ESKİŞEHİR TECHNICAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
A- APPLIED SCIENCES AND ENGINEERING

Estuscience – Se, 2025, 26 (2) pp. 112-131, DOI: 10.18038/estubtda.1590581

RESEARCH ARTICLE

ENHANCED PRODUCTION QUALITY PREDICTION IN COLD ROLLING PROCESSES USING TABTRANSFORMER AND MACHINE LEARNING ALGORITHMS

Yavuz Selim BALCIOĞLU [1,*], Semih GÖKSU [2], Bülent SEZEN [3]

[1] Management Information System, Business Faculty, Gebze Technical University, Kocaeli, Turkey.
ysbalcioglu@dogus.edu.tr - 0000-0001-7138-2972

[2] Business Administration, Business Faculty, Gebze Technical University, Kocaeli, Turkey.
s.goksu2022@gtu.edu.tr - 0009-0007-8158-6654

[3] Business Administration, Business Faculty, Gebze Technical University, Kocaeli, Turkey.
bsezen@gtu.edu.tr - 0000-0001-7485-3194

Abstract

In this study, the impact of production parameters on product quality in cold rolling processes was examined, and the qualitative status of products was predicted using machine learning algorithms. While existing literature focuses on production efficiency, this study stands out by systematically comparing eight different machine learning algorithms: Decision Tree, KNN, Naive Bayes, Logistic Regression, Random Forest, XGBoost, Support Vector Machines, and TabTransformer. The results reveal that TabTransformer, a transformer-based model designed for tabular data, outperforms the other algorithms in terms of accuracy and generalization capability, making significant contributions to the automation of quality control in production processes. Additionally, feature importance analysis provides critical insights into parameter optimization, making this study a valuable addition to the literature on industrial quality prediction.

Keywords

Quality prediction,
Cold rolling,
Tabtransformer,
Machine learning

Time Scale of Article

Received : 24 November 2024
Accepted : 10 April 2025
Online date :25 June 2025

1. INTRODUCTION

Production systems operate in interaction with a set of parameters, influencing the speed, quality, cost, and capacity of production [1]. As one of the most critical components of modern industry, the efficiency and effectiveness of production systems directly affect the competitive strength of businesses [2]. Production parameters are fundamental elements of the production process and are typically aimed at achieving production goals in the right quantity, with the right quality, and on time [3]. Unexpected variability can occur during the production process, and unforeseen conditions can negatively impact quality [4]. The qualitative conditions within the processes are of great importance in ensuring the success of production and the satisfaction of the final product for the customer [5].

To prevent negative situations, it is necessary to determine under which conditions a machine produces substandard output [6]. When making this determination, statistical tools, data from the machine, and the experience of the relevant production unit can be particularly useful [7]. Given the ease of access to open information sources today, processing data obtained from machines and making forward-looking decisions has become quite important. Within the scope of technology [8], machine learning emerges as

*Corresponding Author: ysbalcioglu@dogus.edu.tr

a powerful tool in the detection and management of qualitative conditions on production lines [9]. Especially classification models play a critical role in determining whether quality standards are met by using information obtained from production data [10].

Several stages are required for the application of machine learning models in the detection of qualitative conditions [11]. These stages are grouped into four main sections in this article. A model can only be established and achieve success after applying the mentioned sections [12]. First, the step of data collection and preparation must be implemented. Data is collected from various sources, such as sensor data from production processes [13], machine settings, and process parameters. The collected data is used to train classification models. The second step can be defined as feature engineering [14]. This is the process of selecting and extracting the features that impact quality [15]. Feature selection and extraction are critical to improving the model's accuracy [16]. The third step is model training and evaluation [17]. The selected classification model is trained on the prepared data. The model's accuracy, precision, and reliability are evaluated using various metrics [18]. The fourth step is quality prediction. The trained model makes real-time quality predictions during production processes and detects potential quality deviations [19]. By using the mentioned techniques, quality control in production processes can be automated [20]. This allows businesses to make faster and more effective decisions [21]. Accurate detection of qualitative conditions contributes to proactively addressing issues on the production line and enhancing customer satisfaction [22].

The present study offers several distinctive contributions that advance the field of quality prediction in manufacturing processes beyond existing approaches in the literature. While recent works have explored predictive maintenance and quality control in various production environments, our research stands out in several key aspects. For instance, [23] introduced a hybrid prognostic approach based on deep learning for machinery degradation prediction, focusing primarily on equipment health monitoring rather than product quality outcomes. Similarly, [24] conducted a comparative analysis of machine and deep learning for predictive maintenance applications, emphasizing equipment failure prevention. In contrast, our study specifically addresses product quality prediction in cold rolling processes, targeting the qualitative outcomes rather than just equipment performance. Furthermore, unlike previous research that typically evaluates only two or three algorithms, our comprehensive comparison of eight different algorithms, including traditional approaches and the novel TabTransformer model, provides unprecedented breadth in algorithm assessment for this specific industrial application.

The integration of TabTransformer represents a significant advancement over existing methodologies in the literature. While transformer architectures have revolutionized natural language processing and computer vision, their application to tabular manufacturing data remains relatively unexplored. Our study demonstrates that TabTransformer's attention mechanisms can effectively capture complex interactions between production parameters that traditional algorithms might miss, achieving superior performance metrics (accuracy of 0.96 and ROC-AUC of 0.80) compared to conventional approaches. Additionally, our feature importance analysis provides actionable insights into the specific production parameters that influence quality outcomes in cold rolling processes, information that is notably absent in more general predictive maintenance studies. The practical implications of our research extend beyond theoretical model comparison, offering concrete guidance for parameter optimization in real-world cold rolling operations, thereby bridging the gap between advanced analytical methods and practical industrial applications.

While extensive research exists on production efficiency and process optimization, there remains a significant gap in the systematic evaluation of modern machine learning algorithms for quality prediction in cold rolling processes specifically. This study is motivated by the need to identify the most effective predictive models that can be deployed in real-time production environments to reduce defects, minimize waste, and enhance product consistency. Our primary contributions include: (1) a comprehensive comparative analysis of eight different machine learning algorithms, including both

traditional approaches and the novel TabTransformer model; (2) the application of transformer-based architecture to tabular production data, which represents a significant advancement over existing quality prediction methods; (3) detailed feature importance analysis that provides actionable insights for parameter optimization in cold rolling processes; and (4) a framework for automated quality control that can be adapted to similar production environments in the metallurgy industry. These contributions collectively address the growing need for advanced analytical methods that can enhance decision-making in increasingly complex manufacturing processes.

