

Lab Project
Final Report
April 3, 2025

Team 2 - Feature Importance in Time
Series For CNC Machine Energy
Consumption

Kavyashree Byalya Nanjegowda – 249291
Nitin Bharadwaj Nataraj – 248993
Ranjith Mahesh – 249303

Team 2 - Feature Importance in Time Series For CNC Machine Energy Consumption

– Lab Project: Final Report –

Kavyashree Byalya Nanjegowda – 249291

Nitin Bharadwaj Nataraj – 248993

Ranjith Mahesh – 249303

April 3, 2025

Abstract

This project aims to support energy-efficient manufacturing by investigating the application of various feature importance techniques to identify key factors influencing energy usage in Computer Numerical Control (CNC) operations. By leveraging time-series data and machine learning models such as Feedforward Neural Networks (FNN), Long Short-Term Memory (LSTM), Random Forests (RF), Decision Trees (DT) and XGBoost (XGB), we analyzed the effectiveness of interpretability methods including Integrated Gradients (IG), Permutation Importance (PI), Windowed Feature Importance in Time (WINIT), and Local Interpretable Model-Agnostic Explanations (LIME). The study involved systematic dataset preprocessing through correlation-based filtering, followed by method-specific model training and feature ranking. Each technique was evaluated across different datasets in terms of test loss, execution time, and interpretability. Comparative analysis revealed that Integrated Gradients (FNN with Correlation) and WINIT (LSTM with Correlation) consistently delivered high accuracy, offering balanced interpretability and performance. The findings provide valuable insights for energy-aware process optimization and establish a scalable framework for applying explainable AI in industrial environments.

Contents

1	Introduction	5
1.1	Overview of CNC Machines	5
1.2	Energy Efficiency and its Relevance in Modern Manufacturing	5
1.3	Role of Feature Importance Analysis in Energy Prediction	5
1.4	Overview of Employed Feature Importance Techniques	6
1.4.1	Integrated Gradients (IG)	6
1.4.2	Permutation Importance (PI)	7
1.4.3	WINIT (Windowed Feature Importance in Time)	7
1.4.4	LIME (Local Interpretable Model-agnostic Explanations)	8
1.5	Overview of Dataset	8
2	Task Description and Project Goals	9
2.1	Task Description	9
2.2	Project Goals and Specific Objectives	10
2.3	Broader Goals and Impact	10
3	Scientific/Technical Status at Project Start	11
3.1	Team Expertise	11
3.2	Computational Resources	11
3.3	Software Tools	11
4	Literature Foundations and Methodological Insights	12
5	Project Plan	13
5.1	Key Tasks and Workflow Breakdown	14
6	Project Execution and Scientific/Technical Results	15
6.1	Dataset Understanding and Preprocessing	16
6.1.1	Initial Cleanup and Feature Filtering	16
6.1.2	Target Selection	17
6.1.3	Feature Scaling	17

6.1.4	Correlation-Based Filtering	17
6.2	Implementation of Feature Importance Techniques	22
6.3	Execution Methodology	22
6.4	Comparison and Analysis of Results	23
6.4.1	Technique-wise Best Performers	24
6.4.2	Technique Suitability by Dataset Type	28
6.5	Team Contribution and Collaboration	29
6.5.1	Literature Survey and Problem Understanding	29
6.5.2	Individual Implementation Responsibilities	29
6.5.3	Evaluation and Comparative Analysis	30
7	Comparison of Project Plan and Execution	30
7.1	Challenges and Deviations from Initial Plan	30
7.2	Adaptations Made During Project Execution	30
8	Usability and Limitations of Results	31
8.1	Practical Applications of the Results	31
8.2	Constraints and Limitations	31
9	Conclusions	32
9.1	Lessons Learned	32
9.2	Future Work	33

List of Figures

1	CNC Machine with key components (source: www.cncmasters.com)	5
2	Project plan Gantt Chart	13
3	Overview of Proposed Methodology	16
4	Correlation Heatmap for DMC2_AL_CP1 Dataset	18
5	Correlation Heatmap for DMC2_AL_CP2 Dataset	19
6	Correlation Heatmap for DMC2_S_CP1 Dataset	20
7	Correlation Heatmap for DMC2_S_CP2 Dataset	21
8	Home Page of Streamlit Application	23
9	Test Loss, Execution Time per Dataset for IG	24
10	Test Loss, Execution Time per Dataset for WinIT	25
11	Test Loss, Execution Time per Dataset for LIME	26
12	Test Loss, Execution Time per Dataset for PI	27

1 Introduction

1.1 Overview of CNC Machines

Computer Numerical Control (CNC) machines are pivotal components in modern manufacturing, automating tasks such as milling, drilling, cutting, and turning with exceptional precision. These systems follow detailed CAD/CAM instructions, ensuring repeatability, minimal human intervention, and high accuracy. However, these benefits come with significant energy demands, making energy consumption a key consideration in large-scale production environments. With sustainability becoming an industrial imperative, optimizing energy usage in CNC machining is not only economically beneficial but also environmentally responsible.

