iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.81467

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Performance Enhancement of Steel Plate Fault Detection using Synthetic Data

Er. Shreeraj Khatiwada
Makawanpur Multiple Campus, Nepal

Abstract: ***The research contributes to enhancing the performance of steel plate fault detection by harnessing the capabilities of synthetic data generation techniques. This research is driven by a significant research gap, the increasing demand for high-quality synthetic data and the concurrent need for robust deep learning models that can benefit from such data. The main goal was to assess Tabular GAN's ability to create synthetic data that accurately mirrors original dataset patterns. Notably, the duration of training epochs was identified as a key factor significantly impacting data quality, emphasizing the iterative nature of GAN training. Exploring TGAN's ability to generate synthetic data with visual validations, such as Kernel Density Estimation (KDE) plots and histograms, conclusively established the remarkable similarity between the distributions of synthetic and original data, thereby reinforcing the effectiveness of TGAN in capturing underlying data patterns. It is systematically investigated the interplay of hyperparameters, including batch size, training epochs, hidden layers and test dataset size, in order to optimize the DNN model's effectiveness. Notably, the conventional assumption regarding test dataset size was challenged, revealing that judicious selection allows the DNN to maintain robust generalization performance even with smaller test datasets. Early stopping techniques were adeptly employed to strike a balance between capturing data patterns and mitigating overfitting. The research outcomes are pivotal, showcasing the immense potential of synthetic data in enhancing fault detection. The combination of TGAN generated synthetic data and a finely tuned DNN model achieved exceptional accuracy of 87.4% with minimized loss, surpassing conventional methods. These results furnish valuable insights into data generation techniques and model optimization, underscoring the transformative impact of synthetic data in real-world fault detection scenarios.***

Keywords: Synthetic Data, Material Informatics, Artificial Neural Network, Deep Neural Network, Fault Detection, TGAN, Confusion Matrix.

I. INTRODUCTION

Material informatics is a field that employs data mining techniques and computational methods to analyse and comprehend materials. By merging principles from materials science with artificial intelligence (AI), it aims to forecast material behaviour and uncover novel materials. The objective of material informatics is to expedite material discovery and development through data-driven strategies. It involves leveraging extensive data from experiments, simulations, and other sources to extract valuable insights. These insights are then analysed using machine learning, statistical, and computational techniques. The insights acquired through this analysis serve as a compass for the formulation of novel materials, enhancing the efficiency of current ones, and forecasting material behaviour across different circumstances [1].

Among various materials, steel is used widely in various engineering applications and its strength and toughness depend on its carbon percentage [1]. Steel plates find application in the production of machinery and ships; however, they are vulnerable to a range of catastrophic failures [2], therefore, the identification of faults becomes a pivotal concern within the realm of material science. Early detection of faults, including cracks, corrosion, or deformations, can pre-empt steel failures or collapses, resulting in significant time and cost savings [3, 4]. Additionally, regular fault detection can prolong the lifespan of the steel structure and improve overall safety for the people who use it [4]. By integrating machine learning into inspection and monitoring processes of steel plate, it becomes possible to significantly minimize the effort and resources wasted, leading to savings in both time and money [5]. There are various methods used in the steel industry to detect defects or predict the failure of steel plates such as Finite Element Analysis (FEA) [6], Stress-Strain Analysis [7], Fracture Mechanics [8], Probabilistic Methods [9], Machine Learning [10], Artificial Intelligence [11], and Non-destructive Testing (NDT) [12]. These methods include computer simulation, analysis of material properties, statistical models [13], machine learning algorithms, and techniques that check the integrity of the plate without damaging it. Out of above listed different techniques, automation of fault detection techniques using data mining methods is emerging as potent approach for detecting faults with new technology, like robots, smart systems, and machine learning, the process of fault detection can be done more quickly and efficiently [14], and these new technologies can help to find problems faster and at a lower cost.

©IJRASET: All Rights are Reserved

Artificial data is valuable for augmenting limited real-world datasets, increasing diversity, balancing imbalanced data, and creating training data for rare events or edge cases [15]. Artificial data serves several purposes, such as bias removal in training data, enlarging small datasets to improve robustness, and protecting privacy [16]. Privacy protection is another crucial application, where artificial data is designed to resemble some aspects of real data while preserving privacy [16]. It is observed that Tabular GAN works sufficiently for continuous data [17], as our dataset is a continuous dataset, we will employ TGAN [18] for the generation of artificial data.

In this research, we address the critical task of enhancing fault detection in steel plates by employing cutting-edge artificial intelligence techniques. Specifically, we harness the power of Artificial Neural Networks (ANN) [19] and Deep Neural Networks (DNN) [20] to analyse and classify defects within steel plates, benefiting from their ability to capture intricate patterns within data. We utilize an artificially generated data, created using methods like TGAN, to improve detection accuracy and robustness. This innovative approach not only ensures the structural integrity and safety of steel components in engineering applications but also offers a cost-effective and efficient solution for fault detection. Deep Neural Networks, with their multiple hidden layers, excel in autonomously learning hierarchical features, aligning well with the complexity of identifying subtle defects in steel plates. This data-driven methodology efficiently uses synthetic data, enabling the robust fault detection. To sum up, this research presents a promising prospect for enhancing the accuracy of steel plate fault detection using artificial data, which has significant implications for safety and cost-efficiency in engineering applications.

II. LITERATURE REVIEW

In the research by Abdullahi et al. [14], the authors applied machine learning to diagnose faults in steel plates by using three well-established machine learning models including Logistic Regression, Naive Bayes, and Support Vector Machine. The dataset utilized for training and evaluating the models was sourced from the UCI machine learning repository and accuracy, precision, and recall were employed as evaluation metrics to assess the performance of the models. The findings revealed that the Logistic Regression model attained the highest accuracy rate (94.5%) and precision score (0.756). In contrast, the Support Vector Machine model exhibited the highest recall score of 0.317.