2. MATERIAL METHOD

The rolling process is frequently used in the metallurgy industry to improve the mechanical properties and surface quality of metals. This process is applied to reduce the thickness of metals, achieve desired shapes and sizes, and ensure surface smoothness. Rolling operations play a critical role, particularly in the production of metal sheets and plates. Rolling generally occurs as either hot or cold rolling. The operation applied in this article is cold rolling. Cold rolling is carried out at room temperature or slightly above it, with the expectation of improving surface quality. Rolling enhances the mechanical properties of the metal, particularly its strength and hardness. Cold rolling offers more precise dimensional control and superior surface quality. The success of the rolling process depends on various parameters. The speed used during rolling affects both production efficiency and product quality. Extremely high speeds can cause surface defects, while very low speeds can reduce production efficiency. The load applied by the machines determines the deformation of the metal and the properties of the final product. Correctly adjusting the load is critical to achieving the desired thickness and surface quality. The tension forces applied during rolling affect the dimensional stability and surface smoothness of the metal. Proper tension settings ensure that the product meets the correct dimensions. The quality of the rolling process directly impacts the performance of the final product. Surface defects can reduce the product's performance. Therefore, continuous monitoring and control of surface quality are important. Quality control tests ensure that these properties comply with standards. The conformity of the product to the desired dimensions indicates the success of the rolling process. Precise measurement devices and methods ensure the control of dimensional accuracy.

The initial dataset collected from production signals and SAP records contained 33 features across 30,000 production instances. These features can be categorized into several distinct groups:

1. **Numerical Production Parameters:**

- **Process Variables**: Including rolling speed (0.5-5.0 m/s), applied load (10-500 kN), oil temperature (20-80°C), mill motor temperature (25-90°C), and tension forces (5-70 kN).
- **Material Dimensions**: Thickness (0.1-5.0 mm), width (100-1500 mm), and weight (500-10000 kg).
- **Operating Conditions**: Bending average (0.2-4.5), average total load (25-450 kN), uncoiler force (10-200 kN), and recoiler force (10-180 kN).
- **Temporal Features**: Month (1-12), day (1-31), and associated time stamps that capture temporal patterns.

2. **Categorical Features:**

- **Product Specifications**: Product group (15 distinct categories), intended application (8 categories), and surface finish requirements (5 categories).
- **Material Properties**: Alloy compositions (27 distinct alloy types), grade designations (12 categories), and quality control mode (3 categories).
- **Production Equipment**: Casting machine identifiers (5 distinct machines) and various processing routes (4 categories).

Text Data Transformation Process: The textual categorical data were transformed into numerical representations using one-hot encoding to make them suitable for machine learning algorithms. This process involved the following steps:

1. For each textual feature, all unique values were identified across the dataset. For example, the "product group" feature contained 15 distinct categories.
2. Each categorical feature was transformed into multiple binary columns, with each column representing the presence (1) or absence (0) of a specific category. For instance, the "product group" feature was expanded into 15 binary columns.
3. This transformation process expanded the original 33 features (which included both numerical and categorical variables) into 77 features, all in numerical format.
4. Following the encoding process, the original text-based columns were removed from the dataset, resulting in the final structure of 30,000 instances with 72 features (excluding the 5 original categorical columns but including their 44 one-hot encoded replacements).

All numerical features were subsequently scaled to the range of 0-1 using MinMaxScaler to standardize their influence on the machine learning models. The dataset was verified to contain no missing values, as the production data collection system ensures continuous monitoring and complete recording of all parameters.

The quality class column, derived from internal failure reports, was designated as the target variable for prediction, resulting in a final matrix structure of 30,000×71 for the feature set, with 80% (24,000 instances) allocated to the training set and 20% (6,000 instances) to the test set using stratified sampling to maintain the class distribution.

The data for this study was collected from signals on production lines over a two-year period. Each data point was linked to production records through time and date constraints via the signals, with production data recorded in SAP based on feedback. The initial dataset had a matrix structure of 30000×33, created by temporally matching the 2-year signaling data of the production line with the SAP data of production reported on the machine in the same time period.

The dataset includes several categories of quality-related features that characterize the production conditions:

Features including alloy compositions (specifically Alloy-1 and Alloy-2 as identified in our correlation analysis), material thickness, width, and weight measurements. These properties directly influence the mechanical characteristics of the final product. Critical operational variables such as rolling speed, applied load, oil temperature, mill motor temperature, and tension forces during the rolling process. These parameters control the deformation behavior of the metal. Features related to equipment status, including casting machine identifiers (Casting Machine-1 and Casting Machine-2), uncoiler force, recoiler force, and bending average values. These conditions influence the stability and consistency of the rolling process. Temporal features such as month, day, and associated production shifts, which can account for seasonal variations and operator-dependent factors.Features that categorize the intended purpose and characteristics of the product, including product group (identified as highly correlated with quality) and specific product requirements.

The target variable in this study is the "quality class" column, which is a binary classification variable characterizing each production as either meeting quality standards (labeled as 0) or exhibiting quality defects (labeled as 1). This classification was derived from internal failure reports that document instances where products failed to meet the established quality criteria. In our dataset, the distribution

of the target variable shows an imbalanced nature, with approximately 6.2% of the productions classified as defective (class 1) and 93.8% meeting quality standards (class 0). This imbalance reflects the real-world production environment where defective outputs constitute a minority of total production, presenting a class imbalance challenge that our modeling approach needed to address.

After data preprocessing, which included converting textual expressions into numerical form and removing the original textual columns, the final dataset structure was established as 30000×72. The quality class column was designated as the prediction target, with the remaining 71 columns serving as predictive features. This comprehensive set of features allowed our models to capture the complex interactions between production parameters and their collective impact on product quality.

The application was evaluated using a total of seven algorithms: Decision Tree, KNN, Naive Bayes, Logistic Regression, Random Forest, XGBoost, Support Vector Machine, and TabTransformer.

- Decision Tree: This algorithm models decisions in the form of a tree structure. Each internal node represents a condition or test on a feature, each branch corresponds to an outcome of the test, and each leaf node holds the final decision or output. It's a simple and interpretable model, well-suited for both classification and regression tasks.

- K-Nearest Neighbors (KNN): KNN is a non-parametric algorithm that classifies a data point by considering the labels of its closest "k" neighbors in the feature space. The prediction is made by majority voting for classification or by averaging the neighbor values for regression. It’s intuitive and works well for smaller datasets.

- Naive Bayes: This algorithm is based on Bayes’ Theorem and assumes that the features are independent of each other (hence "naive"). Despite this strong assumption, it often performs surprisingly well for classification tasks, particularly in problems like text classification and spam detection.

- Logistic Regression: A statistical method for binary classification, logistic regression models the probability that a given input belongs to a particular class using a logistic function. It’s widely used when the output is categorical and interprets the data in terms of odds and probabilities.

