International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Semiconductor Fault Diagnosis Using Deep Learning-Based Domain Adaption

Krutthika Hirebasur Krishnappa[1], Murigendrayya M. Hiremath[2], Manasa R.[3], Madhura R.[4], Navya Holla K.[5], P. Gomathi[6]

Submitted: 11/10/2023 **Revised**: 10/12/2023 **Accepted**: 21/12/2023

Abstract: In contemporary industries, quality control in semiconductor manufacturing is crucial. Recent years have seen the effective development and use of intelligent data-driven condition-monitoring techniques in industrial applications. The current approaches generally assume that the testing and training data are drawn from the same distribution, despite the condition monitoring performance being quite promising. The acquired data are typically prone to varied distributions in different operating situations in practice due to the difference in the process of manufacturing, which considerably degrades the performance of the data-driven approaches. This research presents a domain adaption approach for fault diagnosis in semiconductor production that is deep learning-based in order to address this problem. The deep neural network's learned high-level data representation is optimized for the maximum mean discrepancy measure. The implemented method appears to offer an efficient and generalized fault diagnosis methodology for quality inspection, according to experimental results using a dataset from a real-world semiconductor manufacturing facility. When compared with existing methods such as VAE-IDF, FDC, SCSDAE, and KABSL, the implemented Deep learning-based domain adaption (DL-DA) achieves 99.56% accuracy in fault diagnosis detection in semiconductors.

Keywords: *Deep learning, domain adaption, fault diagnosis, semiconductor manufacturing.*

1. Introduction

The intermediate stages of accurate quality inspection are becoming increasingly crucial in recent years due to the quick development in technologies of manufacturing semiconductors to ensure efficient process control [1]. The wafer fabrication techniques used in the production of semiconductors are a series of labor-intensive, highly complex, and lengthy products that require numerous equipment, process steps, and recipes [2]. Prior to process control, proper equipment control should be done to ensure the necessary wafer quality in semiconductor manufacture [3]. Ensuring that all standards are followed is essential to minimizing the likelihood of system problems (for instance, the locations of wires for wireless power supplies in the rail) [4]. In the production of semiconductors, Fault Detection, and Classification (FDC) is a crucial component of Advanced Process Control (APC). Monitoring and analyzing process data variations are the primary goal of FDC to spot abnormalities and pinpoint potential underlying causes [5]. The FDC model may perform poorly in terms of generalization for data from a new recipe if it was trained on data from existing recipes. Every time a recipe change takes place, the FDC model must be retrained using the new recipe data. However, this process takes a long time because it takes countless efforts to collect enough labeled training data [6]. For quality enhancement in the semiconductor manufacturing sector, statistical process control (SPC) has received widespread recognition. The general assumption in SPC applications is that product quality can be accurately evaluated using linear to nonlinear models for univariant or multivariate quality criteria [7]. The class imbalance dataset is not the only algorithmic model that has a challenging task when dealing with problems with early defect identification semiconductor devices [8].

Deep learning approaches have been widely used in the process sector to support

[1]*Southern University and A&M College, Baton Rouge, Louisiana 70807*
[2]*Department of Medical electronics Engineering, Dayananda Sagar College of Engineering, Bangalore - 560078, Karnataka, India.*
[3, 4, 5]*Department of Electronics and Communication Engineering, Dayananda Sagar College of Engineering, Bangalore -560078, Karnataka, India.*
[6]*Department of Mathematics, BMS College of Engineering, Bengaluru -560019, Karnataka, India.*

decision-making and knowledge discovery utilizing sensor data [9]. Various sensors that monitor process and tool conditions are also included in modern semiconductor fabrication tools [10]. In general, supervised and unsupervised classifiers in ML for wafer map defect pattern identification (WMDPI) fall into these two groups. When class labels for wafers are accessible, supervised learning models are utilized to categorize the given unknown data into several known classes using the information gained from the previously available training data set. When the class labels are unknown, however, unsupervised learning models are utilized to categorize the wafer data [11]. This study's main goal is to reduce overfitting and time consumption tasks so that they can operate in harmony to improve bad wafer identification [12]. Two methods can be used to recognize patterns on wafer maps: recognition of Model-based patterns and feature extraction-based pattern recognition are: A predetermined function of a probability distribution is utilized for every pattern in the model-based pattern recognition, and the optimal model is chosen by analyzing models using the information criterion. Unique faulty pattern characteristics are first extracted for feature extraction-based pattern identification, and then various pattern categorization methods are used to categorize these patterns. Different techniques, such as the closet neighbor method, correlogram, Radon transformations, etc., can be used to extract the features [13]. When a fault is diagnosed, erroneous sensor signal patterns are seen at the excursion time that the model has predicted, and these patterns match the results of manual identification performed by subject-matter experts [14]. The following list summarizes this work's main contributions,

1. For decision-making, information from many sources (controllers and sensors) is used.
2. Framework of 'Data modeling' is created for:

a. Segmenting the data gathered from various sources according to the operational circumstances. For example, various steps in the process of etching have various properties, hence a set of criteria is needed to properly segregate the data from various steps.
b. Data synchronism between many sources is done to account for network delay, inconsistent time stamps, various sample rates, etc.
c. Data fusion done effectively, where 1. The analytical procedure requires less computation, 2. Different sensors no longer depend on one another, and 3. The model can be trained using fewer samples.

3. A real-world dataset collected from a semiconductor etching process that has induced a domain shift is used to test the implemented methodology.

The remainder of this paper is outlined in the section that follows, the literature review is described in Section 2, the implemented method of fault diagnosis is presented in Section 3, the result and comparative analysis are described in Section 4, and the paper concludes with the conclusion in Section 5.0

Literature Survey

Moslem Azamfar et al [15] implemented a deep learning-domain adaption method for fault diagnosis in semiconductor manufacturing. The classification of health conditions and feature extraction were done using deep convolutional neural networks. The implemented domain adaption method increases the model generalization by transferring knowledge from the training and testing domain, and it reduces overfitting. However, the availability of high-quality labeled data in the source domain remains a crucial requirement for effective domain adaption and time consumption.

