Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2023) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Industry 4.0 and Smart Manufacturing

DQ-DeepLearn: Data Quality Driven Deep Learning Approach for Enhanced Predictive Maintenance in Smart Manufacturing

Nitesh Bharot[a,*], Priyanka Verma[a], Mirco Soderi[a], John G. Breslin[a]

[a]*Data Science Institute, University of Galway, Galway, H91FP6V, Ireland*

Abstract

In the realm of smart manufacturing, predictive maintenance plays a pivotal role in ensuring equipment reliability, minimizing downtime, optimizing costs, and reducing product failure rate by detecting faulty products in the early stage. However, the efficacy of predictive maintenance hinges on the quality of training data employed for predictive modeling. Inadequate data quality can lead to erroneous predictions and, consequently, ineffective maintenance strategies. This research paper introduces a solution that leverages deep learning and data quality management to enhance predictive maintenance in smart manufacturing. Our proposed methodology encompasses two major components such as data quality management and faulty product detection. Data quality management includes data preprocessing, feature reduction, and data balancing, whereas faulty product detection is done using deep learning techniques. By combining these elements, a predictive model capable of accurately forecasting faulty products in early stages to reduce economic losses is developed. The proposed approach effectively addresses data quality issues and is tested on the SECOM dataset which indicates that it surpasses traditional models with an accuracy of 96.4% and perfect recall. Ultimately, this research contributes to the advancement of predictive maintenance and deep learning in the context of smart manufacturing, benefiting both industrial practitioners and researchers alike.

Keywords: Smart Manufacturing; Industry 4.0; Predictive Maintenance; Data Quality Management; Deep Learning

1. Introduction

In smart manufacturing industries, the detection and prevention of faulty products are critical factors that contribute to maintaining high-quality output and reducing waste [1]. Traditional maintenance approaches often rely on reactive measures, leading to increased downtime, higher costs, and compromised product quality [2]. However, with the advancements in technology and the emergence of smart manufacturing, predictive maintenance has gained significant attention as a proactive strategy to address these challenges.

* Nitesh Bharot
E-mail address: nitesh.bharot@universityofgalway.ie

Predictive maintenance involves scrutinizing data from multiple sources, like sensors, production machines, & quality control systems, to anticipate equipment failures, faulty products, and schedule maintenance activities accordingly [2]. By leveraging historical data, predictive maintenance models can identify patterns, anomalies, and indicators of potential faults, enabling timely interventions to prevent the production of faulty products.

However, the effectiveness of predictive maintenance heavily depends on the quality of data used to train the models [3]. Poor data quality, characterized by missing values, outliers, inconsistencies, and measurement errors, can significantly impact the efficacy of predictive maintenance models. When unreliable data is utilized, it can lead to erroneous predictions, false alarms, and ultimately ineffective maintenance strategies [4].

To address these challenges, a data quality-driven approach for predictive maintenance in smart manufacturing industries is proposed. This approach focuses on ensuring high-quality data by employing comprehensive data quality assessment techniques, data cleaning, preprocessing, and feature engineering methods. By integrating these data quality management techniques with the powerful capabilities of deep learning, we aim to enhance the accuracy, reliability, and effectiveness of predictive maintenance models in detecting faulty products.

At the start KNN imputer is employed to handle missing data, Principal Component Analysis (PCA) for feature reduction, CTGAN for class imbalance removal, and RNET50, a deep learning model, for classification. The proposed methodology involves a series of steps that aim to address data quality issues and improve the accuracy of the predictive maintenance model. In this proposed approach, we specifically leverage the RNET50 architecture for training the predictive maintenance model. By utilizing RNET50, complex patterns and correlations within the data could be efficiently captured, enabling the accurate identification of faulty products in real-time or on a periodic basis.

