Fault Diagnosis of Intelligent Production Line Based on Digital Twin and Improved Random Forest

Kai Guo, Xiang Wan, Lilan Liu *, Zenggui Gao and Muchen Yang

Abstract: Digital twin (DT) is a key technology for realizing the interconnection and intelligent operation of the physical world and the world of information and provides a new paradigm for fault diagnosis. Traditional machine learning algorithms require a balanced dataset. Training and testing sets must have the same distribution. Training a good generalization model is difficult in an actual production line operation process. Fault diagnosis technology based on the digital twin uses its ultrarealistic, multisystem, and high-precision characteristics to simulate fault data that are difficult to obtain in an actual production line to train a reliable fault diagnosis model. In this article, we first propose an improved random forest (IRF) algorithm, which reselects decision trees with high accuracy and large differences through hierarchical clustering and gives them weights. Digital twin technology is used to simulate a large number of balanced datasets to train the model, and the trained model can be transferred to a physical production line through transfer learning for fault diagnosis. Finally, the feasibility of our proposed algorithm is verified through a case study of an automobile rear axle assembly line, for which the accuracy of the proposed algorithm reaches 97.8%. The traditional machine learning plus digital twin fault diagnosis method proposed in this paper involves some generalization, and thus has practical value when extended to other fields.

Keywords: fault diagnosis; digital twin; random forest; hierarchical clustering; transfer learning

1. Introduction

Intelligent manufacturing is the engine driving the development of future industry. With the rapid development of communication technology and information technology [1], based on cloud computing, big data, and the Internet of Things [2], a new round of industrial transformation represented by intelligent manufacturing has begun globally. Digital twin (DT) technology supports the core technology system of intelligent manufacturing, promotes technology integration, and is considered to be the key to realizing intelligent manufacturing. Digital twin technology no longer refers to three-dimensional static models, but refers to the use of physical models, sensor updates, operating history, and other data to map physical entities to virtual space, so as to reflect the whole life cycle process of the corresponding physical entities and provide a reference for system analysis and decision-making [3]. A digital twin model will continuously collect real-time data from the physical production line and use real-time data and historical data for model training, model verification, model updating, and feedback of the final results to the physical production line to achieve production control. The physical production line will be optimized based on the results of the virtual production line (such as the fault diagnosis results in this paper).

With the development of science and technology, the intelligent production line is developing towards a high degree of automation and integration. As the production process becomes increasingly complex, the possibility of related failures in the entire system will increase during the continuous operation of the production line. Minor failures in the
production process may cause irreparable losses. At the same time, with the introduction of more large-scale mechanized equipment in the production line, the equipment has a production order relationship. As a part of the production line, the stability and safety of a single piece of equipment directly affects the continuous operation of the whole production line [4]. With the development of machine learning technology, the acquisition of knowledge is not completely dependent on expert experience, but uses some intelligent methods based on data. The data-based fault diagnosis method has strong practicability and universality. Data-based intelligent methods can deeply mine high-dimensional data and extract high-value information hidden in the data. This high-value information can reflect the essence of the object. Therefore, we can establish the corresponding fault diagnosis model to effectively complete various fault diagnosis tasks. By using the data generated in the production line to establish a model, it is possible to judge the running state of the equipment in real time. Mining the valuable information behind the data is why data-driven fault detection methods have attracted widespread attention in recent years. There are many classic algorithms in this field that significantly improve the efficiency and accuracy of fault diagnosis, such as Bayesian networks [5], support vector machines (SVMs) [6], artificial neural networks (ANNs) [7], decision trees [8], etc. [9–11]. Yang et al. proved that random forest (RF) has a higher accuracy rate than ANNs and SVMs, and so it is used in machine fault diagnosis [12].

In recent years, an increasing number of scholars have used machine learning methods for multi-class fault classification. For example, Kadri et al. [13] proposes a hybrid algorithm based on binary ant colony (BACO) and SVM. The BACO algorithm is used to select the appropriate feature subset and optimize the parameters of the SVM to improve the classification accuracy. Wan et al. [14] proposed a method of combining spark and improved random forest to solve the problem of fault diagnosis. By eliminating decision trees with lower accuracy, a decision tree with higher accuracy and faster accuracy can be established. Then, a series of experiments can verify the effectiveness of the algorithm. Chen et al. [15] used the unsupervised training of a denoising autoencoder to obtain a deep neural network for fault diagnosis. Azadeh et al. [16] used a genetic algorithm and particle swarm optimization algorithm to optimize the super parameters of an SVM and applied them to the fault diagnosis of a centrifugal valve. The above studies have proven the effectiveness of machine learning methods for fault diagnosis.

Ensemble learning is a method of machine learning, which uses multiple models to solve the same problem. The core idea of ensemble learning is to build multiple models, and then fuse the decision results of each model to get better results than a single model. By combining multiple models, the error of a single model may be compensated by other models, so the overall performance of the ensemble is better than that of a single model [17]. In an actual fault diagnosis experiment on an elevator motor system, Niu et al. [18] adopted four algorithms fusing majority voting, which verifies the feasibility of the algorithm through an experiment and proves the great potential of ensemble learning in practical applications. Random forest [19] is one of the most classic algorithms in ensemble learning. Random forest [19] is one of the most classic and most popular algorithms in ensemble learning [17]. Random forest has been used in various fields since it was first proposed. The random forest algorithm has a simple structure, superior performance, and strong interpretability. Compared with other machine learning methods, it performs better in many classification and regression tasks [20]. Due to the randomness, random forest is not easy to overfit; it has good antinoise ability and is not sensitive to outliers or abnormal points. Yang et al. studied the possibility of applying the random forest (RF) algorithm to machine fault diagnosis and proposed a hybrid method combined with a genetic algorithm to improve the accuracy of the classification [12].

