'National Conference on Advancements in Computer Science and Engineering'
International Journal of Scientific Research in Computer Science,
Engineering and Information Technology | ISSN : 2456-3307 (www.ijsrcseit.com)

Deep Learning-Based Data Analysis for Intelligent Manufacturing Processes Using Deep BiLSTM Techniques

A. Sivakumar[1], R. Kamalakannan[2]
[1]Department of Mechanical Engineering, Anna University, Chennai, Tamil Nadu, India
[2]Department of Mechanical Engineering, M. Kumarasamy College of Engineering, Tamil Nadu, India

ABSTRACT

Automated visual inspection in the semiconductor industry seeks to boost the identification of defects in materials and workmanship by harnessing the benefits of artificial intelligence and data sensing technologies, allowing businesses to earn from higher yields and easier fabrication. High tech analytics-based optimization techniques used in manufacturing operations are equipped with smart production. With the prevalent utilisation industry 4.0 detectors in production processes, the supply for robust data approaches is rising. Employing mass production dataset with artificial intelligence and deep learning can lead to more efficient and intelligent automation. The actual data gleaned that during industrial automation leaps outcomes in a mass data platform. Observing a really huge amount of data in legitimate with such an elevated data rate for inevitable responsibility to fix detection and control during production is incredibly hard. The dimensions are inherently unpredictable and intricate. This obscures existing approaches of trying to locate this hidden information. We present an extensive interpretation of industrial automation using resilient data processing and deep learning techniques in this research. We propose a hybrid integrative framework based on Deep Bidirectional Long Short Term Memory (DeepBiLSTM). In the fully-connected part of DeepBiLSTM, maxout receptors are used to solve the problems of removing and igniting dimensions for attempting to learn about semiconductor manufacturing processes and negotiating with a variety of indications throughout this hybrid learning integrated model developed. During the data preprocessing phase, the recidivism data normalisation heuristic is used to reduce over-fitting. Explore how we developed an intelligent feature selection algorithm for DeepBiLSTM training the model. We hope to provide manufacturers with access to effective forecasting by providing a cutting- edge solution for curtailing production processes and gaining perspective across multiple dimensions.

Keywords— Deep Bidirectional Long Short Term Memory (DeepBiLSTM, Deep learning, Intelligent Manufacturing, Root- Mean-Square-Error (RMSE)

I. INTRODUCTION

The industrial sector has seen enormous progress in the form of massive major paradigm shifts over the last few decades. The production internet of things [1, 2] and machine learning (ML) to enable industrial machinery to self-optimize [3-7] have been decided to embrace by the industrial revolution 4.0. Utilizing computer control

Copyright: © the author(s), publisher and licensee Technoscience Academy. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License, which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited

over production processes can helps to combine mechanisms became more smarter. In broad sense, industry 4.0 is a data-driven solution that enables use of IoT devices and smart monitoring sensors. Mobilizing innovative equipment in production, such as IoT combined with cloud applications, offers access to a wealth data at numerous levels, such as the manufacturing firm, industrial machinery, and production methods. Machine learning enables us to morph massive amounts of production statistics into real production insights. Industrial production can be controlled by cutting-edge machine learning and artificial intelligence (AI), and tasks can be implemented derived from empirical findings to boost productivity whereas reducing expenses.

Manufacturing that is both economical and environment benign has become a highest concern for both industry and academia. In sequence to do either, it is critical to identify aspects that play a vital role in processes and outcomes. Figure 1 depicts a practical framework for product development and data predictive analysis. The system is divided into laminae and can be thought of as a computer-integrated production model in which computer vision can control the entire manufacturing process. During the Business planning phase, all decisions about finished products are made. The operational managerial level manages operational decisions concerning performance enhancement. At the monitoring level, numerous sensor-based monitoring strategies, such as intrusion detection systems, are used. Finally, at the production process and detection levels, it is decided to perform data acquisition and real-time computation.

Figure 1. Numerous sorts of automation and data analytics

The strategy of this work aids in reducing and manufacturing threats while also inspiring long-term development of semiconductor technology. Relocating to integrated functional optimal sizing is a difficult task; unfortunately, integrating machine intelligence into increased automation and semiconductor fabrication may benefit from two very different cost reductions as well as product increased efficiency. Intelligent manufacturing data analysis focuses on product life cycle management, production process strategic planning, manufacturing interoperability, and process design. The Internet of Things (IoT), cloud technology, and big data are recent technological developments that have the potential to accelerate and simplify the manufacturing process, resulting in innovative industry 4.0 [8]-[12]. Such progressions may expedite the development of fabricating designs into automation system transfer speed, aiding self-adaptive, self-sensing, and self-organizing smart properties. To obtain such solutions, dozens of new issues must be resolved, including volume of data, quality of data, and data amalgamation.

Sensory signals are regarded as a single asset in traditional fault diagnosis and diagnostic test systems [13]. These characteristics are then fed into a framework, which is used to validate the implementation status. The challenge with this approach is that it falls short to identify critical characteristics in semiconductor technology, which may result in data sensory loss. Sensory data includes noise, anomalies, and missing data in addition to

heterogeneous structures. To tackle these issues, we propose a smart and innovative algorithm with a feature extraction method. Quality of product forecasting is an extremely unbalanced categorization in the broad sense, but semiconductor production has none of that. To be more specific, the raw data is unstable because the likelihood of manufacturing industry failures during training is quite low. To mitigate this risk, an extremely unbalanced strategy for increasing prediction accuracy must be used. The following sections may show us how and why. In this regard, we developed an integrated DeepBiLSTM-based algorithm to solve an optimization problem while also defining a predictive control solution by extracting the most important variables and using their feature values in classification models. There are two types of metaheuristic optimization algorithms: hazardous release and heuristic optimization. Methods of decomposition and dominance. This research presents a Deep Learning based decomposition method. This technique can be used for any type of production evaluation, such as feature extraction/selection, versatile optimization, and defect diagnosis. We specifically investigate the aforementioned:To begin, how to propose a hybrid model for nonlinearity analysis based on an optimization algorithm combined with DeepBiLSTM; to second, where to integrate DL combined with LSTM functionality to implement a highly versatile and personalised intelligent production landscape; and finally, whether integrating with Deep Learning can outperform existing techniques.