Youngju Kim et al. [16] implemented an FD model that was resistant to drift by using a variational autoencoder (VAE) to simulate process drift. The implemented model encrypts certain time-varying information across distinct hidden layers as process drift was characterized by time-varying information. In a manufacturing context where different process drifts occurred, the FD model increased prediction performance in terms of accuracy and decreased false alarms. However, the performance of defect detection is negatively impacted by VAE's inability to adjust to process drifts in semiconductor fabrication, leading to substantial differences in data distribution.

Dong Hwan Kim & Sang Jeen Hong [17] implemented Fault Detection Classification (or control) (FDC), a new multifunction integrated algorithm by the ensemble algorithm. To determine the origin of the abnormality in the process parameters, classification, and fault detection were examined using the information of plasma state and optical emission spectroscopy (OES) data. Batch normalization was performed prior to the activation function because it had the advantages of preventing overfitting, resolving gradient vanishing issues, and accelerating learning. However, there is a restriction

to the implemented FDC that was it did not accurately depict the process condition.

Jianbo Yu et al. [18] implemented a stacked convolutional sparse denoising auto-encoder (SCSDAE) was a novel feature learning method to recognize wafer map patterns (WMPR) in processes of semiconductor manufacturing, which the features directly derived from images. SCSDAE achieves a high accuracy rate, produces greater results on the simulation, and exhibits extremely good performance for WMPR. However, insufficient or biased training results in inferior performance or generalization problems, although SCSDAE required big labeled data for optimal performance.

Tobias Schlosser et al. [19] implemented an SH-DN-based hybrid multistage system as a denoise autoencoder (DAE)-based anomaly detection technique. The DAE learns a primary representation of typical wafers from equipment sensor inputs and uses that information as the basis for a one-class classification model. Higher accuracy in fault detection was achieved by the SH-DN method because of their capacity to learn intricate patterns and features from large datasets. However, denoising the autoencoder of SH-DN required large, diverse labeled datasets, but manufacturing of semiconductors faces difficulties in defect rates, data collection, and privacy concerns.

Yi Zhang et al. [20] implemented sequential oversampling discrimination (SOSD) to address the imbalance issue of fault detection for batch processes essentially time-varying dynamics and with strong nonlinearity. The technique of padding and masking outperformed the better last-point copying technique in performance by effectively handling batch durations with shorter sequences. However, the sequential oversampling method caused a delay in the detection of faults, causing perhaps compromising product quality, prolonged exposure, and increased manufacturing costs.

Junliang Wang et al. [21] implemented a knowledge augmented broad system (KABLS), which offers a selective sampling network of multichannel to decouple the mixed-type faults, with a broad selective sampling module, and knowledge module. The benefit of KABLS appears from the result, which shows that it still has the highest detection accuracy of mixed-type defects and improved classification accuracy. However, the KABLS was difficult to structure complicated domain-specific knowledge, resulting in a knowledge representation that was insufficient or incomplete.

There are some limitations in semiconductor fault diagnosis using deep learning, that are mentioned above such as due to insufficient training data, autoencoders were overfitted, which causes poor generalization and performance degradation on unobserved data in industrial processes, insufficient or biased training results in inferior performance or generalization problems, although SCSDAE required big labeled data for optimal performance.

3. Methodology

The implemented method consists of main five steps, which are data partition, data modeling, designing a CNN architecture, training the implemented CNN model with weight-decay, and testing the model. Overcoming the limitations of insufficient training data leads to overfitting in autoencoders, resulting in poor generalization and performance degradation in industrial processes, so the CNN architecture was implemented. The implemented method of fault diagnosis flowchart is represented in Figure 1.

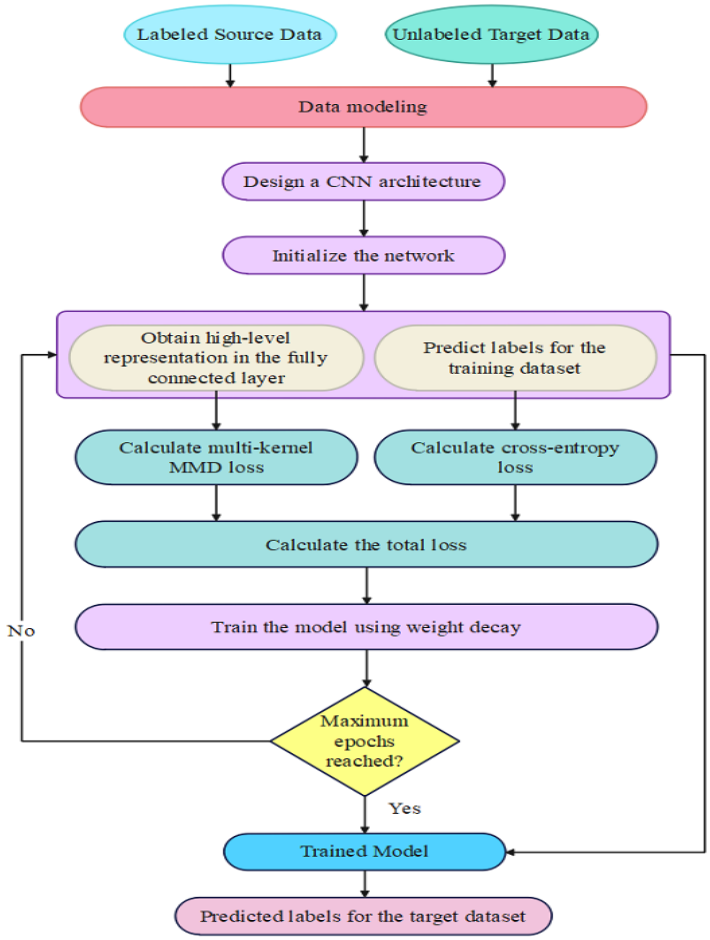

Fig 1. The implemented method of fault diagnosis flowchart

3.1. Data partitioning

The initial stage of the method involves splitting the raw time-domain sensor data into two steps are, unlabeled data (target domain data) and labeled data (source domain data). Additionally, training and testing are the two sets divided from the target dataset, the CNN model is trained using one subset, and the trained model is tested using the other set.