However, traditional ensemble learning also has disadvantages. Zhou et al. [21,22] have found that there are differences among the classifiers in ensemble learning, and the results of some classifiers will reduce the overall accuracy. Using the selective integration method, it is better to select a good classifier than selecting all classifiers. For example,
Zhou et al. [21,22] proved that it may be better to ensemble many instead of all of the neural networks and decision trees from the point of view of regression and classification, and proposed an algorithm called GASEN and GASEN-b, which uses a genetic algorithm to improve weights. Lee et al. [23] proposed a multi-objective instance weight transfer learning network to solve the problem of fault diagnosis. The effectiveness of the proposed algorithm was verified by industrial robot and spot-welding experiments.

Most traditional machine learning algorithms used for classification must satisfy two preconditions to perform well [24]: (1) the training dataset contains balanced information; (2) the training dataset and the test dataset satisfy the same feature distribution. These conditions are difficult to meet in the actual production. In the actual operation of the production line, most of the data are obtained under normal conditions, so fault data are difficult to obtain, and the loss of the production line is irreparable. In addition, if the production conditions change, the training dataset and the test dataset do not satisfy the same feature distribution. The accuracy of the model will decline sharply, so a model trained in a previous stage cannot be used in the current stage. The cost of retraining the model will increase [25].

Therefore, in this paper, we propose a method combining digital twin with an improved random forest algorithm for fault diagnosis on an intelligent production line. The contributions of this paper are as follows:

- We propose an improved random forest algorithm which first constructs multiple decision trees by bootstrap sampling, then refilter the decision trees by a hierarchical clustering method, calculate the differences between each tree, select the highest accuracy of each tree for integration, and finally conduct weighted voting. An improved random forest algorithm is constructed. The method is applied for fault diagnosis verification on a rear axle assembly line for automobiles, which proves that our method has higher accuracy than random forest.

- In order to solve the problem of an unbalanced training dataset and the lack of a high volume of fault data in practical applications, we propose using digital twin to simulate a large amount of balanced data, and then transferring it to the physical production line. The data in the virtual space are used as a training set to train the improved random forest algorithm, and then hierarchical clustering is used to refilter the decision tree. Through the transfer of learning applied to the actual production line, we can then use real data to fine-tune and train a reliable model.

- The effectiveness of our proposed method was verified by the actual automobile rear axle assembly line. The accuracy and efficiency of the DT + IRF algorithm are better than those of an ordinary machine learning algorithm.

In Figure 1, we present an overview of this paper. At the physical entity layer, we will establish three-dimensional models of the tighten robot, loading robot, unloading robot, conveyor belt, four-wheel positioning device, etc., using lightweight tools for model optimization and building real-time mapping digital twin scenes. The connection between the background service and the source data storage system is established by calling relevant interfaces, so that the background service can obtain the status signals collected by PLC and sensors and display them in the visual interface. Then we will verify the data reliability to ensure that the data is not lost during transmission. Finally, the data is input into the algorithm for decision-making. A 3D visualized fault diagnosis system is established through digital twin technology and an IRF algorithm, and the fault diagnosis results are analyzed through data in real time. We will introduce the basic theory in Section 2. We describe the proposed DT + IRF method in Section 3. In Section 4 we use an automobile rear axle assembly line to verify the feasibility of the algorithm and present the experimental results. In Section 5 we compare our algorithm with others. Section 6 gives the conclusions of this paper. The abbreviations in the paper can be seen in the Abbreviation part.
2. Random Forest and Its Improvements

2.1. Random Forest Algorithm

Random forest [19] is a machine learning algorithm that was proposed by Breiman in 2001. The algorithm has a very high accuracy rate. It is not easy to overfit and has good antinoise ability. In addition, it is not sensitive to abnormal points, is easy to implement, and performs quickly in training. Because of its simple structure and superior performance, it has been widely used in various fields for classification and regression tasks. Random forest is achieved through bagging and random selection. Random forest uses a random sampling method to generate multiple subtraining sets and corresponding test sets from the original data. Repeated resampling means that duplicate data exist in each training subset, which avoids the problem of falling into local extremes. Then, different decision tree models are trained by bagging data to obtain the final decision. Unlike decision trees, the random forest algorithm randomly selects features. The process of random forest, from sampling to final voting, is shown in Figure 2 [26].
The construction process of random forest is as follows:

1. Randomly select \( n \) subtraining sets and corresponding test sets from the original dataset.
2. When each node of the decision tree needs to be split, \( m \) attributes are randomly selected. Select one attribute from the \( m \) attributes according to the method of information gain as the split attribute.
3. Repeat step 2 until it can no longer be split.
4. Repeat steps 1–3 to build a large number of decision trees to form a random forest.

2.2. Hierarchical Clustering Algorithm Rescreening Decision Trees

The hierarchical clustering algorithm performs hierarchical decomposition according to the similarity between data points to create a nested clustering tree with a hierarchical structure. Whether a hierarchical clustering algorithm is agglomerative or divisive mainly depends on the process of hierarchical decomposition. The bottom-up hierarchical decomposition corresponds to the agglomerative method and the top-down hierarchical decomposition corresponds to the divisive method. Agglomerative hierarchical clustering has lower complexity than divisive hierarchical clustering [27]. The specific steps of agglomerative hierarchical clustering are as follows:

1. Treat each dataset as a cluster, calculate the distance between each pair of data points, and finally obtain the distance matrix.
2. Find the two closest clusters according to the distance matrix, and then merge them. Update the distance matrix.
3. If the data are in one cluster or reach the preset number of clusters, stop. Otherwise, repeat step 2.

In this paper, the linkage criterion used is average linkage, which can be expressed as follows:

\[
d_{\text{avg}}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} \text{Dis}(x, y)
\]  

(1)

where \( C_i \) and \( C_j \) are given clusters; \( \text{Dis} \) is the similarity calculation formula between two points, and the specific formula will be given below.

