamma_k d_i}}$, d_i=1-$\|x_i$-$c_k\|/r_k$, $t(\delta_k,x)$ is denoted by t_k or $t_k(d_i)$ for short. The margin between two samples belonging to different covers is enlarged by using $t_k(d_i)$. For the cover δ_k, it is seen as the coordinate axis in the Cartesian coordinate system, and t_k is the corresponding coordinate of each x. And the vector T is named as a topology feature of the corresponding sample.

Considering γ_k, it is required that the smaller margin width, the larger γ_k. Thus, we have Equation (12),

$$\gamma_k = \frac{\bar{\gamma}}{1+e^{-\eta(\Delta r_k - r_6)}} \tag{12}$$

where $\bar{\gamma} \geq 1, \eta > 0$, in Equation (13),

$$\Delta r_k = \max\{m_k\}/m_k \tag{13}$$

m_k is the margin width of δ_k, and the set $\{\Delta r_k\}$ is sorted, γ_6 is its 60^{th} percentile.

3.3. Weighted Linear Classifier

Two sets of samples are denoted by Ω_1, Ω_2, and their class labels are {1, -1}. Based on the topology feature T, a binary linear classifier is designed and used to classify T , i.e., Equation (14),

$$F(T) = w^T T + b \tag{14}$$

where $w_k = s_{\delta k}\,\omega_k^2$, $s_{\delta k} = \begin{cases} 1 & \delta_k \subseteq \Omega_1 \\ -1 & \delta_k \subseteq \Omega_2 \end{cases}$. Let s_δ=[$s_{\delta 1}$, $s_{\delta 2}$, …, $s_{\delta k}$,…]^T.

For Equation (14), parameters are obtained by solving the optimization problem Equation (15),

$$\begin{aligned}\min_{\omega,b,\xi} J(\omega,b,\xi) &= \frac{1}{2}\sum_k \omega_k^2 + \frac{1}{2}b^2 \\ &+ \frac{C}{2}\left[\lambda_w \sum_i \mu_i \xi_i^2 + (1-\lambda_w)\sum_i o_i\right]\end{aligned} \tag{15}$$

where $o_i = \max\{0, 1 - y_i(w^T x_i + b) - \xi_i^2\}$, y_i is the label of a sample x_i, $\lambda_w\in(0, 1)$, ξ_i is a slack variable, μ_i>0 is a weight factor of ξ_i. Let $\xi = [\xi_1, \xi_2, \cdots, \xi_i, \cdots]^T$, μ=[μ_1, μ_2, …, μ_i,…]^T.

The optimization problem is solved by using the Adam method [21]. Gradients w.r.t. ω, b, ξ are obtained through Equation (16).

$$\begin{aligned}\nabla_\omega &= \omega + C(1-\lambda_w)\sum_i(-y_i s_\delta \odot \omega \odot H)s_{oi} \\ \nabla_b &= b + C(1-\lambda_w)\sum_i -y_i s_{oi} \\ \nabla_\xi &= C\lambda_w \mu \odot \xi - 2C(1-\lambda_w)\xi \odot s_o\end{aligned} \tag{16}$$

where $\odot$ means vector dot product, $s_{oi} = \begin{cases} 1 & o_i > 0 \\ 0 & o_i \leq 0 \end{cases}$, s_0=[s_{01}, s_{02}, …, s_{0i}, …]^T.

Suppose that a sample $x_i \in \Omega_k$ is covered by more than one covers $\{\delta_{kj}\}$. The distance between it and the cover boundary is obtained by using Equation (17),

$$d_{bi} = \max_j \{1 - \|x_i - c_{kj}\| / r_{kj}\} \quad (17)$$

The weight factor μ_i is determined by using Equation (18),

$$\mu_i = 1 - \mu_0 d_{bi} \quad (18)$$

where, $\mu_0 \in (0,1]$.

Considering the optimization problem in Equation (15), for the given w, the second and third items are contradictory. In other words, the second item increases while the third one decreases, as λ_w decreases, and the second item is bigger than the third one when λ_w is smaller. Generally, $\lambda_w \in (0.6, 0.7)$. Compared with SVM [9, 17, 18, 21, 33, 39, 41, 42, 43], TSIC adopts a unified weight rather than a weight for each sample. Each cover is at the set boundary, and it can be considered a support cover.

- **REMARK 1**: Considering real-time feasibility of TSIC, computational complexity of TSIC is evaluated. TSIC consists of three steps: metric learning, topology feature extraction and a weighted classifier. In the step of metric learning, its computational complexity is O(m, n) where $L \in R^{m \times n}$. For topology feature extraction, its computational complexity O((m+1)N) is obtained from Equation (11) where N is the number of covers. From Equation (14), computational complexity of the weighted classifier is $O(N)$. Thus, computational complexity of TSIC depends on N, n. So TSIC can be used for real-time applications.

3.4. Generalization Performance Analysis

Considering a linear classifier, the VC dimension is used to describe generalization performance. For n-D linear classification, we introduce lemma 1 firstly.