In the study [12], the authors delve into the potential of decision tree ensembles for the detection of faults in steel plates. Through their analysis, the authors discovered that the use of ensembles can significantly enhance the accuracy of predictions. The Random Subspace and AdaBoost.M1 methods were found to be the most successful, achieving prediction accuracy rates of 83%. However, this is not a good prediction rate as Abdullahi et al. [14] achieved a higher prediction rate using logistic regression. They have also used multivariate feature selection method relief [36] to remove the features. Even though, they have not achieved a good prediction rate (82%).

In the study [5], researchers conducted an extensive evaluation of several machine learning algorithms aiming to determine the most effective algorithm for steel plate fault detection. Among the algorithms considered, the decision tree algorithm achieved an accuracy rate of 77.27%, accompanied by a Root Mean Square Error (RMSE) of 0.203. Adaboost, another algorithm under investigation, also demonstrated an accuracy of 77.27% but had a slightly higher RMSE of 0.232. K-Nearest Neighbor (KNN) exhibited an accuracy rate of 79.08% and an RMSE of 0.236. Support Vector Machine (SVM) achieved an accuracy of 75.16%, with an RMSE of 0.308. However, the standout performer among these algorithms was the random forest algorithm, which delivered the highest accuracy of 79.23% and the lowest RMSE of 0.203. These results emphasize that random forest is the most promising choice for steel plate fault detection, offering superior accuracy and precision compared to the other algorithms considered in the study.

In the study titled [13], researchers employed the research including C5.0 decision tree (C5.0 DT) with boosting, MLPNN with pruning, and Logistic Regression (LR) with a step-forward approach. Notably, the C5.0 decision tree with boosting algorithm exhibited outstanding diagnostic performance, achieving an impressive 97.25% on the training subset and an even higher accuracy of 98.09% on the test subset. These results highlight the effectiveness of C5.0 decision tree with boosting as a powerful tool for steel plate fault diagnosis, offering high accuracy and reliability in this critical domain.

In study [37], a C5.0 decision tree model demonstrated outstanding fault classification accuracy for steel plates, achieving 95.56% on the training dataset and 95.66% on the testing dataset. It outperformed alternative models, emphasizing its potential for enhancing quality control in the steel industry.

In the research [38], two datasets were utilized, including Toy dataset consisting of three categories, each with 2000 samples, and a faults dataset with three categories containing 670 samples for each category (k scratch, Bumps, and other faults). The primary focus of this study was to address the issue of data insufficiency in fault diagnosis.

To overcome this challenge, the researchers employed the WGAN-GP for data augmentation. The results demonstrated significant improvements in classification accuracies. Furthermore, the study suggested the potential for further enhancements through the utilization of Tabular GAN (TGAN) and larger Deep Neural Network (DNN)-based classifiers to increase both data diversity and quantity. These improvements in data augmentation ultimately led to enhanced model performance, promising advancements in fault diagnosis processes.

III.METHODOLOGY

A. System Model

As shown in Fig 1 the dataset is arbitrarily splitted into training and testing set. The training data is resized normalized as preprocessing and is used to train for the model. The final trained model is evaluated on the original test dataset based on the mentioned performance metrics.

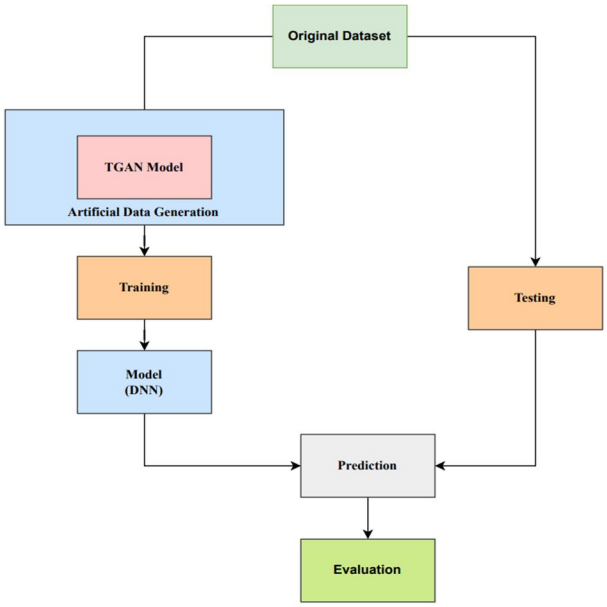


Figure 1: The Proposed Model

B. Data Collection and Description

For this study we will use data set provided by Semeion, Research Centre of Sciences of Communication. The data set [39, 40] provides information about the faults in steel plates that could potentially affect the occurrence of faults. This data can be used to develop a machine learning model that can accurately classify the faults in steel plates based on the given attributes. The dataset is one of the available datasets utilized to classify steel plate's faults into seven distinct types: Pastry, Z-Scratch, K-Scratch, Stains, Dirtiness, Bumps, and Other-Faults. The dataset consists of 1941 instances with distinct fault types label where each instance of the dataset has 27 independent variables and a fault type.

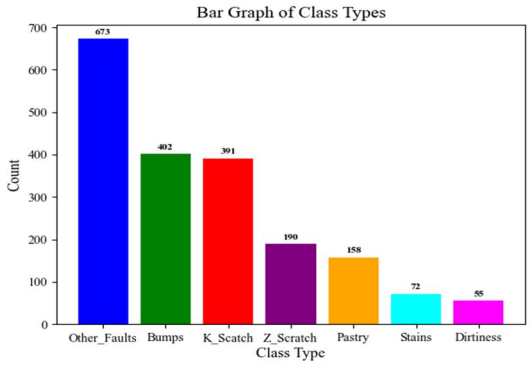


Figure 2: Steel plate's faults dataset with class distribution

C. Data Pre-processing

1) Reverse Encoding

Reverse encoding is the process of converting encoded data back to its original readable format. It is often employed when encoded data needs to be interpreted, analysed, or utilized in a meaningful way. The specific method for reverse encoding depends on the encoding scheme used. For example, in the case of one-hot encoding [41], which represents categorical variables as vectors, reverse encoding involves transforming the vectors back into their original categorical values. This is achieved by identifying the active (1) element in each one-hot encoded vector and mapping it back to its corresponding category or not active (0). Reverse encoding enables the recovery of the original data, restoring its semantic meaning and facilitating easier interpretation, analysis, and application in various domains, as shown in Table 1 and 2.