3.2. Data modeling

Before training the diagnosis model, there are three main processes for modeling the data, as follows:

1. Regime separation: Because of changes in operational circumstances during the etching process, the properties of the signal's measurement may significantly change. Therefore, before training the fault diagnosis model, it is crucial to carry out a process of regime separation and data reconstruction.
2. Signal alignment: The length of the signal measurement from the process of etching on a wafer may differ from the length of the signal measurement from other wafers. To adjust every sensor across various experiments and wafers, a signal alignment is necessary. For signal alignment in this study, the Cross-Correlation method has been applied.

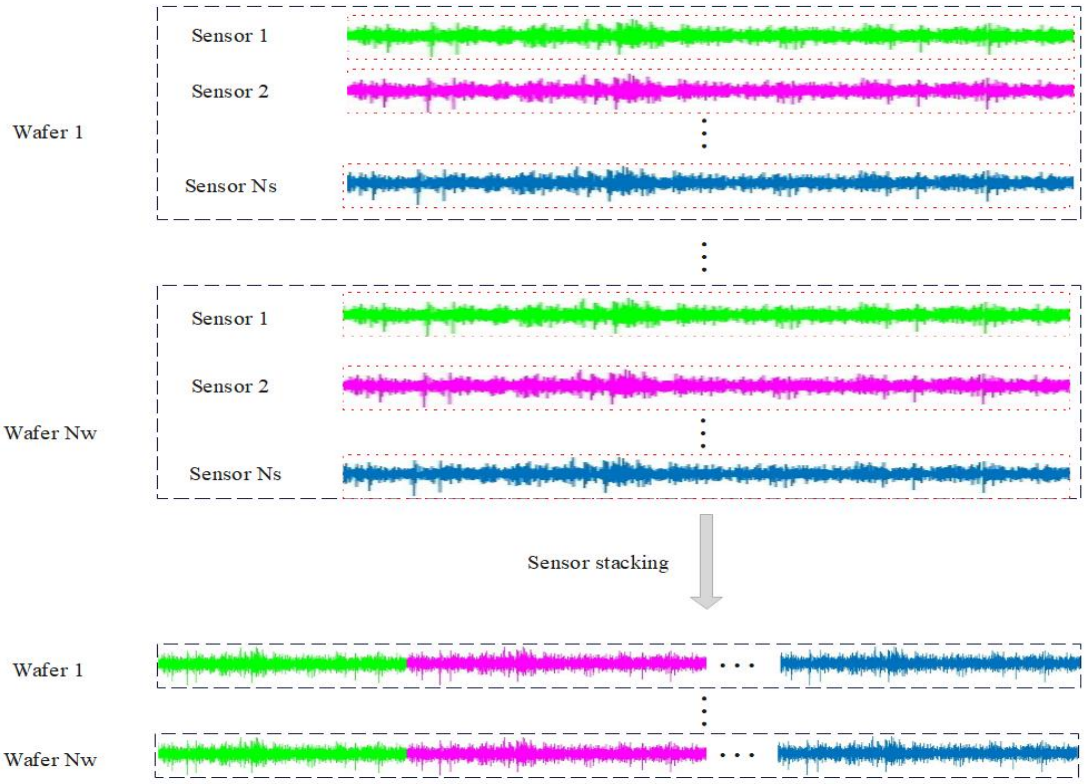

Fig 2. Generating 1D data arrays for each wafer and from all sensors.

3. Sensor Fusion: The implementation of a multi-channel architecture of 1D CNN is possible, but it requires more samples for training and is statistically expensive to determine the dependence and relationship between the various channels. Thus, a basic stacking procedure is implemented to enable the usage of a 1D CNN architecture for sensor fusion with just one channel. In this procedure, the influence of a contact point between two successive sensor data as well as the impact of sensor order are minimized by using the proper stride and padding settings. Figure 2 shows the stacking procedure across numerous wafers.

3.3. Constructing a CNN model

In this stage, the source and target domain data are used to build an architecture of CNN to remove the representation high-level feature. Figure 3 depicts the implemented CNN network's organizational structure for wafer failure diagnosis. Four layered 1D convolutional layers make up the feature extraction architecture. The batch size, filter number, filter length, and kernel size employed in this study are actually estimated to be 32,15, 5, and 15 accordingly.

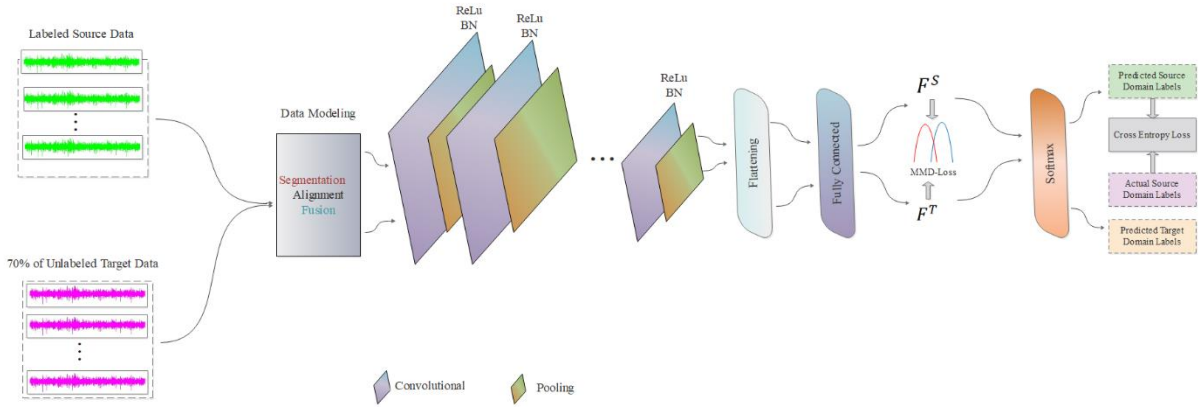

Fig 3. The network architecture of implemented deep neural network in the phase of training

In the FC layer, a function dropout with a rate of 0.5 was applied to avoid overfitting. Every convolutional layer is followed by a max-pooling layer, which reduces the data dimension while retaining important spatial information. To predict each wafer's health state, a softmax layer is used in the final stage. Rectified linear unit (ReLU) activation functions were used in this deep architectural design to address the issues caused by disappearing gradients during training. First, attention is given to the convolutional deep learning pattern, which aims to minimize the error of empirical classification on the supervised source-domain data. The popular cross-entropy loss function L_c is utilized in this study and is described as Equation (1),

$$L_c = -\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{j=1}^{2} 1\{y_i = j\}\log\frac{e^{x_i^{S,j}}}{\sum_{m=1}^{2} e^{x_i^{S,m}}} \quad (1)$$

Where the training samples number represents n_s, $y_i$is the label of the i-th source sample, taking as input the i-th labeled sample in the source domain, and $e^{x_i^{S,j}}$ is the j-th element of the output vector.

