

RESEARCH ARTICLE

OPEN ACCESS

SMART-INSPECTION SYSTEM ON ASSEMBLY PROCESS OF PIN-THROUGH COMPONENTS USING MACHINE LEARNING

Carlos Americo de Souza Silva¹, Jorge Eduardo Santos Penedo², Edson Pacheco Paladini³ and Waldir Sabino da Silva Junior⁴

^{1,3} Industrial and Systems Engineering Department, University Federal of Santa Catarina – UFSC, Florianópolis, Brazil.

^{2,4} Electronic and Telecommunication Department, University Federal of Amazonas – UFAM, Manaus, Brazil.

¹<https://orcid.org/0000-0003-2632-7717> , ²<https://orcid.org/0000-0003-0421-0569> , ³<https://orcid.org/0000-0002-8651-0970> ,
⁴<https://orcid.org/0000-0003-3095-0042> 

Email: camericoss@gmail.com, jorjeh.penedo@gmail.com, edson.paladini@ufsc.br, waldirjr@ufam.edu.br

ARTICLE INFO

Article History

Received: January 02, 2025

Revised: January 20, 2025

Accepted: January 25, 2025

Published: February 28, 2025

Keywords:

Support vector machine,
Defect classification,
Machine Learning,
K-Nearest neighbor,
Decision tree.

ABSTRACT

This paper proposes using machine learning techniques to implement a failure mode classifier for automatic fail classification in pin-through hole (PTH) connector terminals in printed circuit boards (PCB). The Support Vector Machine (SVM), K-nearest neighbor (KNN), and Decision Tree (DT) algorithms were used. It was evaluated using a dataset of real images from manufacturing multimedia centers for the algorithm training phase. Subsequently, it thoroughly evaluated the results of the metrics obtained from each trained model. The main objective is to select the model with the best precision in predicting two failure modes to be implemented at the automotive factory and improve the inspection phase to reduce the defect and rework rates. The failure mode classifier trained with the SVM algorithm obtains the best precision, with an accuracy of 99% in predicting the dataset of tested images. KNN and DT achieved 78% and 79% accuracy, respectively, but DT was unstable. The final decision was to implement the SVM algorithm that obtained the best accuracy in decision-making for the failure modes evaluated in the research.



Copyright ©2025 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Printed circuit boards (PCBs) are essential in the manufacturing of electronic devices. In recent years, the demand for more sophisticated products with more embedded functions has made PCBs more complex, requiring higher quality and the application of lean thinking theory in production lines [1].

In manufacturing, the search for defect-free products has demanded more sophisticated inspection methods [2], using methodologies and algorithms capable of extracting knowledge from data [3]. PCB inspection is a crucial process to ensure the reliability and quality of the product before it is made available to the end consumer. Inspection is often performed visually by human operators, which can result in variations in the classification of defects due to physical and emotional inconsistencies of each operator [4],[5]. This has led industries to seek more efficient inspection methods to identify defects in the early stages of production [6].

Automatic Optical Inspection (AOI) has been used in industry to assist in identifying defective components in PCBs [7].

AOI systems generally employ defect inspection methods by scanning the board and analyzing it using techniques such as local feature matching with a standard image [8] and morphological image comparison to detect defects, achieving excellent results. However, problems with reflective materials can cause false failures [9].

With the sophistication and miniaturization of components inserted in PCBs, the challenges for fault detection become increasingly complex [10]. Detecting the absence of terminal projections and recognizing components and their similarities are complex tasks by manual visual inspection [11]. This increasingly requires traditional image classification algorithms and convolutional neural network models for defect detection [9]. Studies based on automatic visual inspection for detecting PCB faults through Machine Learning [7] and convolutional neural networks [8] have gained significant space and attention within the scientific community in recent years. According to [1], several methods have been proposed to detect and classify a variety of defects in PCBs. These methods are increasingly used in industry for decision-making, enabling the transformation of traditional

manufacturing to Industry 4.0 [12],[13],[14]. This article proposes methods to detect and classify defects without the projection of power connector terminals using machine learning algorithms. The analyzed PCB dataset was collected from a real production line of an industry installed in the Manaus Industrial Pole located in Brazil. Subsequently, the accuracy of the best method for implementation in detecting the absence of projection of pins will be analyzed.

II. THEORETICAL FRAME

The enumerations of citations in the body of the article must be sequenced in the order in which they appear, according to the example shown below.

The sequence of actions is structured so that the dynamics of the image dataset classification include control mechanisms capable of compensating for possible disturbances. In this way, the classifier can generate a correct output, even if there are interferences in its learning process [15]. This is achieved by comparing the actual prediction values (output) with the input values (test images).

Control systems are physical models that show the dynamics of a system and are usually composed of blocks that can be analyzed mathematically [16]. The block diagram of the Control System for classification is shown in Figure 1.

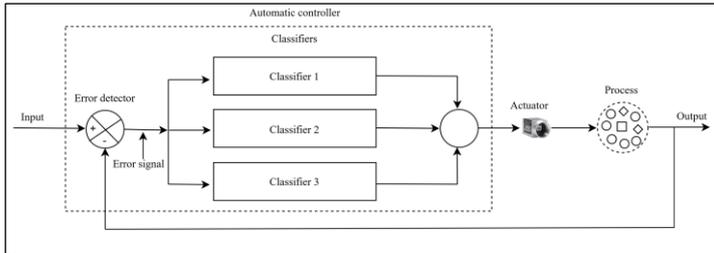


Figure 1: System diagram. Source: Authors, (2025).

The dynamics of the classification system start with the reference data. To further perform the classification process of the model for each machine learning algorithm to compare the output data with the reference image to verify the failure mode. The acting error signal provides feedback to the system to reduce the error and prevent external changes from affecting the system's behavior. Then it will obtain information from the best failure mode classifier to predict the image classification for deciding the OK or NOK state of the inspected PCB.

II.1 BIBLIOMETRIC ANALYSIS ON MACHINE LEARNING AND SMART INSPECTIONS FOR PTH COMPONENTES

A bibliometric analysis was performed to analyze the dynamics of research evolution, considering machine learning algorithms used for quality inspection of the manufacturing process in the context of Industry 4.0 and Quality 4.0. The final search was realized in December 2024 on the Scopus database and Web of Science Database with the terms "Pin Through-hole" or "PCB" and "Machine Learning" or "SVM" or "KNN" or "Decision Tree" or "smart-vision inspection", applied in the titles, abstracts, and keywords of the articles. For the portfolio, only articles with publications in English were considered.

Based on the adopted methodology, 220 articles were found in qualitative synthesis; from the articles selected for content analysis from the timespan 2019 to 2024, quantitative analyses

were developed with the Bibliometrix tool of the R Studio® software, following the procedure developed by [17].

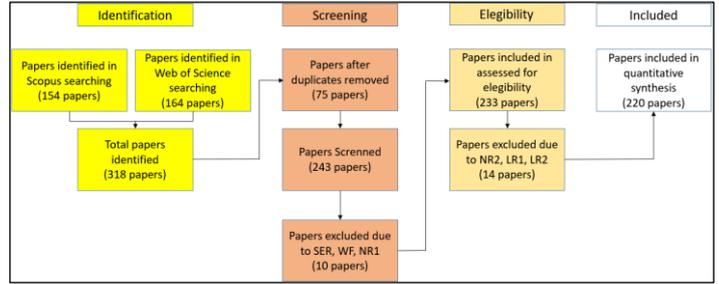


Figure 2: Prisma methodology. Source: Authors, (2025).