The accuracy of the random forest algorithm is related to the base classifier. The existing conclusion shows that [28] the higher the prediction accuracy and the greater the difference between the base classifier, the better the ensemble effect. Conversely, the
lower the accuracy and difference of the base classifier, the worse the final result will be. The higher the classification accuracy of each base classifier, the higher the confidence of the final voting results and the accuracy of the random forest algorithm. If the similarity between two base classifiers is high, there will be repeated voting, which will lead to a decrease in the accuracy of random forest. Therefore, we improve the traditional random forest algorithm through a hierarchical clustering algorithm to select a base classifier with high accuracy and low similarity. The algorithm flowchart is shown in Figure 3.

**Figure 3.** Flowchart of the algorithm.

If the initial data are \( X = \{x_1, x_2, \ldots, x_K\} \), where sample \( x \in \mathbb{R}^m \) and \( K \) is the number of samples, \( N \) training datasets \( \{X_1, X_2, \ldots, X_N\} \) are obtained by bootstrap resampling. The corresponding decision tree is \( \{T_1, T_2, \ldots, T_N\} \). The decision tree of this paper adopts C4.5, and the classification attribute of nodes adopts the attribute with the largest information gain rate. The generated classification rules are easy to understand and have a high accuracy rate.

After the completion of RF construction, the accuracy of each tree is calculated through the test set. The difference between each pair of trees will be calculated to form a matrix \( S \):

\[
S = \begin{bmatrix} a & b \\ c & d \end{bmatrix}
\]

where \( a \) and \( d \) represent being classified correctly and incorrectly, respectively; \( b \) and \( c \) represent accurate classification by only one of the base classifiers at the same time.

The difference between a pair of trees \( (T_i \) and \( T_j) \) is calculated by the following formula:

\[
\text{Dis}_{ij} = 1 - \frac{a + d}{a + b + c + d}
\]
The distance matrix \(D\) in hierarchical clustering is represented by the similarity between trees:

\[
D = \begin{bmatrix}
\text{Dis}_{1,1} & \cdots & \text{Dis}_{1,N} \\
\vdots & \ddots & \vdots \\
\text{Dis}_{N,1} & \cdots & \text{Dis}_{N,N}
\end{bmatrix}
\]  
(4)

The decision tree in the random forest is taken as a single sample point, and the dissimilarity between any two trees is calculated. According to the hierarchical clustering algorithm, the two trees with the smallest distance are found. Then they are combined into a new tree, and this process is repeated until they reach the preset value. Finally, the decision tree with the highest precision in each cluster is selected to generate a new random forest.

The new random forest classification accuracy matrix is as follows:

\[
A = \begin{bmatrix}
\text{acc}_{1,1} & \cdots & \text{acc}_{1,n} \\
\vdots & \ddots & \vdots \\
\text{acc}_{m,1} & \cdots & \text{acc}_{m,n}
\end{bmatrix}
\]  
(5)

In the formula, \(i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n\). \(m\) represents the category of the classification algorithm, \(n\) represents the tree of the new random forest decision tree, \(\text{acc}_{i,j}\) represents the accuracy of the decision tree \(j\) for category \(i\) classification. Then, the weight of each tree is calculated according to the accuracy rate:

\[
\omega_{i,j} = \frac{\text{acc}_{i,j}}{\sum_{j=1}^{n} \text{acc}_{i,j}}
\]  
(6)

\[
W = \begin{bmatrix}
\omega_{1,1} & \cdots & \omega_{1,n} \\
\vdots & \ddots & \vdots \\
\omega_{m,1} & \cdots & \omega_{m,n}
\end{bmatrix}
\]  
(7)

In the formula, \(\omega_{i,j} i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n\) represents the weight of the \(j\)-th tree for the category \(c_i\). For data \(X_{\text{new}}\), if \(\varphi_j(X_{\text{new}}) = i\), this means that the classification result of the classifier is \(c_i\). The vote of the \(j\)-th classifier for the classification result \(c_i\) of the production line fault diagnosis data is given by \(\text{vot}_{i,j}\):

\[
\text{vot}_{i,j} = \begin{cases}
1 & \text{when } \varphi_j(X_{\text{new}}) = i \\
0 & \text{otherwise}
\end{cases}
\]  
(8)

where \(i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n\).

Finally, the resulting voting matrix is:

\[
\text{Vot} = \begin{bmatrix}
\text{vot}_{1,1} & \cdots & \text{vot}_{1,m} \\
\vdots & \ddots & \vdots \\
\text{vot}_{n,1} & \cdots & \text{vot}_{n,m}
\end{bmatrix}
\]  
(9)

The final voting result is:

\[
\omega_{\text{vot}} = \sum_{j=1}^{n} \omega_{i,j} \times \text{vot}_{i,j}
\]  
(10)

where \(i = 1, 2, \ldots, m\). Finally, the result with the most votes is selected as the classification result of the production line fault diagnosis, \(\text{Result}(i)\):

\[
\text{Result}(i) = \arg\max \{\omega_{\text{vot}}\} i = 1, 2, \ldots, m
\]  
(11)
3. Digital-Twin-Assisted Fault Diagnosis Method Using Transfer Learning

3.1. Digital-Twin-Assisted Transfer Learning

In 2003, Dr. Michael Grieves [29] first proposed the digital twin concept in a product lifecycle management course at the University of Michigan. Digital twin serves as a bridge between the physical world and the virtual world. It has the characteristics of real-time synchronization, high fidelity, and faithful mapping. With the development of high-performance computing, system simulation, the industrial Internet of Things, big data, and machine learning technology, digital twin has become a research hotspot. DT refers to the construction of a one-to-one real-time mapping model corresponding to a physical entity in a virtual space. The concept of Cyber-Physical Systems (CPS) is the basis of DT. The difference between DT and CPS is that DT emphasizes real-time simulation of physical entities. Digital twin is a digital model that integrates multiple physical fields, including the real working conditions of physical entities, materials, and data obtained through real-time sensors. Tao et al. [30] discussed the role of big data in supporting intelligent manufacturing and summarized the historical viewpoint of the data lifecycle in the manufacturing industry. Kendrik et al. [31] selected 123 representative projects and 22 supplementary works to analyze DT from the perspective of product lifecycle management and business innovation and confirmed eight potential future uses of DT. Rojek et al. [32] presented the results of research on the development of digital twins of technical objects. Piltan et al. [33] used intelligent DT combined with machine learning for bearing-anomaly detection and crack-size identification. Resman et al. [34] proposed a five-step approach to planning data-driven digital twins of manufacturing systems and their processes.