Table 1: Dataset before reverse encoding

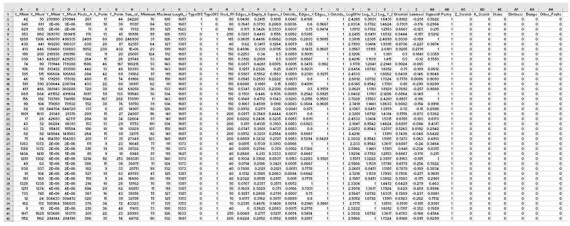
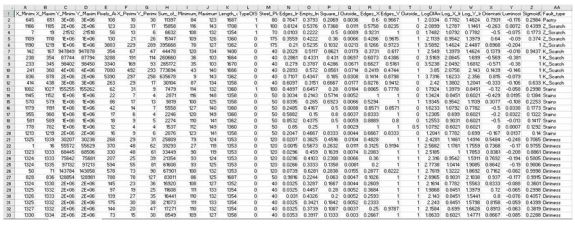


Table 2: Dataset after reverse encoding



2) Data Generation Using TGAN

The integration of TGAN (Tabular Generative Adversarial Network) into our research offers great potential for generating synthetic data closely mirroring real-world datasets. TGAN excels in capturing intricate statistical relationships within tabular data, preserving essential structural and distributional characteristics. Its data-driven approach adapts to dataset-specific features, enabling the detection of anomalies and faults, enhancing data quality, enabling more accurate analysis, improved machine learning model training, and deeper insights into our research objectives.

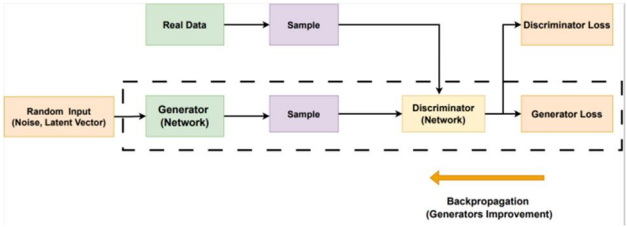


Figure 2: TGAN Architecture

D. Synthetic Data Quality Evaluation

1) Histogram Plot

In this work, a histogram plot was utilized with the arguments of bins, range, and density. The bins argument determined the number of intervals in which the data was divided for the histogram, providing insights into the frequency of values in each bin. To generate histogram, we used auto as a bin size. The range argument specified the range of values considered for the histogram, allowing for a focused analysis of specific data subsets. By setting the density argument to True, the histogram was normalized to have a total area of 1, enabling comparisons between histograms with different bin sizes.

2) Kernel Density Estimation

By choosing an appropriate kernel function and bandwidth, KDE provides a flexible and powerful tool for understanding the distribution of data and making predictions based on it. In this work, KDE is employed as a method for assessing data similarity.

E. Training the Model Using DNN

For training the model with a Deep Neural Network (DNN), we have utilized TensorFlow, a library for neural network development. The DNN model is structured as a sequential neural network, which means the data flows through the layers sequentially. It starts with an input layer with 16 units and a Rectified Linear Unit (ReLU) activation function. The shape of the input data is determined from Xtrain. The network then has hidden layer of 8 units, using ReLU activation functions. These layers progressively reduce the dimensionality of representations. Finally, there's an output layer with a number of units equal to number of classes, which employs the SoftMax activation function. This makes it suitable for multi-class classification tasks, where it outputs probability distributions over different fault types. The model's architecture and layer sizes can be customized according to the specific problem and dataset requirements. This configuration was tailored for multi-class fault type classification, optimizing the DNN's performance for precise fault classification.

F. Model Evaluation

A confusion matrix is a vital tool in assessing a machine learning model's performance, particularly for classification tasks. It summarizes the model's predictions in terms of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These metrics are used to calculate accuracy, precision, recall, and the F1-score, offering a comprehensive evaluation of the model's performance. Accuracy is a straightforward metric that measures overall correctness and is valuable when the goal is to maximize correct classifications without a strong focus on distinguishing between false positives and false negatives.

IV. Results and Discussion

A. Data Generation Using Tabular GAN

Table 3: Data samples with their quality score

Epoch	Sample	Overall Quality Score(%)	Column Shapes(%)	Column Pair Trends(%)
300	20000	83.7	78.2	89.19
	200000	83.75	78.31	89.19
	2000000	83.75	78.32	89.19
1000	20000	84.6	79.2	89.58
	200000	84.91	79.33	89.76
	2000000	85.04	80.09	90.03
5000	20000	89.53	87.37	91.69
	200000	89.55	87.38	91.73
	2000000	89.58	87.38	91.77
10000	20000	90.11	87.38	92.84
	200000	90.18	87.48	92.89
	2000000	90.19	87.46	92.91
30000	20000	91.03	89.64	92.36
	200000	91.03	89.68	92.53
	2000000	91.03	89.68	92.67

It shows an analysis of Tabular GAN results for generating synthetic data closely resembling real-world data. Using a benchmark dataset with 7 fault types and 27 independent variables, experiments varied the number of training epochs from 300 to 30,000. Longer training consistently improved data quality, aligning with established principles of deeper training for better results. TGAN's iterative training process allows it to better approximate the data distribution and capture statistical patterns, resulting in synthetic data closely matching real data characteristics. TGAN also demonstrated robustness and reliability across different training configurations, making it a versatile solution for data generation tasks.

The study revealed an intriguing finding regarding data sample quantity. Increasing the number of generated data samples had a relatively minor impact on data quality and the preservation of column pair trends, contrary to expectations. This underscores the significance of prioritizing training duration (epochs) over the quantity of data samples, indicating that extended training to comprehensively capture the data distribution is more effective for enhancing data quality and preserving critical data patterns. Understanding this relationship is valuable for those using TGAN for data generation, allowing them to allocate resources more effectively by focusing on training duration as a primary driver of success.