Additionally, the domain distribution discrepancy loss L_d, is indicated in this study using the MMD metric and is defined in Equation (2),

$$L_d = MMD_k(F^S, F^T) \quad (2)$$

Where the distributions in the fully-connected layer of the high-level representations of the source data and the target domain denoted F^S and F^T , respectively.

As a result of integrating the objectives in Equations (1) and (2), and problem in general optimization can be written in Equation (3),

$$\min L_{opt} = L_c + L_d \quad (3)$$

Each epoch during training can include updating the network parameters as follows in Equation (4),

$$\theta \leftarrow \theta - \delta\left(\frac{\partial L_c}{\partial \theta} + \alpha\frac{\partial L_d}{\partial \theta}\right) \quad (4)$$

Where ∂ represents the learning rate, and θ represents the network models parameters.

3.4. Train the model using weight-decay

Over-fitting and time consumption is an essential problem with CNN's classification methods. The implemented CNN architecture was therefore trained using weight-decay to minimize these issues. Weight decay can have an impact on the speed the network converges and how many training iterations are needed for it to perform at an adequate level. Weight decay may help with convergence by promoting smaller weights, which would prevent the network from overfitting. This might perhaps result in fewer training iterations, which would speed up training. However, in comparison to other elements like the network architecture, dataset size, and available computational resources, the impact of weight decay on training time is typically negligible. Since the strength of gradients varies exponentially over layers in a deep model, this is not immediately applicable to the current situation. In order to take into account, the relative importance of the local parameters inside the layer, normalize the gradient norm $|g_j^t|$ to have mean 0 and standard deviation (std) 1 within each layer at each iterationtion. The normalized gradient-norm $g_j^{\sim t}$ is given by Equations (5) and (6),

$$g_j^t = \frac{\partial \rho(w^t)}{\partial w_j} \quad (5)$$

ρ denoted data fidelity with respect to the parameter w_j at iteration t.

$$g_j^{\sim t} = \frac{|g_j^t| - \mu_l^t}{\sigma_l^t} \quad (6)$$

Where σ_l^t represents the deviation of standard and mean of all the gradient norms for the parameters within layer l at iteration t, and l represents the layer that includes the parameters μ_l^t and w_j. The following data-driven regularity results from the assumption that the degree of regularity θ_j^t for each parameter w_j at iteration t follows a distribution of the residual in Equation (7),

$$\theta_j^t \propto g_j^{\sim t} \quad (7)$$

Where each parameter's degree of regularization is inversely proportional to the gradient's norm. $S(x; \propto) = 2/1 + \exp(-\propto x)$, Where $\propto \in \mathbb{R}$ is a control parameter for the steepness of the function value transition, is the scaled sigmoid function that we utilize to determine the adaptive regularization. The scaled sigmoid function S of the normalized gradient norm $g_j^{\sim t}$ then determines the relative level

of regularization θ_j^t for each parameter w_j at iteration t as follows in Equation (8):

$$\theta_j^t = S\left(g_j^{\sim t}; \propto\right) = \frac{2}{1+\exp\left(-\propto g_j^{\sim t}\right)} \quad (8)$$

Where according to the gradient norm, the slope of the decay rate transition denotes $\propto$, and $g_j^{\sim t}$ is normalized to have standard deviation 1 and mean 0 and since θ_j^t ranges from 0 to 2 and its average is 1.

3.5. Testing the model

The trained CNN model is given the partial target domain data at this stage for model testing, and the fault detect diagnostic model's performance is analyzed to that of competing approaches.

4. Experimental Setup and Results

In this experiment, the PC used for all trials has a Core i5 processor, 16 GB of RAM, and an NIVIDIA GeFoce TX 2080 Ti graphics card. The model training time is shortened by using GPU computing Tensorflow programming. The implemented DL-DA (Deep-Learning-based Domain Adaption) method is compared with the RNN (Recurrent Neural Network), CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory), and GAN (Generative Adversarial Network), the effectiveness of the implemented method is examined here in terms of accuracy, precision, sensitivity, specificity, recall, and f-measure.

4.1. Evaluation parameters

For the assessment of fault diagnosis, the following three factors are utilized, as follows in Equations (9), (10), (11), (12), (13), and (14),

- Accuracy = It is the proportion of accurate predictions to all input samples and it is calculated using the below equation,

$$\text{Accuracy} = \frac{TP+TN}{(P+N)} \times 100 \quad (9)$$

- Specificity = It is the proportion of actual negatives to all other negative effects.

$$\text{Specificity} = \frac{TN}{(FP+TN)} \quad (10)$$

- Sensitivity = The measure of a model's sensitivity measures its ability to it predicts true positives in each of the categories that are accessible.

$$\text{Sensitivity} = \frac{TP}{(TP+FN)} \quad (11)$$

- Precision: Positive prediction's accuracy is gauged by a static called precision. It is equated to the total number of accurate forecasts divided by the sum of accurate predictions and false positive predictions.

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad (12)$$

- Recall: The wide range of positive predictions is measured by the recall. It is equated to the number of true positive predictions divided by the amount of false negative predictions plus true positive predictions.

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (13)$$

- F-measure: It is a single metric that captures both features by combining precision and recall.

$$\text{F-measure} = \frac{(2*\text{Precision}*\text{Recall})}{(\text{Precisioon}+\text{Recall})} \quad (14)$$

where the number of fault instance denote P, the number of correctly predicted fault instances denote TP, the incorrectly predicted fault samples denote FP, the number of normal instances denote N, and the number of correctly predicted normal instances denote FN.