Several categorization algorithms are perform to use the extracted features, and the one with the lowest classification failure rate is chosen. There is also a comparison between the proposed alternative and traditional techniques. In terms of production system accuracy and stability, the integrated strategy outperforms the others. This framework can also be used to identify flaws without requiring specialized skills. During this integration, a few issues arose, such as dealing with data sets, exploration, and enslavement in an optimization method. Contingencies are explored throughout the manuscript to ensure quality and safety.

Prospect and Research Contributions

Deep learning, artificial intelligence, and the Internet of Things, as well as technologically advanced embedded systems, have the potential to be a promising alternative for a productive, cost- effective manufacturing process. Semiconductor manufacturing is a multi- stage, multi-disciplined, intricate, time- consuming, and expensive process. Defective products are the result of manufacturing flaws. As a result, as illustrated in Figure. 1, defining failure causes is critical to successful policy decisions and a daunting task during the business planning stage. It is indeed possible to accomplish this by exploring production stages and attempting to retrieve specific manufacturing features. As an outcome, feature extraction and fault diagnosis appear to be critical. As a consequence, we're interested in incorporating a feature extraction method into electronics manufacturing. The solution entails creating a performance monitoring system based on machine learning and artificial intelligence algorithms in order to improve production process service outcomes by retrieving the most important features. By interpreting these features, we can quickly identify the root cause of a defect (manufacturing processes). One such efficient model contributes to cost reduction and increased productivity.

While feature extraction is the most challenging component of this work, certain data-related concerns, such as unlabeled data and anomalies, must be resolved initially. These data preprocessing stages are poised to transform raw data into actionable and insightful information which can be used to identify patterns from the data and, as an outcome, impose efficient methods. Researchers used a synthesised minority over-sampling algorithm to significantly increase the tiny proportion of faulty cases and assign a huge price to grossly misrepresenting faulty products than standard products to meet the unbalance classification. A symbolic importance level was determined based on these criteria, and anomalies were identified and eliminated. After

that, the initial data set is subjected to a feature selection technique. The goal of feature extraction is to keep important features while reducing large amounts of data. These characteristics are then used to distinguish between patterns.

The dynamic feature selection model proposed here is based on a hybrid algorithm that combines a meta-heuristic and an artificial neural network. We used a binary LSTM to determine the optimal number of features and their associated costs, which was then used to build forecasting models. In a selection method, our primary goal is to find a solution with low-cost values. A deep network, which is an embedded part of a feature selection algorithm, was used to outline the cost function. The optimal solution is divided into several stages, including the selection phase, crossover, mutation, and population formation [14]. A critical component of GA is feature selection, which includes functions such as parent string categorization, reassortment, and mutation. The goal of reproductive development is to select the most expensive chromosomes in the population to produce offspring for the next generation. To address exploration and exploitation while avoiding premature convergence, we proposed a selection scheme that combines various crossover operations. The aforementioned problem is inextricably linked to a lack of diversification. Throughout the screening process, the proposed solution emphasises the importance of cost scalability and updates the evolutionary pressure. We compromise exploration and exploitation by integrating and adapting the probabilities of crossover operators. The sections that follow look at how to evaluate exploratory and exploitative prices. As a result, offspring are generated by distributing such potential outcomes across the selection pool with the help of a hybrid slot machine pick controller. Selected features are fed into a predictive model to determine the failure condition. It is important to note that the method has two competing goals: reducing the number of features and improving the performance of the classifier. As a result, the results of the proposed model are compared to those of previous methods. We discovered that our literary strategy was more efficient and effective than others after performing research. To clarify, the overall goal is to propose whether incorporating additional layers of training into the architecture improves prediction, as well as to investigate a feature selection method based on DeepBiLSTM and an efficient classification algorithm to investigate manufacturing processes.

The following is the research structure: Section II looks into related works. Methodologies and Proposed model are both included in Section III. The experimental research design, data collection methods, and preparation procedures are all described in Section IV. Finally, Section V presents the conclusion of the paper.

II. RELATED WORK

Manufacturing data has recently increased exponentially due to the rapid evolution of high-throughput technologies [15]. Because traditional methods to data processing are practically impossible owing to excessive dimensions, suggesting an efficient and productive data analysis strategic plan has become extremely crucial. To that end, machine learning can aid in the development of strategies for automatically identifying patterns in large datasets. Continuous monitoring of mechanisms that can be associated with various issues, such as noisy signals, is the key to leveraging manufacturing data. Principal component analysis (PCA), linear discriminant analysis (LDA), and canonical correlation analysis (CCA) are useful in dealing with noise and redundant and irrelevant features and should be regarded as a pre - processing phase of fabricating analysis of data, resulting in better insights and rigorous determinations [16]. Previous research on production fault identification has concentrated on extracting and classifying the most relevant features using the aforementioned techniques. The

three types of feature selection methods are filter, wrapper, and embedded methods. The features are selected by the filter methods. Wrapper methods select features based on predictor performance. At last, instead of splitting the data into training and testing sets, embedded methods incorporate variable selection into the training process.

To minimise cost and complexity, the authors of [17] extracted features using PCA. They were employing extraction process features to develop a classification model that determines whether a semiconductor device appears to be defective or normal. To solve the problem, they used the k-nearest neighbours (KNN) algorithm. Cherry et al. [18] developed a multiway PCA-based model for monitoring stream data (MPCA). [19] created a decision tree algorithm to investigate different kinds of faulty gadgets. In [20], a KNN technique was used to estimate feature similarities, and Euclidean distance was considered. By defining similarity metric based on Mahalanobis distance, Verdier et al. improved the performance of a KNN algorithm tailored for fault detection in semiconductor manufacturing [21]. In [22], a support vector machine (SVM) is being used to pinpoint semiconductor screw ups. To address the issue of high dimension, the authors created an analysis based on an RBF kernel. For fault diagnosis, [23] employs an incremental clustering method. To infer a manufacturing process, a Bayesian model has been proposed. The authors looked into the underlying causes of manufacturing problems. Regrettably, it appears that their strategy is heavily reliant on an analyst's knowledge of the specific field. A convolution neural network was proposed by Zheng et al. [24]. They converted multivariate time series data to univariate data. The features were then extracted for the classification process, which was carried out using an MLP-based method. Lee et al.