Figure 2 shows the PRISMA methodology used on the research [18].

The final search string is presented as follows:

String to Scopus database- TITLE-ABS-KEY (“Pin Through-hole” OR “PCB”) AND (“machine learning” OR “SVM” OR “KNN” OR “Decision Tree” OR “smart-vision inspection”)

String to Web of Science database - TS= (“Pin Through-hole” OR “PCB”) AND (“machine learning” OR “SVM” OR “KNN” OR “Decision Tree” OR “smart-vision inspection”)

Figure 3 shows the temporal evolution of publications in the selected portfolio. 2016 was the first paper in which a publication appeared in an indexed journal in the considered databases. [19] presented a system with a neural network to predict the skew factor of PCB laminate designs. [20] described a model to predict the production cycle time of high-mixed PCB based on machine learning methods.

Since 2019, the number of publications has grown consistently. Between 2019 and 2024, publications increased by 162%. This analysis shows the growing interest in smart inspection using machine learning algorithms.

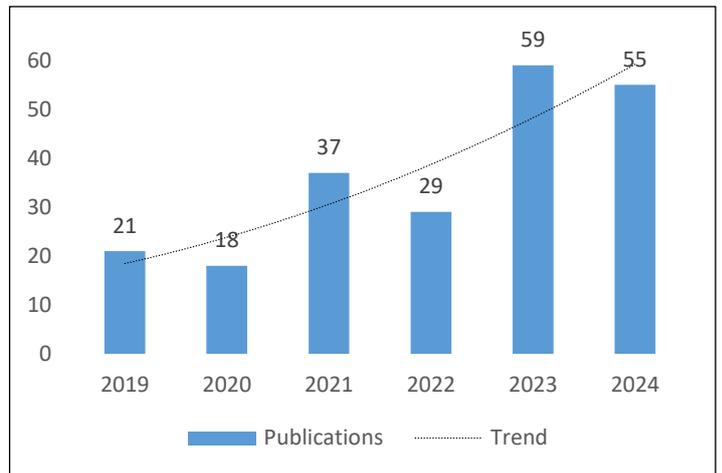


Figure 3: Temporal evolution of publications. Source: Authors, (2025).

Figure 4 shows the main countries with associated studies in the research area, categorized by publications authored by only one country (in blue) and several countries (in red). China presents great performance with 60 publications, followed by USA with 46, Germany with 19, India with 17, and the other countries with 7 or fewer publications.

statistical learning theory is dimension, which defines the most significant number of points that can be separated in various ways [27].

II.3.1 SEPARATION HYPERPLANE

By using the structural risk minimization principle to identify the optimal hyperplane to maximize the margin of the closest examples, SVMs create a series of hyperplanes whose dimension boundaries can be processed [27]. One example is the patterns for linearly separable classes, in which the class y_i can only receive values +1 and -1 [28]. Equation (2) illustrates the decision surface of a hyperplane to perform class separation.

$$(\omega \cdot x) + b = 0, \omega \in \mathbb{R}^n, b \in \mathbb{R} \quad (2)$$

The ω gives the adjustable weight vector and b gives the threshold. Illustrated in equation (3).

$$\begin{cases} (\omega \cdot x) + b \geq 0, \text{ para } y_i = 1 \\ (\omega \cdot x) + b < 0, \text{ para } y_i = -1 \end{cases} \quad (3)$$

The closest data point is called the separation margin. Figure 8 illustrates the optimal hyperplane obtained using the maximum class separation margin.

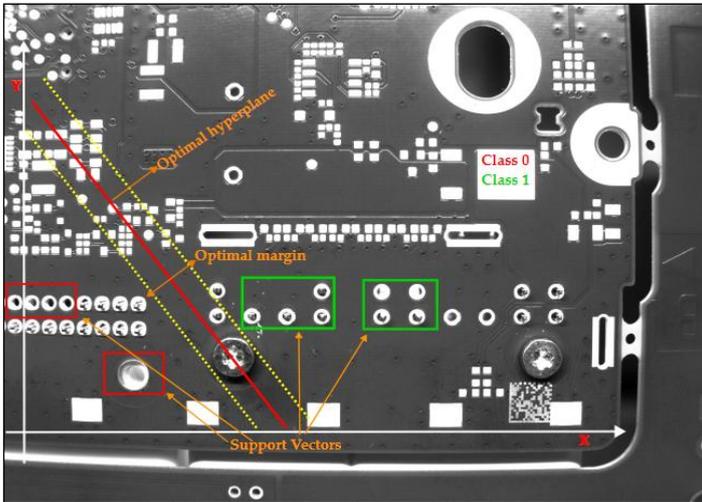


Figure 8: Definition of the optimal hyperplane. Source: Authors, (2025).

Figure 8 illustrates the maximum margin separator, represented by the solid red line, and the margins, represented by the dashed lines. The support vectors are the holes highlighted by the dashed circle and the connector terminals highlighted by the green squares closest to the separator.

II.4 K-NEAREST-NEIGHBOR - KNN

The k -nearest neighbors (KNN) classifier is a classical classification algorithm that uses nonparametric methods. Its basic concept is determining class labels based on their k nearest neighbors [29]. KNN classifies the K points of the closest training set to find K elements with the smallest distance. Figure 9 illustrates the definition of KNN.

Figure 9 illustrates the representation of the data already trained with its classifications previously defined for the Pin and Hole classes. In summary, the distance of the new object will be determined by defining the k neighbors but closest to the category. When K is defined as having three or four occurrences, it will be classified as Pins since two of the three closest neighbors are Pins.

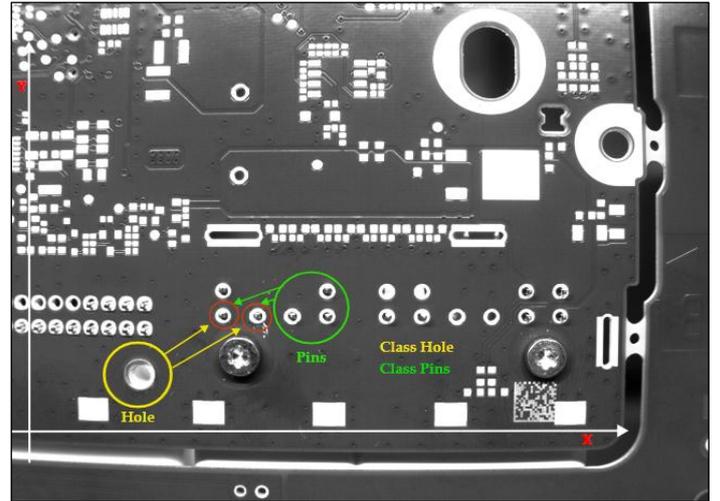


Figure 9: Definition of the KNN. Source: Authors, (2025).