- **Lemma 1 [37, 40]**: Let n-D vectors x belong to a sphere of the radius R, then the set of Δ-margin separating hyperplanes has VC dimension d_{vc} bounded by the inequality.

$$d_{VC} \le \min([\frac{R^2}{\Delta^2}], n) + 1 \quad (19)$$

- **Theorem 1**: Considering TSIC, a vectors $\hat{x}$ belongs to a sphere of the radius R, the dimension of T is n. Then TSIC has VC dimension d_{vc} bounded by the inequality.

$$d_{VC} \le \min([\frac{1}{16}\bar{\gamma}^2 \|w\|^2 \bar{\lambda}_M R^2 \Sigma_r^n], n) + 1 \quad (20)$$

where $\bar{\lambda}_M$ is the maximum eigenvalue of M, $\Sigma_r^n = \sum_k \frac{1}{r_k^2}$.

- **Proof**:

Suppose that T is on a sphere of the radius R_t, and the classifier satisfying

$$|w^T T + b| \ge 1 \quad (21)$$

We deduce Equation (19) from lemma 1, i.e.

$$d_{VC} \le \min([R_t^2 \|w\|^2], n) + 1 \quad (22)$$

Considering two samples x_i, x_j, we obtain Equations (20) and (21) from the mean value theorem.

$$t_k(d_i) - t_k(d_j) = \frac{\partial t_k(d_0)}{\partial d_0}(d_i - d_j) \quad (23)$$

$$\begin{aligned} \frac{\partial t_k(d)}{\partial d} &= \gamma_k t_k(d)(1 - t_k(d)) \\ &\le 0.25\gamma_k \le 0.25\bar{\gamma} \end{aligned} \quad (24)$$

Thus, it is obtained that;

$$\begin{aligned} \|t_k(d_i) - t_k(d_j)\| &\le 0.25\bar{\gamma}\|d_i - d_j\| \\ &\le \frac{0.25\bar{\gamma}}{r_k}\|x_i - x_j\| \end{aligned} \quad (25)$$

From Equation (22), we have

$$\begin{aligned} R_t^2 &= 0.25 \max_{i,j}(\|T_i - T_j\|^2) \\ &= 0.25 \max_{i,j}(\sum_k \|t_k(d_i) - t_k(d_j)\|^2) \\ &\le \frac{1}{64}\bar{\gamma}^2 \max_{i,j}(\sum_k \frac{\|x_i - x_j\|^2}{r_k^2}) \\ &\le \frac{1}{16}\bar{\gamma}^2 \sum_k \frac{R_s^2}{r_k^2} \end{aligned} \quad (26)$$

Let $\Sigma_r^n = \sum_k \frac{1}{r_k^2}$, we have

$$R_t^2 \le \frac{n}{16}\bar{\gamma}^2 R_s^2 \Sigma_r^n \quad (27)$$

It is worth noting that $\hat{x}^T M \hat{x} \le \bar{\lambda}_M \|\hat{x}\|^2$ where $\bar{\lambda}_M$ is the maximum eigenvalue of M. Because $R_s^2 = \max_i\{\hat{x}_i^T M \hat{x}_i\}$, we have,

$$R_s^2 \le \bar{\lambda}_M R^2 \quad (28)$$

Substituting Equations (24) and (25) into Equation (19), we have

$$d_{VC} \le \min([\frac{1}{16}\bar{\gamma}^2 \|w\|^2 \bar{\lambda}_M R^2 \Sigma_r^n], n) + 1 \quad (29)$$

4. Experiment Validation

In this section, feasibility and effectiveness of TSIC are verified. At the beginning of the section, TSIC is applied in excavator activity classification. And then TSIC is compared with other classifiers based on public data sets, and comparison experiments are carried out and used to assess the performance of TSIC. At last, impact of TSIC parameters on generalization performance is discussed. All experiments are carried out on win-10 platform, in MATLAB 2020a.

4.1. Activity Recognition Experiments and Discussion

Experiments of excavator activity recognition are carried out in an airport construction project (See Figure 5). The soil on the construction site is mostly soft sand, which can easily cause excavator tilting or collision accidents. The excavator model is SE245LC, and made by Shantui of a Chinese construction machine company. Positions and yaw angles of the excavator are captured by using a real-time kinematic global navigation satellite system made by Hi-Target, the model of the IMU sensor is HWT9073 made by WitMotion Shenzhen Co., Ltd., and the accuracy of the laser range sensor is 2mm. The model of the industrial computer is SpecialControl MEC-T8762, its information is as follows, Intel i3 CPU, Windows 7, 4G memory.

Figure 5. Airport construction experiments.

3000 driving samples record activity data of three excavators, and they are filtered by using the method of empirical mode decomposition [24]. And according to driving specifications they are divided into two classes, i.e. normal activities and abnormal activities. The data are used to construct the 10-D vector shown in Table 1, and describe excavator activities. Additionally, information of excavator drivers is listed in Table 2.

Table 2. Excavator driver information.

Driver	Length of employment (year)	Qualification
driver 1	25	advanced
driver 2	13	intermediate
driver 3	20	intermediate

In order to demonstrate feasibility of TSIC for activity classification, a method of 5-fold cross validation is adopted to train and test TSIC. Parameters of TSIC are listed in Table 3.