Through this research, the aim is to address the challenges associated with data quality in predictive maintenance and provide an innovative solution for detecting faulty products in smart manufacturing industries. By combining data quality management techniques, deep learning methodologies, and the RNET50 architecture, we envision improved operational efficiency, reduced downtime, enhanced product quality, and optimized maintenance strategies. The results and insights obtained from this study can benefit industrial practitioners and researchers seeking to harness the full potential of predictive maintenance in the context of smart manufacturing.

The rest of the paper is organised as follows: Section 2 delves into the related work and our proposed approach is described in section 3. Section 4 begins with the system configurations and metrics involved and analyses the results. Finally, section 5 concludes the paper and discusses scope of future work.

2. Related Work

In recent years, the application of DL techniques for predictive maintenance & data quality management has grabbed more attention from researchers & practitioners. Several studies have addressed challenges related to feature selection, class imbalance, and predictive modeling [5]. This section gives an overview of the relevant literature in these areas.

Anghel et al. [6] compared and applied two alternative prediction models on the SECOM dataset. Redundant and null data removal is done during preprocessing. Oversampling is used to solve the class imbalance problem. Then it uses two methods for feature selection; the Multivariate Adaptive Regression Spline (MARS) algorithm and Support Vector Machine (SVM) technique. Once primary attributes are chosen, MARS uses Gradient Boosting Trees (GBT) and neural networks leading to a conclusion that NN is the better strategy.

In another study, Kerdprasop et al. [7] developed a new method called the MeanDiff algorithm for preprocessing SECOM datasets. The algorithm compares value differences, forms clusters based on passing or failing instances, and ranks features based on estimated value differences. Additionally, columns with duplicate data and more than fifty-five percent of empty values are excluded. It compares the performance of Naive Bayes (NB), decision trees, k-Nearest Neighbor (k-NN), and logistic regression algorithms on clean data. NB performs best but has many false positives. Decision trees, on the other hand, minimize false positive and false negative results.

Munirathinam et al. [8] explored how to use machine learning approaches to develop accurate models for error detection. Features with constant values & with missing values (greater than fifty-five percent) are removed using feature selection techniques. In addition, PCA and statistical chi-square analysis are used. Techniques such as expertise, correlation analysis, and variable component analysis are employed to address issues such as overfitting, increased throughput, and reduced predictive accuracy. The main drawback is that oversampling duplicates existing samples, making the model prone to overfitting. Deep learning is needed to enhance the prediction of anomalous classes. A DL approach called Deep Belief Network (DBN) supports both supervised & unsupervised learning. A DBN has various nodes & layers depending on the dataset. Using gradient descent makes it difficult to achieve global optimization.

Tayaba Abbasi et al. [9] used a recurrent neural network with long short-term memory to perform predictive maintenance on turbocharger compressor engines, a key component of machinery in the oil and gas industry. Similarly, a study by Narul Afiqah A. Majid et al. [10], highlights the utility of mechatronics in predictive maintenance through a method called Vibration Fault Simulation System (VFSS). The paper concludes vibration is the most dependable parameter in maintenance prediction. The study by Xiang Li et al. [11], employs the random forest regression model for implementing an IoT equipment maintenance predictive model, highlighting its superior performance.

The related work in the field of predictive maintenance and data quality management has explored various techniques for feature reduction, class imbalance handling, and predictive modeling, but still, there is a scope to improve the performance of predictive maintenance. Hence, we proposed DQ-DeepLearn which enables efficient predictive maintenance.