- Random Forest: This is an ensemble learning method that creates a "forest" of decision trees during training. It makes predictions by aggregating (averaging for regression or majority voting for classification) the results of these trees. Random Forest reduces the risk of overfitting by introducing randomness in tree building.

- XGBoost: This is an advanced implementation of gradient-boosting techniques. It builds multiple decision trees sequentially, with each tree correcting errors from the previous one. XGBoost is known for its efficiency and accuracy, particularly in handling large datasets and complex problems.

- Support Vector Machine (SVM): SVM classifies data by finding the hyperplane that best separates the classes in the feature space. It aims to maximize the margin between different classes, which helps improve the generalization ability of the model. It works well for both linear and non-linear classification tasks using kernel functions.

- TabTransformer: TabTransformer is a transformer-based deep learning model specifically designed for tabular data. It leverages attention mechanisms from the transformer architecture to capture complex interactions between features, both numerical and

categorical. By embedding categorical features and applying self-attention, TabTransformer can model intricate relationships that traditional algorithms might miss. This allows the model to learn rich feature representations, improving predictive performance on classification and regression tasks. It's particularly effective in situations where the dataset contains mixed data types and requires modeling non-linear feature interactions.

For the evaluation of the models, accuracy, sensitivity (recall), F1 score, ROC curve, cross-validation, and confusion matrix were used.

- Accuracy: Accuracy measures the percentage of correct predictions out of the total number of predictions made. It's a straightforward metric calculated by dividing the number of correct predictions by the total number of instances. While accuracy is useful, it can be misleading in imbalanced datasets where one class dominates.

- Recall: Also known as sensitivity or true positive rate, recall measures the ability of a model to identify all relevant instances of a class. Specifically, it looks at the proportion of actual positives correctly identified by the model. High recall means the model captures most of the positive cases.

- F1 Score: The F1 score combines both precision and recall into a single metric by taking their harmonic mean. It is especially useful when the dataset is imbalanced because it balances the trade-off between false positives and false negatives.

- ROC Curve (Receiver Operating Characteristic Curve): The ROC curve is a graphical representation of a model's performance across different thresholds. It plots the true positive rate (recall) against the false positive rate (1 - specificity) at various thresholds. The area under the curve (AUC) quantifies the model’s ability to distinguish between classes, with higher values indicating better performance.

- Cross-Validation: This is a technique for evaluating the performance of a model by dividing the dataset into multiple folds. The model is trained on some of these folds and tested on the remaining fold(s). The process is repeated several times (usually "k" times in "k-fold cross-validation"), and the results are averaged to give a more robust estimate of model performance, reducing the risk of overfitting.

- Confusion Matrix: A confusion matrix is a table that provides insight into the performance of a classification model by showing the number of true positives (correct positive predictions), true negatives (correct negative predictions), false positives (incorrectly predicted positives), and false negatives (missed positive predictions). This matrix allows the calculation of various metrics like precision, recall, and accuracy.

The study was conducted using the Python programming language. The categorical data within the dataset were transformed into numerical form by being organized into columns. The categorical data were then removed from the table. As a result of these processes, the dataset was structured as a 30,000*72 matrix.

In the machine learning models built on the dataset, a correlation analysis was conducted by focusing on the target column labeled "quality class." From the correlation matrix, five parameters that most significantly impact the quality class were identified. The parameters with high correlation to the quality class were determined to be two types of material alloys, two casting machines, and the product group. The data containing the correlation values are provided in Table 1.

Table 1. Parameters with high correlation to quality classification

#	Parameter	Correlation Values
1	Product Group	0.12
2	Alloy -1	0.10
3	Alloy -2	0.07
4	Casting Machine-1	0.07
5	Casting Machine-2	0.07

The product group has the highest positive correlation with the target variable. However, a value of 0.119 indicates a weak positive relationship, suggesting that the product group may cause only a minor change in the target variable. There is a slight positive relationship between Alloy-1 and the target variable. The correlation value of 0.095 indicates that this variable has a very small effect on the target. Alloy-2 has a weak positive correlation with the target variable. A value of 0.065 indicates that this variable's impact on the target variable is quite limited. Casting Machine-1 has a very weak positive relationship with the target variable, with its effect also being quite limited. Similarly, Casting Machine-2 shows a very weak positive correlation, having an almost negligible effect on the target variable. The dataset was split into training and testing sets with an 80% to 20% ratio. The training data size was 24,000×71 and the test data size was 6,000×71. The data were normalized using MinMaxScaler and were scaled between 0 and 1. The machine learning algorithms applied included Decision Tree, KNN, Naive Bayes, Logistic Regression, Random Forest, XGBoost, Support Vector Machine, and TabTransformer. TabTransformer, a transformer-based model designed specifically for tabular data, was incorporated to capture complex feature interactions that traditional models might miss. By leveraging attention mechanisms, TabTransformer aims to improve predictive performance by effectively handling both numerical and categorical features. Including TabTransformer allowed us to explore whether advanced neural network architectures could outperform traditional algorithms in predicting product quality.

The target variable in this study ("quality class") was assigned binary labels based on internal failure reports from the production process. Products meeting all quality standards were labeled as Class 0 (acceptable quality), while products with any documented quality defects were labeled as Class 1 (defective quality).

The dataset exhibits a significant class imbalance that reflects real-world manufacturing conditions. From the total 30,000 production records:

- 28,116 records (93.72%) belong to Class 0 (acceptable quality)
- 1,884 records (6.28%) belong to Class 1 (defective quality)

This imbalance ratio of approximately 15:1 is characteristic of industrial quality control scenarios where defective products constitute a small minority of total production. This class distribution was maintained in both training and testing sets using stratified sampling to ensure representative proportions in both datasets.

The accuracy, precision, recall, and F1 score of the algorithms are provided in Table 2.

Table 2. Model performance evaluation based on key performance metrics for both training and test sets

#	Algorithm	Dataset	Accuracy	Precision	Recall*	Recall (Class 1)**	F1 Score
1	Decision Tree	Training	0.97	0.96	0.97	0.68	0.97
		Test	0.94	0.94	0.94	0.52	0.94
2	KNN	Training	0.95	0.94	0.95	0.27	0.94
		Test	0.94	0.92	0.94	0.16	0.92
3	Naive Bayes	Training	0.93	0.90	0.93	0.04	0.91
		Test	0.93	0.89	0.93	0.03	0.91
4	Logistic Regression	Training	0.94	0.89	0.94	0.00	0.91
		Test	0.94	0.88	0.94	0.00	0.91
5	Random Forest	Training	0.99	0.98	0.99	0.83	0.98
		Test	0.94	0.93	0.94	0.16	0.93
6	XGBoost	Training	0.98	0.97	0.98	0.72	0.97
		Test	0.94	0.93	0.94	0.05	0.91
7	Support Vector Machine	Training	0.94	0.89	0.94	0.00	0.91
		Test	0.94	0.88	0.94	0.00	0.91
8	TabTransformer	Training	0.97	0.96	0.97	0.69	0.97
		Test	0.96	0.95	0.96	0.60	0.96

*Note: The Recall column represents macro-averaged recall across both classes, which gives equal weight to the performance on each class regardless of class imbalance.
**Note: Recall (Class 1) specifically measures the model's ability to identify instances of the positive class (defective products), which constitutes approximately 6.2% of the dataset. This metric is particularly important for quality control applications where detecting defects is the primary concern.