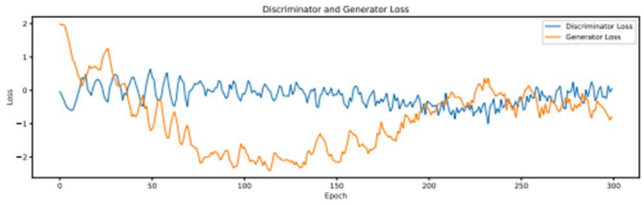


Figure 3: Generator and Discriminator Loss Plot (Epoch = 300)

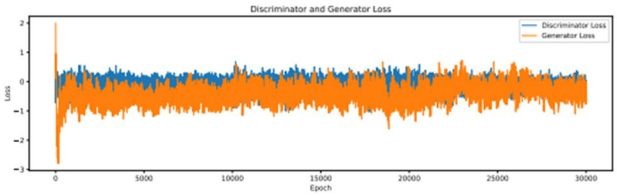


Figure 4: Generator and Discriminator Loss Plot (Epoch = 30000)

The study closely monitored the loss functions of the generator and discriminator networks in TGAN training, illustrating an iterative process where the generator aims to minimize its loss, and the discriminator seeks to enhance discrimination capabilities. Converging loss functions were observed, aligning with improved data quality in TGAN-generated synthetic data with extended training. Surprisingly, data quality significantly improved even after loss function convergence, highlighting the importance of continued training beyond convergence in enhancing the realism and fidelity of the generated data. This is a noteworthy and intriguing finding from the experiments.

Through Kernel Density Estimation (KDE) plots and histograms, the study visually validated TGAN's capabilities, revealing striking similarities between the original dataset and synthetic data distributions. This confirmed TGAN's proficiency in replicating intricate underlying data structures, not just surface-level statistics, highlighting its effectiveness in generating complex, realistic data. These visual confirmations strengthen the assertion that TGAN is a valuable tool for data generation, with the potential to enhance various domains by faithfully preserving data distribution and statistical patterns.

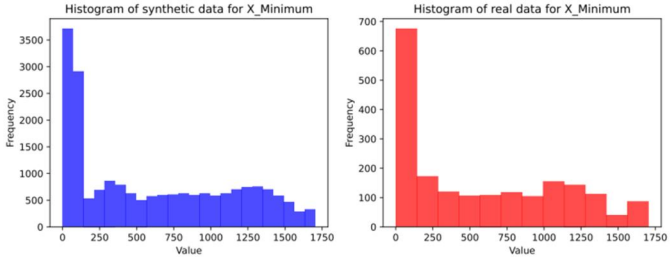


Figure 5: Histogram plot of X Minimum for both synthetic and real data

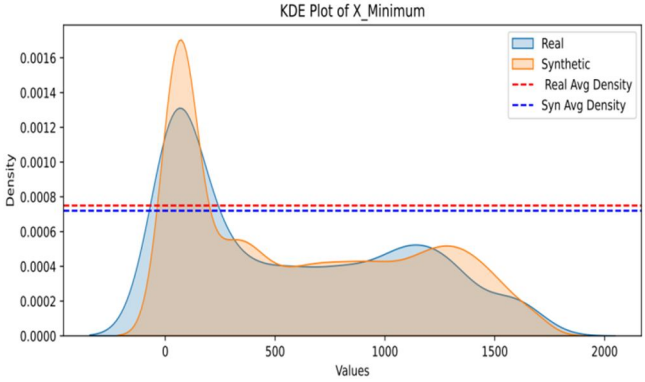


Figure 6: KDE plot of X Minimum for both synthetic and real data

The study introduced the Kolmogorov-Smirnov (KS) complement statistic as a critical tool for evaluating data quality in the context of synthetic data generation for the steel plate dataset. The positive outcomes obtained through the KS complement further support the central finding that TGAN's synthetic data generation is not limited to steel plate data but extends to diverse domains. This application of a rigorous statistical measure underscores TGAN's potential to generate high-quality synthetic data across various fields, emphasizing its adaptability and reliability in faithfully replicating real-world data characteristics. The research not only demonstrates TGAN's efficacy for steel plate data but also highlights its transformative potential in a wide range of applications, making it a valuable asset in data generation.

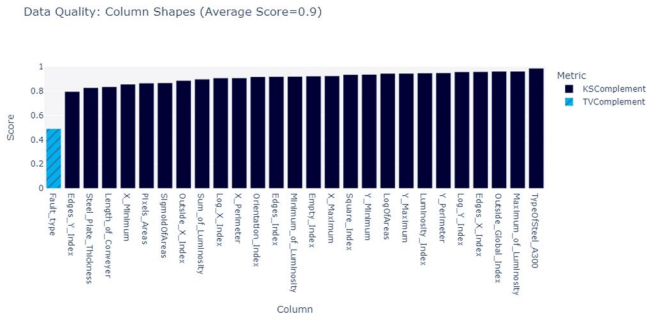


Figure 7: Overall data quality score and column wise trends for epoch of 30000 with synthetic data of 20000

B. DNN Model Performance

As we know, the effectiveness of DNN models in addressing intricate tasks depends upon a multitude of factors that collectively influence their performance. In this study, we systematically explored these influential factors, encompassing hyperparameter tuning, the interplay of batch sizes (32, 64, 128, 256, 512) and epochs (200, 500, 1000), as well as adjustments in the test dataset size (0.2, 0.3, 0.4) in conjunction with varying numbers of hidden layers (1, 2, 3, 4). We designed our model architecture using TensorFlow Keras Sequential, incorporating multiple dense layers with SoftMax activation for a classification task. In our exploration of hyperparameter tuning, which included variations in batch sizes, epochs, test dataset sizes, and hidden layers, a noteworthy discovery emerged.

1) The Test Dataset Size Trade-Off

In our research, we explored the influence of test dataset size on model evaluation, finding that it significantly impacts the reliability of performance metrics. Larger test datasets (e.g., 40% of the total data) offer more robust evaluations but reduce available training data, while smaller test datasets (e.g., 20%) may result in noisier metrics but allow for a larger portion of data for model learning. We discovered that, among various combinations of epochs, batch sizes, and hidden layers, a test dataset size of 0.2 yielded the best testing accuracy, indicating that our deep neural network maintained strong generalization even with a larger test dataset. This underscores the need to carefully balance the trade-off between test dataset size and training data availability to optimize model performance and generalization.