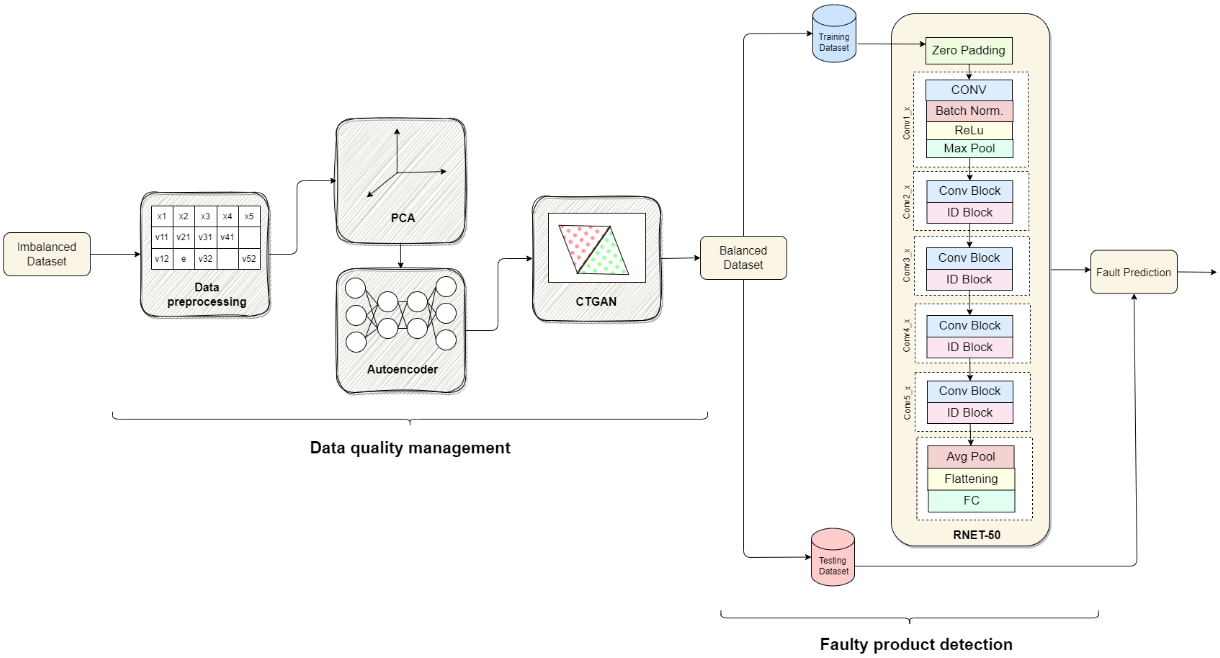


Fig. 1. Proposed model design

3. Proposed approach

Predictive maintenance aids in the forecast of the mechanical errors of the machine thus adding longevity to the system. One of the major factors contributing to predictive maintenance is data quality. Noisy, inaccurate data could lead to faulty detection hence our proposed approach consists of data quality management as its first phase. It involves refining the data thus helping to make the knowledge extraction easier. It then proceeds with faulty product detection as the second phase. The workflow of the proposed approach designed is described in Fig. 1.

3.1. Data quality management

The data quality management phase handles missing values, noisy features, and class imbalance problems. It also helps to reduce the computation time by reducing the number of features using PCA. Additionally, it aids to characterize those features using an autoencoder system. This phase commences with the data preprocessing component which involves data imputation thus handling missing values. Followed by preprocessing comes the dimensionality reduction of components. It helps to eliminate noisy features and enhances data knowledge. At the end of this phase, a data balancing technique is applied to overcome the class imbalance problem. This approach aims to enhance operational efficiency, minimize downtime, and optimize maintenance strategies in smart manufacturing environments.

3.1.1. Data preprocessing

It involves a series of steps to prepare and transform raw data into a suitable syntax for analyzing and modeling. The proposed approach involves data pruning and data imputation for data preprocessing. This component of data quality management works on quality enhancement by applying preprocessing techniques such as min-max scaling, label encoding, KNN imputation, and outlier detection. It enables bringing the data values to the same level and filling in the missing values using KNN. The goal of data imputation is to maintain the completeness and integrity of the dataset by providing estimated values for missing entries. This component also helps to convert categorical values to numeric easing the system modeling. Additionally, it also uses data pruning which helps to streamline the dataset, making it more manageable and enhancing the performance of subsequent data processing and modeling steps. Finally, it uses Isolation Forest to reduce the number of outliers in the dataset.