By approximating the k values, the distance between points x and y is calculated using the Euclidean or Manhattan distance [30],[31]. The calculation of the space between the distances of the objects is demonstrated by the equations (4 and 5).

The Euclidean distance between point x and y is given by equation (4):

$$d(x, y) = \sqrt{(x_1, y_1)^2 + (x_2, y_2)^2 + \dots + (x_n, y_n)^2} \quad (4)$$

The smaller distance between points x and y was determined by measuring the Euclidean distance.

The Manhattan distance between points x and y is given by the equation (5):

$$(x, y) = |x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n| \quad (5)$$

The sum of the absolute differences between points x and y in all dimensions of space will determine the distance to Manhattan.

II.5. DECISION TREE

Nonparametric models for data classification and prediction based on supervised learning are known as decision trees [32]. Decision trees use the splitting strategy, which means that the training data set is divided into several smaller subsets until one of the subsets is of the same class or until the class is the predominant one [33].

The decision tree is constructed from the compactly organized data, which recursively classifies new examples. This creates a data structure [32], corresponding to a node or leaf as a class or decision node that can test several attributes. When each result creates a new subtree [33]. A decision tree is shown in Figure 10.

The decision tree nodes are represented by the NOK attribute and distributed in the tree according to their level. The segments that define the nodes to which each attribute belongs are used to test the values. The attributes of the categorical type are validated using the equal sign, as shown in Figure 10 by the white circles, where each circle is the attribute. Decision trees use algorithms to identify the value assigned to the node and represent quantitative values in a specific range of values. These algorithms also determine the branches' division into subsamples comparable to the variable resulting from the classification [33].

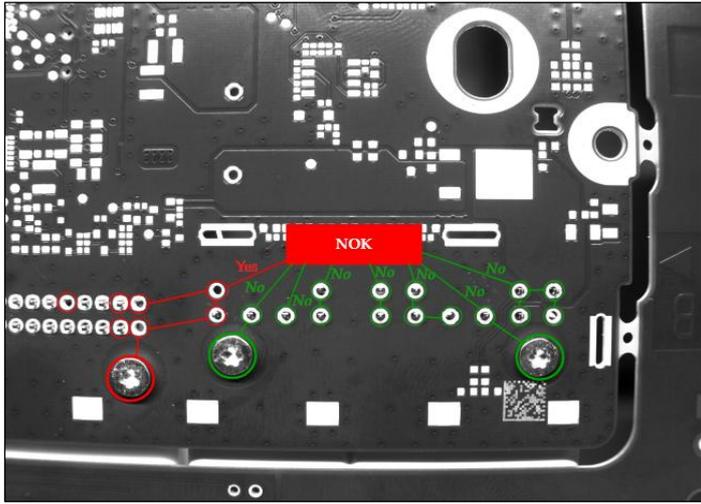


Figure 10: Decision tree model.
Source: Authors, (2025).

III. RESEARCH STAGE 1 - CLASSIFIERS

In this section, the experiments and results used to carry out this work are presented. Accuracy comparisons were made between the machine learning algorithms applied in the training and classification of the database of failure modes in PCBs.

III.1 IMAGE PRE-TREATMENT

The two classes and the number of images for training, testing, and validation of the classifiers are presented in Table 1.

Table 1: Definition of classes for classification.

Classes	Training	Validation	Testing
NOK	380	95	95
OK	380	95	95
Totals of Imagens	760	190	190

Source: Authors, (2025).

The images were captured in grayscale and used for supervised training. Since the process involves object detection, the images need to be cataloged. Figure 11 demonstrates the defined areas of interest.

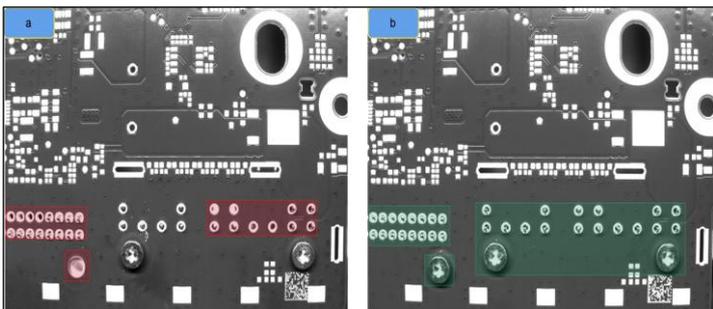


Figure 11: Definition of the image area of interest.
Source: Authors, (2025).

The standardization of classes was implemented because the terminals and screws do not present differences in shape or color. Each class uses images with a resolution of 1280 x 720 pixels for training and validation. The definition of the classes is presented in Figure 12.

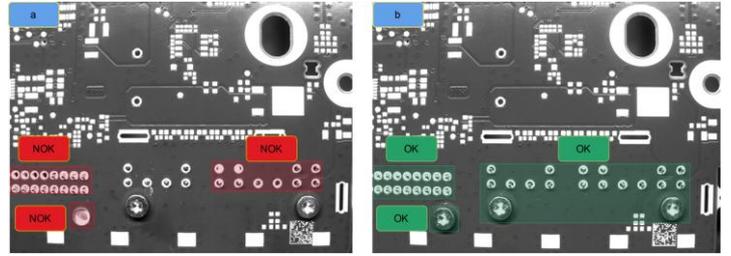


Figure 12: Definition of classes.
Source: Authors, (2025).

III.2 FAILURE MODES DEFINED IN RESEARCH

The acquisition of the dataset to detect the absence of projection of the connector terminals and classification of failure modes were obtained on the production line through the capture of 1140 images of black and white PCBs with different failure modes, using criteria from Failure Mode and Effects Analysis (FMEA) [34], [35],[36],[37],[38], [39].

The creation of only two classes for failure mode detection was considered to represent PCBs without failures and PCBs with failures, as mentioned in Table 2.

Table 2: Failures Modes.

Items	Defect Type	Pictures	Description	Specification
1	Missing Pin and Screw		No evidence of pins and screw	Nok
2	Missing Pin and Screw		No evidence of pins projection and missing screw	Nok
3	Missing Screw		Missing screws (2x)	Nok
4	Missing Pin, Screw and Connector		No evidence of pins projection and missing screw and connector	Nok
5	Missing Pin		There is no evidence of pin projection	Nok
6	Golden Sample		Solder splashes on metal component surfaces impact form, fit or function.	Ok

Source: Authors, (2025).

IV. EXPERIMENTS

During the experiments, three machine learning algorithms were used to build a failure mode detection model, to classify samples, and to determine the class corresponding to the failure mode:

I. To create the failure mode detector, we used the support vector machine (SVM) algorithm in the first experiment with a linear kernel since the 600 images in the database are linearly separable. The classifier performs the classification of each data sample to classify each training sample into its corresponding NOK or OK class. After completion, the classifier will be able to make new predictions of failure modes in the latest samples of PCB images.

II. The decision tree algorithm was used to perform the second experiment. The algorithm created a failure mode detector from an empty tree, iteratively searching for the best attribute to divide the data. The algorithm used the 600 images in the database to achieve this goal. If the data were divided and belong to the same class, a leaf will be created with the NOK or OK label. After training, the classifier predicts the class of new samples of PCB images.