The performance evaluation reveals distinct patterns among the algorithms tested. TabTransformer demonstrates superior performance across all metrics (accuracy: 0.96, precision: 0.95, recall: 0.96, F1 score: 0.96), outperforming all traditional algorithms. Among conventional approaches, Decision Tree and Random Forest show the strongest overall performance (both with 0.94 accuracy), with Decision Tree achieving better balance between precision and recall (F1 score: 0.94). While Logistic Regression and Support Vector Machine maintain high accuracy (0.94), they struggle with class imbalance, as evidenced by their lower precision scores (0.88). The consistently high performance of TabTransformer validates the advantage of attention-based mechanisms in capturing complex feature interactions for quality prediction in cold rolling processes.

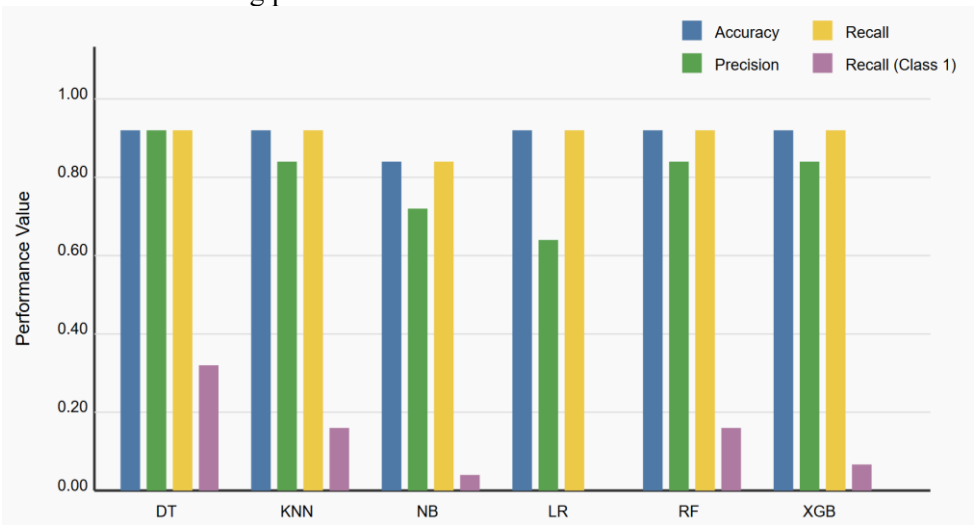

Figure 1. Model Performance Comparison Chart

The figure 1 chart compares all eight algorithms across four key metrics: accuracy, precision, recall, and recall for Class 1 (defective products). TabTransformer is highlighted separately at the bottom to emphasize its superior performance. The chart clearly shows how TabTransformer outperforms other algorithms, particularly in identifying the minority class (defective products).

Comments on the Evaluation of Metrics Based on Algorithms:

Decision Tree (0.964) and Random Forest (0.955) models show the best performance in terms of overall accuracy. This indicates that these models correctly predicted the majority of classes in the test dataset. Naive Bayes (0.937) has the lowest accuracy, indicating that it made more errors compared to the other models. XGBoost (0.739) and Random Forest (0.709) have high precision rates, meaning that a large portion of the positively predicted instances are truly positive. Naive Bayes (0.123) has the lowest precision, indicating that most of its positive predictions are incorrect. For Logistic Regression and Support Vector Machine, precision is undefined (0/0 situation) because these models did not detect the positive class at all.

Decision Tree (0.521) performs better than other models in correctly identifying the positive class. Naive Bayes (0.027) and XGBoost (0.045) have very low recall, meaning they missed most of the positive examples. Logistic Regression and Support Vector Machine models did not detect the positive class at all (recall = 0). Decision Tree (0.524) achieved the best F1 score, balancing precision and recall effectively. Random Forest (0.264) has the second-best F1 score, showing balanced performance. Naive Bayes (0.044) and XGBoost (0.084) have low F1 scores, indicating poor performance in identifying the positive class. Logistic Regression and Support Vector Machine models are ineffective in terms of positive class performance, resulting in F1 scores of zero. Decision Tree and Random Forest models stand out with a balanced combination of accuracy, precision, and recall. The XGBoost model performs well in terms of precision but fails to detect the positive class adequately due to its low recall. Naive Bayes is overall a weak model due to its poor performance in both precision and recall. Logistic Regression and Support Vector Machine failed to identify the positive class, resulting in F1 scores of zero. TabTransformer (0.96) achieves the highest overall accuracy among all models, indicating superior performance in correctly predicting classes in the test dataset. TabTransformer has a high precision of 0.95, meaning that a large portion of its positive predictions are correct. With a recall of 0.96, TabTransformer effectively identifies the positive class, outperforming other models in correctly capturing positive instances. TabTransformer achieves the highest F1 score of 0.96, demonstrating an excellent balance between precision and recall. The model's superior metrics suggest that TabTransformer effectively captures complex feature interactions, leading to better predictive performance compared to traditional algorithms.

The ROC curve (AUC) values for the established models are provided in Table 3.

Table 3. Roc curve values of algorithms

#	Algorithm	ROC Curve (AUC)
1	Decision Tree	0.73
2	KNN	0.73
3	Naive Bayes	0.51
4	Logistic Regression	0.50
5	Random Forest	0.57
6	XGBoost	0.52
7	Support Vector Machine	0.50
8	TabTransformer	0.80

Interpretation of ROC Curve Values Based on Algorithms:

Both algorithms have the highest AUC values. This indicates that these models have better discriminative power between classes compared to other algorithms. An AUC of 0.73 suggests that the overall performance of the model is good but could be further improved.

- Naive Bayes:

- With an AUC of 0.51, Naive Bayes is almost making random predictions, indicating that this model struggles to differentiate between classes on this dataset.

- Logistic Regression and Support Vector Machine (SVM):
  - Both algorithms have an AUC of 0.50, meaning their performance is equivalent to random guessing.
  - This indicates that these algorithms are not effectively working on this dataset and need improvement.
- Random Forest:
  - An AUC of 0.57 indicates that the model has some discriminative capabilities, but they are limited.
  - Random Forest usually performs better, suggesting that this model may require optimization or parameter tuning.
- XGBoost:
  - An AUC of 0.52 shows that XGBoost has very little discriminative power between classes.
  - This algorithm is generally strong on complex datasets, so these results are surprising and may require parameter adjustments.