3.1.2. Feature reduction

This component deals with dimensionality reduction which in turn reduces the computation time of the model. Moreover, it allows us to vizualize the data with reduced number of features using PCA and AutoEncoder system. PCA is famous for analyzing data & for reducing dimensions of data along with maintaing the significant information of data intact.

Algorithm 1 PCA

Input: M: Data matrix with dimensions (no. of samples, no.of features)
k: Number of desired principal components
Output: Transformed data matrix with shape (n_samples, k)

```
1: Standardize the data:
      for each feature j:
      compute mean_j = mean(M[:, j])
      compute std_j = σ(M[:, j])
      for each sample i:
      M[i, j] = (M[i, j] - mean_j) / σ
2: Compute the covariance matrix:
      covariance_matrix = compute_covariance_matrix(M)
3: Perform Eigendecomposition:
      eigenvalues, eigenvectors = perform_eigendecomposition (covariance_matrix)
4: Sort eigenvalues and eigenvectors:
      sorted_indices = sort_indices(eigenvalues, descending=True)
      sorted_eigen_values = eigen_values[sorted_indices]
      sorted_eigen_vectors = eigen_vectors[:, sorted_indices]
5: Select principal components:
      selected_eigenvectors = sorted_eigenvectors[:, :k]
6: Transform data:
      transformed_data = M.dot(selected_eigenvectors)
7: Return transformed data:
      return transformed_data
```

PCA works to identify the principal components (PCs), which are a set of orthogonal axes that capture the maximum variance. The first PC_1 represents the direction along which the data varies the most, while each subsequent principal component represents a new direction that captures the remaining variance. These PCs are computed as linear combinations of the original features, with coefficients determined through eigendecomposition as described in Algorithm 1.

After reducing the number of features using PCA, we aim to refine the data using auto encoders. Auto encoders are neural networks that compress input data into a lower-dimensional representation before reconstructing the original data. By analyzing the compressed representation, important features can be identified and selected, while less relevant ones are discarded. This reduced feature set can be used to train new models, leading to simpler and more interpretable models with potential improvements in generalization and avoidance of overfitting. The AutoEncoder system used for the proposed system is depicted through Fig. 2.

3.1.3. Data Balancing

Class imbalance is a common challenge in predictive maintenance datasets, where the quantity of the minority class is significantly smaller than the majority class. This imbalance can tend to form biased models. Various methods have been proposed to overcome this problem like Synthetic Minority Oversampling Techniques (SMOTE) and adaptive synthetic sampling (ADASYN). They are popular oversampling techniques that generate synthetic samples for the minority class. However, these methods may introduce noise or fail to capture the true data distribution. In this work, we employed Conditional Tabular GAN (CTGAN), a GAN-based approach specifically designed for generating synthetic samples in tabular data.

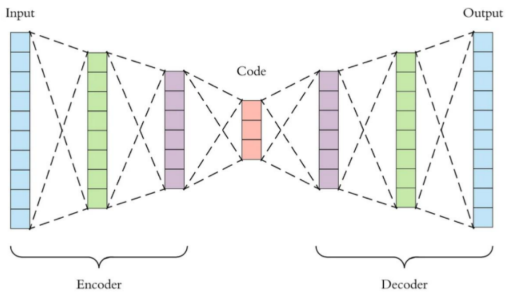


Fig. 2. AutoEncoder System

CTGAN has shown promising results in handling class imbalance by generating synthetic samples that closely resemble the minority class distribution. By combining the original and synthetic samples, we aim to create a balanced dataset that enables more accurate predictive modeling. A CT-GAN is a generator-discriminator-based mechanism where a generator functions to generate the synthetic data and the discriminator aims to identify the fake and real data. The addition of classification loss to conditional GAN and a unique conditional vector encoding that effectively encodes mixed variables and aids in handling highly skewed distributions for continuous variables are two essential components of CT-GAN. It works on minimizing the loss of the discriminator process and the generator process as described in equations 1 & 2 respectively.

