A Hybrid Clustering–Classification Approach for Analyzing Machine Usage Patterns in Industrial IoT Data

Farideh Tavakoli
Matr. Num. 0001115734
Digital Transformation Management
Course: Machine Lerarning
University of Bologna
Email: farideh.tavakoli@studio.unibo.it

***Abstract*—This work presents a hybrid machine learning framework for understanding and predicting machine behaviors using industrial IoT data. The proposed method integrates unsupervised K-Means clustering for behavioral segmentation with supervised classification for pattern recognition. Distinct machine usage profiles were first identified through clustering and then used to train predictive models, including Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN). Experimental results show that all models achieved high accuracy, with the SVM outperforming others in generalization and stability. The final SVM pipeline enables real-time classification of new IoT data, supporting predictive maintenance and data-driven decision-making in smart manufacturing environments.**

1. Introduction

Industrial machines continuously generate large amounts of IoT data through embedded sensors that monitor their daily operations. These sensors record how long each machine spends in various states such as execution (active production time), ready (idle but operational), failure (unplanned downtime), and power-off (planned stop or shutdown). In the company under study, a wide range of machines are used across different production lines, each categorized into a specific subtechnology according to its functional purpose and technical characteristics.

As the volume of machine data has grown rapidly, manual analysis has become infeasible. Understanding the hidden behavioral patterns within this data is crucial for optimizing performance, planning maintenance, and improving production reliability.

The goal of this project is to analyze machine usage data to uncover distinct performance patterns, establish benchmarks, and identify machines operating under inefficient or potentially critical conditions. Machines that operate efficiently, with high execution time and minimal downtime, serve as benchmarks of optimal performance, while those with frequent or prolonged failures are considered underperformers.

The dataset used for analysis contains approximately 36,106 records, each representing a machine's daily aggregated operational summary. The features include operational durations, state transition counts (how often machines move between states), and contextual identifiers such as date, serial number, and subtechnology type.

To address these challenges, a hybrid data-driven framework was developed, combining unsupervised and supervised learning techniques. Clustering methods such as K-Means were first applied to uncover natural groupings and behavioral patterns among machines, which were then used to generate labels for training classification models. This integrated approach enables both the discovery of underlying operational structures and the prediction of machine behavior in future unseen data.

By grouping machines with similar operational behavior and comparing them within their subtechnology category, we aim to understand performance variability and pinpoint machines that may require maintenance or attention. This analysis supports the creation of a data-driven alert and recommendation system, helping maintenance teams take preventive actions, improve reliability, and reduce downtime and bring measurable business value to the company.

The remainder of this report is organized as follows. Section 2 reviews related work and existing methodologies. Section 3 presents the proposed framework following the Cross-Industry Standard Process for Data Mining (CRISP-DM) approach. Section 4 discusses the experimental results and analysis. Finally, Section 5 concludes with the main findings, limitations, and future research directions.

2. Related Work

Unsupervised clustering algorithms have been widely used to identify machine usage patterns and detect anomalies in industrial IoT data. In particular, K-Means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) can group multivariate sensor readings into clusters that represent different equipment behaviors, while flagging points that deviate from these clusters as potential outliers [1]. K-Means has been applied to group similar sensor behavior and identify deviations, for example, in automotive manufacturing, K-Means was used for process optimization and to cluster sensor readings for anomaly detection [2].

Researchers often prefer K-Means for its simplicity and speed on large datasets, as it partitions data into distinct clusters that can represent different machine operating states [3]. However, a fixed K-Means clustering can sometimes mask rare events, while it yields clear cluster boundaries, it may rely heavily on cluster centroids and not immediately highlight subtle outlier behavior that signals early-stage faults [2]. To address this, density-based clustering methods like DBSCAN have been explored for IoT maintenance data. DBSCAN can find arbitrarily shaped clusters and naturally treats sparse outliers as "noise," making it well-suited for fault detection in complex sensor patterns [3]. Overall, clustering methods have proven to be valuable in predictive maintenance for grouping similar sensor behaviors, detecting outlier conditions, and revealing underlying structures in machine operation data that correlate with maintenance needs.