Transfer learning [35,36] refers to the use of knowledge obtained through learning in massive adjacent data domains to realize the transfer of knowledge so as to solve similar problems in new data domains faster and more effectively. During the operation of an actual production line, due to a lack of training data in case of failure, most of the data obtained are on the normal operation of the production line. This makes it difficult to obtain a relatively balanced dataset, and thus it is difficult to train a classifier with generalization.

In transfer learning, the data domain from which the knowledge is learned is called the source domain (Dos), and the data domain used as the predictive output is called the target domain (Dot). For a task, \( T_a = \{ Y, f(\cdot) \} \), where \( Y \) is the label space, \( Y = \{ y_1, \ldots, y_m \} \), \( f(\cdot) \) is the target prediction function, which cannot be directly observed, and needs to pass training data \( \{ x_i, y_i \} \) to learn, and \( f(\cdot) \) predicts the corresponding output label \( f(x_i) \) according to the current data \( x_i \). \( f(x_i) \) can be written as the conditional distribution function \( P(y_i|x_i) \), and the source domain can be written as:

\[
D_s = \{(x_{s1}, y_{s1}), \ldots, (x_{sn}, y_{sn})\} \quad x_{si} \in \chi_s
\]  

(12)

In machine learning, this represents the data domain where the training data are located, and \( y_{si} \in Y_s \) is the corresponding class label. Similarly, the target domain can be written as follows:

\[
D_t = \{(x_{t1}, y_{t1}), \ldots, (x_{tn}, y_{tn})\} \quad x_{ti} \in \chi_t
\]  

(13)

Equation (2) indicates the data domain where the test set is located, and \( y_{ti} \in Y_s \) is the corresponding label output. The task learned in the source domain is called the Source task (Tas), and the task applied in the target domain is called the Target task (Tat).

When the target task has less high-quality training data, the learning technology is transferred from some previous tasks to the target task. In this case, transfer learning can save a lot of labeled data in the target domain [37].

During the operation of the actual production line, because most of the time the equipment is operating normally, it is impossible to acquire a large amount of fault data. The loss caused by equipment failure is huge. In this paper we combine digital twin technology and transfer learning technology to obtain more balanced and sufficient data for model training.
The transfer learning process based on digital twins is as follows. First, the source task learner is trained through a large amount of balanced virtual data generated by the digital twin workshop. It then transfers the parameters of the source task learner to the target task learner in another field (a physical workshop). In the target domain, only a small number of balanced labeled data are used as training data to obtain the target task learner. By encoding the learned knowledge into shared parameters through transfer learning, without reconstructing the structure of the learner, the training speed and robustness will be improved. The Digital Twin Assisted Transfer learning framework is shown in Figure 4.

Figure 4. Digital Twin Assisted Transfer learning framework.

3.2. Proposed DT + IRF Method

In an actual factory, the DT + IRF method we propose can be summarized by the following process. First, engineers can use the manufacturing platform to build virtual models corresponding to physical entities. A high-fidelity virtual model is the key point. It includes a geometric model, a function model, and an information model [38]. (1) The geometric model maps the physical entities of the real world. It is used to express the relationship attributes and logic of digital twins. The topological model can be used to describe the kinematic relationship. (2) The function model is used to represent the behavior and function in the virtual model. These include data acquisition, synchronization, and storage, as well as simulation functions and fault diagnosis. (3) The information model connects the physical entity and the virtual model. The data collected by different devices are heterogeneous. This will cause difficulties in data connection. The meaning of various information will be defined in the information model.

When the digital twin scene is built, real-time mapping can be generated with physical entities. We started by training the fault diagnosis model. The overall framework of the proposed algorithm in this paper is shown in Algorithm 1. In the digital twin scenario, the data are obtained by adjusting the parameters generated by the physical entity and the simulation cases in different states, and the corresponding tags are manually added. Then we use the algorithm to train the model. First, the training data are divided into a large number of simulated data and a small number of real data. Among them, the simulated data are balanced, but the actual data lack fault data. The test set is composed of data from the physical production line. In the training model stage, a large number of simulation data are divided into a training set \((X_{v, \text{train}}, Y_{v, \text{train}})\) and a test set \((X_{v, \text{test}}, Y_{v, \text{test}})\), and the training set \((X_{v, \text{train}}, Y_{v, \text{train}})\) of simulation data is used to train the random forest classifier 1. In the test set \((X_{v, \text{test}}, Y_{v, \text{test}})\) results, the hierarchical clustering algorithm is used to pick
the base classifier with the best performance in each cluster of the trained classifier 1 to form a new IRF1. The parameters in IRF1 are used to initialize the new classifier IRF2, and then the real data \((X_P, Y_P)\) in the training set are used to fine-tune IRF2. Finally, IRF2 is tested with the test set \((X_P, Y_P)\), and the test results are obtained.

When everything described above is ready, we input the data generated by the physical entity into the algorithm for real-time fault diagnosis and analysis. Through the fault diagnosis results, the location of the fault can be quickly and accurately determined, and maintenance can be carried out through corresponding solutions.

Algorithm 1  Pseudocode of the Proposed Algorithm.