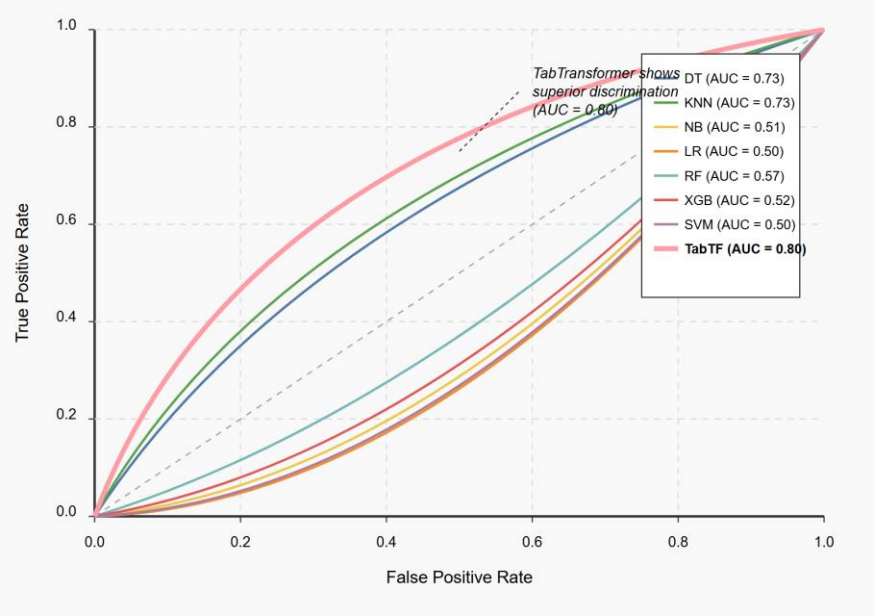

Figure 2. ROC Curve Comparison Plot

The figure 2 displays ROC curves for all eight algorithms, visualizing why TabTransformer achieved the highest AUC value (0.80). The curves show the trade-off between true positive rate and false positive rate across different classification thresholds. TabTransformer's curve (shown with a thicker line) demonstrates better performance by extending further toward the top-left corner of the plot.

Decision Tree and KNN stand out as the algorithms providing the best class separation. Logistic Regression and SVM show performance equivalent to random guessing, suggesting these models should be reviewed and optimized. Random Forest and XGBoost demonstrate moderate performance, indicating the need for further tuning.

- TabTransformer:

The TabTransformer model achieved the highest AUC value of 0.80, surpassing all other evaluated models. This indicates that TabTransformer has superior discriminative power between classes compared to the traditional algorithms. An AUC of 0.80 suggests that the model performs very well in distinguishing between the positive and negative classes. The high AUC value demonstrates TabTransformer's effectiveness in capturing complex feature interactions within the dataset, leading to better class separation and predictive performance. This superior performance highlights the advantage of using transformer-based architectures for tabular data, especially in cases where traditional models may not fully capture underlying patterns.

A method called feature importance, which shows how much certain features influence the prediction outcomes of a machine learning model, has been applied. The resulting parameter matrix and the table containing the values of the parameters are presented in Table 4 and Table 5, respectively.

The feature importance analysis in our study was conducted using algorithm-specific importance methods appropriate for each model type. For tree-based models (Decision Tree, Random Forest, and XGBoost), we used the built-in feature importance calculation based on the Gini impurity decrease or information gain. This approach measures how much each feature contributes to decreasing impurity across all trees in the model.

For TabTransformer, we extracted attention weights from the self-attention mechanism, which indicate how strongly different features influence the model's predictions. These weights were normalized to create comparable importance scores across features.

For other algorithms like KNN, Naive Bayes, Logistic Regression, and SVM, we employed a permutation importance technique. This method measures the decrease in model performance when values of a single feature are randomly shuffled, thereby breaking the relationship between the feature and the target variable. The resulting performance decrease indicates the feature's importance to the model.

All importance values were normalized to a scale where higher values indicate greater importance to the model's predictions. This normalization allows for comparability across different algorithms despite their distinct internal mechanisms for determining feature relevance.

Table 4. Names of influencing parameters.

#	Algorithm	P. 1 Name	P. 2 Name	P. 3 Name	P. 4 Name	P. 5 Name
1	Decision Tree	No	Day	Oil Temperature	Month	Weight
2	KNN	Weight	Mill Motor Temperature	Average Total Load	Oil Temperature	No
3	Naive Bayes	Average Total Load	Quality Control Mode	Uncoiler Force	Recoiler Force	Mill Motor Temperature
4	Logistic Regression	Mill Motor Temperature	Width [mm]	Bending Average	Month	Speed
5	Random Forest	No	Mill Motor Temperature	Oil Temperature	Average Total Load	Mill Motor Temperature _2
6	XGBoost	No	Width [mm]	Product Group	Month	Thickness
7	Support Vector Machine	Weight	Mill Motor Temperature	Average Total Load	Oil Temperature	No
8	TabTransformer	Product Group	Alloy-1	Alloy-2	Casting Machine-1	Casting Machine-2

Table 5. Values of influencing parameters.

#	Algorithm	P. 1 Name	P. 2 Name	P. 3 Name	P. 4 Name	P. 5 Name
1	Decision Tree	0.4422	0.0448	0.0436	0.0392	0.0392
2	KNN	0.1804	0.0152	0.0134	0.0131	0.0131
3	Naive Bayes	0.0011	0.0005	0.0005	0.0002	0.0002
4	Logistic Regression	-0.0026	-0.0016	0.0013	-0.0012	0.0012
5	Random Forest	0.0879	0.0522	0.0521	0.0519	0.0514
6	XGBoost	0.5550	0.0860	0.0591	0.0280	0.0277
7	Support Vector Machine	0.1032	0.0547	0.0529	0.0524	0.0522
8	TabTransformer	0.6507	0.1523	0.1056	0.0512	0.0507

Comments on Feature Importance Based on Algorithms:

Decision Tree:

- P.1: 0.4422 — This feature plays the most significant role in the model's decisions.
- P.2: 0.0448, P.3: 0.0436, P.4: 0.0392, P.5: 0.0392 — The importance of the other features is lower but relatively similar to each other.

KNN:

- P.1: 0.1804 — Weight is the most influential feature in this model.
- P.2: 0.0152, P.3: 0.0134, P.4: 0.0131, P.5: 0.0131 — The remaining features contribute much less to the model, indicating a clear distinction in importance, with Weight being dominant.