Supervised classification algorithms have also been extensively applied for predictive maintenance, especially when historical sensor data labeled with normal or failure states are available [1]. Support Vector Machines (SVM) and Decision Trees are common choices for predicting equipment health status or potential failures. They can learn decision boundaries or rules that distinguish healthy from faulty conditions, often achieving high accuracy in forecasting failures based on early indicators [1]. K-Nearest Neighbors (KNN) has been used for predictive maintenance by comparing current sensor data to similar past cases, but it often causes more false alarms than SVM or decision tree models. In comparative studies, decision trees achieved the best accuracy ($\approx 98\%$) with almost no false positives, while SVM and KNN also performed well (AUC > 0.95) but had drawbacks, SVM was slower, and KNN required larger storage and showed higher false alarm rates [4]. Overall, decision trees are often praised for their interpretability and efficiency in industrial maintenance contexts, whereas SVMs handle high-dimensional sensor data well [5] and KNN provides a simple benchmark despite its sensitivity to noise.

Hybrid approaches have emerged that combine unsupervised clustering with supervised learning to improve predictive maintenance results. In such frameworks, clustering first partitions the data into behavioral groups, and the resulting cluster labels are then used to train classification models. For instance, researchers have proposed using K-Means to cluster similar operating conditions of machines, and subsequently training an SVM (or other classifier) on each cluster to predict failures within that context [5]. This layered strategy leverages the strengths of both methods: the clustering step uncovers structure in the data, while the classification step provides targeted predictions given that structure. Recent work demonstrates the effectiveness of this approach. In a study on industrial compressor data, incorporating cluster-derived features (from an unsupervised pre-clustering step) led to notable improvements in supervised classification performance. Classifiers including SVM, decision tree ensembles, and KNN all achieved higher recall and F1-scores when the model was informed by preceding clustering results [6]. In particular, KNN and SVM benefited significantly from the pre-clustering stage, showing that a hybrid unsupervised-supervised pipeline can enhance detection of abnormal conditions while reducing misclassification rates. This body of related work highlights that the combination of clustering for pattern discovery and classification for prediction is a promising direction in IoT-driven predictive maintenance, enabling more robust and accurate monitoring of equipment health [5] [6].

Overall, previous research demonstrates the effectiveness of both unsupervised and supervised learning approaches for predictive maintenance. However, many existing studies rely solely on either clustering for pattern recognition or classification for fault prediction, often requiring pre-labeled data that may not always be available in real industrial settings. To address this limitation, the present study proposes a hybrid methodology that first employs unsupervised clustering to label machine behavior patterns and then uses these labels to train supervised classifiers for predictive analysis. This approach bridges the gap between exploratory pattern discovery and predictive modeling, enabling a scalable, data-driven system for monitoring equipment health. The following section details the proposed method, including data preprocessing, clustering, labeling, and classification stages.

3. Proposed Method

This section outlines the methodology for analyzing industrial IoT data to identify operational patterns, establish performance benchmarks, and detect underperforming machines. The approach follows the CRISP-DM framework, comprising the phases of Business Understanding, Data Understanding, Data Preparation, Modeling, and Evaluation.

3.1. Business Understanding

The objective of this study is to uncover distinct behavioral patterns among industrial machines using clustering and classification techniques. Machines with similar operational profiles are grouped into behavioral categories such as *highly productive*, *normal operation*, *idle–stable*, or *fault-prone*. This segmentation enables early detection of underperforming machines and supports data-driven maintenance planning.

Because each subtechnology type exhibits different functional characteristics and operational ranges, that is the expected range of machine operational metrics (e.g., state durations or transition counts), benchmarks are established separately within each subtechnology. In this framework, clusters represent behavioral categories whose definitions emerge from data rather than predefined labels. By identifying these data-driven patterns, the methodology provides a foundation for targeted performance monitoring and actionable maintenance recommendations.

3.2. Data Understanding

3.2.1. Data Collection and Description. The dataset used in this study was compiled from multiple IoT data sources, including SQL Server Management Studio, a CRM database, and Azure Data Explorer. These sources were integrated in Power BI, where initial transformations and data cleaning were performed before exporting the unified dataset for analysis in Python. To ensure data privacy, machine serial numbers and subtechnology identifiers were anonymized in compliance with GDPR regulations.