1: Input:
2: Train dataset:
3: (1) Large number of simulation data \((X_v, Y_v)\)
4: (2) A small amount of physical monitoring data \((X_P, Y_P)\)
5: Testing dataset: Physical monitoring dataset \((X_P, Y_P)\)
6: Training Period:
7: Step 1: Split the dataset \((X_v, Y_v)\) into \((X_v, Y_v)_\text{train}\) and \((X_v, Y_v)_\text{test}\)
8: Step 2: Train the RF classifier 1 with the dataset \((X_v, Y_v)_\text{train}\)
9: Step 3: Use a hierarchical clustering algorithm to select a new IRF classifier 1 composed of the trees with the highest accuracy in each cluster within the dataset \((X_v, Y_v)_\text{test}\)
10: Step 4: Use the parameter of IRF1 as the initial parameter of the IRF classifier 2
11: Step 5: Train the IRF classifier 2 with a small amount of physical monitoring data \((X_P, Y_P)\)
12: Testing Period:
13: Step 6: Test the physical monitoring dataset \(X_P\)
14: Output: Physical monitoring data prediction label \(Y_P\).

4. Experiments and Analysis

4.1. Data Description

The assembly line studied in this paper is an intelligent assembly line based on the background of Industry 4.0 and was built in accordance with intelligent manufacturing production standards. Aiming at the intelligent assembly of automobile rear axles, this line adopts technologies such as assembly production digitization, breaking through key technologies such as multisource heterogeneous data perception and collection, assembly process quality traceability, and online monitoring. The integrated application is based on digital-twin-based rear-axle assembly process virtual and real mapping and synchronization technologies, automatic loading and unloading systems, and other systems to realize the intelligent assembly of the automobile rear axle. As shown in Figure 5, there is a real-time synchronization scenario involving digital twin.

The assembly line realizes the assembly and production of an automobile’s rear axle through four steps: manual assembly, manual pretightening, mechanical arm tightening, and mechanical equipment adjustment. It involves a high degree of automation, including an RFID product traceability system, and the production rhythm reaches 60 jph. Since the assembly line has been in operation for two years, due to the fatigue life of some equipment, a variety of faults are prone to occur in daily production, which affects the normal production. Therefore, it is of practical value to diagnose assembly line equipment faults in a timely manner and provide fault solutions. In this work, the main equipment fault of the intelligent assembly line is analyzed.

The digital twin shop floor, including the structure, characteristics, behaviors, and rules of its physical shop floor, is established. The digital twin workshop is connected to the physical workshop through a programmable logic controller (PLC) and sensors and collects information about each device through the PLC and sensors. There are 15 industrial robots in the assembly line, 13 of which are responsible for tightening, generating data such as posture, torque, angle, current, and voltage, and two are responsible for handling materials, generating posture, grasping, and lowering signals, and current and voltage data. The conveyor belt motor generates current and voltage and the rolling, rotating, and occupying
signal data generated by the conveyor belt. The current and voltage are generated by the four-wheel alignment device. We selected one of the 13 tightening robots and two handling robots for analysis. The description of each device is given in Table 1.

Figure 5. The real-time synchronization scenario of a digital twin.

Table 1. Product line fault diagnosis sample parameters.

|                  | Number of Training Samples in Digital Twin Shop Floor | Number of Testing Samples in Digital Twin Shop Floor | Number of Training Samples in Physical Shop Floor | Number of Testing Samples in Physical Shop Floor | Label |
|------------------|-------------------------------------------------------|------------------------------------------------------|--------------------------------------------------|--------------------------------------------------|-------|
| Normal           | 1400                                                   | 600                                                  | 2100                                             | 900                                              | 0     |
| Conveyor belt motor failure | 1400                                                   | 600                                                  | 350                                              | 150                                              | 1     |
| Tightening robot failure | 1400                                                   | 600                                                  | 350                                              | 150                                              | 2     |
| Loading robot failure | 1400                                                   | 600                                                  | 350                                              | 150                                              | 3     |
| Unloading robot failure | 1400                                                   | 600                                                  | 350                                              | 150                                              | 4     |
| Four-wheel positioning device sensor failure | 1400                                                   | 600                                                  | 350                                              | 150                                              | 5     |

Table 1 shows the selected physical production line and digital twin production line data samples under different fault types. Note that, in the sample data in this table, it is difficult to obtain fault data due to the fact that the equipment rarely fails during the physical manufacturing process. Therefore, the dataset of the physical production line is unbalanced, and most of the sample cases are generated during normal operation. However, in the digital twin workshop, through the synchronous mapping of the digital twin, adjusting the equipment parameters or constructing a vivid simulation case, data under different failure conditions can be obtained. We implemented simulation of the manufacturing process according to Unity3D and a PLC. In the digital twin scene, the data collected by the sensor and PLC were adjusted in reverse. For example, the fault data when the robot arm moves to the wrong position were simulated. We simulated the data collected by various sensors, such as current, voltage, occupation signal, and so on. It should be noted that the actual production line was not running in order to simulate the fault state. The data generated are stored in the historical database (this paper uses the influxDB database to store historical data and simulation data). The number of data samples in the digital twin workshop dataset was balanced. The dataset was constructed in this way to ensure that it was consistent with the actual situation as much as possible.

There was a total of 10,000 virtual data points generated by the digital twin workshop and they were divided 7:3 into the training set and the test set. There were 5000 actual data points generated by the physical workshop. There were more data in the normal operation of the production line; the proportion of fault data was small, and the data were not evenly
distributed. There were 46 characteristic data points, such as temperature, current, voltage, vibration, angle of each axis, and torque. In addition, in order to study the effect of transfer learning in different working conditions, the data generated by the digital twin workshop were from one month earlier than the data generated by the physical workshop, making the data distribution slightly different. In the following, we use DT to represent the data generated on the digital twin shop floor and the physical shop floor. For example, the dataset used in the DT + IRF algorithm represents the data generated in the digital twin shop floor and the physical shop floor.

Based on Unity3D [39] virtual simulation software, we integrate the simulation of the working status of the rear axle assembly line of the automobile, the acquisition of status data, the verification of data reliability, the intelligent analysis of the fault status, and then the results of the matching item dataset and the status of each station are displayed. Running status, fault location warning, and fault solutions form a complete fault diagnosis system for the rear axle assembly line of the automobile.

It can be seen from Figure 6 that, when the fault diagnosis system obtains the sensor signal of the production line, the system will display the current state data, analyze the reliability of the data, and input the data into the algorithm model introduced in this article to determine the current state of the device. Then, the fault location is revealed.