[25] evaluated the performance of various fault detection models, including feature extraction and classification techniques. They revealed that developing an algorithm based on features that aren't suitable for a particular model can significantly reduce classifier performance. Consider both the feature extraction and classification stages at the same time to improve model performance.

The majority of previous research has concentrated on the use of PCA and KNN algorithms for manufacturing categorization. On the other hand, PCA- based strategies project features into another space predicted by the linear mixture of original features. As a result, in the original feature space, they are uninterpretable [26]. Furthermore, the majority of PCA-related research has concentrated on linear PCA, which is inefficient for investigating non- linear patterns. Although these techniques attempt to account for as much variation as possible among manufacturing variables, incorrect parameter selection, such as principal components, may result in significant data loss. KNN is a memory- based classifier. As a result, in the case of high-dimensional data sets, its performance degrades dramatically with data size. An efficient global search method (e.g., evolutionary computation (EC) techniques) should be considered to address feature selection problems more effectively [27]. Because of their ability to perform global searches, these methods are well-known. Derrac et al. [28] proposed a cooperative co- evolutionary algorithm for feature selection based on a GA. The proposed method solves the problem of feature selection in a single step. However, it should be noted that EC algorithms are stochastic methods that can yield different results when different starting points are used. As a result, instabilities plague the proposed model. Zamalloa et al. [29] used a GA-based method to rank features. As a result, features have been chosen based on the rank orders. The proposed method may result in data loss, which is one potential disadvantage of this work. Furthermore, the relationship between features has not been taken into account in this solution. We proposed an alternative based on a dynamic feature selection method comprised of various modes to provide information on critical variables for fault diagnosis in order to alleviate

the following concerns. We've included ANN in our model to investigate the nonlinear relationship between features. Advanced computing and AI can improve manufacturing efficiency by providing higher levels of intelligence and low-cost sensing [30]. There are two perspectives on the smart manufacturing process. For example, the manufacturing industry has grown to be a significant contributor to the service industry, and the line between cyber and physical systems is becoming increasingly blurred. As a result, architectural approaches such as service- oriented architectures (Cloud manufacturing) can be considered in manufacturing modes and systems. Manufacturing resources can be effectively aggregated and processed/monitored in such distributed and heterogeneous systems based on an efficient serviceoriented manufacturing model. These solutions have the potential to enable large- scale analysis while also increasing productivity. To reveal insights, a successful model requires several steps, such as data cleansing and data transformation. Because data quality affects analysis, using a data preprocessing procedure is critical. The following are the characteristics of this type of debate.

III. METHODOLOGY

3.1. Deep Learning Overview

Deep learning is an artificial intelligence subset that mimics the human brain's data production and pattern generation in order to make sound decisions. It is a subset of ML that enforces ML algorithms using a laminar form of ANN and offers more advanced modelling techniques. Deep neural learning structures are made up of several interconnected layers. It can learn from and convert input data to different levels of abstraction.

3.1.1. Recurrent Neural Network

As a result, RNN can be used to learn features from sequence data. It allows for the storage of data in hidden layers as well as the acquire of previous instances from a few time steps ago. In RNN, a latest update rule is used to calculate the hidden states at different time steps. Using the same activation function, the current hidden state can be calculated as a vector in two parts (e.g., sigmoid or tanh function). The first part is calculated using the input, while the second part is derived from the previous time step's hidden state. The target output can then be calculated using the current hidden state and a softmax function. The learned representation of the input is the hidden state after processing. The hidden state is the learned representation of the input sequential data after processing the entire sequence, and a conventional multilayer perceptron (MLP) is added on top to map the obtained representation to the target.The RNN will now perform the following actions:

- By using the same weights and biases across all layers, RNN reduces the complexity of increasing parameters and memorising previous outputs by feeding each output into the next hidden layer.
- As a result, these three layers can be combined into a single recurrent layer, with all of the hidden layers having the same weights and biases. Equation 1 is a formula for calculating current state.

$ht = f(ht\text{-}1, xt)$ (1)

Where ht denotes the current state t-1 symbolises the previous state and xt indicates the state of input. Equation 2 is used to calculate the activation function application formula (tanh).

$ht = \tanh(Whh\ ht\text{-}1 + WxhXt)$ (2)

Where Whh stands for recurrent neuron weight. The weight at the input neuron is denoted by Wxh. The formula for calculating output is given in Equation 3:

Yt= Why ht (3)

Where yt denotes the output layer, and Why denotes the weight at the output layer.

3.1.2. Long Short Term Memory (LSTM)

Several variants have been developed to address the issue of Vanishing and Exploding Gradients in a Deep Recurrent Neural Network. The Long Short Term Memory Network is one of the most well-known (LSTM). In theory, an LSTM recurrent unit attempts to "recollect" all of the network's foreknowledge while "ignoring" extraneous information. This is achieved by incorporating a number of activation feature layers known as "gates" for a variety of reasons. Each LSTM recurrent unit also helps to preserve an internal cell state vector, which intuitively describes the data that the previous LSTM recurrent unit chose to keep. Figure 2 depicts the operation of the LSTM. A Long Short Term Memory Network is made up of four distinct gates, each representing a distinct goal, as described below:

1. The Forget Gate(f) explicitly states how far back in time previous data should be erased..
2. The Input Gate(i) regulates the amount of data written to the Internal Cell State.
3. The Input Modulation Gate (g): is commonly considered a sub- component of the input gate, and much LSTM literature ignores it, assuming it is contained within the input gate. It modifies the information written to the Internal State Cell by the Input gate by adding nonlinearity and making the information zero-mean. Because zero-mean input converges faster, learning time is reduced. Although this gate's actions are less important than the others and are frequently treated as a finesse-providing concept, including it in the structure of the LSTM unit is good practise.
4. Output Gate(o) specifies which output (next Hidden State) should indeed be manufactured from the current Internal Cell State.