The dataset contains approximately 36,106 daily records, each representing the aggregated operational behavior of a single machine over one day. Key attributes include operational durations across four machine states: *EXE*, *READY*, *FAIL*, and *POWER_OFF*. Additional features capture the frequency of state transitions (e.g., *N_Change_EXE_FAIL*, *N_Change_FAIL_READY*), as well as contextual variables such as manufacturing date, shipment date, and subtechnology type.

3.2.2. Data Exploration. Exploratory data analysis revealed that most numeric features were strongly **right-skewed**, as shown in Figure 1. This distribution indicates long tails toward higher values, a common pattern in machine usage data, where most units operate under normal conditions (short durations and few transitions), while a smaller subset experiences prolonged failures or abnormal behaviors. Boxplots confirmed this trend, showing narrow interquartile ranges and numerous high-value outliers.

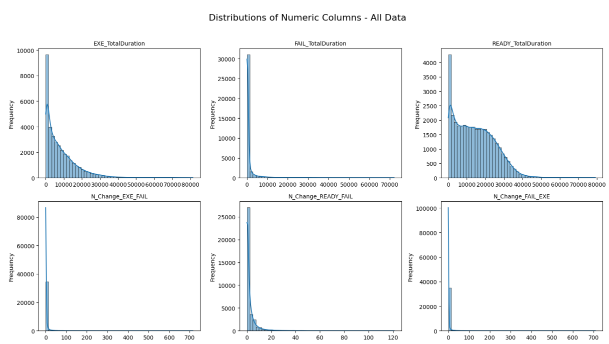

Figure 1. Distribution of key numeric features.

Because these outliers likely represent genuine abnormal events rather than data errors, they were retained for analysis. To reduce their overly strong effect on model training, scaling techniques were evaluated, and **MinMax scaling** was ultimately selected, as it experimentally produced clearer and more stable cluster boundaries.

3.3. Data Preparation

The data preparation phase ensured that the dataset was clean, consistent, and properly formatted for clustering and classification.

3.3.1. Data Cleaning. This step involved addressing missing values, verifying data types, identifying and handling outliers, and selecting relevant subsets for analysis.

Handling Missing Values. Missing values were first identified and assessed to understand their causes. Discussions with domain experts revealed that missing operational durations typically indicated zero time spent in a specific state rather than a recording error. Therefore:

- Missing operational duration values were imputed with zeros.
- Rows where all operational duration features were zero were removed, as they represented days when the machine was inactive or disconnected from the IoT system.
- Records missing `Subtechnology_Name_ITA` or `Shipment Date` were dropped, as they are missing at random or correspond to the machines still in production.

Outlier Detection. Numeric features exhibited strong right skewness, with many low-duration values and few extreme ones. Because these extremes likely represented meaningful rare events (e.g., long failures or abnormal operations), outliers were retained rather than removed. A *logarithmic transformation* was tested to reduce their influence, but it degraded clustering quality by compressing meaningful behavioral variation. Therefore, the transformation was discarded, and outlier effects were later mitigated through appropriate feature scaling.

Data Type Verification. All columns were verified for correct data types: numerical features were cast as integers or floats, while temporal fields (`Date`, `Shipment Date`, and `Manufacturing Date`) were converted to `datetime` objects for consistency.

Relevant Data Selection. Non-informative columns such as identifiers (`Serial Number`, `Date`) and metadata (`Shipment Date`, `Manufacturing Date`) were excluded from modeling features but retained for reference and interpretability in the final results. The analysis focused on two representative subtechnologies, `sub_3` and `sub_4`, chosen for their sufficient data volume and operational relevance.

3.3.2. Data Construction. After cleaning, feature transformation and scaling were performed to enhance model interpretability and performance.

Feature Engineering. Operational durations were converted from seconds to minutes to improve readability for domain experts. Transition count features were preserved to maintain behavioral integrity, as they capture how often machines shift between operational states.

Feature Scaling. Because features varied widely in magnitude (e.g., durations in thousands vs. transition counts in tens), scaling was essential. Both *RobustScaler* and *MinMaxScaler* were tested. Although robust scaling mitigated outlier effects, it led to poorer clustering separation. Therefore, *MinMax scaling* was selected, as it provided clearer cluster separation and more interpretable results.