![Fault diagnosis system](image)

**Figure 6.** Fault diagnosis system.

4.2. Results Analysis

Bharathidason [28] proved the effectiveness of screening irrelevant high-performance trees for classification tasks in random forests. Bharathidason [28] used AUC accuracy as an indicator to evaluate the performance of each tree, then selected good decision trees from a large number of trees according to the value of high AUC accuracy. The selected trees are clustered according to the correlation between the trees. As a result of clustering, each cluster contains similar or highly correlated group of trees. We refer to the settings in [28]. In the initial state, M decision trees are generated for classification. Then, through the hierarchical clustering algorithm, the decision tree with the highest accuracy in each cluster is selected to form an improved random forest, in which there are N decision trees. \( M:N = 3:1 \). For example, if there are 100 trees in the IRF, then there are 300 trees in the initial random forest. Table 2 shows the comparison between the RF and DT + IRF algorithms using the dataset in this paper.

| Algorithm | Prediction Accuracy |
|-----------|---------------------|
| RF        | 88.7%               |
| DT + IRF  | 89.0%               |

Table 2. Comparison of prediction accuracy between DT and DT + IRF algorithms.
Table 2. Comparison of prediction accuracy between DT and DT + IRF algorithms.

| Datasets Trees | Physical Shop Floor + Digital Twin Shop Floor | RF   | DT + IRF |
|----------------|---------------------------------------------|------|----------|
|                |                                              | 10   | 61.4%    | 74.9%    |
|                |                                              | 20   | 78.5%    | 84.7%    |
|                |                                              | 30   | 84.0%    | 90.3%    |
|                |                                              | 40   | 84.8%    | 92.9%    |
|                |                                              | 50   | 86.1%    | 94.2%    |
|                |                                              | 60   | 86.9%    | 94.9%    |
|                |                                              | 70   | 87.2%    | 95.7%    |
|                |                                              | 80   | 88.1%    | 96.5%    |
|                |                                              | 90   | 88.4%    | 97.1%    |
|                |                                              | 100  | 88.7%    | 97.8%    |

Next, we used the DT + IRF algorithm for the fault diagnosis analysis of the automobile rear axle assembly line. After data preprocessing, 300 classification trees were generated [40,41]. Figure 7 shows the relationship between the accuracy of random forest and the number of decision trees. It can be seen from Figure 7 that, when the number of decision trees is greater than 100, even if the number of decision trees is greatly increased, the accuracy change is very small. The complexity of constructing a random forest is proportional to the number of decision trees. There are not as many decision trees as possible. With the increase in the number of decision trees, the computational efficiency will continue to decline, and the time spent will continue to increase. When the number of decision trees is 300, the accuracy rate is 90.8%, and then it fluctuates slightly. At the same time, when the number of decision trees reaches a certain level, the interpretability of the random forest will be weakened. Therefore, we set the number of initially generated decision trees to 300, and the number of decision trees to 100 after hierarchical clustering.

![Figure 7. Accuracy of different decision trees in random forest.](image)

According to Section 3.2, the training set was used to train the model to form a random forest through bootstrap sampling, and the accuracy of each base classifier was calculated through the test set. Figure 8 gives the accuracy of each classification tree in the random forest. It can be seen that the classification accuracy of each base classifier is roughly between 65% and 75%, and the accuracy of individual classification trees can reach about 80%. There are also individual classification trees whose accuracy rate is low, and the classification accuracy rate is only about 60%, which is not accurate enough for fault diagnosis.
Therefore, we used the proposed method to cluster the trained classification trees according to the similarity of the classification results through hierarchical clustering, and then selected the batch of classification trees with the highest prediction accuracy from each category to participate in the integration. We set the cluster of clustering to 100 by validation [40,41]. The clustering results for 300 classification trees are shown in Figure 9. It can be seen that there are about 70 base classifiers in the eleventh cluster, which shows that these trees are very close to the results of the test set, and the similarity of these trees is very high. In order to improve the accuracy of the final classification results and the difference of each base classifier, we selected the best-performing tree in each cluster to form a new random forest, and the result of the classification was used as the weight of the weighted voting for online fault diagnosis.

A confusion matrix was used to represent the results of different algorithms in order to explain the experiment better. Figure 10a shows the IRF + DT algorithm proposed in this paper. The model was pretrained through virtual data, and then the decision tree was rescreened through a hierarchical clustering algorithm, the parameters of the model were transferred to the new model through transfer learning, and the new model was trained through a training set of actual data. Finally, the accuracy of the model in the test set of actual data was as high as 97.8%. Figure 10b shows the random forest model that used the

Figure 8. Accuracy of each base classifier.

Figure 9. Clustering result for classification trees.
data from the digital twin shop floor and the physical production line. The accuracy rate was 90.8%. These results show the effectiveness of the proposed IRF + DT algorithm in the automotive rear-axle assembly line. The IRF + DT algorithm more accurately distinguished the fault status of the equipment, which shows that data distribution and insufficient data sampling affect the performance of the IRF algorithm.

![Confusion matrix for fault classification of automobile rear-axle assembly line](image)

**Figure 10.** Confusion matrix for fault classification of automobile rear-axle assembly line: (a) IRF + DT algorithm; (b) RF algorithm.

It can be seen from the confusion matrix that the accuracy of the model can reach 100% in the case of normal state, conveyor belt fault, and four-wheel positioning device fault if there is a large volume of data generated by the digital twin. This is because there is a large amount of normal state data, and the characteristics of the conveyor belt and four-wheel positioning device are relatively simple. However, for the tightening robot, the loading robot and the unloading robot are still misclassified. The model cannot easily distinguish between the robots. This is caused by the similarity of the characteristics between the robots.