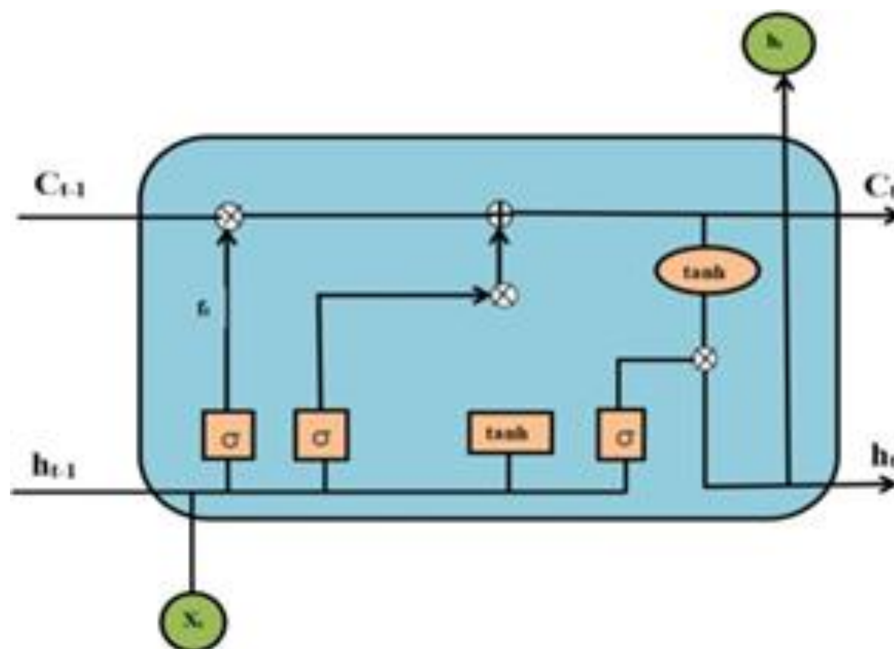

Figure 2. LSTM Operation

Figure 2 shows the operation of LSTM. The function of long-term short- term memory is divided into three stages.

Step 1: Determine the number of time series to be retrieved.

The LSTM's first step is to determine which cell data should be removed during that time step. This is decided by the sigmoid function. The prior state (ht-1) and the current input xt are used to evaluate the function. where ft = forget gate determines which information from the previous time step to delete.

$ft = \sigma (Wf . [ht-1, xt] + bf)$ (4)

Step 2: Determine how much this unit adds value to the current state.

The second layer is divided into two sections. The sigmoid function is one excellent illustration, and the tanh function is another. The sigmoid function determines which values are allowed to pass (0 or 1).

The tanh function designates weightage to the values passed, determining their level of importance (-1 to 1).

$it = \sigma (Wi . [ht-1, xt] + bi)$ (5)

$Ct = \tanh (Wc . [ht-1 , xt] + bc$ (6) it = input gate decides which information to let through based on its significance in the current time step. The input gate analyses the important information with the current input at x(t)

Step 3: Determine how much of the current cell state is output.

The final step is to select an outcome. To begin, a sigmoid layer is used to ascertain which parts of the cell state should be production. After passing through tanh, the cell state is multiplied by the sigmoid gate production to push the attributes among -1 and 1. Ot = output gate empowers data transfer in to impact the results of a time step.

$Ot = \sigma (W0 [ht-1, xt] + b0)$ (7)

$ht = Ot * \tanh (Ct)$ (8)

3.2. PROPOSED SYSTEM

This work makes use of data from a semiconductor production plant, semiconductor manufacturing (SECOM). It is comprised of multiple implementation findings, such as wafer fabrication datasets, and contains 590 features (operation measurements). The desired feature is important for several reasons (Fault and Excellence), alluding to manufacturing's standing, and is embedded as zero and one. Data cleansing is the first step in data analysis, and it tries to address a wide range of data quality problems such as noise, anomalies, lack of consistency, and missing data. As a result of incorrect data gathering, we had to deal with missing data and noise. These can have a deleterious impact on subsequent operations.

To achieve better integration, the DeepBiLSTM neural network pertains min- max scalar to the entire training set. All metrics are reduced to 0-1 whereas the data processing efficiency of the feature proposed method is enhanced, leading to better forecasting. The standardised data values are as follows:

$mr = (xo - xmi) / (xma - xmi)$ (9)

Where, mr is our resultant value and the original cell value is xo , xmi symbolises the column's minimum value , xma is the column's maximum value.