Table 3. Parameters of TSIC.

item	λ_c	N_b	α	α_0
value	0.4	3	0.995	0.05
item	C	λ_w	$\bar{\gamma}$	μ_0
value	2500	0.665	1.5	1
item	ρ	λ_M		
value	0.5	0.001		

Training Accuracy (TRA) and Testing Accuracy (TEA) are shown in Figures 6 and 7. As shown in Figure 6, it is concluded that the TRA of TSIC is not the best, but it is comparable to that of other classifiers. As shown in Figure 7, it is obtained that the TEA of TSIC is better than others. Moreover, it has a smaller range of variation.

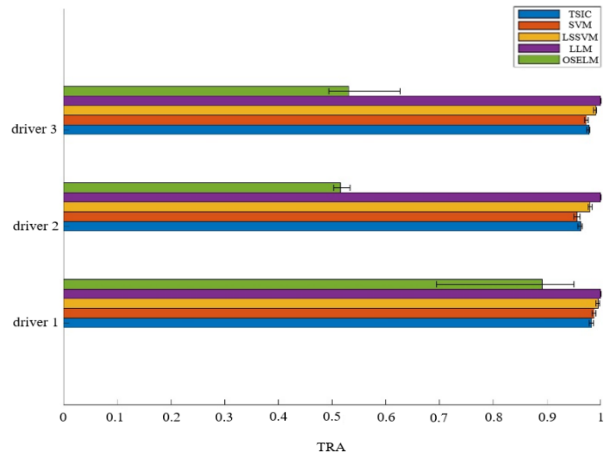


Figure 6. TRA of activity recognition.

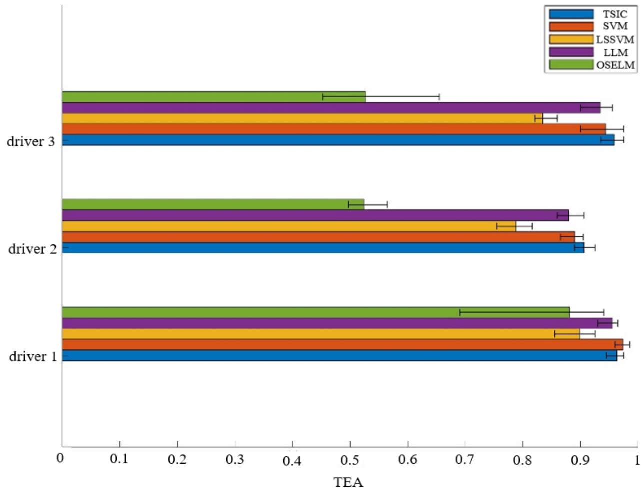


Figure 7. TEA of activity recognition.

From Figures 6 and 7, TSIC works well for excavator activity recognition. It is obvious that TRA and TEA of the driver 1 are better than that of the other two drivers, and the variances are smaller. That is because that the driver has longer working experience than the other two ones.

4.2. Comparison Experiment Results

In this section, TSIC is compared with other classifiers, i.e. SVM, LSSVM, ELM with kernel (KELM), Online Sequential ELM (OSELM), and CCNN. The code of KELM and OSELM are downloaded from http://www.extreme-learning-machines.org/, and the LSSVM is downloaded from http://www.esat.kuleuven.be/sista/lssvmlab/. A Gaussian function is used for a kernel function of SVM, LSSVM, KELM and OSELM. 10 public data sets, downloaded from the UCI machine learning repository [3, 7], are used to train and test the classifiers, and they are listed in Table 4, in which set names, donors, Sample Numbers (SN), and Attribute Numbers (AN) are shown.

Table 4. Information of public data sets.

Set names	**Donor**	**SN.**	**AN.**
Glass	Vina spiehler	214	10
Ionosphere	Vince sigillito	351	35
Iris	R. A. fisher	100	4
Liver disorders	Richard S. forsyth	345	7
Mammographic mass	Matthias elter	961	6
Monk's problems	Sebastian thrun	432	8
Pima indians diabetes	Vincent sigillito	768	8
Sonar	Terry sejnowski	208	61
German	Hofmann	1000	20
Wdbc	Wolberg willian H. nick street W.	569	30

In order to compare the classifiers, a 5-fold cross-validation method is adopted to train and test the classifiers. Parameters of TSIC are listed in Table 3, and the kernel parameter σ is optimized by using the function fitcsvm. Parameters of CCNN refer to [41]. Experiment results, i.e., TRA, TEA, and numbers of hidden-layer nodes, support vector and dimension of T (the three parameters are denoted by NSD), are shown in Figures 8, 9, and 10.

From Figure 8, it is obtained that although the TRA of TSIC is not the best, but it is comparable to that of SVM. From Figure 9, for most of the data sets, TSIC has the best TEA and means of TEA. Compared with CCNN, accuracy confidence intervals of TSIC are much tighter, for most data sets. As shown in Figure 10, the NSD of TSIC is the smallest. So, compared with Figures 8 and 9, it is easy to deduce that TSIC has better generalization performance. From theorem 1, we have the same conclusion.