3.4. Feature Selection

After data cleaning and construction, feature selection was performed to eliminate redundant or uninformative variables prior to clustering and classification.

Variance Analysis. Variance thresholding was explored, but after applying *MinMaxScaler* (which normalizes each feature to the $[0, 1]$ range), many features showed very small numerical variance, and columns that were constant within a subset became exactly zero-variance. Using logarithmic/robust scaling increased variance numerically, but degraded clustering separability. Since the features capture meaningful behavior, none were removed by variance thresholding.

Correlation Analysis. To minimize redundancy, a Pearson correlation matrix was computed. When two features had a correlation coefficient above 0.7, one was removed to retain only the most representative variable. This step improved feature independence and enhanced the stability of subsequent clustering and classification models.

3.5. Data Modeling

The modeling phase combines **unsupervised clustering** and **supervised classification** to identify and predict machine behavioral patterns. First, clustering algorithms were applied to discover natural groupings of machines based on operational characteristics. The resulting cluster assignments were then interpreted and used as class labels to train predictive models, forming a hybrid framework that links exploratory pattern discovery with classification of new, unseen data.

3.5.1. Clustering Phase. The **K-Means** algorithm was selected as the primary clustering method for its simplicity, scalability, and ability to segment machines into distinct behavioral groups based on feature similarity. The optimal number of clusters (k) was determined using two complementary techniques:

- **Elbow Method:** Identifies the point where additional clusters yield diminishing reductions in within-cluster variance.
- **Silhouette Analysis:** Evaluates how well each data point fits within its cluster compared to others, with higher scores indicating clearer separation.

Separate K-Means models were trained for each subtechnology (`sub_3` and `sub_4`), ensuring that differences in machine design and usage patterns were captured independently. Principal Component Analysis (PCA) was then applied to reduce dimensionality and visualize the clusters in a three-dimensional space.

The **DBSCAN** algorithm was also tested to validate cluster robustness and detect irregular or noisy behaviors. While DBSCAN effectively identified sparse outlier points, these samples were too few and inconsistent to represent generalizable behavioral groups. Moreover, since DBSCAN labels some points as noise (without assigning them to any cluster), its output was unsuitable for producing complete labels required for supervised classification. In contrast, K-Means assigns every observation to a cluster, providing comprehensive and interpretable labeling for downstream modeling. Therefore, K-Means was selected as the final clustering method.

Each cluster was then analyzed based on mean operational durations and transition counts to confirm distinct behavioral patterns. These validated clusters formed the behavioral categories used as labels in the subsequent supervised classification phase.

3.5.2. Labeling and Classification Phase. After clustering, the identified groups were interpreted and appended to the dataset as categorical labels. These labels were then used as target variables in supervised learning models to predict machine behavioral categories for unseen data.

Three classification algorithms were evaluated to predict machine behavior categories derived from the clustering phase. To prevent data leakage, all models were implemented within a unified *Pipeline* combining feature scaling and the classifier, ensuring that the `MinMaxScaler` was fitted only on the training folds during cross-validation. Model optimization was performed using `GridSearchCV` with stratified 5-fold cross-validation, maximizing the macro-averaged F1-score. Because both missed failure detections (false negatives) and false alarms (false positives) carry significant operational costs, the F1-score was selected as the primary evaluation metric, balancing precision and recall to ensure reliable performance.

- **SVM:** Selected for its robustness and effectiveness in high-dimensional spaces. Both `linear` and `rbf` kernels were tested, with hyperparameters C, γ, and class weighting (`balanced`) tuned over logarithmic scales.
- **Decision Tree:** Chosen for its interpretability and ability to model non-linear decision boundaries. The grid search explored parameters such as `criterion` (`gini`, `entropy`), `max_depth`, `min_samples_split`, `min_samples_leaf`, and `max_features`.
- **KNN:** Used as a distance-based baseline model. Hyperparameters including `n_neighbors` (1–50), `weights` (`uniform`, `distance`), and `p` (1, 2), corresponding to Manhattan and Euclidean distances, were systematically evaluated. Scaling via `MinMaxScaler` ensured fair distance computations.

Each model was evaluated on the test set using accuracy, and weighted precision, recall, and F1-score. Confusion matrices were used to visualize the model's performance for each class, while learning curves illustrated model generalization by comparing training and cross-validation scores across varying sample sizes.