5. Discussion

In order to test the fault classification performance of the proposed algorithm, it was compared with several classic algorithms in machine learning, such as KNN, ANN, SVM, and RF. In the algorithm test, the selected training set is shown in the table. However, no matter what the training set is, the test set is the dataset generated by the physical production line. The results are shown in Table 3. Three common single classifier parameters were set as follows: the number of k-nearest neighbors of the KNN model was set to 5, and the Euclidean distance calculation method was adopted. The number of ANN network layers was set to 3. The accuracy of using ANN alone was only 74.6%. Using the transfer learning method to transfer ANN parameters, combined with DT, the final test accuracy rate can reach 92.7%. The number of LSTM network layers was set to 4. The number of LSTM neurons in the first and second layers was 128, and that in the third layer was 64. The number of softmax classifiers in the last layer was 6. We used dropout to prevent the overfitting of models and selected an Adam optimizer. For SVM, we chose a radial basis function (RBF) as the kernel function. The number of decision trees in a random forest was 100. In DT + IRF, the initial number of decision trees was 300. After hierarchical clustering screening, it had 100 decision trees. The results are shown in Table 3. In KNN, ANN, LSTM, SVM, and RF, using only the physical production line training set gives an average accuracy of 79.98%. Just using the data simulated by the digital twin workshop as the training set gives an average accuracy of 72.4%, using both leads to an average accuracy of 83.98%. These results indicate that different data distributions and the amount of data will affect...
the results of the algorithm. It can be seen that the accuracy of the integrated method has been greatly improved compared to the single model, and the proposed DT + IRF method further improves the accuracy of the integrated method, proving its effectiveness.

Table 3. The accuracy of each classifier.

| Datasets Algorithm | Physical Shop Floor | Digital Twin Shop Floor | Physical Shop Floor + Digital Twin Shop Floor |
|--------------------|----------------------|-------------------------|---------------------------------------------|
| KNN                | 81.6%                | 73.7%                   | 85.3%                                       |
| ANN                | 71.1%                | 65.3%                   | 74.6%                                       |
| DT + ANN           |                      |                         |                                             |
| LSTM               | 79.6%                | 70.9%                   | 83.9%                                       |
| SVM                | 83.4%                | 76.8%                   | 87.4%                                       |
| RF                 | 84.2%                | 75.3%                   | 88.7%                                       |
| DT + IRF           |                      |                         | 97.8%                                       |

6. Conclusions

In an actual production process, faults need to be found quickly and accurately, and the classification of faults is of great significance. Therefore, in order to improve the accuracy of fault diagnosis, we have proposed an improved random forest algorithm based on ensemble selection. In addition, in order to solve the problem of data imbalance in the actual production process, the lack of a large amount of fault data, and the algorithm needing to retrain the model when changing the working conditions, we proposed a method that combines digital twins and transfer learning. Finally, combining the above methods, the DT + IRF method was proposed to enhance the fault diagnosis ability and extend the fault diagnosis cycle to the entire product lifecycle. Experiments showed that the method is more flexible, effective, and accurate, and has a certain generalizability. Through the fault twin system established in this paper, faults can be found and located quickly, and solutions can be provided. This could make intelligent manufacturing more effective and sustainable. Our future work will focus on verifying the generalization of our algorithm in other fields. We will use more algorithms to verify the effectiveness of transfer learning combined with digital twins.

Author Contributions: K.G. conceived and drafted the original manuscript. X.W. collected the original data and did the computer analysis. L.L. approved the final version of the paper and worked on funding acquisition. Z.G. and M.Y. built the digital twin scene. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the project “Development and application of key technologies for car intelligent chassis assembly line,” grant number 19511105200.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the Shanghai Key Laboratory of Intelligent Manufacturing and Robotics and all members of the CIMS laboratory for their support.

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

Abbreviations

DT Digital Twin
RF Random Forest
IRF Improved Random Forest
ANN Artificial Neural Network
References

1. Tao, F.; Cheng, Y.; Da, X.L.; Zhang, L.; Li, B.H. CCIoT-CMfg: Cloud Computing and Internet of Things-Based Cloud Manufacturing Service System. *IEEE Trans. Ind. Inform.* 2014, 10, 1435–1442.

2. Kusiak, A. Smart manufacturing must embrace big data. *Nature* 2017, 544, 23–25. [CrossRef]

3. Schluse, M.; Priggemeyer, M.; Atorf, L.; Rossmann, J. Experimentable digital twins—Streamlining simulation-based systems engineering for Industry 4.0. *IEEE Trans. Ind. Inform.* 2018, 14, 1722–1731. [CrossRef]

4. Chen, Z.; Liu, Y.; Liu, S. Mechanical state prediction based on LSTM neural network. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 3876–3881.

5. Bennacer, L.; Amirat, Y.; Chibani, A.; Mellouk, A.; Ciavaglia, L. Self-diagnosis technique for virtual private network combining Bayesian networks and case-based reasoning. *IEEE Trans. Autom. Sci. Eng.* 2015, 12, 354–366. [CrossRef]

6. Islam, M.M.; Kim, J.M. Reliable multiple combined fault diagnosis of bearings using heterogeneous feature models and multiclass support vector Machines. *Reliab. Eng. Syst. Saf.* 2019, 184, 55–66. [CrossRef]

7. Palacios, R.H.C.; Goedtel, A.; Godoy, W.F.; Fabri, J.A. Fault identification in the stator winding of induction motors using PCA with artificial neural networks. *J. Control Autom. Electr. Syst.* 2016, 27, 406–418. [CrossRef]

8. Han, S.; Choi, H.J.; Choi, S.K.; Oh, J.S. Fault diagnosis of planetary gear carrier packs: A class imbalance and multiclass classification problem. *Int. J. Precis. Eng.* 2019, 20, 167–179. [CrossRef]

9. Chen, Y.Q.; Fink, O.; Sansavini, G. Combined fault location and classification for power transmission lines fault diagnosis with integrated feature extraction. *IEEE Trans. Ind. Electron.* 2018, 65, 561–569. [CrossRef]

10. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* 1991, 21, 660–674. [CrossRef]

11. Guo, X.; Chen, L.; Shen, C. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. *Measurement* 2016, 93, 490–502. [CrossRef]