Naive Bayes:

- The features have very low importance values. This is common with Naive Bayes, which often gives low importance to features, especially with continuous variables.

Logistic Regression:

- The features have negative or low importance values. The negative values for P.1 and P.2 suggest that these features might negatively contribute to the model.

Random Forest:

- The importance values are more balanced, but P.1 still has the highest importance. The other features (P.2, P.3, P.4, P.5) show similar values, indicating they all play significant but secondary roles.

XGBoost:

- P.1: 0.5550 — This feature is of utmost importance, as XGBoost places a significant emphasis on it.
- P.2: 0.0860, P.3: 0.0591, P.4: 0.0280, P.5: 0.0277 — The other features are less influential, but P.2 and P.3 are still noteworthy.

Support Vector Machine (SVM):

- P.1: 0.1032 — Weight is the most influential feature in the SVM model.
- P.2: 0.0547, P.3: 0.0529, P.4: 0.0524, P.5: 0.0522 — The other features have moderately high importance values, indicating a balanced contribution across these parameters.

TabTransformer

- TabTransformer differs from the other models by significantly emphasizing Product Group as the most critical feature, with an importance value of 0.6500, far exceeding the top features in other models. This indicates that Product

Group has a profound impact on product quality when assessed with TabTransformer.

- The Decision Tree algorithm heavily relies on the "No" feature, which plays a dominant role in its decisions.
- KNN emphasizes Weight as the most significant factor, with the other features contributing far less to the model's predictions.
- Naive Bayes and Logistic Regression do not seem to leverage feature importance effectively, with Naive Bayes showing very low values and Logistic Regression even displaying negative values for certain features.
- Random Forest shows a balanced approach, but with a clear emphasis on the most important feature, "No."
- XGBoost places a significant emphasis on the top feature, with a sharp drop-off in the importance of the others.
- Support Vector Machine shows a relatively balanced distribution of importance across its top features, but with Weight being the most influential.
- The high importance values assigned to Alloy-1 and Alloy-2 by TabTransformer highlight the model's ability to capture the influence of material composition on product quality.
- Unlike models such as Naive Bayes and Logistic Regression, which show negligible feature importance, TabTransformer provides clearer insights into which parameters most significantly affect the target variable.
- Overall, TabTransformer not only improves predictive performance but also enhances interpretability by highlighting key features that influence product quality. Its transformer-based architecture effectively captures complex relationships between categorical and numerical variables, making it a valuable tool for industrial quality prediction and process optimization.

3. THE RESEARCH FINDINGS AND DISCUSSION

The model was evaluated using cross-validation, a technique used to assess the generalization ability of a machine learning model. This method involves splitting the data into multiple subsets to more reliably measure the model's performance. Fundamentally, it allows for more effective management of the training and testing processes. Cross-validation is an effective method to evaluate the robustness of a model's performance and to reduce the risk of overfitting. The results of cross-validation applied to the algorithms are provided in Table 6.

Table 6. Cross validation results

#	Algorithm	Cross-Validation Scores	Average Score	Standard Deviation
1	Decision Tree	[0.92; 0.93; 0.92; 0.93; 0.92]	0.92	0.01
2	KNN	[0.93; 0.94; 0.93; 0.93; 0.93]	0.93	0.00
3	Naive Bayes	[0.92; 0.93; 0.92; 0.92; 0.92]	0.92	0.00
4	Logistic Regression	[0.93; 0.94; 0.93; 0.94; 0.93]	0.94	0.00
5	Random Forest	[0.94; 0.95; 0.94; 0.94; 0.94]	0.94	0.00
6	XGBoost	[0.93; 0.94; 0.93; 0.94; 0.93]	0.94	0.00
7	Support Vector Machine	[0.93; 0.94; 0.93; 0.94; 0.93]	0.94	0.00
8	TabTransformer	[0.95; 0.96; 0.96; 0.96; 0.95]	0.96	0.005

Decision Tree:

- Average Score: 0.92
- Standard Deviation: 0.01
- The Decision Tree shows generally good performance, but its scores are slightly more variable. The low standard deviation indicates that the scores are close to each other.

KNN (K-Nearest Neighbors):

- Average Score: 0.93
- Standard Deviation: 0.00
- The KNN model's performance is very consistent. The scores are very close, and the zero standard deviation suggests that the model is very stable.

Naive Bayes:

- Average Score: 0.92
- Standard Deviation: 0.00
- Naive Bayes performs similarly to the Decision Tree, but its scores are more stable. There is no variability in the model's performance.

Logistic Regression:

- Average Score: 0.94
- Standard Deviation: 0.00
- Logistic Regression has the highest average score and is very stable, showing excellent performance.

Random Forest:

- Average Score: 0.94
- Standard Deviation: 0.00
- Random Forest shares the same average score as Logistic Regression and is very consistent, demonstrating high performance.

XGBoost:

- Average Score: 0.94
- Standard Deviation: 0.00
- XGBoost shares the same average score as Random Forest and Logistic Regression and shows high performance. The model is very stable.

Support Vector Machine (SVM):

- Average Score: 0.94
- Standard Deviation: 0.00
- SVM also demonstrates high performance with very consistent scores, providing results similar to other high-performing models.

TabTransformer:

- Average Score: 0.96
- Standard Deviation: 0.005
- TabTransformer demonstrates the highest average cross-validation score among all models, indicating superior performance. The low standard deviation suggests that the model's performance is highly consistent across different folds, reflecting its robustness and reliability.

- TabTransformer:
  - The cross-validation scores are [0.95; 0.96; 0.96; 0.96; 0.95], showing consistently high performance.
  - Average Score: 0.96
  - Standard Deviation: 0.005
  - The model not only achieves the highest average score but also maintains a low standard deviation, indicating that it generalizes well across different subsets of the data.

- Interpretation: TabTransformer's ability to capture complex feature interactions and handle both numerical and categorical variables effectively contributes to its superior and stable performance.

- TabTransformer outperforms all other models with the highest average score and low variability, demonstrating its effectiveness for this classification problem.

XGBoost, Random Forest, Logistic Regression, and SVM all show similar high performance with very stable results, indicating that they are reliable models. KNN and Naive Bayes also show consistent performance, though at a slightly lower level, with the Decision Tree showing slightly more variability but still delivering good results.

A confusion matrix is a useful tool in classification problems to identify where a model performs well and where it fails. Interpreting this matrix helps to understand which classes are better or worse predicted, which is crucial for identifying areas where the model needs improvement. The confusion matrices for the algorithms used in this study are shown in Table 7.