4.2. Result

Table 1. Performance of the implemented method's accuracy

Methods	Accuracy (%)
LSTM	89.21
CNN	91.04
RNN	90.82
GAN	95.67
DL-DA	**99.56**

The implemented DL-DA method is compared to the RNN, CNN, LSTM, and GAN in term of accuracy. Performance of the implemented method's accuracy shown in Table 1. The graphical representation of the performance of the implemented method's accuracy is shown in Figure 4. In comparison with the other methods of LSTM, CNN, RNN, and GAN with values of 89.21%, 91.04%, 90.82%, and 95.67%, the accuracy obtained reveal that the implemented DL-DA method attains greatest value with 99.56 %.

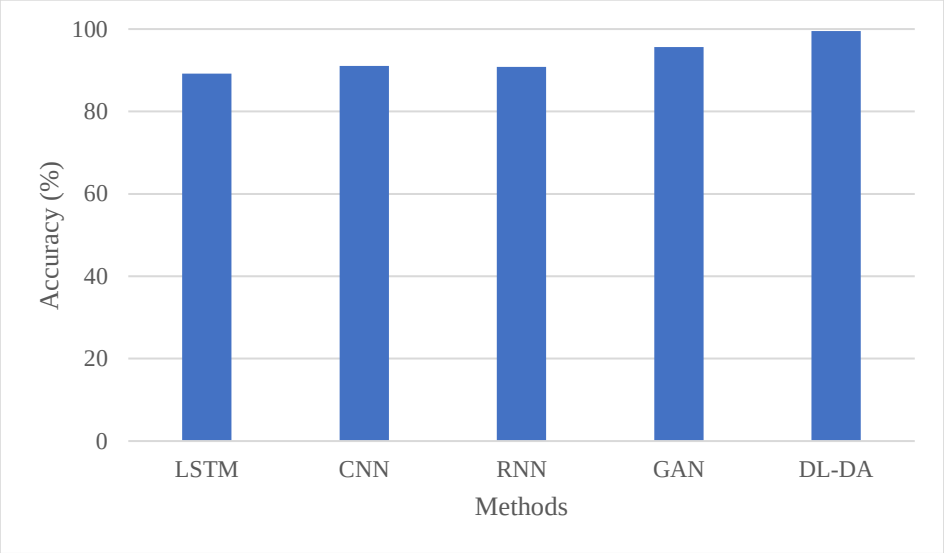

Fig 4. The graphical representation of the performance of the implemented method's accuracy

Table 2. Performance of the implemented method's precision

Methods	Precision (%)
LSTM	90.12
CNN	87.60
RNN	94.43
GAN	91.74
DL-DA	**98.73**

The implemented DL-DA method is compared to the RNN, CNN, LSTM, and GAN in term of precision. Performance of the implemented method's precision shown in Table 2. The graphical representation of the performance of the implemented method's precision is shown in Figure 5. In comparison with the other methods of LSTM, CNN, RNN, and GAN with values of 90.12%, 87.60%, 94.43%, and 91.74%, the precision obtained reveal that the implemented DL-DA method attains greatest value with 98.73 %.

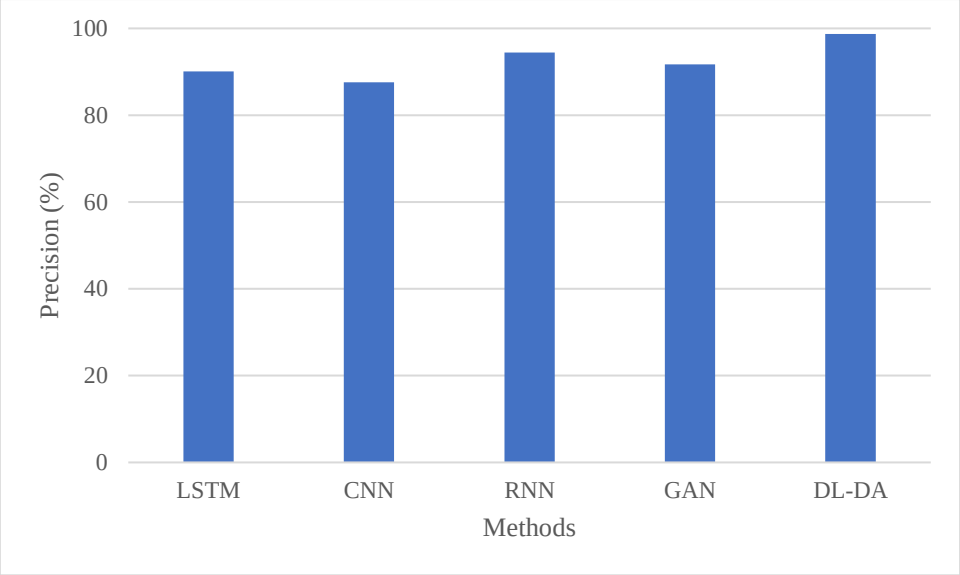

Fig 5. The graphical representation of the performance of the implemented method's precision

Table 3. Performance of the implemented method's sensitivity

Methods	Sensitivity (%)
LSTM	92.29
CNN	91.00
RNN	90.47
GAN	90.85
DL-DA	**97.31**

The implemented DL-DA method is compared to the RNN, CNN, LSTM, and GAN in term of Sensitivity. Performance of the implemented method's sensitivity shown in Table 3. The graphical representation of the performance of the implemented method's sensitivity is shown in Figure 6. In comparison with the other methods of LSTM, CNN, RNN, and GAN with values of 92.29%, 91.00%, 90.47%, and 90.85%, the sensitivity obtained reveal that the implemented DL-DA method attains greatest value with 97.31 %.