III. The KNN algorithm was used in the third experiment. Data was given to measure the test point of a specific value or label to predict the training set, and then the nearest k points were selected to make the class prediction based on the label of its neighbors to create the failure mode detector. The cross-validation method was used to select the k parameter. The algorithm will have access to the 600 images in the database. If the points in the classification belong to the same class, the nearest neighbor will be labeled as NOK or OK. After training, the classifier will be able to make the predictions.

The experiments were performed using the diagram of the classification methods suggested in Section 2. The objective of the experiments is to verify the efficiency of the algorithms in predicting failure modes. After the training, the metrics generated during the training will be compared to define which models will be implemented in the production line.

V. VALIDATION OF FAILURE MODE DETECTION

The confusion matrix is a popular method for evaluating machine learning algorithm metrics, such as precision, accuracy, and ROC curves [40],[41], whose values are found through the confusion matrix illustrated in Table 3.

Table 3: Confusion Matrix

		Is there an image failure mode?	
		True	False
Was the algorithm detecting the failure mode in the image?	True	True Positive (TP)	False Negative (FN)
	False	False Positive (FP)	True Negative (TN)

Source: Authors, (2025).

The variable (TP) corresponds to the number of failure modes classified as good, as shown in the confusion matrix in Table 3. The variable (TN) corresponds to the approved failure modes. The number of failures (OK) classified as non-failures (NOK) is represented by the variable (FN). In contrast, the number of failures (NOK) classified as failures (OK) is represented by the variable (FP). The variables (TP and TN) indicate the hits that the classifier obtained in its result, while the variables (FN and FP) indicate the errors caused in the classification of the classifier.

Accuracy, sensitivity, and specificity were used to measure the performance of the classifiers. The adequate number of positive and negative samples represents the precision of the model.

VI. RESULTS AND DISCUSSIONS

The paper's main objective is to obtain the accuracy of the best machine learning algorithms for developing a failure mode classifier to perform automatic visual inspection of the projection of the terminals of PTH components.

VI.1 SVM

The experiment metrics, obtained from the SVM classification, are demonstrated through the confusion matrix and

learning curve generated after training the failure mode model, according to Figure 13.

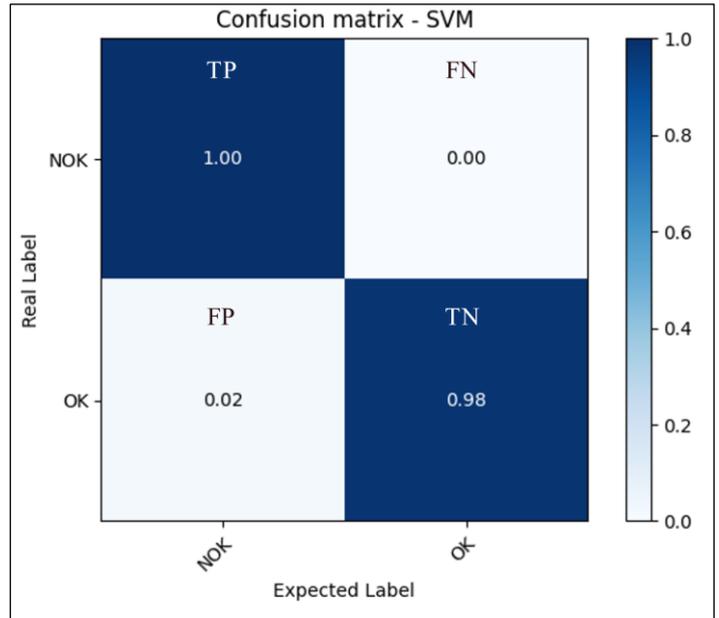


Figure 13: Definition of classes for SVM.

Source: Authors, (2025).

The TP variable demonstrates the accuracy of the failure mode classification of PCBs classified as NOK with a precision of 1.00. The TN variable represents the classification accuracy of OK PCBs with a precision of 0.98. The FP represents the number of OK PCBs classified as NOK with a precision rate of 0.02, demonstrating a small error in the prediction. The FN demonstrates the number of NOK PCBs classified as OK with a precision of 0.00, demonstrating that the classifier did not make a mistake in this prediction.

The classifier performance metrics were obtained using the confusion matrix data of the trained SVM. The most used metrics for evaluating machine learning models are learning and ROC curves, accuracy, specificity, and sensitivity [40]. Figure 14 illustrates the results of the SVM classifier metrics.

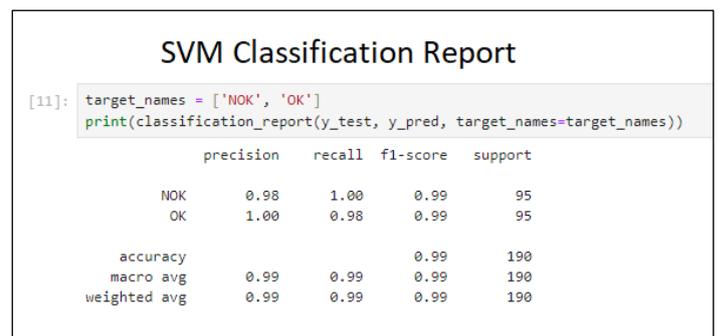


Figure 14: SVM model evaluation metrics.

Source: Authors, (2025).

The accuracy of the model reflects its performance during training and learning. The accuracy calculated is the total number of correct answers divided by the total number of images in the database, demonstrating the model's ability to make correct predictions. The accuracy of the SVM was 99%. Precision considers only true positive values, preventing false positive values from introducing biased errors in the result. The recall metric indicates the frequency with which the image is correctly identified

as belonging to a given class. The f1-score, the harmonic mean between precision and recall, evaluates the quality of the model's training. This metric is fundamental in imbalanced datasets. Figure 15 illustrates the accuracy of the learning curve when testing images classified by the linear SVM algorithm.

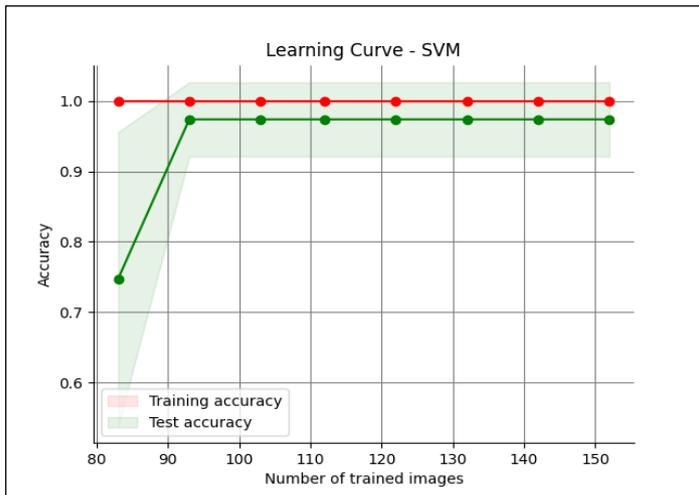


Figure 15: Learning curve for SMV. Source: Authors, (2025).