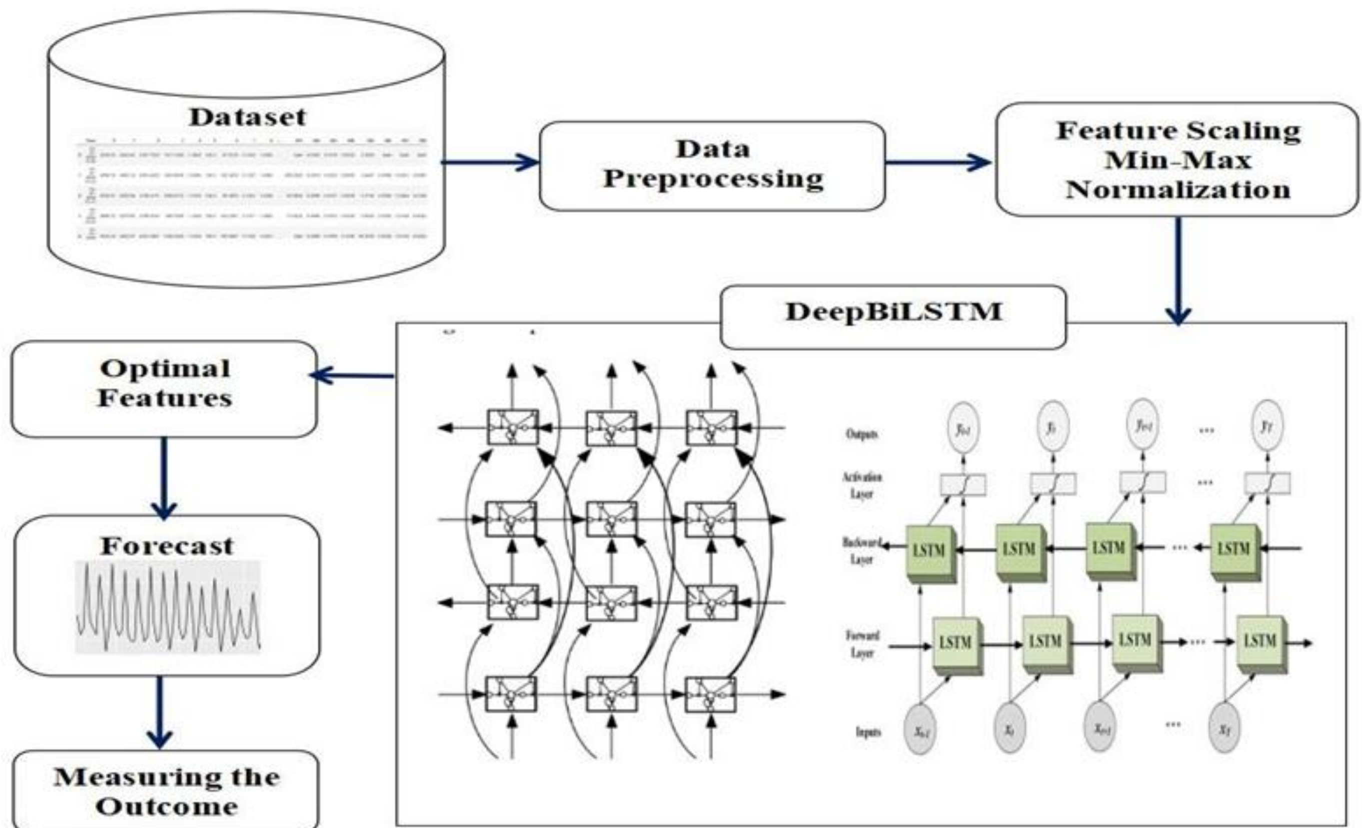

Figure 3. Proposed Evaluation for Smart Manufacturing Process Architecture

Deep-bidirectional LSTMs are an augmentation of the outlined LSTM modeling techniques that hire 2 Long short - term memory throughout pertaining to the training dataset. In round one, the data stream is subjugated to an LSTM (i.e., forward layer). The reverse form of the training data is nourished into the round Model LSTM in the second instance (i.e., backward layer). By using LSTM numerous times greatly enhances huge learning interconnections, which enhances the accuracy and reliability. A Smart Manufacturing Process Architecture Evaluation Proposition is depicted in Figure

The data set was split into 2 parts: training and testing, with 70percent of each data set used for training and 30percent of each data set used to test model accuracy. Deep learning techniques reveal significant "loss" features. Dropping is, in hypothesis, a profit from making incorrect predictions. In other utterances, if the model accurately forecasts the future, the loss cost is zero. The goal is to identify a set of weights and biases that reduce loss values as a result. The Root-Mean-Square-Error (RMSE) is used in this particular instance to evaluate forecasting accuracy in relation to the loss function used by DeepBiLSTM network. The RMSE measures the difference between actual and predicted values. The RMSE formula is as follows:

$RMSE = \sqrt{1} \sum(Yi - Yp)2$
N

Where N signifies the size of data, Yi embodies the observed value, and Yp resembles the predicted value. The major benefit of RMSE is that it stigmatises large errors. Forecast values are scaled in the same units as score values. Moreover, we used the rate of reduction in RMSE as a metric of improvement, which can be calculated.

The overall "feed-forward" Deep Learning Algorithms allow training the working model by only relocating through one way without trying to take any feedback from various data input into consideration. Proposed models, in specific, move from input (left) to output (right) without taking into consideration any input from heretofore trained data. As an outcome, any layer's output has no effect on the layer's training process (i.e., no memory). These neurons can be used to model the data factor distribution and thus the main factor to correlation coefficients data analysis. In other phrases, these networks define clear, which entailed scoping data input to yield data. This type of deep net is frequently used in data analysis. DeepBiLSTM techniques are among the models proposed.

RNNs, on the other hand, recognise portions of the original dataset using an input technique in which training results not only from inputs and outputs (as feed- forward), but also from a network loop to retain a few data and thus features like recollection. In contrast to feedforward Deep networks, response neural networks are dynamic, with their states changing continuously until they reach equilibrium and are thus optimised. State equilibrium is retained until latest inputs disrupt it. A simple RNN's major weakness is that it cannot save and thus does not remember long inputs.

The proposed DeepBiLSTM networks are a type of LSTM network that trains the intended framework from inputs to outputs as well as outputs to inputs. A DeepBiLSTM model may feed a set of input time series to an LSTM model (validation layer), then repeat the training with another LSTM model in reverse order of the input time series. Traditional LSTM models have been shown to outperform DeepBiLSTM designs [3]. The proposed DeepBiLSTM algorithm depicts the algorithms used in the experiment results described in this study. Let us acknowledge the combination of optimization features (LSTM and BiLSTM). When a significant analysis is obtained, the spinning techniques are used to train the designs. The forecasting value is added to the training set and the design is re-trained as a result of estimating the model and comparing its value to the actual value.

The Proposition for the DeepBiLSTM Algorithm

Input: Timescale

Output: Individual [features selected (a binary vector)] as an outcome

Step 1: Import the requisite packages, then access the provided CSV as a dataset and follow the instructions elsewhere on this document.

a. Analyze a few factoids and the sculpt of the dataset.

b. Inspect for missing data. If there are any null values, replace them.

c. Random effect analysis - examine the frequency count of the desired column as well as the distribution of the first few features (sensors). d. Perform multivariate analysis analysis and look for correlations. e. Remove any unnecessary columns.

Step 2: SECOM data preprocessing

Step 3: Data Loading and Preparation

Step 4: Envision the data

Step 5: Depict the evaluation of semiconductor fabrication data.Apply the filter function

Step 6: Prepare the Data.

```
scaler =
MinMaxScaler(feature_range=(0, 1)) scaled_data =
scaler.fit_transform(dataset)
```

Step 7: Separate the reliant column ("Fault and Excellence") from the data set. Similarly, divide the data into two sections: training and testing (70:30 split) And split the dataset into training and testing set (70:30 split)
size ← len(series) * 0.70 train ← series[0...size]
test ← series[size...len(size)]
Step 8: Render the random seed a fixed value.
set random.seed(7)
Step 9: A model must be created and compiled before it can be integrated.
model = Sequential() model.add(TimeDistributed(Dense(
1, activation='sigmoid')))
model.add(Bidirectional(LSTM(...), input_shape=(...)))
model.compile(loss='mean_squared
_error',optimizer='adam', metrics=['acc']) for each i in range(epoch) do model.fit(X, y, epochs=1,
shuffle=False)
model.reset_states()
end for return model
Step 10: Evaluate the Model
loss. acc = model.evaluate(X ,y, verbose=0)
Step 11: Make Predictions with the Model yh=model.predict_classes(X,
verbose=0)