12. Yang, B.S.; Di, X.; Han, T. Random forests classifier for machine fault diagnosis. *J. Mech. Sci. Technol.* 2008, 22, 1716–1725. [CrossRef]

13. Kadri, O.; Mouss, L.H.; Mouss, M.D. Fault diagnosis of rotary kiln using SVM and binary ACO. *J. Mech. Sci. Technol.* 2012, 26, 601–608. [CrossRef]

14. Wán, L.; Gong, K.; Zhang, G.; Yuan, X.; Li, C.; Deng, X. An Efficient Rolling Bearing Fault Diagnosis Method Based on Spark and Improved Random Forest Algorithm. *IEEE Access* 2021, 9, 37866–37882. [CrossRef]

15. Chen, Z.; Li, Z. Research on fault diagnosis method of rotating machinery based on deep learning. In Proceedings of the 2017 Prognostics and System Health Management Conference (PHM-Harbin), Harbin, China, 9–12 July 2017; pp. 1–4.

16. Azadeh, A.; Saberi, M.; Kazem, A.; Ebrahimipour, V.; Nourmohammadzadeh, A.; Saberi, Z. A flexible algorithm for fault diagnosis in a centrifugal pump with corrupted data and noise based on ANN and support vector machine with hyper-parameters optimization. *Appl. Soft Comput.* 2013, 13, 1478–1485. [CrossRef]

17. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wires Data Min. Knowl.* 2018, 8, e1249. [CrossRef]

18. Niu, G.; Lee, S.-S.; Yang, B.-S.; Lee, S.-J. Decision fusion system for fault diagnosis of elevator traction machine. *J. Mech. Sci. Technol.* 2008, 22, 85–95. [CrossRef]

19. Breiman, L. Random forests. *Mach. Learn.* 2001, 45, 5–32. [CrossRef]

20. Fernandez-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *J. Mach. Learn. Res.* 2014, 15, 3133–3181.

21. Zhou, Z.H.; Wu, J.; Tang, W. Ensembling neural networks: Many could be better than all. *Artif. Intell.* 2002, 137, 239–263. [CrossRef]

22. Zhou, Z.H.; Tang, W. Selective Ensemble of Decision Trees. *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 476–483.

23. Lee, K.; Han, S.; Pham, V.H.; Cho, S.; Choi, H.-J.; Lee, J.; Noh, I.; Lee, S.W. Multi-Objective Instance Weighting-Based Deep Transfer Learning Network for Intelligent Fault Diagnosis. *Appl. Sci.* 2021, 11, 2370. [CrossRef]

24. Nathalie, J.; Shaju, S. The class imbalance problem: A systematic study. *Intell. Data Anal.* 2002, 6, 429–449.
25. Kwak, J.; Lee, T.; Kim, C.O. Incremental clustering-based fault detection algorithm for class-imbalanced process data. *IEEE Trans. Semicond. Manuf.* 2015, 28, 318–328. [CrossRef]

26. Sun, Y.; Zhang, H.; Zhao, T.; Zou, Z.; Shen, B.; Yang, L. A New Convolutional Neural Network with Random Forest Method for Hydrogen Sensor Fault Diagnosis. *IEEE Access* 2020, 8, 85421–85430. [CrossRef]

27. Yu, J.; Yan, X. Multiscale intelligent fault detection system based on agglomerative hierarchical clustering using stacked denoising autoencoder with temporal information. *Appl. Soft Comput.* 2020, 95, 106523. [CrossRef]

28. Bharathidason, S.; Venkataesanwar, C.J. Improving classification accuracy based on random forest model with uncorrelated high performing trees. *Int. J. Comput. Appl.* 2014, 101, 26–30. [CrossRef]

29. Grieves, M.; Vickers, J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary Perspectives on Complex Systems*; Springer: Cham, Switzerland, 2017; pp. 85–113.

30. Tao, F.; Qi, Q.; Liu, A.; Kusiak, A. Data-driven smart manufacturing. *J. Manuf. Syst.* 2018, 48, 157–169. [CrossRef]

31. Kendrik, Y.; Hong, L.; Pai, Z.; Chun-Hsien, C. A state-of-the-art survey of digital twin: Techniques, engineering product lifecycle management and business innovation perspectives. *J. Intell. Manuf.* 2020, 31, 1313–1337.

32. Rojek, I.; Mikolajewski, D.; Dostatni, E. Digital Twins in Product Lifecycle for Sustainability in Manufacturing and Maintenance. *Appl. Sci.* 2021, 11, 31. [CrossRef]

33. Piltan, F.; Kim, J.-M. Bearing Anomaly Recognition Using an Intelligent Digital Twin Integrated with Machine Learning. *Appl. Sci.* 2021, 11, 4602. [CrossRef]

34. Resman, M.; Protner, J.; Simic, M.; Herakovic, N. A Five-Step Approach to Planning Data-Driven Digital Twins for Discrete Manufacturing Systems. *Appl. Sci.* 2021, 11, 3639. [CrossRef]

35. Pan, S.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 2010, 22, 1345–1359. [CrossRef]

36. Zhang, R.; Tao, H.; Wu, L.; Guan, Y. Transfer learning with neural networks for bearing fault diagnosis in changing working conditions. *IEEE Access* 2017, 5, 14347–14357. [CrossRef]

37. Huang, H.; Yang, L.; Wang, Y.; Xu, X.; Lu, Y. Digital Twin-driven online anomaly detection for an automation system based on edge intelligence. *J. Manuf. Syst.* 2021, 59, 138–150. [CrossRef]

38. Unity Game Engine. Available online: http://www.unity3d.com (accessed on 20 August 2021).

39. Liu, M.; Lang, R.; Cao, Y. Number of trees in random forests. *Comp. Eng. Appl.* 2015, 51, 126–131.

40. Oshiro, T.M.; Perez, P.S.; Baranauskas, J.A. How Many Trees in a Random Forest? *Lect. Notes Comput. Sci.* 2012, 7376, 154–168.