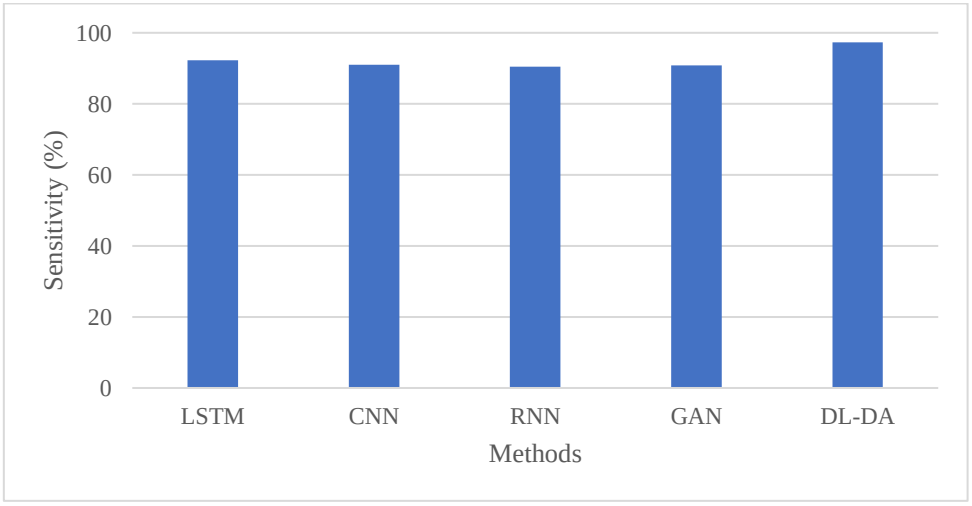

Fig 6. The graphical representation of the performance of the implemented method's sensitivity

Table 4. Performance of the implemented method's specificity

Methods	Specificity (%)
LSTM	91.64
CNN	90.82
RNN	92.46
GAN	88.72
DL-DA	**95.35**

The implemented DL-DA method is compared to the RNN, CNN, LSTM, and GAN in term of specificity. Performance of the implemented method's specificity shown in Table 4. The graphical representation of the performance of the implemented method's specificity is shown in Figure 7. In comparison with the other methods of LSTM, CNN, RNN, and GAN with values of 91.64%, 90.82%, 92.46%, and 88.72%, the specificity obtained reveal that the implemented DL-DA method attains greatest value with 95.35 %.

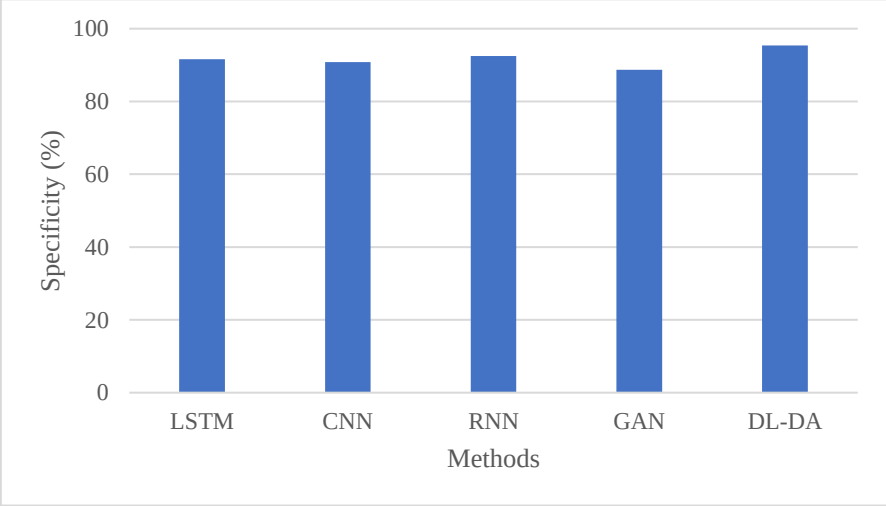

Fig 7. The graphical representation of the performance of the implemented method's specificity

Table 5. Performance of the implemented method's recall

Methods	Recall (%)
LSTM	92.03
CNN	95.69
RNN	96.41
GAN	93.85
DL-DA	**99.05**

The implemented DL-DA method is compared to the RNN, CNN, LSTM, and GAN in term of recall. Performance of the implemented method's recall shown in Table 5. The graphical representation of the performance of the implemented method's recall is shown in Figure 8. In comparison with the other methods of LSTM, CNN, RNN, and GAN with values of 92.30%, 95.69%, 96.41%, and 93.85%, the recall obtained reveal that the implemented DL-DA method attains greatest value with 99.05 %.

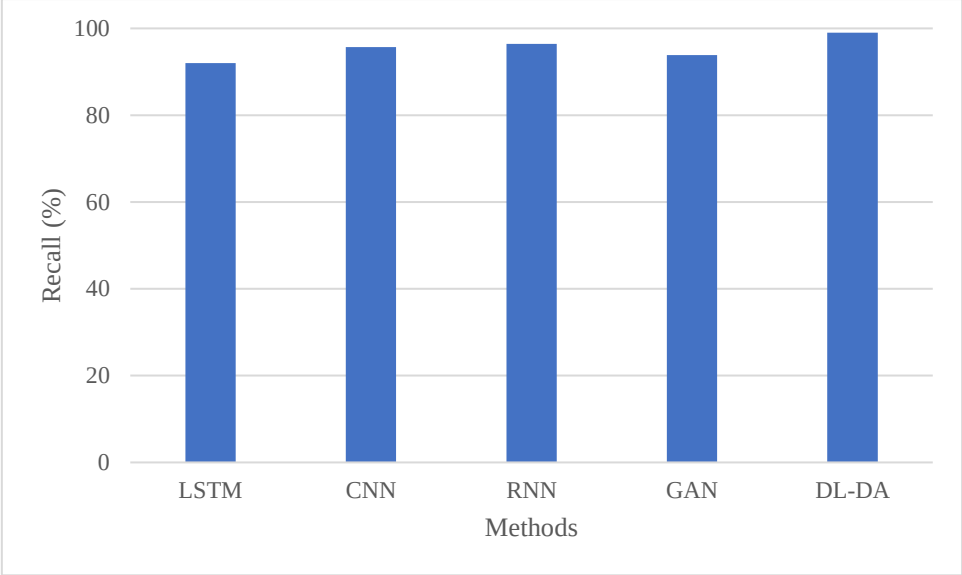

Fig 8. The graphical representation of the performance of the implemented method's recall

Table 6. Performance of the implemented method's f-measure

Methods	F-measure (%)
LSTM	94.18
CNN	89.36
RNN	97.54
GAN	93.72
DL-DA	**98.90**

The implemented DL-DA method is compared to the RNN, CNN, LSTM, and GAN in term of f-measure. Performance of the implemented method's f-measure shown in Table 6. The graphical representation of the performance of the implemented method's f-measure is shown in Figure 9. In comparison with the other methods of LSTM, CNN, RNN, and GAN with values of 94.18%, 89.36%, 97.54%, and 93.72%, the f-measure obtained reveal that the implemented DL-DA method attains greatest value with 98.90 %.