Figure 15 shows the accuracy of the model's learning curve. It is noticeable that the training accuracy increases with the number of images used in the algorithm. As we approach 93 tested images, it is evident that the accuracy has increased, remaining consistent and stable, with an accuracy of 99% at the end of training.

VI.2 KNN

The KNN algorithm was used in the second experiment to train the failure modes. In this scenario, the same database was used under the same conditions and quantity mentioned in the first experiment. Figure 16 illustrates the confusion matrix generated after completing the KNN training.

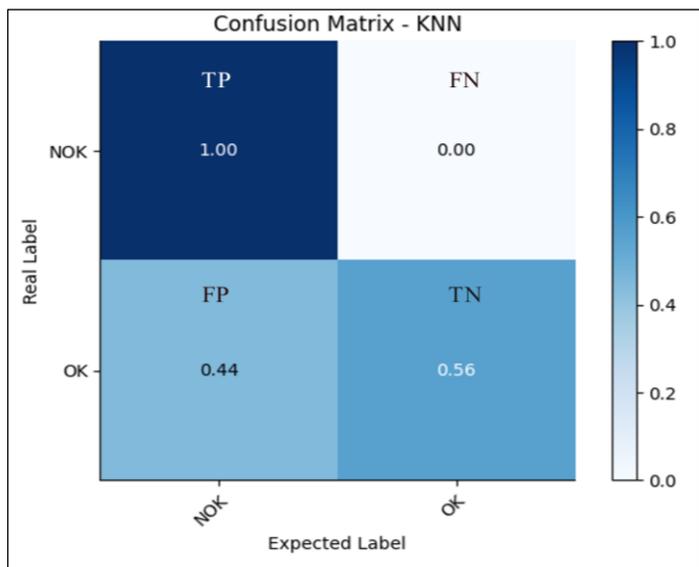


Figure 16: Definition of classes for KNN. Source: Authors, (2025).

The variable (TP) represents the number of PCBs classified as NOK with an accuracy rate of 1.00. The variable (TN) represents the PCBs approved OK, with an accuracy of 0.56. The variable (FP) represents the total number of PCBs OK and classified as

NOK, with an accuracy of 0.44. Moreover, the variable (FN) represents the number of PCBs classified as NOK and classified as OK, with an accuracy rate of 0.00. The results of the metrics are illustrated in Figure 17.

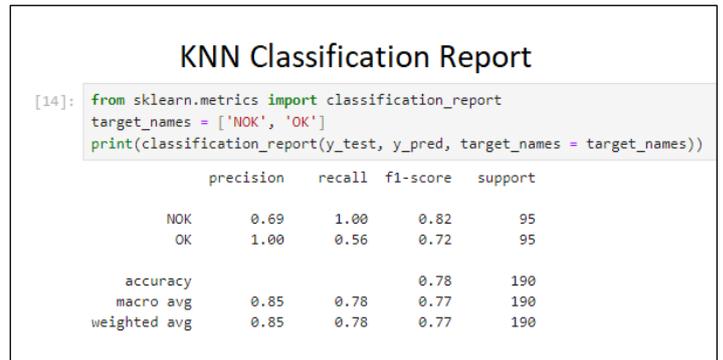


Figure 17: KNN model evaluation metrics. Source: Authors, (2025).

In the model's accuracy of the KNN was 78%, the precision of values for NOK failed PCBs was 0.69, and for OK PCBs, it was 1.00; only the true positive values were used. In the recall, NOK PCBs had 1.00, while OK PCBs had 0.56. This demonstrated the frequency of an image in a specific class. The final score of the model training can be seen in the f1-score metric, where NOK PCBs had 0.82 and OK PCBs were 0.72. Figure 18 shows the training and testing learning curve accuracy of images classified by KNN.

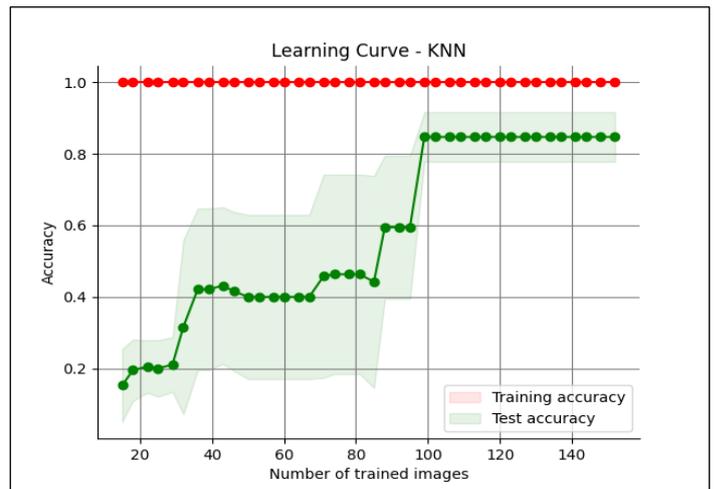


Figure 18: Learning curve for KNN. Source: Authors, (2025).

Figure 18 shows the accuracy of the learning curve of the model in the second experiment, where the accuracy of the training increases with the number of images used in the algorithm training. The stability of the model is noticeable after 100 images were tested, remaining stable and constant. At the end of the training, the accuracy was 78%.

VI.3 DECISION TREE

The third experiment used the decision tree algorithm to perform failure mode training. The dataset was used in the same quantity and conditions as in the second experiment. The confusion matrix created after the decision tree algorithm training was completed is shown in Figure 19.

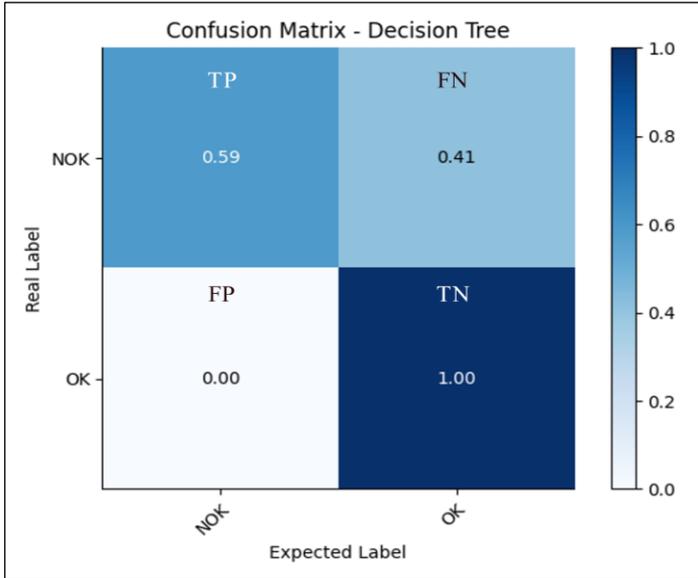


Figure 19: Definition of classes for DT. Source: Authors, (2025).

The variable (TP) represents the number of PCBs classified as NOK with an accuracy rate of 0.59. The variable (TN) represents the PCBs approved OK, with an accuracy of 1.00. The variable (FP) represents the total number of PCBs OK and classified as NOK, with an accuracy of 0.00. And the variable (FN) represents the number of PCBs classified NOK and classified as OK, with an accuracy rate of 0.41. The results of the metrics are illustrated in Figure 20.