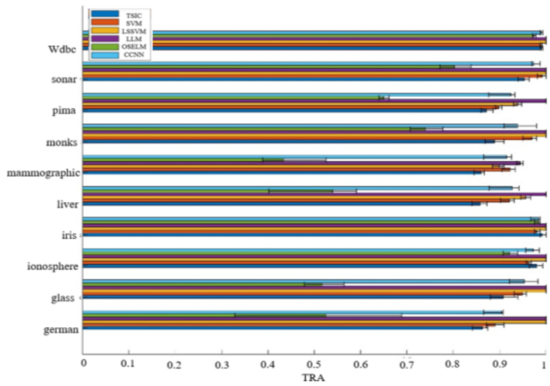


Figure 8. TRA of experiment results.

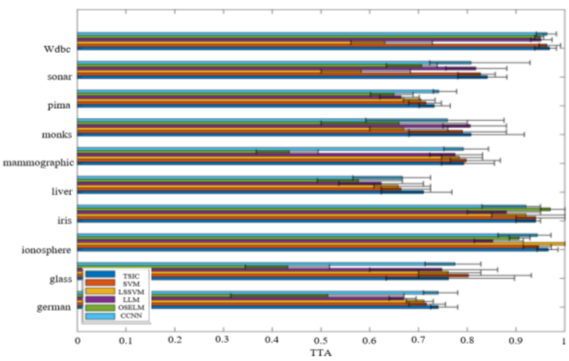


Figure 9. TEA of experiment results.

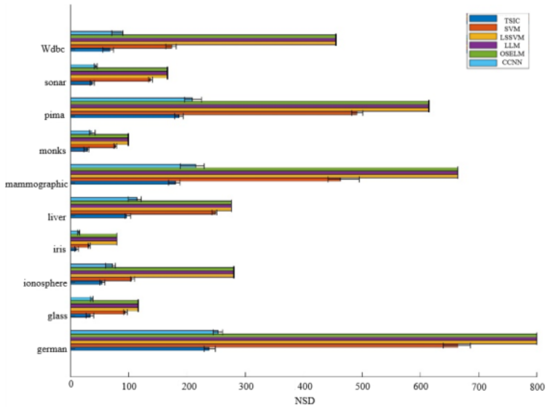


Figure 10. NSD of experiment results.

4.3. Larger Dataset Based Experiments

To verify the generalization ability of TSIC on large datasets, the set, named MAGIC gamma telescope, is employed, and its donor is P. Savicky. Additionally, its number of instances is 19020, and its number of attributes is 11. And in the experiment, TSIC is compared with the method of deep forest [44], and its parameters of TSIC can refer to Table 3. The window size of the deep forest is 6. Experiment results are shown in Table 5.

Table 5. Larger dataset based experiment results.

TSIC		**Deep forest**	
TRA(%)	**TEA(%)**	**TRA(%)**	**TEA(%)**
90.52±0.002	83.86±0.007	81.05±1.50	83.50±0.78

From Table 5, it is concluded that TEA of TSIC is comparable to that of the deep forest, and TRA of TSIC is better than that of the deep forest. Therefore, TSIC has better generalization ability for large datasets.

4.4. Discussion of TSIC Parameters

In this section, impact of the TSIC parameters on generalization performance is discussed. The parameters are λ_M, λ_c, C and λ_w. For a discussed parameter, other parameters refer to Table 3.

4.4.1. Discussion of λ_M

The weight factor λ_M is used to increase generalization of distance metric learning. A range of λ_M is given, i.e., {0, 0.00001, 0.0001, 0.001, 0.01, 0.1, 0.5, 1, 1.5, 2.0, 2.5, 3}. Five data sets are used for experiments, and TRA, TEA, and NSD of TSIC are obtained and shown in Figure 11.

From Figure 11, it is obtained that λ_M is beneficial for increasing generalization of TSIC, but it can't be too big for some sets, e.g., the set wdbc. Considering the set ionosphere, NSD is the biggest when λ_M =0.5, and TEA is the smallest. For the sets sonar and monks, the bigger λ_M, the better generalization performance. Considering the set glass, TEA decreases as λ_M increases, and it is the minimum when λ_M=0.5. Meanwhile, TRA is the minimal, and NSD is the maximal when λ_M is near 0.5.

Thus, λ_M cannot adopt a uniform value for different sets.

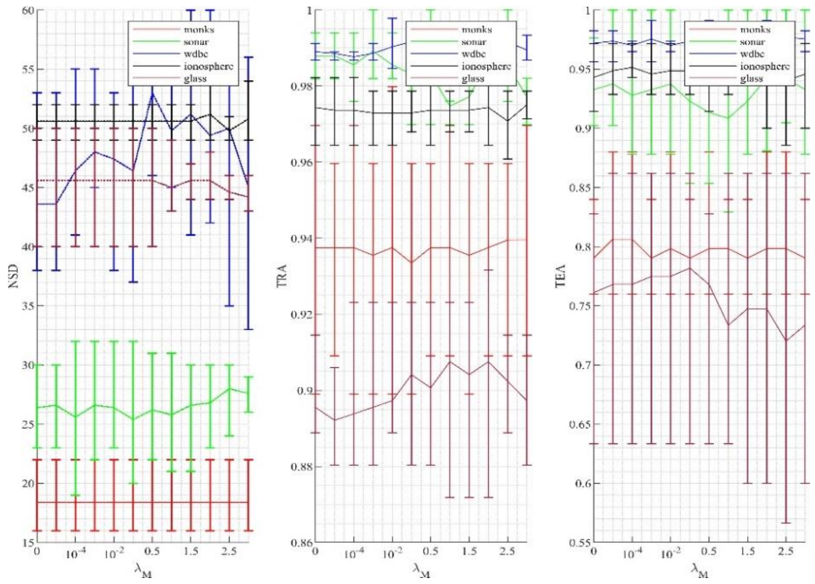


Figure 11. Impact of λ_M on TSIC.