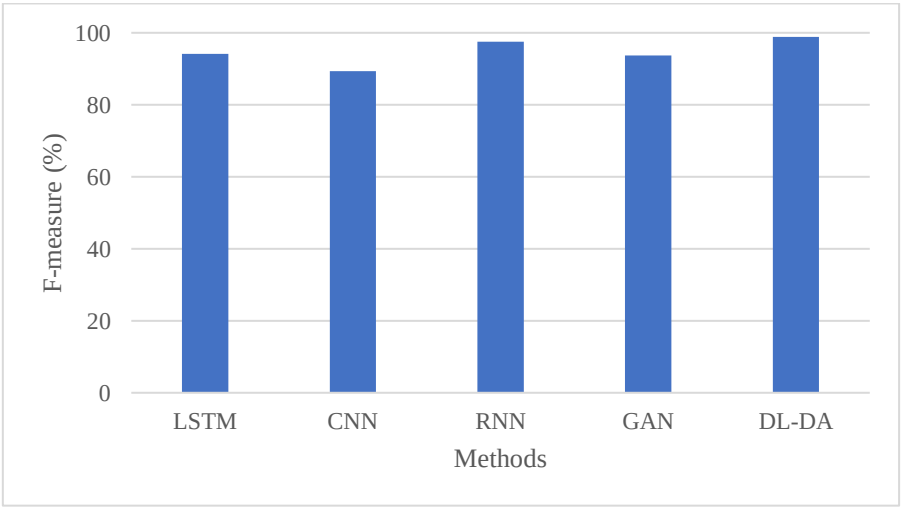

Fig 9. The graphical representation of the performance of the implemented method's f-measure

4.3. Comparative analysis

Table 7. The comparative analysis of existing and implemented model

Author	Method	Accuracy (%)
Youngju Kim et al. [3]	VAE-IDF	96.9
Dong Hwan Kim & Sang Jeen Hong [4]	FDC	93.6
Jianbo Yu et al. [5]	SCSDAE	94.81
Junliang Wang et al. [16]	KABSL	97.91
Implemented method	DL-DA	99.56

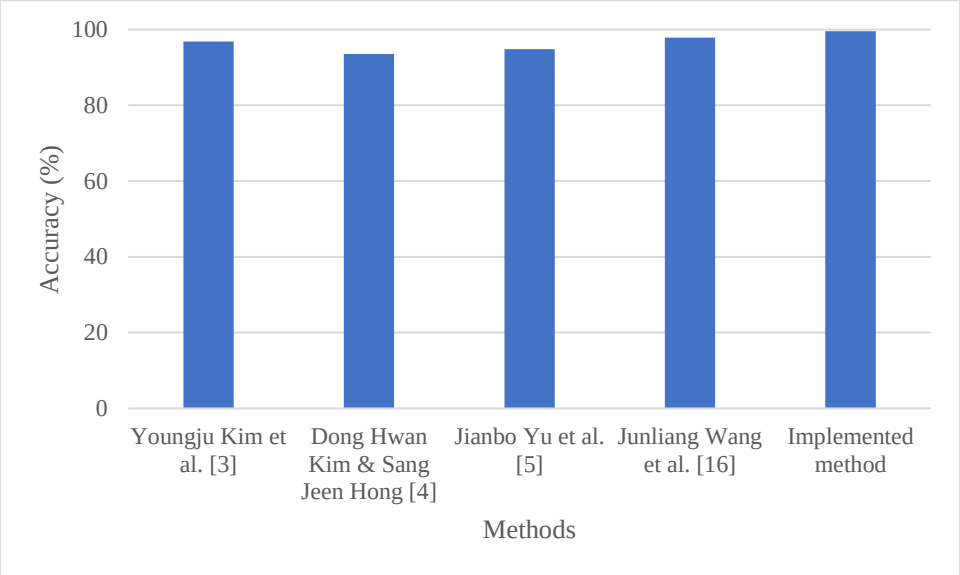

Fig 10. Graphical representation of The comparative analysis of existing and implemented model

In this section, the proposed model's performance has been tested by using the parameter with accuracy. The existing and implemented models' comparative analysis is shown in Table 7. The model is proposed to compare the current approaches that are available in the field in this comparison study. According to accuracy, a comparison study is displayed in Table 7 and the graphical representation in Figure 10 that the implemented method outperformed all other models, including VAE-IDF, SCSDAE, FDC, and KABSL. Due to weight-decay reduce the time consumption, the current model Deep learning-based domain adaption obtains the highest accuracy. Figure 10 shows that when the Deep learning-based domain adaption is the implemented model to be compared with the prior model, the proposed performance model shows high accuracy because of the implemented CNN architecture was therefore trained using weight-decay to minimize the issues, and it gives better results of the effective detection of fault semiconductors.

4.4. Discussion

This section provides the Semiconductor Fault Diagnosis using deep learning and compared those results with existing methods in comparative analysis section 4.1. The major goal of this study is to detect fault diagnosis of semiconductors and to reduce the time consumption with weigh-decay. Information from numerous sources (controllers and sensors) is used to make decisions. Segmenting the information obtained from multiple sources based on the circumstances of use. To appropriately separate the data from distinct processes, a set of criteria is required to consider, for instance, the etching process's many steps have different features. To evaluate the adopted methodology, a dataset in real-world obtained from a domain-shifted semiconductor etching procedure is used. The

representation of high-level data that the deep neural networks learn is optimized for the maximum mean discrepancy metric. Experimental results using a dataset from an actual semiconductor manufacturing facility present that the implemented method appears to provide an effective and generalized defect diagnostic tool for quality inspection. When compared with existing methods VAE-IDF [3], FDC [4], SCSDAE [5], and KABSL [16], the implemented Deep learning-based domain adaption achieves 99.56% accuracy fault diagnosis detection in semiconductor.