Table 4: Result based on Test Size

Test Size	Epoch [Early Stopped]	Batch Size	Hidden Layer	Testing Accuracy	Testing Loss
0.2	**200 [183]**	**128**	**2**	**0.874**	**0.39**
0.3	1000 [281]	512	2	0.867	0.411
0.4	500 [228]	512	2	0.863	0.418

2) DNN's Layer

The research systematically explored the impact of hidden layer architecture on the performance of the Deep Neural Network (DNN) model. Among various configurations ranging from 1 to 4 hidden layers, it was found that models with two hidden layers consistently delivered optimal testing accuracy and loss, demonstrating robust performance across different test dataset sizes, batch size and epochs as shown in table below. This underscores the pivotal role of hidden layer architecture in influencing DNN model performance. The study's DNN model consisted of an input layer with 16 neurons utilizing ReLU activation functions, and subsequent hidden layers consists of 8 neurons, also employing ReLU activation functions. The final output layer, activated by 'softmax,' enabled multi-class fault type classification. These insights emphasize the importance of judiciously selecting the number of hidden layers to strike a balance between model complexity and generalization capacity.

Table 5: Result based on DNN Layer

Hidden Layer	Epoch [Early Stopped]	Batch Size	Test Size	Testing Accuracy	Testing Loss
1	200 [113]	64	0.2	0.868	0.4
2	**200 [183]**	**128**	**0.2**	**0.874**	**0.39**
3	200 [168]	512	0.2	0.864	0.399
4	200 [133]	512	0.2	0.854	0.595

3) Generalization and the Impact of Batch Size

Our research delved into the impact of varying batch sizes on Deep Neural Network (DNN) performance, maintaining other parameters as constants. Surprisingly, we challenged the conventional belief that smaller batch sizes inevitably lead to superior model generalization. Instead, we identified a compelling pattern: a batch size of 128 yields optimal performance, emphasizing the intricate relationship between batch size and DNN performance. Coupled with two hidden layers, specific epoch settings, and a test dataset size of 0.2, this configuration achieved the highest testing accuracy and lowest testing loss. These findings underscore the critical role of batch size selection in designing effective DNN models, urging a data-dependent approach for optimal hyperparameter settings.

Furthermore, our study challenged the conventional wisdom regarding batch size and generalization, revealing that larger batch sizes, notably 256 and beyond, consistently delivered robust and consistent model performance. Smaller batch sizes, while initially showing faster convergence, failed to maintain this advantage over time. We recommend using a test dataset size of 0.2 and configuring the model with two hidden layers for peak performance, challenging traditional assumptions and advocating for a data-dependent and refined approach to batch size selection in hyperparameter tuning. By tuning all the hyperparameters, we achieved the optimal results based on batch size, as illustrated in the table below:

Table 6: Result based on Batch Size

Batch Size	Hidden Layer	Epoch [Early Stopped]	Test Size	Testing Accuracy	Testing Loss
32	1	200 [69]	0.2	0.867	0.4
64	1	200 [113]	0.2	0.868	0.399
128	**2**	**200 [183]**	**0.2**	**0.874**	**0.39**
256	2	500 [173]	0.2	0.866	0.404
512	2	500 [244]	0.2	0.865	0.404

4) Performance Evaluation and Insights

After conducting an exhaustive exploration of hyperparameters, we identified the optimal configuration for the DNN's model. This optimal setup included two hidden layers, a batch size of 128, and a test dataset size of 0.2 when using the synthetic data generated by TGAN. Remarkably, when we applied the same configurations to the original dataset, it becomes evident that in comparison to the original dataset, the synthetic dataset significantly enhances the model's performance by clearly increasing testing accuracy and minimizing loss. Comparison of these two datasets with their performance can be viewed on below table.

Table 7: Performance comparison between original and synthetic data

Batch Size	Hidden Layer	Test Size	Original Dataset		Synthetic Dataset	
			Testing Accuracy	Testing Loss	Testing Accuracy	Testing Loss
512	2	0.2	0.708	0.724	0.864	0.41
256	2	0.2	0.71	0.708	0.858	0.422
128	**2**	**0.2**	**0.717**	**0.65**	**0.874**	**0.391**
64	2	0.2	0.751	0.667	0.867	0.399
32	2	0.2	0.73	0.68	0.863	0.405

This outcome underscores the potency of fine-tuning DNN models using synthetic data, as it not only matches the performance of real data but, in some instances, exceeds it.

Furthermore, we analysed the training and testing accuracy and loss curves, confusion matrices, and AUC-ROC for both the synthetic and original datasets. These evaluation tools are crucial in comprehending the DNN model's classification performance, guiding further model enhancements, and facilitating data-driven decision-making in the realm of machine learning.

The training and testing accuracy and loss curves for both datasets are displayed below:

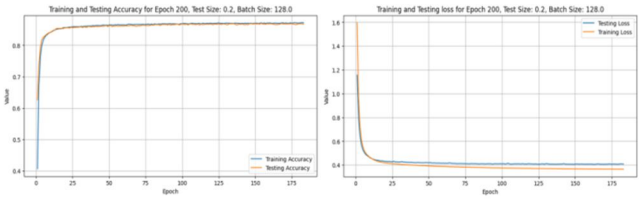


Figure 8: Training Testing Accuracy and Loss Curve of Synthetic Dataset

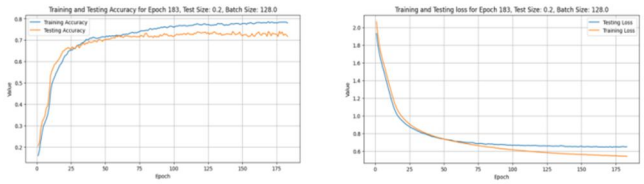


Figure 9: Training Testing Accuracy and Loss Curve of Original Dataset

These curves provide insights into how the model learns and generalizes over time, helping us to fine-tune its performance. In addition, we analysed confusion matrices for both datasets, offering detailed insights into the model's classification performance, including metrics such as true positives, true negatives, false positives, and false negatives for each class. Moreover, we utilized the AUC-ROC metric to assess the model's class discrimination abilities. The AUC-ROC curve visually represents the model's capability to distinguish between different classes, which is especially valuable when dealing with imbalanced datasets. Incorporating these evaluation tools and the insights derived from our extensive experimentation has not only enhanced our understanding of the model's performance but also provided valuable guidance for the development of high-performing machine learning models in similar contexts. The combination of optimized hyperparameters and comprehensive evaluation metrics positions us to make informed decisions and refine our models effectively.