Among all classifiers, the SVM achieved the highest overall F1-score and demonstrated the best balance between bias and variance, making it the final model selected for deployment. The optimized SVM pipeline, including the feature scaling step, was saved using `joblib` for reproducibility and deployment. This integrated pipeline allows new IoT machine data to be directly scaled and classified into behavioral categories.

4. Results

This section presents the experimental results of the proposed hybrid framework, including the outcomes of data exploration, clustering, and classification. The analysis was performed separately for each subtechnology, with representative results from `sub_3` shown below.

4.1. Feature Correlation Analysis

Before clustering, feature correlations were examined to identify redundancy among numeric variables. Figure 2 shows the correlation heatmap for `sub_3` after scaling. Strong positive correlations were observed among transition related features, such as `N_Change_EXE_READY` and `N_Change_READY_EXE`, indicating predictable bidirectional state transitions. Highly correlated variables were removed to reduce redundancy and improve clustering clarity.

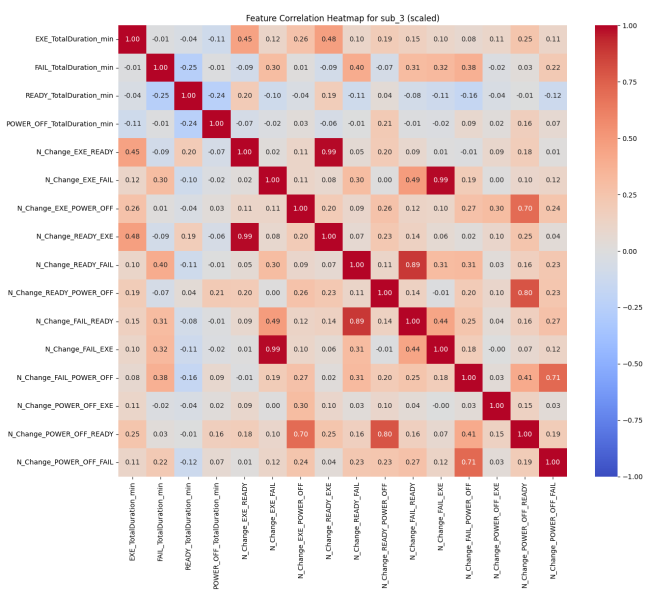

Figure 2. Feature correlation heatmap for `sub_3` (scaled data).

4.2. Clustering Evaluation

The optimal number of clusters was determined using the elbow method and silhouette analysis (Figure 3). Both indicated $k = 4$, where further increases yielded minimal improvement in within-cluster variance. The corresponding silhouette score ($s \approx 0.284$) confirmed a reasonable balance between compactness and separation, so $k = 4$ was selected for further analysis.

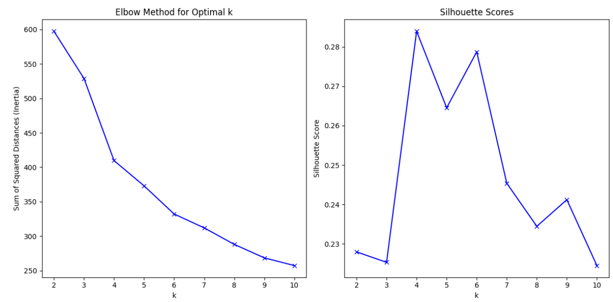

Figure 3. Elbow method (left) and silhouette analysis (right) for optimal k.

4.3. Clustering Visualization and Interpretation

The final K-Means model with $k = 4$ was applied to the `sub_3` dataset. The data were projected onto three principal components using PCA (Figure 4), revealing distinct and meaningful cluster separations among machines.