$$Loss\_Disr(\delta_d) = -\log(o_1) - \log(1 - o_2) \tag{1}$$

$$Loss\_Gen(\delta_g) = -\log(o_2) \tag{2}$$

where :
o_1 = discriminators output probability for real sample and
o_2 = discriminators output probability for synthetic sample

3.2. Faulty product detection

After completion of the Data quality management phase, we proceed with the predictive analysis which is done through the RNET-50 model. Inspired by RESNET-50, RNET-50 uses residual blocks, which combine identity mappings and residual mappings to learn the difference between the input and the desired output. It heavily relies on convolutional layers for feature extraction and applies batch normalization and Rectified Linear Unit (ReLU) activation to enhance training and introduce non-linearity. It contains two main components: identity mapping and residual mapping as described in equations 3 & 4 respectively. It helps to understand whether the system would be faulty by passing the obtained quality management data to the RNET-50 system.

$$I(x) = x \tag{3}$$

$$R(x) = R'(x) - x \tag{4}$$

where :
x = input of the residual block and,
R'(x) = learned residual mapping

4. Performace Metrics & Evaluation

4.1. Experimental Setup

The proposed approach was implemented in Python using TensorFlow and the designed autoencoder model is implemented using Keras. The proposed framework is implemented and evaluated using Python 3 on a MacBook Pro having an Apple M1 Pro chip with a 10_core Central Processing Unit and 16_core Graphics Processing Unit, 16-GB RAM, and 1-TB SSD. The parameters for the autoencoder system and RNET-50 are described in Table 1.

4.2. Dataset Description

SECOM [12] dataset is used to evaluate the proposed approach contains 591 features, where one of the attributes defines whether the product is passing or failing the quality test. It consists of 1567 samples where every tuple is recorded after analyzing the product through the quality test using a 1567*591 matrix. Within 1567 tuples, 104 are determined as a 'fail' class (encoding 1) and the remaining as 'pass' (encoding 0). A detailed description of the dataset is described in Table 2.

4.3. Evaluation Metrics

The proposed framework was evaluated quantitatively using the metrics accuracy, precision, F-Score, recall, and AUC. The ratio of accurately classifying positive and negative samples is known as accuracy. The recall is defined as the proportion of correctly categorized positive instances to samples with positive examples. The ratio of accurately anticipated positive examples is known as precision. A rather thorough assessment index is the F-Score, which is the weighted harmonic average of precision and recall. True Positive Rate (TPR) and False Positive Rate (FPR) can be drawn on the Y-axis and the X-axis, respectively, to produce ROC curves. The region under the ROC curve is known

Table 1. Model description

Auto Encoder	No. of layer in encoder or decoder = 3, activation =‘relu’ (all), epochs= 64, batch_size = 32, optimizer = ‘adam’, metrics = ‘Mean absolute error’, loss = ‘mean absolute error’	
RNET-50	Conv1	7X7, 64, strides 2
	Conv2_x	3X3 max pool, stride 2 $\begin{bmatrix} 1X1, & 64 \\ 3X3, & 64 \\ 1X1, & 256 \end{bmatrix} X3$
	Conv3_x	$\begin{bmatrix} 1X1, & 128 \\ 3X3, & 128 \\ 1X1, & 512 \end{bmatrix} X4$
	Conv4_x	$\begin{bmatrix} 1X1, & 256 \\ 3X3, & 256 \\ 1X1, & 1024 \end{bmatrix} X6$
	Conv5_x	$\begin{bmatrix} 1X1, & 512 \\ 3X3, & 512 \\ 1X1, & 2048 \end{bmatrix} X3$
	Average pool, 100-d fc, softmax FLOPs 3.8X10	

Table 2. Dataset description

No. of features	590
No. of classes (pass & fail)	2
No. of ‘pass’ instances	1463
No. of ‘fail’ instances	104
Total no. of instances	1567

as AUC (area under curve). AUC may be used to evaluate the classifier’s performance quantitatively and intuitively.

$$Accuracy = \frac{e + b}{e + f + g + h} \tag{5}$$

$$Precision = \frac{e}{e + g} \tag{6}$$

$$Recall(TPR) = \frac{e}{e + h} \tag{7}$$

$$F - Score = \frac{2 * e^2}{2 * e^2 + e * (g + h)} \tag{8}$$

where:
e = true positive, f = true negative, g = false positive, h = false negative.