4.4.2. Discussion of λ_c

In this section, impact of λ_c on TSIC is analyzed. A range of λ_c, {0, 0.1, 0.2, …, 0.9, 1}, is tried to obtain TRA, TEA, and NSD. In order to reduce impact of L, let L be a unit matrix. Four data sets are used for experiments, and results are shown in Figure 12.

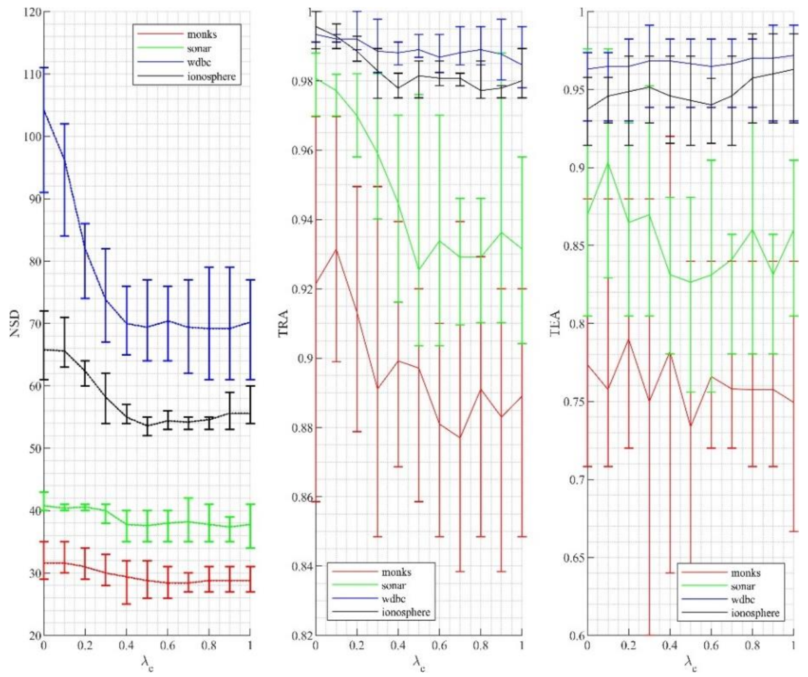
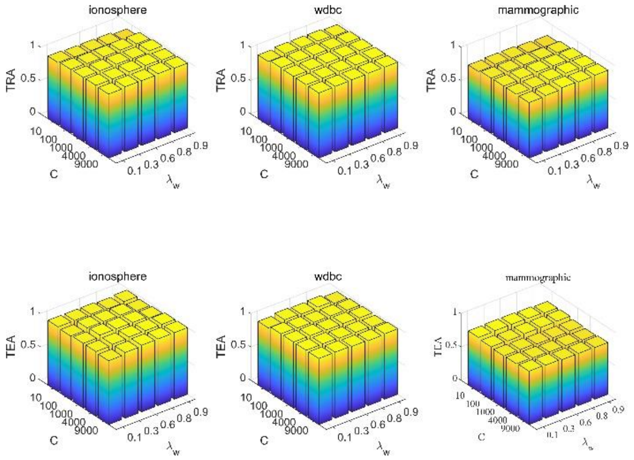


Figure 12. Impact of λ_c on TSIC.

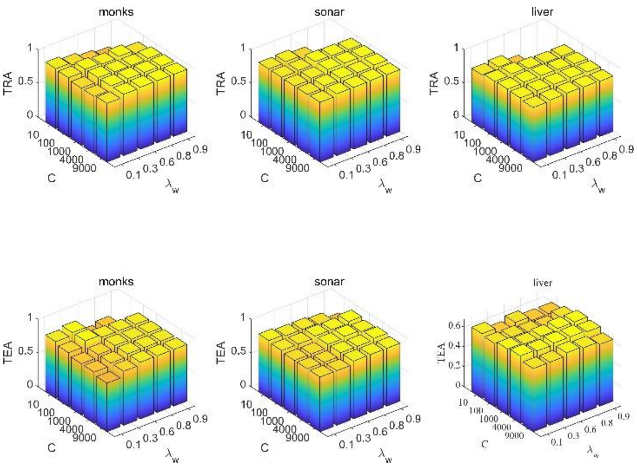
From Figure 12, it is obvious that the bigger λ_c, the smaller NSD. TRA decreases as λ_c increases. For the data sets wdbc and ionosphere, the TEA increases as λ_c increases. But for the other two sets, the TEA decreases as λ_c increases. In other words, the bigger λ_c, the better generalization performance. Therefore, generalization performance of TSIC is affected by λ_c. Considering the sets monks and sonar, the TRA and TEA are better when λ_c is smaller. It is because that the bigger λ_c, the smaller distance from a covered sample to its corresponding boundary, and there is no significant changes of their NSD. It leads to insufficient discrimination of the hidden layer output, and increases risk of covering different-class samples.