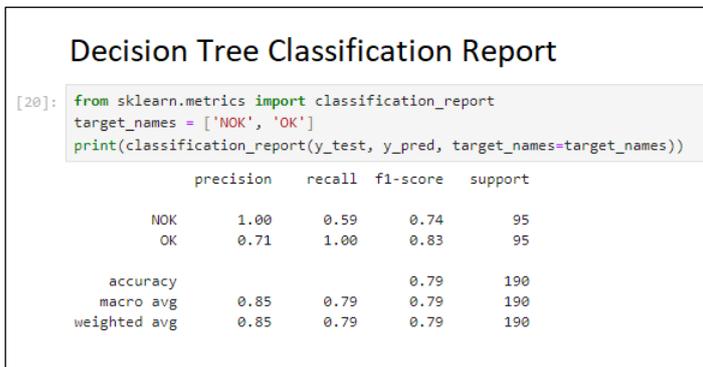


Figure 20: Decision tree model evaluation metrics. Source: Authors, (2025).

In the model, the accuracy of the DT was 79%, the precision of values for NOK failed PCBs was 1.00, and for OK PCBs, it was 0.71; only the true positive values were used. In the recall, NOK PCBs had 0.59, while OK PCBs had 1.00. This demonstrated the frequency of an image in a specific class. The final score of the model training can be seen in the f1-score metric, where NOK PCBs were 0.74 and OK PCBs were 0.83 on the final classification report.

Figure 21 shows the accuracy of the training and testing learning curve of images classified by DT. The training accuracy varies throughout the training process as the number of images the algorithm processes increases. The model demonstrates instability, reaching a final accuracy of 79%.

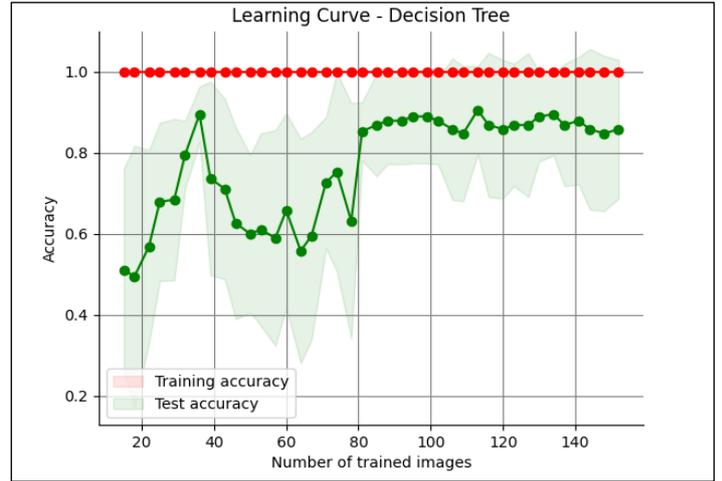


Figure 21: Learning curve for DT. Source: Authors, (2025).

Table 4 presents the results of the classifier metrics. The accuracy of the classification of failure modes by the SVM algorithm was the best. However, the classification performed with the KNN and DT algorithms did not accurately identify the failure modes.

Table 4: Summary of model classification metrics.

Algorithm	Class	Precision	Recall	f1-score
SVM	NOK	0.98	1.00	0.99
	OK	1.00	0.98	0.99
KNN	NOK	0.69	1.00	0.82
	OK	1.00	0.56	0.72
DT	NOK	1.00	0.59	0.74
	OK	0.71	1.00	0.83

Source: Authors, (2025).

The performance of the classification models created in this study was evaluated using the ROC curve. Figure 22 shows the relationship between true and false positive rates at various decision thresholds. This allows us to determine the best-performing area under the curve (AUC) in classifying failure modes. The SVM showed the best performance, with an AUC of 1.0, indicating that the model has excellent accuracy. On the other hand, the KNN and DT algorithms had an average AUC of just over 0.5, indicating a tendency towards misclassification and unsatisfactory performance.

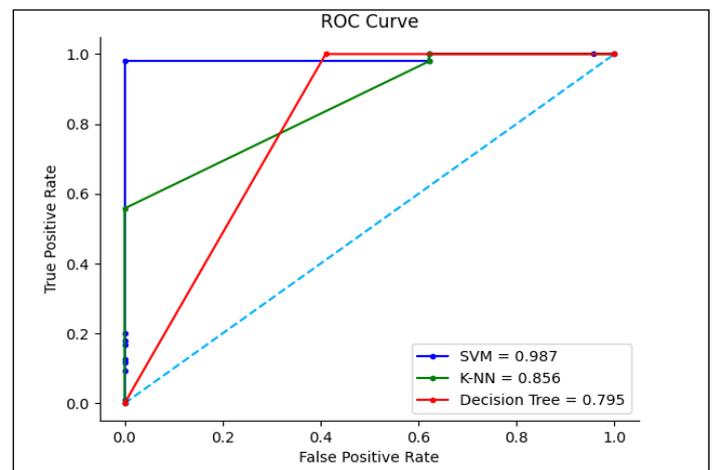


Figure 22: ROC curve of classification models. Source: Authors, (2025).

VII. CONCLUSIONS

The industrial process depends on detecting failure modes in the production of electronic equipment, especially in visual inspection, which is still performed manually by human operators. It is common for this process to present failures due to factors such as fatigue and emotional problems of the human operator, which can affect the inspection result. The industry has sought to incorporate automatic visual inspection systems into its processes to solve these problems.

The experiment carried out with the failure mode classifier, trained using the decision tree algorithm, aimed to categorize the images from the PCB database into the classes "NOK" (non-conforming) and "OK" (conforming). During the process, the model was evaluated for its ability to distinguish between the classes, but the results were unsatisfactory. The final accuracy obtained was 79% in the images tested, indicating poor performance, with significant errors in classification and prediction. This limitation is evidenced in Figure 20, where the learning curve demonstrated a marked instability throughout the training, suggesting that the model could not adequately generalize the failure patterns, which compromised its effectiveness in classification tasks.

The experiment used a failure mode classifier based on the K-Nearest Neighbors machine learning algorithm. After training, the model presented an accuracy of 78% in the tested images, which evidenced the presence of significant errors in classification and prediction. Figure 17 illustrates the model's performance, demonstrating in the learning curve the instability of accuracy in both the test and training data throughout the classification process. This instability suggests that the KNN model faced difficulties in correctly generalizing the failure patterns, compromising its ability to classify the failure modes accurately and reliably.

In the experiment conducted with the SVM (Support Vector Machine) algorithm for classifying failure modes in images from the PCB database, the objective was to predict the "NOK" and "OK" classes. After training, the model achieved an impressive accuracy of 99% on the tested images, indicating no errors in classification and prediction. Figure 14 corroborates these results, showing that the learning curve remained stable, both in the training and testing data. This stability throughout the classification process confirms the high effectiveness of the SVM, evidencing its robust generalization capacity and accuracy. The high performance of the SVM model suggests that it is highly suitable for implementation in a real production environment, where reliability in fault detection is crucial for the quality of the final product. Furthermore, these results highlight the potential of the SVM as a viable and efficient solution to classification challenges in industrial systems, making it a recommended choice for applications that require accuracy and consistency in visual data analysis.