5. Conclusion

To diagnosis throughout the semiconductor manufacturing process, this research implements a method of deep learning-based domain adaption. For feature extraction and classification of health conditions, deep convolutional neural networks are utilized. For optimising the data distributions of various tests, the maximum mean discrepancy measure is used. To demonstrate the usefulness and excellence of the implemented method, research is conducted on a dataset of real-world for monitoring health condition of semiconductor. The findings imply that the implemented method gives a new and suitable means of boosting the generalisation capacity of the fault diagnosis model under various conditions. When compared with existing methods VAE-IDF, FDC, SCSDAE, and KABSL, the implemented Deep learning-based domain adaption achieves 99.56% accuracy fault diagnosis detection in semiconductor. Future research will be done to find ways to make data-driven fault diagnosis algorithms less dependent on supervised data.

Ethics declarations

Conflict of Interest:

On behalf of all the authors, the corresponding author states that there is no conflict of interest.

Funding:

There are no funding agencies related to this work.

Reference

[1] Lee, S., Kim, H.J. and Kim, S.B., 2020. Dynamic dispatching system using a deep denoising autoencoder for semiconductor manufacturing. Applied Soft Computing, 86, p.105904.

[2] Fan, S.K.S., Hsu, C.Y., Tsai, D.M., He, F. and Cheng, C.C., 2020. Data-driven approach for fault detection and diagnostic in semiconductor manufacturing. IEEE Transactions on Automation Science and Engineering, 17(4), pp.1925-1936.

[3] Zhu, F., Jia, X., Li, W., Xie, M., Li, L. and Lee, J., 2022. Cross-chamber data transferability evaluation for fault detection and classification in semiconductor manufacturing. IEEE Transactions on Semiconductor Manufacturing, 36(1), pp.68-77.

[4] Zhakov, A., Zhu, H., Siegel, A., Rank, S., Schmidt, T., Fienhold, L. and Hummel, S., 2020. Application of ANN for fault detection in overhead transport systems for semiconductor fab. IEEE Transactions on Semiconductor Manufacturing, 33(3), pp.337-345.

[5] Zhu, F., Jia, X., Miller, M., Li, X., Li, F., Wang, Y. and Lee, J., 2020. Methodology for important sensor screening for fault detection and classification in semiconductor manufacturing. IEEE Transactions on Semiconductor Manufacturing, 34(1), pp.65-73.

[6] Shim, J., Cho, S., Kum, E. and Jeong, S., 2021. Adaptive fault detection framework for recipe transition in semiconductor manufacturing. Computers & Industrial Engineering, 161, p.107632.

[7] Fan, S.K.S., Hsu, C.Y., Jen, C.H., Chen, K.L. and Juan, L.T., 2020. Defective wafer detection using a denoising autoencoder for semiconductor manufacturing processes. Advanced Engineering Informatics, 46, p.101166.

[8] Ferdaus, M.M., Zhou, B., Yoon, J.W., Low, K.L., Pan, J., Ghosh, J., Wu, M., Li, X., Thean, A.V.Y. and Senthilnath, J., 2022. Significance of activation functions in developing an online classifier for semiconductor defect detection. Knowledge-Based Systems, 248, p.108818.

[9] Mehta, K., Raju, S.S., Xiao, M., Wang, B., Zhang, Y. and Wong, H.Y., 2020. Improvement of TCAD augmented machine learning using autoencoder for semiconductor variation identification and inverse design. IEEE Access, 8, pp.143519-143529.

[10] Jang, J., Min, B.W. and Kim, C.O., 2019. Denoised residual trace analysis for monitoring semiconductor process faults. IEEE Transactions on Semiconductor Manufacturing, 32(3), pp.293-301.

[11] Saqlain, M., Jargalsaikhan, B. and Lee, J.Y., 2019. A voting ensemble classifier for wafer map defect patterns identification in semiconductor

manufacturing. IEEE Transactions on Semiconductor Manufacturing, 32(2), pp.171-182.

[12] Kang, S., 2020. Joint modeling of classification and regression for improving faulty wafer detection in semiconductor manufacturing. Journal of Intelligent Manufacturing, 31(2), pp.319-326.

[13] Nakazawa, T. and Kulkarni, D.V., 2019. Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing. IEEE Transactions on Semiconductor Manufacturing, 32(2), pp.250-256.

[14] Kim, E., Cho, S., Lee, B. and Cho, M., 2019. Fault detection and diagnosis using self-attentive convolutional neural networks for variable-length sensor data in semiconductor manufacturing. IEEE Transactions on Semiconductor Manufacturing, 32(3), pp.302-309.

[15] Azamfar, M., Li, X. and Lee, J., 2020. Deep learning-based domain adaptation method for fault diagnosis in semiconductor manufacturing. IEEE Transactions on Semiconductor Manufacturing, 33(3), pp.445-453.

[16] Kim, Y., Lee, H. and Kim, C.O., 2021. A variational autoencoder for a semiconductor fault detection model robust to process drift due to incomplete maintenance. Journal of Intelligent Manufacturing, pp.1-12.

[17] Kim, D.H. and Hong, S.J., 2021. Use of plasma information in machine-learning-based fault detection and classification for advanced equipment control. IEEE Transactions on Semiconductor Manufacturing, 34(3), pp.408-419.

[18] Yu, J., Zheng, X. and Liu, J., 2019. Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map. Computers in Industry, 109, pp.121-133.

[19] Schlosser, T., Friedrich, M., Beuth, F. and Kowerko, D., 2022. Improving automated visual fault inspection for semiconductor manufacturing using a hybrid multistage system of deep neural networks. Journal of Intelligent Manufacturing, 33(4), pp.1099-1123.

[20] Zhang, Y., Peng, P., Liu, C., Xu, Y. and Zhang, H., 2022. A sequential resampling approach for imbalanced batch process fault detection in semiconductor manufacturing. Journal of Intelligent Manufacturing, pp.1-16.

[21] Wang, J., Gao, P., Zhang, J., Lu, C. and Shen, B., 2023. Knowledge augmented broad learning system for computer vision based mixed-type defect detection in semiconductor manufacturing. Robotics and Computer-Integrated Manufacturing, 81, p.102513.