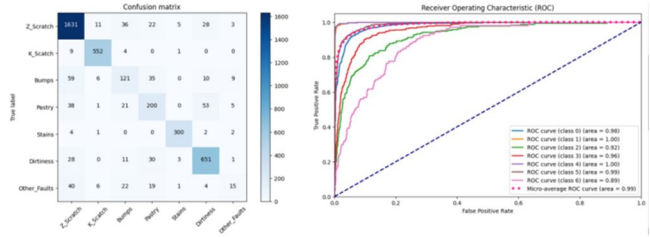


Figure 10: Confusion Matrix and AUC-ROC Curve of Synthetic Dataset

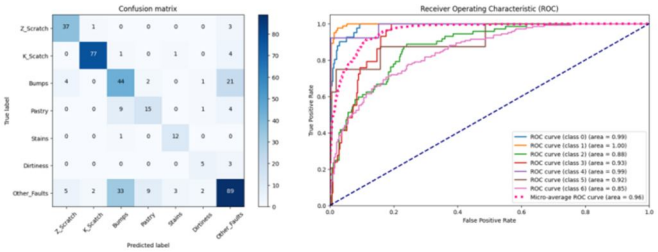


Figure 11: Confusion Matrix and AUC-ROC Curve of Original Dataset

C. Comparison of Performance

In contrast to the study conducted by Gao et al. [38], where they worked with a dataset consisting of three classes (K scratch, Bumps, and Other Faults) totalling 2000 samples and used a WGAN-GP-based data augmentation model for data generation along with SGD-LR as a classifier, their reported accuracy for the provided dataset is 73.04%. On the other hand, in our work, we leveraged a

Tabular GAN for synthetic data generation and dealt with a dataset of 20,000 samples that consist of seven distinct fault classes. We employed a DNN for model evaluation, resulting in an accuracy of 87.4%. This comparison showcases that our model outperforms Gao et al.'s approach in multiple aspects. Firstly, our model handles a larger and more complex dataset with seven classes, demonstrating its scalability. Secondly, despite the dataset's inherent imbalance, our model achieves a significantly higher accuracy of 0.874, showcasing its ability to effectively enhance accuracy even in such challenging scenarios. Therefore, our approach proves superior in terms of both dataset complexity and accuracy improvement, making it a robust choice for handling imbalanced dataset scenarios.

Table 8: Performance comparison between two data augmentation model

Data Augmentation Model	Data Samples	Fault Class	Classifier	Accuracy
WGAN-GP	2000	3	SGD-LR	73.04%
TGAN	20000	7	DNN	87.4%

In comparing the performance of the three models-Random Forest, Deep Neural Network (DNN), and TabNet - across both real and synthetic datasets, several distinctions become evident. The Random Forest algorithm achieved an accuracy of 78.1% on the original dataset but exhibited signs of overfitting. When applied to the synthetic dataset, its accuracy improved to 83.3%. Our proposed DNN model obtained 71.7% accuracy on the real dataset, yet demonstrated remarkable adaptability when trained on the synthetic dataset, achieving 87.4%. By contrast, the TabNet architecture performed competitively on the original dataset with 77.4% accuracy, closely aligning with Random Forest. However, its accuracy declined to 68.7% when trained on synthetic dataset, suggesting limitations in handling augmented data. Overall, while Random Forest and TabNet performed reasonably well on real data, the DNN model emerged as the most robust and versatile, particularly when leveraging synthetic data for enhanced classification performance.

Table 9: Performance comparison of proposed model with Random Forest and TabNet

	Original Dataset [Accuracy]	Synthetic Dataset [Accuracy]
Random Forest	78.1%	83.3%
TabNet Architecture	77.4%	81.7%
Proposed Model	71.7%	87.4%

V. CONCLUSION

In this research, we take on a journey to explore the potential of TGAN in data generation and thoroughly assessed the performance of a DNN model. Our primary focus was on using TGAN to generate synthetic data from a benchmark dataset of 1941 records with 7 fault types as the dependent variable and 27 independent variables as features. We discovered that the duration of training epochs significantly impacted data quality, highlighting the iterative nature GAN training.

Furthermore, our experiments demonstrated the robustness of TGAN across different configurations, with training duration outweighing the quantity of data samples generated. Visualizations through Kernel Density Estimation (KDE) plots and histograms visually confirmed the similarity between synthetic and original data distributions, affirming TGAN's proficiency in replicating data patterns. We introduced the Kolmogorov Smirnov (KS) complement statistic as a rigorous evaluation tool, consistently reinforcing TGAN's versatility.

Additionally, our research extended to exploring the performance of an DNN model. We examined the relationships between batch size, training epochs, and test dataset size, discovering that smaller batch sizes facilitated faster convergence but didn't consistently result in better generalization. We determined optimal training epoch numbers to balance data pattern capture and overfitting mitigation. We also challenged the convention of favouring larger test datasets, demonstrating that careful consideration allowed the DNN to maintain competitive generalization performance even with smaller test datasets.

In conclusion, our research provides valuable insights into TGAN's capabilities in data generation and the results of DNN model performance. These findings contribute to a deeper understanding of data generation techniques and offer guidance for model training and evaluation. This knowledge serves as a foundation for more effective data generation and model development across a broad spectrum of applications in the evolving field of artificial intelligence.