4.4. Results Analysis

This section presents the comparison of the proposed approach for the faulty product detection on the SECOM dataset. Figure 3 shows the comparison of selected feature reduction technique (PCA+AE) for data quality management with other existing feature extraction/reduction technique with the RNET-50 classifier, evaluating their performance in terms of precision, recall, and F-Score.

PCA + AE has the highest F-Score among all methods, indicating the best balance between Precision and Recall. It also has the highest Recall of 1.000, meaning it didn't miss any true positives (it found all relevant instances) and a precision of 0.917. Overall, PCA + AE has achieved outstanding results among the four methods. Whereas, information gain, although having the lowest F-Score, still performs decently. It has the lowest recall among all methods, indicating it missed more true positive instances than the others. Similarly, its precision is lower than PCA + AE, meaning it made more false positive errors. ANOVA and Chi-Squared both have high Recall (0.991), indicating they identified almost all true positives. Their F-Scores are close, but lower than PCA + AE, meaning they maintain a good, but not the best, balance between precision and recall. Their precision is slightly lower than PCA + AE, implying more false positive errors.

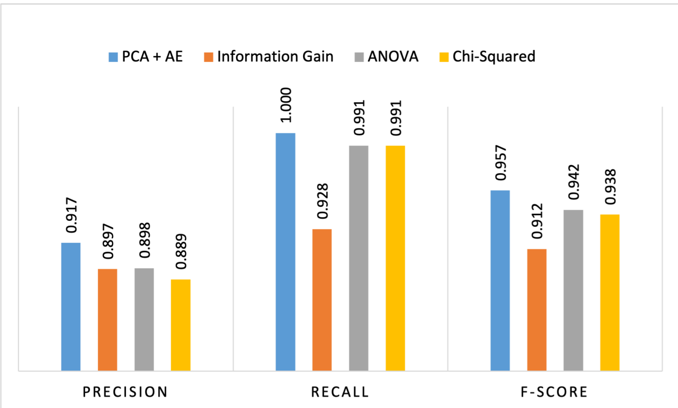
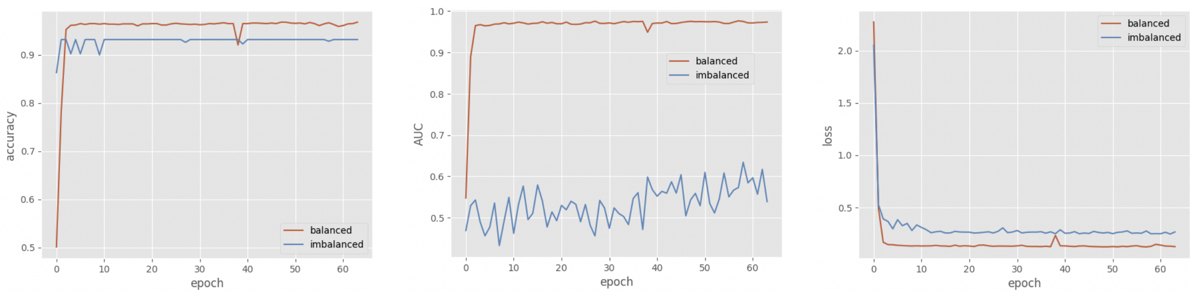


Fig. 3. Comparing selected feature reduction (PCA+AE) with other techniques with RNET-50 classifier



(a) (b) (c)

Fig. 4. Accuracy, AUC, and loss comparison with balanced and imbalance dataset

After selecting the features to further increase the performance, data balancing is performed using CTGAN on the PCA+AE selected feature data and RNET-50 is the classifier used.