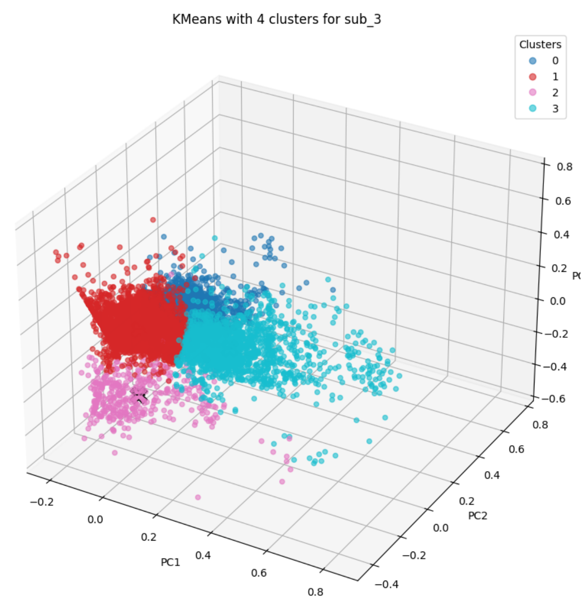

Figure 4. 3D PCA visualization of K-Means clusters for `sub_3`.

Cluster statistics (Table 1) and feature distributions (Figure 5) reveal clear behavioral differences among the four groups. Cluster 3 shows high execution and minimal failure durations (productive machines), Cluster 2 exhibits prolonged failures (fault-prone), Cluster 1 corresponds to idle but stable operation, and Cluster 0 represents balanced normal behavior. These interpretations were used to assign semantic behavioral labels:

- **Cluster 0:** Balanced Use
- **Cluster 1:** Idle
- **Cluster 2:** Faulty
- **Cluster 3:** Productive (Benchmark)

These labeled categories formed the basis for the subsequent supervised classification phase.

TABLE 1. Cluster summary statistics for `sub_3`

(k)	Total_N	Distinct_N	EXE_D	FAIL_D	READY_D	POWER_OFF_D	N_EXE_FAIL	N_READY_FAIL
0	4809	514	1:55:43	0:03:36	6:35:24	0:27:05	1.29	1.53
1	3594	600	1:06:56	0:06:30	1:55:30	1:54:36	1.30	1.24
2	419	62	2:16:57	4:51:55	1:07:08	0:49:10	32.63	21.04
3	2181	332	7:03:29	0:09:17	3:26:35	0:39:47	6.75	2.86

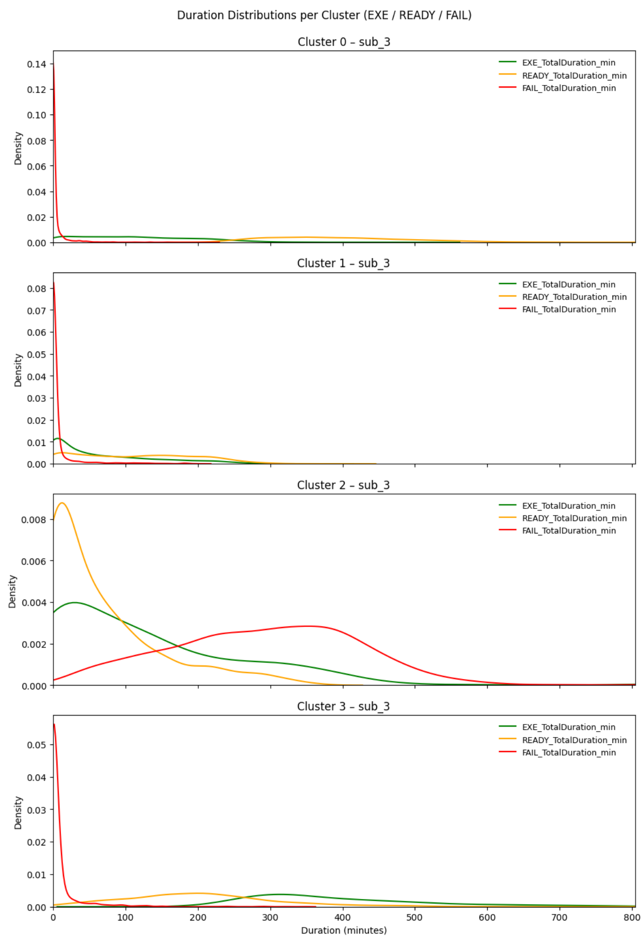

Figure 5. Distribution of key duration features across clusters for `sub_3`.

4.4. Classification Results

Using the behavioral labels derived from clustering, three classifiers, **SVM**, **Decision Tree**, and **KNN**, were trained and tuned via stratified 5-fold cross-validation using macro F1 as the scoring criterion. Final models were evaluated on a held-out test set using accuracy, weighted precision, recall, and F1 (Table 2). All achieved high predictive performance, though their learning dynamics reveal distinct generalization behaviors.