IV. EXPERIMENTAL ANALYSIS

The proposed deep network model is created using data from the SECOM dataset. Since aforementioned, the data set contains nearly 600 features. The scourge of dimensions deals with data sets with a high - dimensional that really can cause serious harm such as overfitting in learning. To tackle these issues, fractal dimension should be diminished, and alternative techniques have been suggested in the literature.
As previously stated, we are dealing with a classifier with numerous variables. Relevant factors for reducing algorithm performance have been widely reported [37]. The use of feature extraction techniques enables the judicious population of features, which aids in the achievement of high trustworthiness throughout. The majority of previous research has focused on feature selection as a single issue, whereas our solution is intrinsically hybrid. This segment contrasted widely disparate strategies, particularly when it comes to traditional feature extraction techniques, with the model discussed in this paper. The goal of this work is to show that intelligent models outperform competing classification techniques. To remove outliers, various scenarios for extracting the features are available. To improve learning accuracy, all strategies were classified as pre-processing tasks. Filtering, wrapping, embedding, and hybriding are the traditional methods [39]. Filter methods can be multivariate or predictor variables. The importance of features was determined using standings methods. Feature subsets are fundamental optimization techniques that select relevant features during the training and testing of a classification model. Embedded methods are executed based on feature dependencies. Finally, the hybrid approach is a progressive integration of other approaches. The models used in this research outcome are then compared.

The first two months' worth of data were split into 70% training data and 30% validation data, with the remaining month's worth serving as test data. Multiple detectors are always present on the freeway; the SECOM data collected from the detectors, as well as the cost of production, are combined to calculate As stated previously, our primary goal is to optimise the production of each iteration (a subset of features) by finding feature set and evaluating possible merits for the input features in order to reduce the measured cost. Figure 4(a) and (b) depicts the Smart Manufacturing Process Evaluation.

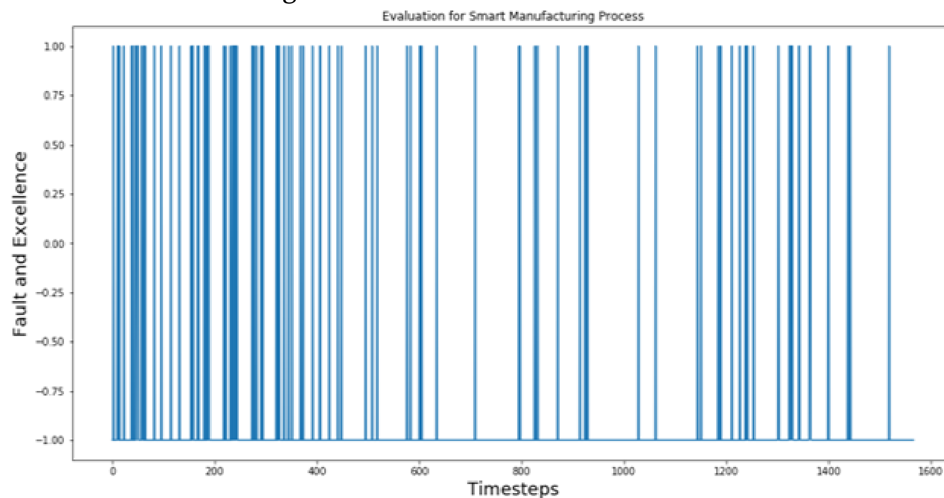

Figure4(a)

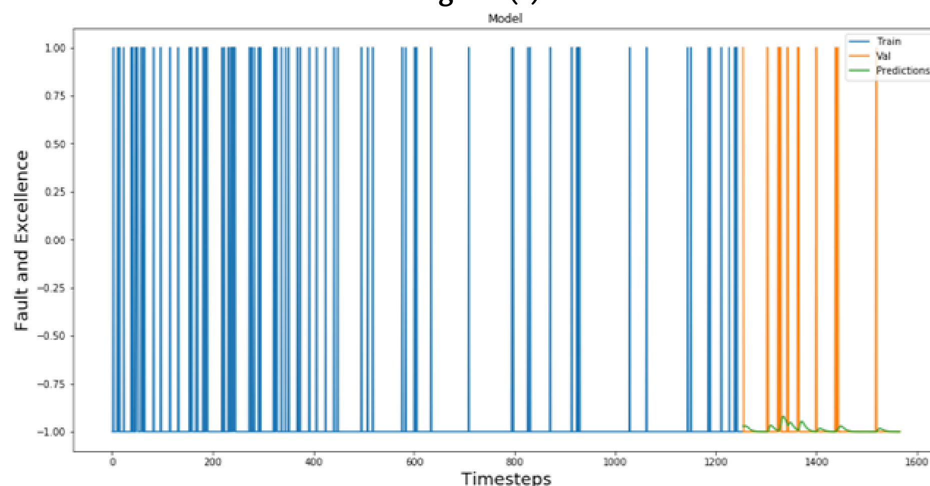

Figure 4(b)

Figure 4 (a) & (b) The Evaluation of Smart Manufacturing Process Evaluation.

At last, we explored various classification techniques and selected the most appropriate one. To accomplish this, a variety of classification approaches, including Gaussian support vector machine, random forest, linear discriminant, k-NN, and SVM with RBF kernel, have been tested. The classification accuracy of the classifiers is used to assess their effectiveness. The ability of each method to correctly predict the correct class is measured and expressed as a percentage. The predictive performance of the classification algorithms under consideration is evaluated using RMSE curves. When the slope nears one, it demonstrates that the classification was done correctly. To prove the efficacy of our proposed solution, we compared the results of this work to previous practises. Furthermore, we would compare the current model to other evolutionary feature selection

algorithms that have been proposed. The model performance improves when the network weights are updated and trained with the number of epochs length.

The prediction accuracy shown in Figure 5 increases with the length of the epochs. Table 1 Shows the RMSE loss of validation and training of the Lasso Regression, ANN & GA, and DeepBiLSTM models at various epochs. The DeepBiLSTM model has a very low RMSE when compared to other models. Lasso Regression and ANN & GA have relatively high RMSE losses, resulting in overfitting and underfitting, which has a significant impact on forecast accuracy. It demonstrates that the RMSE values are exorbitant. It has an impact on the model's prediction accuracy. By employing an elongated LSTM known as DeepBiLSTM, the RMSE values are reduced, and the model prediction efficacy is improved. Table 1 shows that the proposed model achieves 97.5% and 95.2 % training and testing set scores, respectively. Finally, the proposed forecasting methodology is compared to actual and expected output from various models, including Lasso Regression and ANN & GA.