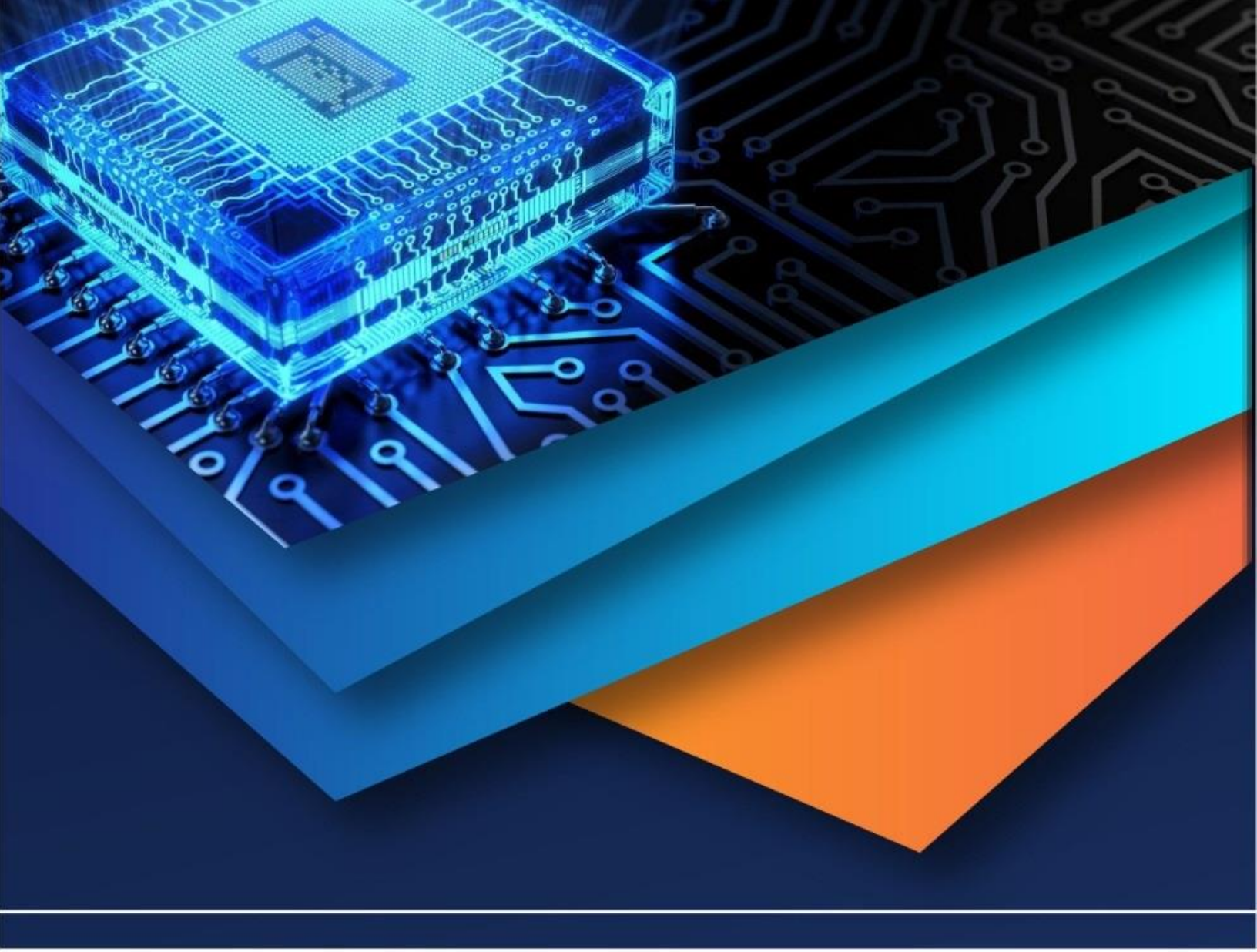
A. Industrial Implementation

Beyond theoretical contributions, the proposed integration of TGAN-generated synthetic data with DNN models carries significant potential for industrial deployment. In steel manufacturing plants, the approach can be embedded into automated inspection systems, enabling real-time fault detection on production lines. By leveraging synthetic data, industries can overcome challenges posed by limited or imbalanced datasets, ensuring that models remain robust under diverse operating conditions. Moreover, coupling this system with IoT-enabled sensors and non-destructive testing devices would allow predictive maintenance, minimizing downtime and reducing operational costs. Such practical implementation not only enhances quality assurance and safety in steel production but also establishes a scalable framework adaptable to other manufacturing sectors where defect detection is critical.