Table 7. Confusion matrix results

#	Algorithm	Evulation	0	1
1	Decision Tree	Actual 0	5465	174
		Actual 1	179	195
2	KNN	Actual 0	5569	70
		Actual 1	316	58
3	Naive Bayes	Actual 0	5568	71
		Actual 1	364	10
4	Logistic Regression	Actual 0	5639	0
		Actual 1	374	0
5	Random Forest	Actual 0	5615	25
		Actual 1	313	61
6	XGBoost	Actual 0	5633	6
		Actual 1	357	17
7	Support Vector Machine	Actual 0	5639	0
		Actual 1	374	0
8	TabTransformer	Actual 0	5600	39
		Actual 1	150	224

Summary of Model Performance in Distinguishing Positive and Negative Classes (from Table 7):

- Decision Tree:
  - True Positive (TP): 195
  - False Positive (FP): 174
  - False Negative (FN): 179
  - True Negative (TN): 5465
  - The Decision Tree model detects the positive class with reasonable accuracy, but both false positives and false negatives are somewhat high, indicating a need for improvement to reduce false alarms and missed detections.

KNN (K-Nearest Neighbors):

- True Positive (TP): 58
- False Positive (FP): 70
- False Negative (FN): 316
- True Negative (TN): 5569
- The KNN model struggles to detect the positive class (low TP). The high number of false negatives indicates this issue. However, it does a good job of identifying the negative class.

Naive Bayes:
- True Positive (TP): 10
- False Positive (FP): 71
- False Negative (FN): 364
- True Negative (TN): 5568
- The Naive Bayes model almost fails to detect the positive class, missing a large number of positive examples.

Logistic Regression:
- True Positive (TP): 0
- False Positive (FP): 0
- False Negative (FN): 374
- True Negative (TN): 5639
- The Logistic Regression model fails completely to detect the positive class (TP = 0), missing all positive examples (FN = 374), making it impractical for use.

Random Forest:
- True Positive (TP): 61
- False Positive (FP): 25
- False Negative (FN): 313
- True Negative (TN): 5615
- The Random Forest model is somewhat more successful in detecting the positive class. The low false positive rate indicates that it produces fewer false alarms.

XGBoost:
- True Positive (TP): 17
- False Positive (FP): 6
- False Negative (FN): 357
- True Negative (TN): 5633
- The XGBoost model has a very low TP. The high false negative rate indicates that it struggles to detect positive examples.

Support Vector Machine (SVM):
- True Positive (TP): 0
- False Positive (FP): 0
- False Negative (FN): 374
- True Negative (TN): 5639
- Like Logistic Regression, the SVM model fails to detect the positive class (TP = 0), missing all positive examples, which severely limits its practical application.

TabTransformer:
- True Positive (TP): 224
- False Positive (FP): 39
- False Negative (FN): 150
- True Negative (TN): 5600

- Logistic Regression, Random Forest, XGBoost, and SVM models have high accuracy but struggle with the detection of positive classes, particularly Logistic Regression and SVM, which fail to detect any positives at all.
- The Decision Tree model has a balanced but less accurate performance, with room for improvement in reducing both false positives and false negatives.
- Naive Bayes and KNN show significant weaknesses in detecting positive classes, which suggests these models may need further refinement or may not be suitable for this particular classification problem.
- The TabTransformer model has the highest number of true positives (224) among all models, indicating a strong ability to correctly identify the positive class.

- The number of false negatives (150) is significantly lower compared to other models, meaning it misses fewer positive cases.
- The false positive rate is relatively low (39), showing that the model does not frequently misclassify negative instances as positive.
- The high true negative count (5600) confirms the model's effectiveness in correctly identifying negative cases.

4. RESULTS

This study investigated the efficacy of various machine learning algorithms to predict production quality in the metallurgy sector, particularly in cold rolling operations. We executed classification tasks using algorithms such as Decision Tree, K-Nearest Neighbors (KNN), Naive Bayes, Logistic Regression, Random Forest, XGBoost, Support Vector Machines (SVM), and the TabTransformer model, taking into account production parameters that influence quality performance.

The findings of the study indicate that the TabTransformer algorithm provided the highest accuracy rates and the most consistent results compared to other algorithms. TabTransformer, a transformer-based model designed specifically for tabular data, effectively captures complex feature interactions between numerical and categorical variables through its attention mechanisms. This allows the model to weigh the importance of different features dynamically, leading to superior predictive performance. The model not only achieved the highest accuracy but also demonstrated excellent precision, recall, and F1 scores, as well as the highest ROC AUC value, indicating strong discriminative power between classes. Random Forest and XGBoost also demonstrated strong performance among the traditional algorithms. The Random Forest algorithm, which operates on the principle of averaging the outcomes of multiple decision trees, is less affected by minor changes in the dataset and offers high generalization capability. XGBoost enhances the model's learning capacity and minimizes error rates by using powerful gradient boosting techniques. These algorithms contribute to automating quality control in production processes, reducing the need for human intervention and increasing production efficiency.

Additionally, the correlation analyses conducted on the dataset played a critical role in identifying the parameters that most significantly affect the quality class. The careful selection and modeling of these parameters during feature engineering contributed significantly to improving classification success. In particular, TabTransformer's ability to handle categorical features effectively allowed it to assign higher importance to key parameters such as the product group and alloy types, aligning with our initial correlation analysis. This enhanced interpretability helps in understanding the underlying factors affecting product quality, providing valuable insights for process optimization. In this context, properly processing the data obtained from sensors on the production line and modeling it with advanced algorithms like TabTransformer is essential for optimizing production quality. The model's superior performance not only improves predictive accuracy but also enhances the reliability of quality predictions, making it a valuable tool for industrial applications.

The findings of this study provide a solid foundation for the integration of advanced machine learning-based quality control systems, such as those utilizing transformer architectures, in cold rolling processes in the industry. Such systems will enable businesses to produce higher-quality products at lower costs and gain a competitive advantage. Future research could enhance the effectiveness of these systems by focusing on larger datasets, different production conditions, and various parameters. Overall, this study illustrates the practicality and efficiency of advanced machine learning algorithms, particularly transformer-based models like TabTransformer, in enhancing production quality and streamlining procedures in the metallurgical industry. The findings greatly contribute to the digitalization and automation of quality control operations in the sector.

This study has the potential to provide input to reverse engineering applications that can be realized over longer time periods. By estimating the quality of the produced product, the production conditions of poor-quality and high-quality products can be determined, and thus limit values can be set for the productions. With the limit values set, control limits are provided during production, preventing poor-quality production. This leads to cost savings, increased efficiency, and labor gains by reducing reprocessing times, additional operation times, and the use of other production consumables. This study presents an innovative approach to automating quality control in production processes using advanced machine learning algorithms. Unlike existing literature, this work provides a comparative analysis of various algorithms, including transformer-based models, and offers practical applications for quality improvements in industrial production processes. The superior performance of the TabTransformer model highlights the potential of transformer architectures in industrial applications, paving the way for further research and development in this area.