4.4.3. Discussion of C and λ_w

In this section, impact of C and λ_w in the cost function in Equation (15) is discussed. A wide range of C is tried, its different values are {10, 100, 1000, 4000, 9000}. Different values of λ_w are {0.1, 0.3, 0.6, 0.8, 0.9}. They result 25 pairs of (C, λ_w). In order to avoid disturbance of L, L is an unit matrix. Six data sets are used for discussion, simulation results are shown in Figure 13.



a) Experiment results for the datasets ionosphere, wdbc, and mammographic.



b) Experiment results for the datasets monks, sonar, and liver

Figure 13. Impact of (C, λ_w) on TSIC.

From Figure 13-a), TRN and TEA are not sensitive to the pair of (C, λ_w). If the parameter λ_w is fixed, the bigger C, the higher TRA. But the distinction of the TRA or the TEA is smaller. From Figure 13-b), if the parameter C is smaller, the TRA and TEA decrease as the parameter λ_w increases. If the parameter C is 9000 and the parameter λ_w is 0.1, the TEA is lower. If the parameter C is near 1000~4000, the bigger λ_w, the better generalization performance. So if the parameter C is near 1000~4000 and the parameter λ_w is near 0.6~0.7, TSIC has better TRA and TEA. Thus, the pair of (C, λ_w) needs to be selected for different data sets.

5. Conclusions

In this paper, TSIC is proposed and used for excavator activity diagnosis. Sample features are explored from input samples by using the improved metric learning, and they are mapped onto a hypersphere. Based on sample distribution analysis, covers are built and used for topology feature extraction. Then a weighted linear classifier is designed to recognize the topology feature. TSIC makes full use of topology and statistical information of samples. Generalization performance is analyzed, and impact of TSIC parameters is discussed. Experimental results demonstrate feasibility and effectiveness of TSIC.

Acknowledgement

This research was funded by the Shandong Provincial Natural Science Foundation No. ZR2022MF345; the Key Technology Ministry of Transport of the People's Republic of China No. 2022-MS6-159; the Support Plan for Shandong Key Laboratory of Transportation Industry; Shandong Provincial Key Laboratory of Transportation Industry.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Akhavian R. and Behzadan A., "An Integrated Data Collection and Analysis Framework for Remote Monitoring and Planning of Construction Operations," *Advanced Engineering Informatics*, vol. 26, no. 4, pp. 749-761, 2012. https://doi.org/10.1016/j.aei.2012.04.004

[2] Akhavian R. and Behzadan A., "Construction Equipment Activity Recognition for Simulation Input Modeling Using Mobile Sensors and Machine Learning Classifiers," *Advanced Engineering Informatics*, vol. 29, pp. 867-877, 2015. https://doi.org/10.1016/j.aei.2015.03.001

[3] Alawneh H. and Hasasneh A., "Survival Prediction of Children After Bone Marrow Transplant Using Machine Learning Algorithms," *The International Arab Journal of Information Technology*, vol. 21, no. 3, pp. 394-407, 2024.

[4] Bae J., Kim K., and Hong D., "Automatic Identification of Excavator Activities Using Joystick Signals," *International Journal of Precision Engineering and Manufacturing*, vol. 20, no. 3, pp. 2101-2107, 2019. https://doi.org/10.1007/s12541-019-00219-5

[5] Chen C., Zhu Z., and Hammad A., "Automated Excavators Activity Recognition and Productivity Analysis from Construction Site Surveillance Video," *Automation in Construction*, vol. 110, pp. 103045, 2020. https://doi.org/10.1016/j.autcon.2019.103045

[6] Cheng C., Rashidi A., Davenport M., and Anderson D., "Activity Analysis of Construction Equipment Using Audio Signals and Support Vector Machines," *Automation in Construction*, vol. 81, pp. 240-253, 2017. https://doi.org/10.1016/j.autcon.2017.06.005

[7] Dua D. and Graff C., UCI Machine Learning Repository, CA: University of California, School of Information and Computer Science, 2019.

http://archive.ics.uci.edu/ml

[8] Gaikwad S., Gawali W., and Yannawar P., "A Review on Speech Recognition Technique," *International Journal of Computer Applications*, vol. 10, no. 3, pp. 16-24, 2010. DOI:10.5120/1462-1976

[9] Ganaie M., Tanveer M., and Lin C., "Large-Scale Fuzzy Least Squares Twin SVMs for Class Imbalance Learning," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 11, pp. 4815-4827, 2022. DOI:10.1109/TFUZZ.2022.3161729

[10] Ghelmani A. and Hammad A., "Improving Single-Stage Activity Recognition of Excavators Using Knowledge Distillation of Temporal Gradient Data," *Computer-Aided Civil and Infrastructure Engineering*, vol. 39, no. 13, pp. 2028-2053, 2024. DOI: 10.1111/mice.13157

[11] Ghelmani A., Torabi G., Hammad A., and Chen C., "Single-Stage Spatiotemporal Activity Recognition of Excavators: A Case Study," *in Proceedings of the 41st International Symposium on Automation and Robotics in Construction*, Lille, pp. 800-807, 2024. https://www.iaarc.org/publications/fulltext/103_ISARC_2024_Paper_181.pdf