REFERENCES

[1] C. Li and K. Zheng, "Methods, progresses, and opportunities of materials informatics," InfoMat, vol. 5, no. 8, Art. no. e12425, 2023.
[2] A. Kelly and K. M. Knowles, Crystallography and Crystal Defects. Hoboken, NJ, USA: Wiley, 2020.
[3] D. Miljković, "Fault detection methods: A literature survey," in Proc. 34th Int. Conv. MIPRO, Opatija, Croatia, 2011, pp. 750–755.
[4] N. Amruthnath and T. Gupta, "A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance," in Proc. 5th Int. Conf. Ind. Eng. Appl. (ICIEA), Singapore, 2018, pp. 355–361, doi: 10.1109/IEA.2018.8387124.
[5] A. K. Srivastava, "Comparison analysis of machine learning algorithms for steel plate fault detection," Int. Res. J. Eng. Technol., vol. 6, no. 5, pp. 1231–1234, 2019.
[6] B. Szabo and I. Babuska, Finite Element Analysis: Method, Verification and Validation. Boca Raton, FL, USA: CRC Press, 2021.
[7] D. Liu et al., "Strain analysis and engineering in halide perovskite photovoltaics," Nat. Mater., vol. 20, no. 10, pp. 1337–1346, 2021.
[8] S. K. Maiti, Fracture Mechanics. Cambridge, U.K.: Cambridge Univ. Press, 2015.
[9] G. C. Calafiore, F. Dabbene, and R. Tempo, "Research on probabilistic methods for control system design," Automatica, vol. 47, no. 7, pp. 1279–1293, 2011.
[10] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," Science, vol. 349, no. 6245, pp. 255–260, 2015.
[11] W. Ertel, Introduction to Artificial Intelligence. Cham, Switzerland: Springer Nature, 2024.
[12] S. M. Halawani, "A study of decision tree ensembles and feature selection for steel plates faults detection," Int. J. Tech. Res. Appl., vol. 2, no. 4, pp. 127–131, 2014.
[13] M. Fakhr and A. M. Elsayad, "Steel plates faults diagnosis with data mining models," J. Comput. Sci., vol. 8, no. 4, pp. 506–512, 2012.
[14] A. Abdullahi et al., "Towards IR4.0 implementation in e-manufacturing: Artificial intelligence application in steel plate fault detection," Indones. J. Elect. Eng. Comput. Sci., vol. 20, no. 1, pp. 430–436, 2020.
[15] A. Triastcyn and B. Faltings, "Generating artificial data for private deep learning," arXiv preprint arXiv:1803.03148, 2018.
[16] J. Jordon et al., "Synthetic data – what, why and how?" arXiv preprint arXiv:2205.03257, 2022.
[17] S. Bourou, A. El Saer, T.-H. Velivassaki, A. Voulkidis, and T. Zahariadis, "A review of tabular data synthesis using GANs on an IDS dataset," Information, vol. 12, no. 9, Art. no. 375, 2021.
[18] D. Zheng et al., "Tabular generative adversarial networks for data synthesis," arXiv preprint arXiv:1708.07963, 2017.
[19] A. Krenker, J. Bešter, and A. Kos, "Introduction to the artificial neural networks," in Artificial Neural Networks: Methodological Advances and Biomedical Applications. Rijeka, Croatia: InTech, 2011, pp. 1–18.
[20] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Explaining deep neural networks and beyond: A review of methods and applications," Proc. IEEE, vol. 109, no. 3, pp. 247–278, 2021.
[21] R. Y. Gupta, S. S. Mudigonda, and P. K. Baruah, "TGANS with machine learning models in automobile insurance fraud detection and comparative study with other data imbalance techniques," 2012.
[22] K. Gurney, An Introduction to Neural Networks. Boca Raton, FL, USA: CRC Press, 2018.
[23] J. Kim and C. D. Scott, "Robust kernel density estimation," J. Mach. Learn. Res., vol. 13, no. 1, pp. 2529–2565, 2012.
[24] Synthetic Data Metrics, DataCebo, Inc., Apr. 2023. [Online]. Available: https://docs.sdv.dev/sdmetrics/. Version 0.9.3
[25] B. Wu, J. Xu, Y. Zhang, B. Liu, Y. Gong, and J. Huang, "Integration of computer networks and artificial neural networks for an AI-based network operator," arXiv preprint arXiv:2407.01541, 2024.
[26] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," arXiv preprint arXiv:1811.03378, 2018.
[27] Y. Bai et al., "Understanding and improving early stopping for learning with noisy labels," in Advances Neural Inf. Process. Syst., vol. 34, pp. 24392–24403, 2021.
[28] A. Abid, M. T. Khan, and J. Iqbal, "A review on fault detection and diagnosis techniques: Basics and beyond," Artif. Intell. Rev., vol. 54, no. 5, pp. 3639–3664, 2021.
[29] S. Liu, Q. Wang, and Y. Luo, "A review of applications of visual inspection technology based on image processing in the railway industry," Transp. Saf. Environ., vol. 1, no. 3, pp. 185–204, 2019.
[30] M.-M. Naddaf-Sh et al., "Defect detection and classification in welding using deep learning and digital radiography," in Fault Diagnosis and Prognosis Techniques for Complex Engineering Systems. Amsterdam, The Netherlands: Elsevier, 2021, pp. 327–352.
[31] M. Shahriari-Kahkeshi, F. Sheikholeslam, and J. Askari, "Adaptive fault detection and estimation scheme for a class of uncertain nonlinear systems," Nonlinear Dyn., vol. 79, no. 4, pp. 2623–2637, 2015.
[32] T. Amanuel, A. Ghirmay, H. Ghebremeskel, R. Ghebrehiwet, and W. Bahlibi, "Comparative analysis of signal processing techniques for fault detection in three-phase induction motor," J. Electron., vol. 3, no. 1, pp. 61–76, 2021.
[33] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," IEEE Trans. Ind. Electron., vol. 62, no. 6, pp. 3757–3767, 2015.

[34] X. Liao, D. Wang, S. Qiu, and X. Ming, "ReDBN: An interpretable deep belief network for fan fault diagnosis in iron and steel production lines," IEEE/ASME Trans. Mechatronics, 2024.
[35] L. Russo, K. Sarda, L. Glielmo, and A. Acernese, "Fault detection and diagnosis in steel industry: A one-class support vector machine approach," in Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC), Melbourne, Australia, 2021, pp. 2304–2309.
[36] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," J. Biomed. Inform., vol. 85, pp. 189–203, 2018.
[37] M. A. A. Kazemi, S. Hajian, and N. Kiani, "Quality control and classification of steel plates faults using data mining," Appl. Math. Inf. Sci. Lett., vol. 6, no. 2, pp. 59–67, May 2018, doi: 10.18576/amisl/060202.
[38] X. Gao, F. Deng, and X. Yue, "Data augmentation in fault diagnosis based on the Wasserstein generative adversarial network with gradient penalty," Neurocomputing, vol. 396, pp. 487–494, 2020.
[39] B. Taşar, "Comparison analysis of machine learning algorithms for steel plate fault detection," Duzce Univ. J. Sci. Technol., vol. 10, no. 3, pp. 1578–1588, 2022.
[40] Steel Plates Faults Data Set, Semeion Research Center of Sciences of Communication, Rome, Italy, 2017. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/steel+plates+faults
[41] Y. Sun et al., "Modifying the one-hot encoding technique can enhance the adversarial robustness of the visual model for symbol recognition," Expert Syst. Appl., vol. 250, Art. no. 123751, 2024.
[42] R. L. Nuzzo, "The box plots alternative for visualizing quantitative data," PM&R, vol. 8, no. 3, pp. 268–272, 2016.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)