Figure 4 shows the comparison of accuracy (a) and AUC (b), with balanced and imbalanced dataset. Results shows that after using CTGAN for data balancing, and increase in accuracy and AUC is seen with the balanced dataset in comparison to imbalance dataset. Moreover, Fig. 4 (c) also shows that the loss is also reduced when using the balanced dataset.

Furthermore, the proposed approach is compared with other ML and DL classifiers such as Multilayer Perceptron (MLP), Random Forest (RF), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Support Vector Machine (SVM) based on their precision, recall, F-Score, and accuracy. Figure 5 shows that RNET-50 outperforms the other classifiers with the highest values for precision, F-Score, accuracy, and perfect recall. This suggests it is most effective at correctly identifying positive cases (high Precision), retrieving all relevant instances (perfect recall), maintaining a superior balance between precision and recall (highest F-Score), and achieving the highest proportion of correct predictions (highest accuracy).

MLP demonstrates perfect recall and also performs well in precision with 0.919, leading to a high F-Score, just slightly lower than RNET-50 which is 0.930. This indicates the MLP model is also performing excellently, but the data quality management of the proposed approach effectively refines the data. This establishes the supremacy of our proposed approach. RF shows a slightly lower Recall with a value of 0.967 compared to RNET-50 and MLP, meaning it failed to identify a small fraction of the relevant instances. However, its precision is higher than MLP, resulting in fewer false positives. The F-Score and accuracy, although slightly lower, show that the RF classifier is still performing well overall.

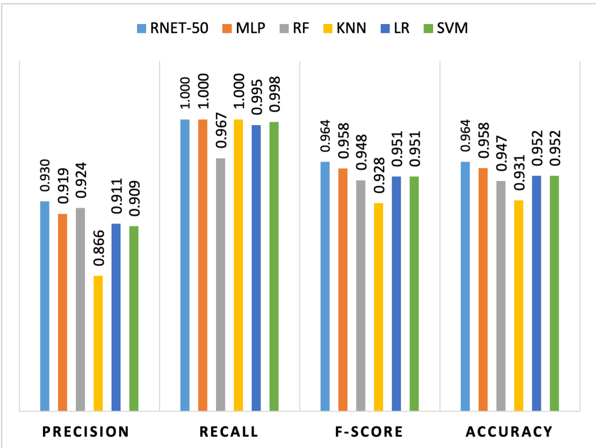


Fig. 5. Comparison of the proposed approach with other classifiers

KNN shows perfect recall but the lowest precision among the classifiers with a value of 0.866, implying it has a higher false positive rate. Despite identifying all the true positives, it incorrectly labels a significant number of negative instances as positive. This impacts the F-Score and accuracy, making KNN the least effective classifier in this comparison.

LR and SVM show similar performance, with nearly perfect recall and high precision. Both provide a good balance between precision and recall, as reflected in the identical F-Scores. Their Accuracy is slightly lower than MLP and RNET-50 but significantly higher than KNN. This implies that both classifiers, while not the top performers, still offer reliable performance.

Thus from the results it can be concluded that the proposed approach with the RNET-50 classifier outperforms the other classifiers and achieves the highest results. Therefore, proving its applicability to be used for data quality management and predictive maintenance.

These results indicate that the proposed approach could be used to aid various industries, such as manufacturing, energy, and transportation, to anticipate and prevent equipment failures by leveraging data analysis, sensor information, and advanced algorithms. Continuously monitoring the condition of machinery and systems, it could support maintenance, henceforth optimizing the timing of repairs or replacements. This approach reduces maintenance costs and maximizes overall operational efficiency, ultimately enhancing productivity and extending the lifespan of critical assets.