[12] Golparvar-Fard M., Heydarian A., and Niebles J., "Vision-Based Action Recognition of Earthmoving Equipment Using Spatio-Temporal Features and Support Vector Machine Classifiers," *Advanced Engineering Informatics*, vol. 27, no. 4, pp. 652-663, 2013. https://doi.org/10.1016/j.aei.2013.09.001

[13] Gong J., Caldas C., and Gordon C., "Learning and Classifying Actions of Construction Workers and Equipment Using Bag-of-Video-Feature-Words and Bayesian Network Models," *Advanced Engineering Informatics*, vol. 25, no. 4, pp. 771-782, 2011. https://doi.org/10.1016/j.aei.2011.06.002

[14] Harichandran A., Raphael B., and Mukherjee A., "Equipment Activity Recognition and Early Fault Detection in Automated Construction through a Hybrid Machine Learning Framework," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, pp. 253-268, 2023. https://doi.org/10.1111/mice.12848

[15] Huang C., Zhou J., Chen J., Yang J., Clawson K., and Peng Y., "A Feature Weighted Support Vector Machine and Artificial Neural Network Algorithm for Academic Course Performance Prediction," *Neural Computing and Applications*, vol. 35, pp. 11517-11529, 2023. https://doi.org/10.1007/s00521-021-05962-3

[16] Huang G., Zhou H., Ding X., and Zhang R., "Extreme Learning Machine for Regression and Multiclass Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 2, pp. 513-529, 2012. DOI:10.1109/TSMCB.2011.2168604

[17] Jain M., Kaur G., and Saxena V., "A K-Means Clustering and SVM based Hybrid Concept Drift Detection Technique for Network Anomaly Detection," *Expert Systems with Applications*, vol. 193, pp. 116510, 2022. https://doi.org/10.1016/j.eswa.2022.116510

[18] Kim D., Lee K., Lee D., and Lee K., "A Kernel-Based Subtractive Clustering Method," *Pattern Recognition Letters*, vol. 26, no. 7, pp. 879-891, 2005. https://doi.org/10.1016/j.patrec.2004.10.001

[19] Kim I., Latif K., Kim J., Sharafat A., and Lee D., "Vision-Based Activity Classification of Excavators by Bidirectional LSTM," *Applied Science*, vol. 13, no. 1, pp. 1-19, 2023. https://doi.org/10.3390/app13010272

[20] Kim J., Kim D., Shah J., and Lee S., "Synthetic Training Image Dataset for Vision-Based 3D Pose Estimation of Construction Workers," *in Proceedings of the Construction Research Congress*, Arlington, pp. 254-262, 2022. https://ascelibrary.org/doi/full/10.1061/9780784483961.027

[21] Kingma D. and Ba J., "Adam: A Method for Stochastic Optimization," *in Proceedings of the International Conference on Learning Representations*, Banff, pp. 1-1, 2014. https://api.semanticscholar.org/CorpusID:6628106

[22] Langroodi A., Vahdatikhaki F., and Doree A., "Activity Recognition of Construction Equipment Using Fractional Random Forest," *Automation in Construction*, vol. 122, pp. 103465, 2021. https://doi.org/10.1016/j.autcon.2020.103465

[23] Lee J., Kim B., Sun D., Han C., and Ahn Y., "Development of Unmanned Excavator Vehicle System for Performing Dangerous Construction Work," *Sensors*, vol. 19, no. 22, pp. 1-17, 2019. https://doi.org/10.3390/s19224853

[24] Li X., Zhang Y., Yu Y., Wang W., and Zhao Z., "Filtering and Denoising of Rock Blasting Vibration Signal with EMD," *Journal of Central South University*, vol. 37, no. 1, pp. 150-154, 2006. https://doi.org/10.1038/s41598-023-47755-9

[25] Liang N., Huang G., Saratchandran P., and Sundararajan N., "A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411-1423, 2006. DOI:10.1109/TNN.2006.880583

[26] Man Z., Lee K., Wang D., Cao Z., and Khoo S., "Robust Single-Hidden Layer Feedforward Network-Based Pattern Classifier," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 12, pp. 1974-1986, 2012. DOI:10.1109/TNNLS.2012.2218616

[27] Man Z., Lee K., Wang D., Cao Z., and Miao., "A New Robust Training Algorithm for a Class of Single Hidden Layer Neural Networks," *Neurocomputing*, vol. 74, no. 16, pp. 2491-2501, 2011. https://doi.org/10.1016/j.neucom.2010.11.033

[28] Mathur M., Aria S., Adams T., and Ahn C., "Automated Cycle Time Measurement and Analysis of Excavator's Loading/Operation Using Smart Phone-Embedded IMU Sensors," *in Proceedings of the ASCE International Conference on Computing in Civil Engineering*, Corvallis, pp. 215-222, 2015. DOI:10.1061/9780784479247.027

[29] Molaei A., Kolu A., Lahtinen K., and Geimer M., "Automatic Recognition of Excavator Working Cycles Using Supervised Learning and Motion Data Obtained from Inertial Measurement Units (IMUs)," *Construction Robotics*, vol. 8, no. 2, pp. 1-20, 2024. https://doi.org/10.1007/s41693-024-00130-0