Table 1. Evaluation metrics for comparing various models

Methodologies	**Dimensions**			
	Testing Accuracy	**Training Accuracy**	**RMSE**	**Features**
DeeepBiLSTM	***97.5***	***95.2***	***0.06***	***50***
Lasso Regression	78.8	74.2	0.11	54
ANN and GA	93.7	90.2	0.8	36

The proposed model outperforms the other two models in Figure 5 in terms of production estimation. Under clearly expresses, the proposed DeepBiLSTM model has the best classification ability, and production forecasting can be done accurately and effectively. Intelligent Manufacturing Systems can be developed using the DeepBiLSTM method's ability to predict from SECOM data.

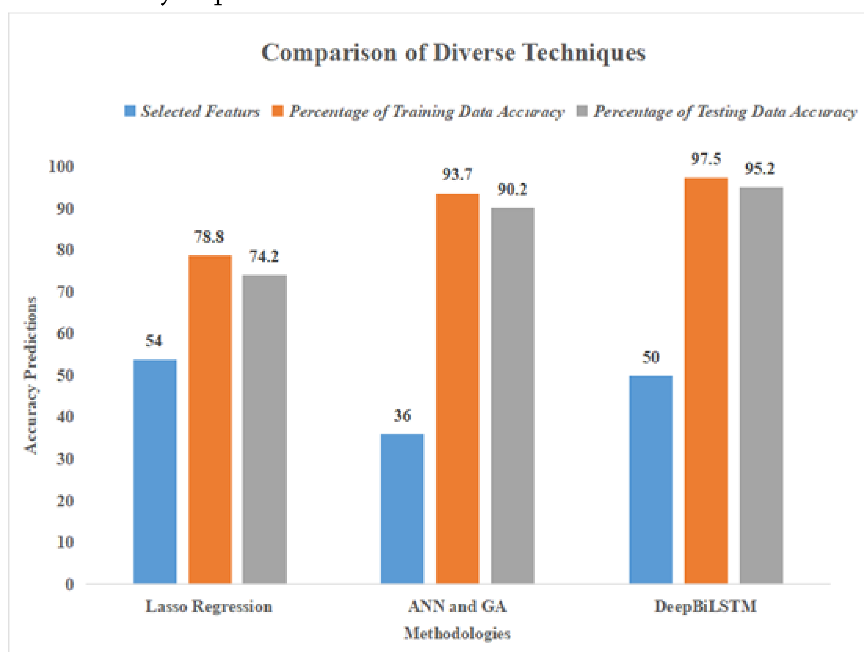

Figure 5 Fabrication Accuracy for Comparing Various Techniques

V. CONCLUSIONS

Manufacturing industries strive to create cost-effective products in order to market successfully. Transitional manufacturing models that use low-cost sensor data can significantly improve industrial intelligence productivity. It aspires to reach a high level of intellectual ability by utilising cutting-edge adequate innovation data processing, predictive analysis, and continuous innovation of Internet access. The terrain of the Industrial Revolution 4.0 requires tactful real-time exposure, observance, and the creation of an effective link between the working population, machinery, and products. The overwhelming majority of their work entails data production analysis using PCA-based methodologies [38]. Those being unable to realise dynamical features in input data and endeavour to extract complex patterns. To overcome this limitation, we proposed an improved feature extraction technique based on hybrid and DeepBiLSTM. We proposed DeepBiLSTM in this paper to model manufacturing time and cost dependencies. According to our findings, DeepBiLSTM is the best network on our datasets, outperforming our previous approach in both testing and training tests. The DeepBiLSTM approach to achieving expressiveness in the talking manufacturing industry has great potential in our opinion.

VI. REFERENCES

[1]. S. Jeschke, C. Brecher, T. Meisen, D. Ozdemir, and T. Eschert, “Industrial internet of things and cyber manufacturing systems,” Industrial Internet of Things, Springer Series in Wireless Technology, Springer, Cham, pp. 3–19, 2017.

[2]. T. Lojka, M. Miskuf M., and I. Zolotova, “Industrial IoT gateway with machine learning for smart manufacturing,” IFIP Advances in Information and Communication Technology, pp. 759– 766, 2017.

[3]. J. Tavcar and I. Horvath, “A Review of the principles of designing smart cyber- physical systems for run-time adaptation: learned lessons and open issues,” IEEE Trans. Systems, Man, and Cybernetics: Systems, vol.49, pp.145–158, 2019.

[4]. B. Chen, D. W. C. Ho, W. Zhang, and L. Yu, “Distributed dimensionality reduction fusion estimation for cyber- physical systems under DoS attacks,” IEEE Trans. Systems, Man, and Cybernetics: Systems, vol. 49, pp.455– 468, 2019.

[5]. P. Palensky, E. Widl, and A. Elsheikh, “Simulating cyber-physical energy systems: challenges, tools and methods,” IEEE Trans. Systems, Man, and Cybernetics: Systems, vol. 44, pp.318– 326, 2014.

[6]. Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, “Review on cyberphysical systems,” IEEE/CAA J. Autom. Sinica, vol.4, pp. 27–40, 2017.

[7]. G. Fortino, F. Messina, D. Rosaci, G. M. Sarne, and C. Savaglio, “A trust-based team formation framework for mobile intelligence in smart factories,” IEEE Trans. Industrial Informatics, 2020.

[8]. M. Ghahramani, M. C. Zhou, and C. T. Hon, “Analysis of mobile phone data under a cloud computing framework,” in Proc. 14th IEEE Int. Conf. Networking, Sensing and Control, Calabria, Italy, 2017.

[9]. M. Ghahramani, M. C. Zhou, and C. T. Hon, “Extracting significant mobile phone interaction patterns based on community structures,” IEEE Trans. Intelligent Transportation Systems, vol.20, pp. 1031–1041, 2019.

[10]. M. Ghahramani, M. C. Zhou, and C. T. Hon, “Mobile phone data analysis: a spatial exploration toward hotspot detection,” IEEE Trans. Automation Science and Engineering, vol.16, pp.351–362, 2019.