Figure 6 illustrates the models' predictive behavior via confusion matrices (left) and learning curves (right), highlighting accuracy and generalization trends. Figure 7 further compares their discriminative performance using multiclass ROC curves.

TABLE 2. Test set performance of the three classifiers.

Model	Accuracy	w_Precision	w_Recall	w_F1
SVM	0.9941	0.9942	0.9941	0.9941
Decision Tree	0.9654	0.9654	0.9654	0.9654
KNN	0.9768	0.9768	0.9768	0.9768

The **SVM** demonstrated the strongest generalization, with a narrow train–validation gap (F1 $\approx$ 0.99–0.98). Its learning curve shows smooth convergence and stable performance, indicating that the linear kernel ($C = 100$) achieved an effective bias–variance balance. The ROC curve (AUC $\approx$ 1.0) confirms near-perfect class separability.

The **Decision Tree** performed well (F1 $\approx$ 0.965) but showed *mild overfitting*, as the training F1 was close to 1.0 while validation dropped to about 0.95. This reflects the tendency of deep trees to memorize training data. Its ROC curve (AUC $\approx$ 0.97) indicates strong, though less stable, discrimination compared to SVM.

The **KNN** achieved balanced performance (F1 $\approx$ 0.977, AUC $\approx$ 0.999) with a small train–validation gap, indicating *moderate overfitting* typical of instance-based learning. Despite this, the model generalized well and effectively distinguished between classes.

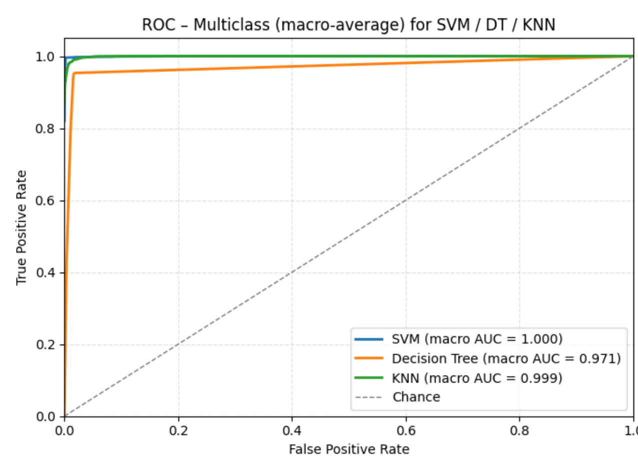

Figure 7. ROC Comparison Across Models

Summary. All models deliver high predictive accuracy, validating the quality of the cluster-derived behavioral labels. Among them, the **SVM** provides the most consistent and stable results, combining excellent accuracy with minimal overfitting. The final SVM pipeline, comprising

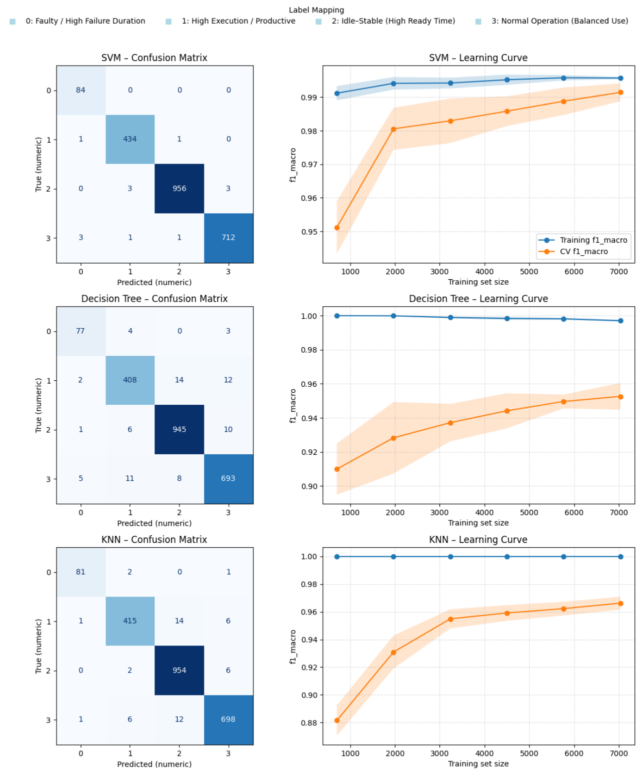

Figure 6. Model Diagnostics: Confusion Matrices and Learning Curves

`MinMaxScaler` and the optimized classifier, was saved using `joblib` for reproducibility and can directly process new IoT machine data without manual preprocessing.