[30] Pradhananga N. and Teizer J., "Automatic Spatio-Temporal Analysis of Construction Site Equipment Operations Using GPS Data," *Automation in Construction*, vol. 29, pp. 107-122, 2013. https://doi.org/10.1016/j.autcon.2012.09.004

[31] Rashid K. and Louis J., "Construction Equipment Activity Recognition from IMUs Mounted on Articulated Implements and Supervised Classification," *in Proceedings of the ASCE International Conference on Computing in Civil Engineering*, Atlanta, pp. 130-138, 2019. DOI:10.1061/9780784482445.017

[32] Seo J. and Kim S., "The Study on Measures for Reducing Safety Accidents of Excavator," *Journal of the Korea Institute of Building Construction*, vol. 8, no. 3, pp. 127-133, 2008. https://doi.org/10.5345/JKIC.2008.8.3.127

[33] Shao Y., Lv X., Huang L., and Bai L., "Twin SVM for Conditional Probability Estimation in Binary and Multiclass Classification," *Pattern Recognition*, vol. 136, pp. 109253, 2023. https://doi.org/10.1016/j.patcog.2022.109253

[34] Shen Y., Wang J., Feng C., and Wang Q., "Dual Attention-Based Deep Learning for Construction Equipment Activity Recognition Considering Transition Activities and Imbalanced Dataset," *Automation in Construction*, vol. 160, pp. 105300, 2024. https://doi.org/10.1016/j.autcon.2024.105300

[35] Suykens J., Brabanter J., Lukas L., and Vandewalle J., "Weighted Least Squares Support Vector Machines: Robustness and Sparse Approximation," *Neurocomputing*, vol. 48, pp. 85-105, 2002. https://doi.org/10.1016/S0925-2312(01)00644-0

[36] Vahdatikhaki F. and Hammad A., "Framework for Near Real-Time Simulation of Earthmoving Projects Using Location Tracking Technologies," *Automation in Construction*, vol. 42, pp. 50-67, 2014. https://doi.org/10.1016/j.autcon.2014.02.018

[37] Vapnik V., *The Nature of Statistics Learning Theory*, Springer, 1995. https://doi.org/10.1007/978-1-4757-2440-0

[38] Wang C., Penga G., and De Baets B., "Embedding Metric Learning into an Extreme Learning Machine for Scene Recognition," *Expert Systems with Applications*, vol. 203, pp. 117505, 2022. https://doi.org/10.1016/j.eswa.2022.117505

[39] Wang H. and Shao Y., "Fast Truncated Huber Loss SVM for Large Scale Classification," *Knowledge-Based System*, vol. 260, pp. 110074, 2023. https://doi.org/10.1016/j.knosys.2022.110074

[40] Wang H., Hu L., and Zhang Y., "SVM Based Imbalanced Correction Method for Power Systems Transient Stability Evaluation," *ISA Transactions*, vol. 136, pp. 245-253, 2023. https://doi.org/10.1016/j.isatra.2022.10.039

[41] Wang M., Lu Y., and Pan W., "An Improved Pattern-Based Prediction Model for a Class of Industrial Processes," *Transactions of the Institute of Measurement and Control*, vol. 44, no. 7, pp. 1410-1423, 2022. https://doi.org/10.1177/01423312211055569

[42] Wang S., Qu Y., Li W., and Qin H., "Face Recognition: Biomimetic Pattern Recognition vs. Traditional Pattern Recognition," *Acta Electronica Sinica*, vol. 32, no. 7, pp. 1057-1061, 2004.

[43] Weinberger K. and Saul K., "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Journal of Machine Learning Research*, vol. 10, pp. 207-244, 2009. https://jmlr.csail.mit.edu/papers/volume10/weinberger09a/weinberger09a.pdf

[44] Zhou Z. and Feng J., "Deep Forest," *National Science Review*, vol. 6, no. 1, pp. 74-86, 2019. DOI:10.1093/nsr/nwy108

Mushu Wang received the Ph.D. degree in Control science and Engineering from University of Science and Technology Beijing in 2019. He works in the School of Information Science and Electrical Engineering (School of Artificial Intelligence) in the Shandong Jiaotong University. His research interests include Modeling and Control of Complex Processes, Pattern Recognition and its Applications, Image Processing.

YuKui Qin is currently a postgraduate student majoring in Robotics Engineering at Shandong Jiaotong University. His research interests include Path Planning and Tracking Control, Artificial Intelligence, Embedded Development and Application.

Tiepeng Bi received the Bachelor's degree in safety engineering from Kunming University of Science and Technology in China. He is currently serving as the project safety manager at Shanghai Baosteel Engineering Consulting Company in China. His research interests include Engineering Safety, Emergency Technology, Smart Safety (Including Big Data, Internet of Things, Artificial Intelligence, Cloud Computing, etc.) and their applications.

Weigang Pan received the Ph.D. degree in Pattern Recognition and Intelligent System from Shandong University in 2013. He works in the School of Rail Transportation in the Shandong Jiaotong University. His research interests include Intelligent Control, Process Control, Intelligent Driving and Manufacturing, Industrial Internet of Things.