[11]. M. H. Ghahramani, M. C. Zhou, and C. T. Hon, “Toward cloud computing QoS architecture: analysis of cloud systems and cloud services,” IEEE/CAA J. Autom. Sinica, vol.4, pp.6–18, 2017.

[12]. D. P. Bertsekas, “Feature-based aggregation and deep reinforcement learning: a survey and some new implementations,” IEEE/CAA J. Autom. Sinica, vol.6, no.1, pp.1–31, Jan. 2019.

[13]. Z. Gao, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault- tolerant techniques, part II: fault diagnosis with knowledge-based and hybrid/active approaches,” IEEE Trans. Industrial Electronics, vol.62, pp.3768– 3774, 2015.

[14]. S. P. Hoseini Alinodehi, S. Moshfe, M. Saber Zaeimian, A. Khoei, and K. Hadidi, “High-speed general purpose genetic algorithm processor,” IEEE Trans. Cybernetics, vol.46, pp.1551– 1565, 2016.

[15]. J. Wan, S. Tang, D. Li, S. Wang, C. Liu, H. Abbas, and A. V. Vasilakos, “A manufacturing big data solution for active preventive maintenance,” IEEE Trans. Industrial Informatics, vol.13, pp.2039–2047, 2017.

[16]. S. M. Meerkov and M. T. Ravichandran, “Combating curse of dimensionality in resilient monitoring systems: conditions for lossless decomposition,” IEEE Trans. Cybernetics, vol.47, pp. 1263–1272, 2017.

[17]. Q. P. He and J. Wang, “Principal component based k-nearest-neighbor rule for semiconductor process fault detection,” in Proc. American Control Conf., Seattle, USA, 2008, DOI: 10.1109/ACC.2008.4586721.

[18]. G. A. Cherry and S. J. Qin, “Principal component based knearestneighbor rule for semiconductor process fault detection,” IEEE Trans. Semiconductor Manufacturing, vol. 19, pp.159–172, 2006.

[19]. S. He, G. Wang, M. Zhang, and D. Cook, “Multivariate process monitoring and fault identification using multiple decision tree classifiers,” Int. J. Production Research, pp.3355–3371, 2013.

[20]. Q. He and J. Wang, “Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes,” IEEE Trans. Semiconductor Manufacturing, vol.20, no.4, pp.345– 354, 2007.

[21]. G. Verdier and A. Ferreira, “Adaptive mahalanobis distance and knearest neighbor rule for fault detection in semiconductor manufacturing,” IEEE Trans. Semiconductor Manufacturing, vol.24, no.1, pp.59–68, 2011.

[22]. R. Baly and H. Hajj, “Wafer classification using support vector machines,” IEEE Trans. Semiconductor Manufacturing, vol. 25, no.3, pp.373– 383, 2012.

[23]. J. Kwak, T. Lee, and C. O. Kim, “An Incremental clustering-based fault detection algorithm for class-imbalanced process data,” IEEE Trans. Semiconductor Manufacturing, vol.28, no.3, pp.318–328, 2015.

[24]. Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. Zhao, “Time series classification using multi-channels deep convolutional neural networks,” in Proc. WAIM, Macau, pp. 298–310, 2014.

[25]. K. Lee, S. Cheon, and C. Kim, “A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes,” IEEE Trans. Semiconductor Manufacturing, vol.30, pp.135–142, May 2017.

[26]. J. Zhang, H. Chen, S. Chen, and X. Hong, “An improved mixture of probabilistic PCA for nonlinear data-driven process monitoring,” IEEE Trans. Cybernetics, vol. 49, 2019.

[27]. B. Xue, M. Zhang, W. N. Browne, and Xin Yao, “A survey on evolutionary computation approaches to feature selection,” IEEE Trans. Evolutionary Computation, vol.20, pp.606–626, 2016

[28]. J. Derrac, S. Garcia, and F. Herrera, “A first study on the use of coevolutionary algorithms for instance and feature selection,” in Proc. Hybrid Artificial Intelligence Systems, Berlin, Germany, 2009.

[29]. M. Zamalloa, G. Bordel, L. J. Rodriguez, and M. Penagarikano, “Feature selection based on genetic algorithms for speaker recognition,” in Proc. IEEE Odyssey Speaker Lang. Recognit. Workshop, USA, 2006, pp. 1– 8.

[30]. L. Dong, S. Chai, B. Zhang, S. K. Nguang, and A. Savvaris, “Stability of a class of multiagent tracking systems with unstable subsystems,”IEEE Trans. Cybernetics, vol.47, pp.2193–2202, 2017.

[31]. S. Wang and X. Yao, “Multiclass imbalance problems: analysis and potential solutions,” IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics), vol.42, pp.1119–1130, 2012.

[32]. H. Liu, M. Zhou and Q. Liu, “An embedded feature selection method for imbalanced data classification,” IEEE/CAA J. Autom. Sinica, vol.6, no.3, pp.703–715, May 2019.

[33]. Q. Kang, X. Chen, S. Li, and M. C. Zhou, “A noise-filtered undersampling scheme for imbalanced classification,” IEEE Trans. Cybernetics, vol.47, no.12, pp.4263–4274, Dec. 2018.

[34]. C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, “DBSMOTE: density based synthetic minority over-sampling technique,” Applied Intelligence, vol.36, pp.1–21, 2011.

[35]. X. Zhang, Y. Zhuang, W. Wang, and W. Pedrycz, “Transfer boostingwith synthetic instances for class imbalanced object recognition,” IEEE Trans. Cybernetics, vol. 48, 2018.

[36]. A. Gupta, Y. Ong, L. Feng, and K. Tan, “Multiobjective multifactorial optimization in evolutionary multitasking,” IEEE Trans. Cybernetics, vol.49, pp.1652–1665, 2017.

[37]. C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, “Joint embedding learning andsparse regression: a framework for unsupervised feature selection,” IEEE Trans. Cybernetics, vol.44, pp.793–804, 2013.

[38]. C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, “A survey on feature selection methods,” Computers & Electrical Engineering, vol.40, no.1, pp.16–28, 2014.

[39]. Mohammadhossein Ghahramani and Yan Qiao and Mengchu Zhou and Adrian O'Hagan and James Sweeney, ‘AI-based modeling and data-driven evaluation for smart manufacturing processes’, IEEE/CAA Journal of Automatica Sinica, Vol.7, pp.1026-1037, 2020.