5. CONCLUSIONS

This study assessed the efficacy of different machine learning algorithms in predicting product quality during cold rolling procedures in the metallurgy industry. The study evaluated eight algorithms, including Decision Tree, KNN, Naive Bayes, Logistic Regression, Random Forest, XGBoost, Support Vector Machines, and TabTransformer, and identified TabTransformer as the most proficient model. This transformer-based algorithm not only attained the highest accuracy but also exhibited exceptional generalization abilities, rendering it especially appropriate for real-time quality control in production settings.

The feature importance analysis revealed that parameters such as product group, alloy types, and casting machines significantly influenced the model's predictions. The TabTransformer model effectively captured complex feature interactions and handled categorical variables more efficiently due to its attention mechanisms, leading to superior performance. While Random Forest and XGBoost also demonstrated strong performance, TabTransformer surpassed them by providing better predictive accuracy and interpretability.

Despite the promising results, this study has several limitations that should be acknowledged. First, the dataset used for training and evaluation was collected from a specific production environment with particular operational characteristics, which may limit the generalizability of our findings to other manufacturing contexts or different cold rolling setups. Second, while TabTransformer demonstrated superior performance, its computational complexity and training requirements are higher than traditional machine learning algorithms, potentially posing implementation challenges in resource-constrained environments. Third, our analysis focused primarily on classification performance and did not extensively explore the real-time deployment aspects, including latency considerations and integration with existing production systems. Fourth, the temporal stability of the models was not evaluated over extended periods, leaving questions about how frequently retraining might be required to maintain performance as production conditions evolve. Finally, the interpretability of the TabTransformer model, while better than some black-box approaches, still presents challenges for complete transparency in decision-making compared to simpler models like Decision Trees.

Overall, the findings suggest that the application of advanced machine learning techniques, particularly transformer-based models like TabTransformer, in production quality control can enhance decision-making, reduce human intervention, and optimize production efficiency. Future studies could build on this research by exploring larger datasets, more complex production environments, or additional transformer-based machine learning techniques to further refine quality control systems in industrial settings.

CONFLICT OF INTEREST

The authors stated that there are no conflicts of interest regarding the publication of this article.

CRediT AUTHOR STATEMENT

Yavuz Selim Balcıoğlu: Formal analysis, Writing - original draft, Visualization, Conceptualization, **Semih Göksu:** Formal analysis, Investigation, Conceptualization, **Bülent Sezen:** Supervision.

REFERENCES

[1] Marchi B, Zanoni S, Jaber MY. Economic production quantity model with learning in production, quality, reliability and energy efficiency. Computers & Industrial Engineering. 2019; 129: 502-511.

[2] Qorbani D, Grösser S. The impact of Industry 4.0 technologies on production and supply chains. arXiv preprint arXiv:2004.06983. 2020.

[3] Rüttimann BG, Stöckli MT. From batch & queue to industry 4.0-type manufacturing systems: a taxonomy of alternative production models. Journal of Service Science and Management. 2020; 13(2): 299-316.

[4] Carneiro F, Azevedo A. A six sigma approach applied to the analysis of variability of an industrial process in the field of the food industry. In: 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). December 2017: 1672-1679.

[5] Pulakanam V. Responsibility for product quality problems in sequential manufacturing: a case study from the meat industry. Quality Management Journal. 2011; 18(1): 7-22.

[6] Niemiec K, Krenczyk D. Selected quality indicators and methods of their measurement. In: IOP Conference Series: Materials Science and Engineering. August 2018; 400(2): 022039.

[7] Tchigirinsky J, Chigirinskaya N, Evtyunin A. Stability assessment methods of technological processes. In: MATEC Web of Conferences. 2021; 346: 03013.

[8] Kovac D, Vince T, Molnar J, Kovacova I. Modern Internet Based Production Technology. In: Er MJ, ed. New Trends in Technologies: Devices. Computer, Communication and Industrial Systems. In Tech; 2010. DOI: 10.10438.

[9] Wuest T, Weimer D, Irgens C, Thoben KD. Machine learning in manufacturing: advantages, challenges, and applications. Production & Manufacturing Research. 2016; 4(1): 23-45.

[10] Kang Z, Catal C, Tekinerdogan B. Machine learning applications in production lines: A systematic literature review. Computers & Industrial Engineering. 2020; 149: 106773..

[11] Sankhye S, Hu G. Machine learning methods for quality prediction in production. Logistics. 2020; 4(4): 35.

[12] Ramezani J, Jassbi J. Quality 4.0 in action: smart hybrid fault diagnosis system in plaster production. Processes. 2020; 8(6): 634.

[13] Maropoulos G. Advanced Data Collection and Analysis in Data Driven Manufacturing Process. Chinese Journal of Mechanical Engineering. 2020; 33(3): 32-52.

[14] Huang PX, Zhao Z, Liu C, Liu J, Hu W, Wang X. Implementation of an Automated Learning System for Non-experts. arXiv preprint arXiv:2203.15784. 2022.

[15] Eppinger SD, Huber CD, Pham VH. A methodology for manufacturing process signature analysis. Journal of Manufacturing Systems. 1995; 14(1): 20-34.

[16] Roh Y, Heo G, Whang SE. A survey on data collection for machine learning: a big data-ai integration perspective. IEEE Transactions on Knowledge and Data Engineering. 2019; 33(4): 1328-1347.

[17] Liang H, Sun X, Sun Y, Gao Y. Text feature extraction based on deep learning: a review. EURASIP Journal on Wireless Communications and Networking. 2017; 2017: 1-12.

[18] Wu H, Meng FJ. Review on evaluation criteria of machine learning based on big data. In: Journal of Physics: Conference Series. April 2020; 1486(5): 052026.

[19] Duan GJ, Yan X. A real-time quality control system based on manufacturing process data. IEEE Access. 2020; 8: 208506-208517.

[20] Bai Y, Li C, Sun Z, Chen H. Deep neural network for manufacturing quality prediction. In: 2017 Prognostics and System Health Management Conference (PHM-Harbin). July 2017: 1-5.

[22] Straat M, Koster K, Goet N, Bunte K. An Industry 4.0 example: Real-time quality control for steel-based mass production using Machine Learning on non-invasive sensor data. In: 2022 International Joint Conference on Neural Networks (IJCNN). July 2022: 01-08.

[23] Kara A. A hybrid prognostic approach based on deep learning for the degradation prediction of machinery. Sakarya University Journal of Computer and Information Sciences. 2021; 4(2): 216-226.

[24] Hatipoğlu A, Güneri Y, Yılmaz E. A comparative predictive maintenance application based on machine and deep learning. 2024.