VIII. AUTHOR'S CONTRIBUTION

Conceptualization: Carlos Americo de Souza Silva and Jorge Eduardo Santos Penedo.

Methodology: Carlos Americo de Souza Silva, Jorge Eduardo Santos Penedo, Edson Pacheco Paladini and Waldir Sabino da Silva Junior.

Investigation: Carlos Americo de Souza Silva and Jorge Eduardo Santos Penedo.

Discussion of results: Carlos Americo de Souza Silva, Jorge Eduardo Santos Penedo, Edson Pacheco Paladini and Waldir Sabino da Silva Junior.

Writing – Original Draft: Carlos Americo de Souza Silva and Jorge Eduardo Santos Penedo.

Writing – Review and Editing: Carlos Americo de Souza Silva and Jorge Eduardo Santos Penedo.

Resources: Carlos Americo de Souza Silva and Jorge Eduardo Santos Penedo.

Supervision: Edson Pacheco Paladini and Waldir Sabino da Silva Junior.

Approval of the final text: Carlos Americo de Souza Silva, Jorge Eduardo Santos Penedo, Edson Pacheco Paladini and Waldir Sabino da Silva Junior.

IX. REFERENCES

- [1] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine Learning for industrial applications: A comprehensive literature review," *Expert Systems with Applications*, vol. 175, 2021, doi: 10.1016/j.eswa.2021.114820.
- [2] M. König and H. Winkler, "Investigation of assistance systems in assembly in the context of digitalization: A systematic literature review," *J Manuf Syst*, vol. 78, pp. 187-199, 2025/02/01/ 2025, doi: <https://doi.org/10.1016/j.jmsy.2024.11.015>.
- [3] T.-C. Tsan, T.-F. Shih, and C.-S. Fuh, "TsanKit: artificial intelligence for solder ball head-in-pillow defect inspection," (in en), *Machine Vision and Applications*, vol. 32, no. 3, p. 66, 2021/05// 2021, doi: 10.1007/s00138-021-01192-8.
- [4] S. Wang and R. J. Jiao, "Smart In-Process Inspection in Human–Cyber–Physical Manufacturing Systems: A Research Proposal on Human–Automation Symbiosis and Its Prospects," *Machines*, vol. 12, no. 12, doi: 10.3390/machines12120873.
- [5] M. Castellani, S. Otri, and D. T. Pham, "Printed circuit board assembly time minimisation using a novel Bees Algorithm," (in en), *Computers & Industrial Engineering*, vol. 133, pp. 186-194, 2019/07// 2019, doi: 10.1016/j.cie.2019.05.015.
- [6] S. Wenjie, Z. Zhijiang, L. Han, and P. Libo, "Research on Visual Inspection Method and Instrument of Solder Joint," (in en), *IFAC-PapersOnLine*, vol. 55, no. 3, pp. 131-136, 2022 2022, doi: 10.1016/j.ifacol.2022.05.023.
- [7] S. S. Zakaria, A. Amir, N. Yaakob, and S. Nazemi, "Automated Detection of Printed Circuit Boards (PCB) Defects by Using Machine Learning in Electronic Manufacturing: Current Approaches," (in en), *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 767, no. 1, p. 012064, 2020/02/01/ 2020, doi: 10.1088/1757-899X/767/1/012064.
- [8] H. Lu, D. Mehta, O. Paradis, N. Asadizanjani, M. Tehranipoor, and D. L. Woodard, "FICS-PCB: A Multi-Modal Image Dataset for Automated Printed Circuit Board Visual Inspection," (in en), 2020/07/17/ 2020. [Online]. Available: <https://eprint.iacr.org/2020/366>.
- [9] Z. Liu and B. Qu, "Machine vision based online detection of PCB defect," (in en), *Microprocessors and Microsystems*, vol. 82, p. 103807, 2021/04// 2021, doi: 10.1016/j.micpro.2020.103807.
- [10] R. S. Peres, J. Barata, P. Leita, and G. Garcia, "Multistage Quality Control Using Machine Learning in the Automotive Industry," *IEEE Access*, vol. 7, pp. 79908-79916, 2019 2019, doi: 10.1109/ACCESS.2019.2923405.
- [11] J. Shen, N. Liu, and H. Sun, "Defect detection of printed circuit board based on lightweight deep convolution network," (in en), *IET Image Processing*, vol. 14, no. 15, pp. 3932-3940, 2020/12// 2020, doi: 10.1049/iet-ipr.2020.0841.
- [12] H. M. Ahmad and A. Rahimi, "Deep learning methods for object detection in smart manufacturing: A survey," *J Manuf Syst*, vol. 64, pp. 181-196, 2022 2022, doi: <https://doi.org/10.1016/j.jmsy.2022.06.011>.
- [13] C. A. d. S. Silva, E. P. Paladini, J. E. S. Penedo, and W. S. d. S. Júnior, "Digital and Smart Production Using Simulation Systems to Improve the Manufacturing Performance," in *XVI Simpósio Brasileiro de Automação Inteligente*, 2023, vol. 1, no. 2, pp. 910-916, doi: 10.20906/SBAI-SBSE-2023/3915. [Online]. Available: <https://dx.doi.org/10.20906/sbai-sbse-2023/3915>
- [14] D. Mehta and N. Klarmann, "Autoencoder-Based Visual Anomaly Localization for Manufacturing Quality Control," *Machine Learning and Knowledge Extraction*, vol. 6, no. 1, pp. 1-17doi: 10.3390/make6010001.
- [15] N. F. P. Dinata, M. A. M. Ramli, M. I. Jambak, M. A. B. Sidik, and M. M. Alqahtani, "Designing an optimal microgrid control system using deep reinforcement learning: A systematic review," *Engineering Science and*

Technology, an International Journal, vol. 51, p. 101651, 2024/03/01/ 2024, doi: <https://doi.org/10.1016/j.jestch.2024.101651>.

[16] A. Turan, "PID controller design with a new method based on proportional gain for cruise control system," *Journal of Radiation Research and Applied Sciences*, vol. 17, no. 1, p. 100810, 2024/03/01/ 2024, doi: <https://doi.org/10.1016/j.jrras.2023.100810>.

[17] M. Aria and C. Cuccurullo, "bibliometrix: An R-tool for comprehensive science mapping analysis," (in English), *Journal of Informetrics*, Article vol. 11, no. 4, pp. 959-975, 2017, doi: [10.1016/j.joi.2017.08.007](https://doi.org/10.1016/j.joi.2017.08.007).

[18] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement," *Annals of Internal Medicine*, vol. 151, no. 4, pp. 264-269, 2009/08/18 2009, doi: [10.7326/0003-4819-151-4-200908180-00135](https://doi.org/10.7326/0003-4819-151-4-200908180-00135).

[19] J. A. Hejase, P. R. Paladhi, R. S. Krabbenhoft, Z. Chen, J. Tang, and D. J. Boday, "A neural network based method for predicting PCB glass weave induced skew," in *2016 IEEE 25th Conference on Electrical Performance Of Electronic Packaging And Systems (EPEPS)*, 23-26 Oct. 2016 2016, pp. 151-154, doi: [10.1109/EPEPS.2016.7835439](https://doi.org/10.1109/EPEPS.2016.7835439).