5. Conclusion

This work presented a complete data-driven framework for understanding and predicting machine operational behaviors using industrial IoT data. The approach combined **unsupervised K-Means clustering** for behavioral segmentation with **supervised classification** for pattern recognition in new data. The clustering phase successfully revealed distinct machine usage profiles, forming the basis for meaningful behavioral labels.

Three classification algorithms, SVM, Decision Tree, and KNN, were trained to reproduce these cluster-derived labels. Comprehensive cross-validation and test evaluation demonstrated that all models achieved strong predictive accuracy, confirming the consistency and interpretability of the discovered clusters. Among them, the **SVM** achieved the highest F1-score and best generalization performance, with negligible overfitting and near-perfect class separability.

The final SVM pipeline, integrating `MinMaxScaler` and the optimized model, was exported using `joblib` to ensure reproducibility and seamless deployment. This enables direct classification of incoming IoT machine data, supporting real-time behavioral monitoring and predictive maintenance.

Overall, the proposed hybrid methodology, linking clustering insights with classification, effectively transforms raw operational data into actionable intelligence. It provides a scalable and interpretable foundation for real-time performance benchmarking and predictive maintenance, enabling early detection of underperforming machines and proactive operational decision-making in smart manufacturing environments.

Limitations and Future Work. Despite its promising results, the study has some limitations. First, the analysis was based on aggregated daily data from a limited number of subtechnologies, which may restrict model generalization across broader equipment types. Second, as the cluster labels were derived from unsupervised learning and subsequently interpreted by domain experts, the process may be subject to semantic labeling bias and limited reproducibility. In addition, as machine behaviors evolve over time, the clustering process should be periodically repeated to prevent *concept drift* and ensure that the behavioral categories remain representative of current operating conditions. Future work will focus on expanding the dataset to include more subtechnologies and temporal granularity, integrating advanced clustering algorithms, and exploring deep learning or ensemble approaches for improved predictive accuracy. Additionally, integrating the framework into an automated monitoring dashboard could enhance real-time decision support in industrial environments.

References

[1] A. SHARMA, B. I. ADEKUNLE, J. C. OGEAWUCHI, A. A. ABAYOMI, and O. ONIFADE, “Iot-enabled predictive maintenance for mechanical systems: Innovations in real-time monitoring and operational excellence,” 2019.

[2] T. Blachowicz, J. Wylezek, Z. Sokol, and M. Bondel, “Real-time analysis of industrial data using the unsupervised hierarchical density-based spatial clustering of applications with noise method in monitoring the welding process in a robotic cell,” *Information*, vol. 16, no. 2, 2025. [Online]. Available: https://www.mdpi.com/2078-2489/16/2/79

[3] I. Hector and R. Panjanathan, “Predictive maintenance in industry 4.0: a survey of planning models and machine learning techniques,” *PeerJ Computer Science*, vol. 10, p. e2016, 2024.

[4] M. Cioch, M. Kulisz, and A. Gola, “Comparison of machine learning methods in predictive maintenance of machines,” *Advances in Science and Technology Research Journal*, vol. 19, no. 11, pp. 33–44, 2025. [Online]. Available: https://doi.org/10.12913/22998624/208284

[5] K. Arun, “Evaluating machine learning algorithms for predictive maintenance in equipment,” *Global Journal of Engineering Innovations and Interdisciplinary Research*, vol. 5, no. 1, pp. 1–10, 2025. [Online]. Available: https://dx.doi.org/10.33425/3066-1226.1059

[6] A. Costa, E. Mastriani, F. Incardona, K. Munari, and S. Spinello, “Predictive maintenance study for high-pressure industrial compressors: Hybrid clustering models,” *arXiv preprint arXiv:2411.13919*, 2024.