5. Conclusion & Future Work

The proposed data quality-driven predictive maintenance approach for faulty product detection relies on the combination of PCA+AE for feature reduction, CTGAN for handling imbalanced data, and RNET-50 as the classifier. The robust performance of this approach underscores its potential as a valuable asset for industry practitioners seeking to optimize product quality and operational efficiency. The ability to effectively minimize false positives and negatives, as indicated by the high precision and perfect recall, makes it an ideal tool for reducing waste and enhancing overall process optimization in the manufacturing industry. By effectively combining data quality-driven methods with advanced deep learning architectures, as demonstrated in this study, transformative changes in predictive maintenance strategies can be realized and result in achieving a high accuracy of 96.4%. This highlights the significance of data quality management in ensuring the reliability and efficacy of predictive maintenance approaches.

However, in the future we would like to focus on enhancing the accuracy of the predictive model and would work on improving the precision values. Even though the proposed approach effectively predicts the instances, it uses a highly complex model RNET-50. Therefore, we would also like to reduce the complexity of the model while maintaining the same system quality.

Acknowledgements

This publication has emanated from research supported in part by a grant from Science Foundation Ireland under Grant Number SFI/16/RC/3918 (Confirm), and also by a grant from SFI under Grant Number SFI 12/RC/2289-P2 (Insight). For the purpose of Open Access, the author has applied a CC BY public copyright licence to any author accepted manuscript version arising from this submission

References

[1] Young H. Chun. Improved method of estimating the product quality after multiple inspections. *International Journal of Production Research*, 54(19): 5686-5696, 2016.

[2] Tiago Zonta, Cristiano André Da Costa, Rodrigo da Rosa Righi, Miromar Jose de Lima, Eduardo Silveira da Trindade, and Guann Pyng Li. Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150: 106889, 2020.

[3] Nitesh Bharot, Mirco Soderi, Priyanka Verma, and John G. Breslin. Improving Product Quality Control in Smart Manufacturing through Transfer Learning-Based Fault Detection. *In 2023 IEEE International Conference on Smart Computing (SMARTCOMP), 2023*, pages 213-215, 2023.

[4] Seokho Kang, Eunji Kim, Jaewoong Shim, Wonsang Chang, and Sungzoon Cho. Product failure prediction with missing data. *International Journal of Production Research*, 56(14): 4849-4859, 2018.

[5] Nitesh Bharot, Mirco Soderi, and John G. Breslin. A Demo of Microservice for Customized Faulty Product Detection System in Smart Manufacturing. *In 2023 IEEE International Conference on Smart Computing (SMARTCOMP), 2023*, pages 180-182, 2023.

[6] Ionut Anghel, Tudor Cioara, Dorin Moldovan, Ioan Salomie, and Madalina Maria Tomus. Prediction of manufacturing processes errors: Gradient boosted trees versus deep neural networks. *In 2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC), 2018*, pages 29-36, 2018.

[7] Kittisak Kerdprasop, and Nittaya Kerdprasop. A data mining approach to automate fault detection model development in the semiconductor manufacturing process. *Int. J. Mech* 5(4): 336-344, 2011.

[8] Sathyan Munirathinam, and Balakrishnan Ramadoss. Predictive models for equipment fault detection in the semiconductor manufacturing process. *IACSIT International Journal of Engineering and Technology* 8(4): 273-285, 2016.

[9] Tayaba Abbasi, King Hann Lim, and Ke San Yam. Predictive maintenance of oil and gas equipment using recurrent neural network. *In Iop conference series: Materials science and engineering*, 495, pages 012067, 2019.

[10] Nurul Afiqah A. Majid, and Asan GA Muthalif. Mechatronics technology in predictive maintenance method. *In IOP Conference Series: Materials Science and Engineering* 260(1), pages 012006, 2017.

[11] Xiang Li, Li Wei, and Jianfeng He. Design and implementation of equipment maintenance predictive model based on machine learning. *In IOP Conference Series: Materials Science and Engineering* 466(1), pages 012001, 2018.

[12] Michael McCann, and Adrian Johnston. SECOM dataset UCI machine learning repository, 2008.