[20] F. Liu, Y. J. Lu, D. B. Li, and R. Chiong, "Wasserstein distributionally robust learning for predicting the cycle time of printed circuit board production," (in English), *COMPUTERS IN INDUSTRY*, vol. 164, JAN 2025, Art no. 104213, doi: [10.1016/j.compind.2024.104213](https://doi.org/10.1016/j.compind.2024.104213).

[21] J. J. Huang, G. H. Tzeng, and C. S. Ong, "Multidimensional data in multidimensional scaling using the analytic network process," (in English), *Pattern Recogn. Lett.*, Article vol. 26, no. 6, pp. 755-767, 2005, doi: [10.1016/j.patrec.2004.09.027](https://doi.org/10.1016/j.patrec.2004.09.027).

[22] I. Retto Uhlmann, S. Ledoux Takeda Berger, C. A. de Souza Silva, and E. M. Frazzon, "Chapter 13 - Digital and smart production planning and control," in *Designing Smart Manufacturing Systems*, C. M. Hussain and D. Rossit Eds.: Academic Press, 2023, pp. 311-343.

[23] C. Zhang, M. Juraschek, and C. Herrmann, "Deep reinforcement learning-based dynamic scheduling for resilient and sustainable manufacturing: A systematic review," *J Manuf Syst*, vol. 77, pp. 962-989, 2024/12/01/ 2024, doi: <https://doi.org/10.1016/j.jmsy.2024.10.026>.

[24] F. C. F. Luiz, G. F. C. Maria, and A. C. F. Marcelo, "Proposta de Implementação em Hardware de SVM multi-kernel para Aplicações em IoT," in *XV Simpósio Brasileiro de Automação Inteligente*, 2021, Online, doi: [10.20906/sbai.v1i1.2764](https://doi.org/10.20906/sbai.v1i1.2764). [Online]. Available: https://www.sba.org.br/open_journal_systems/index.php/sbai/article/view/2764

[25] M. R. Santos, A. Guedes, and I. Sanchez-Gendríz, "SHapley Additive exPlanations (SHAP) for Efficient Feature Selection in Rolling Bearing Fault Diagnosis," *Machine Learning and Knowledge Extraction*, vol. 6, no. 1, pp. 316-341, doi: [10.3390/make6010016](https://doi.org/10.3390/make6010016).

[26] S. K. Baduge et al., "Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications," (in English), *Automation in Construction*, Review vol. 141, 2022, Art no. 104440, doi: [10.1016/j.autcon.2022.104440](https://doi.org/10.1016/j.autcon.2022.104440).

[27] C. Cortes and V. Vapnik, "SUPPORT-VECTOR NETWORKS," (in English), *MACHINE LEARNING*, vol. 20, no. 3, pp. 273-297, SEP 1995, doi: [10.1007/BF00994018](https://doi.org/10.1007/BF00994018).

[28] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189-215, 2020/09/30/ 2020, doi: <https://doi.org/10.1016/j.neucom.2019.10.118>.

[29] S. Zhang, "Challenges in KNN Classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 4663-4675, 2022, doi: [10.1109/TKDE.2021.3049250](https://doi.org/10.1109/TKDE.2021.3049250).

[30] M. Çakir, M. Yılmaz, M. A. Oral, H. Ö. Kazancı, and O. Oral, "Accuracy assessment of RFerns, NB, SVM, and kNN machine learning classifiers in aquaculture," *Journal of King Saud University - Science*, vol. 35, no. 6, p. 102754, 2023/08/01/ 2023, doi: <https://doi.org/10.1016/j.jksus.2023.102754>.

[31] Y. Gonzalez Tejeda and H. A. Mayer, "Deep Learning with Convolutional Neural Networks: A Compact Holistic Tutorial with Focus on Supervised

Regression," *Machine Learning and Knowledge Extraction*, vol. 6, no. 4, pp. 2753-2782, doi: [10.3390/make6040132](https://doi.org/10.3390/make6040132).

[32] A. Coscia, V. Dentamaro, S. Galantucci, A. Maci, and G. Pirlo, "Automatic decision tree-based NIDPS ruleset generation for DoS/DDoS attacks," *Journal of Information Security and Applications*, vol. 82, p. 103736, 2024/05/01/ 2024, doi: <https://doi.org/10.1016/j.jisa.2024.103736>.

[33] D. Colledani, P. Anselmi, and E. Robusto, "Machine learning-decision tree classifiers in psychiatric assessment: An application to the diagnosis of major depressive disorder," *Psychiatry Research*, vol. 322, p. 115127, 2023/04/01/ 2023, doi: <https://doi.org/10.1016/j.psychres.2023.115127>.

[34] A. I. A. Group, *AIAG-VDA FMEA Handbook*, 1st ed. 2019, p. 241.

[35] C. A. D. Silva, I. R. Uhlmann, and E. M. Frazzon, "SCREW TORQUE TRACEABILITY CONTROL: INDUSTRIAL APPLICATION," (in English), *INDEPENDENT JOURNAL OF MANAGEMENT & PRODUCTION*, vol. 11, no. 2, pp. 538-547, MAR-APR 2020, doi: [10.14807/ijmp.v11i2.1038](https://doi.org/10.14807/ijmp.v11i2.1038).

[36] J. Ivančan and D. Lisjak, "New FMEA Risks Ranking Approach Utilizing Four Fuzzy Logic Systems," *Machines*, vol. 9, no. 11, doi: [10.3390/machines9110292](https://doi.org/10.3390/machines9110292).

[37] P. Kuchekar, A. S. Bhongade, A. U. Rehman, and S. H. Mian, "Assessing the Critical Factors Leading to the Failure of the Industrial Pressure Relief Valve Through a Hybrid MCDM-FMEA Approach," *Machines*, vol. 12, no. 11, doi: [10.3390/machines12110820](https://doi.org/10.3390/machines12110820).

[38] L. Han, M. Xia, Y. Yu, and S. He, "A Novel Method for Failure Mode and Effect Analysis Based on the Fermatean Fuzzy Set and Bonferroni Mean Operator," *Machines*, vol. 12, no. 5, doi: [10.3390/machines12050332](https://doi.org/10.3390/machines12050332).

[39] C. A. Silva and E. P. Paladini, "Smart Machine Vision System to Improve Decision-Making on the Assembly Line," *Machines*, vol. 13, no. 2, doi: [10.3390/machines13020098](https://doi.org/10.3390/machines13020098).

[40] E. Richardson, R. Trevizani, J. A. Greenbaum, H. Carter, M. Nielsen, and B. Peters, "The receiver operating characteristic curve accurately assesses imbalanced datasets," *Patterns*, vol. 5, no. 6, p. 100994, 2024/06/14/ 2024, doi: <https://doi.org/10.1016/j.patter.2024.100994>.

[41] J. M. Rožanec et al., "Human-centric artificial intelligence architecture for industry 5.0 applications," *International Journal of Production Research*, vol. 61, no. 20, pp. 6847-6872, 2023/10/18 2023, doi: [10.1080/00207543.2022.2138611](https://doi.org/10.1080/00207543.2022.2138611).