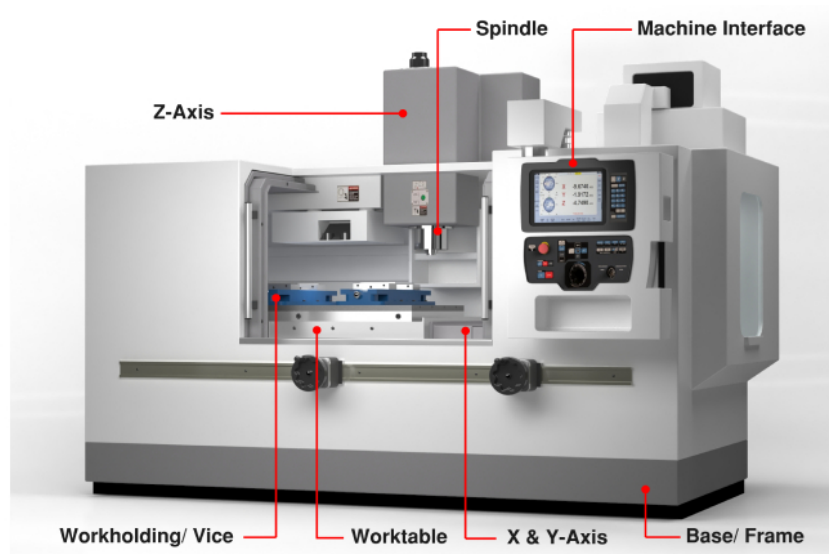


Figure 1: CNC Machine with key components (source: www.cncmasters.com)

1.2 Energy Efficiency and its Relevance in Modern Manufacturing

The manufacturing sector accounts for a substantial portion of global energy consumption. Within this context, CNC machines represent a critical node due to their continuous operation and reliance on high-powered actuators and controllers. Understanding energy usage patterns and their driving factors is essential for effective resource management. By identifying high-impact operational parameters, manufacturers can adopt data-driven strategies to minimize wastage and reduce their carbon footprint. Predictive energy modeling facilitates proactive decision-making, operational planning, and predictive maintenance scheduling.

1.3 Role of Feature Importance Analysis in Energy Prediction

Feature importance techniques offer insight into which variables most influence a machine learning model's output, in this case, energy consumption. These methods help bridge the

gap between model performance and interpretability, offering practical value in process optimization. Feature attribution methods like LIME [5], Permutation Importance provide a model-agnostic lens into the mechanics of prediction, while techniques like Integrated Gradients [7] offer gradient-based interpretability for differentiable models. Newer approaches like WinIT [3] consider temporal dynamics in time series datasets, enhancing explainability. By analyzing time-dependent feature impacts, organizations can identify critical parameters influencing energy usage, optimize control strategies, and achieve sustainable operations.

1.4 Overview of Employed Feature Importance Techniques

In this project, four prominent feature importance techniques were employed to analyze and interpret energy usage patterns in CNC machine operations. These methods were chosen for their varying underlying principles, offering a broad comparative framework for evaluating model interpretability in time-series industrial data.

1.4.1 Integrated Gradients (IG)

Integrated Gradients is a gradient-based attribution method designed for neural networks. It quantifies feature importance by integrating the gradients of a model's predictions with respect to input features, computed along a straight-line path from a baseline input to the actual input. This approach satisfies axioms like sensitivity and implementation invariance, making it theoretically grounded and suitable for complex models such as LSTMs and FNNs. IG is particularly useful in capturing subtle interactions in data, especially in cases involving temporal sequences [7].

Advantages:

- Satisfies theoretical properties, ensuring reliable attributions [7].
- Produces smooth, less noisy attributions by integrating over a path.
- Particularly effective with deep models like FNN and LSTM.
- Captures subtle feature interactions, including nonlinear dependencies.

Disadvantages:

- Applicable only to differentiable models; not model-agnostic.
- Results are sensitive to the choice of baseline input.
- Outputs can be challenging to interpret without domain expertise.

1.4.2 Permutation Importance (PI)

Permutation Importance is a model-agnostic technique that evaluates feature importance by measuring the change in model performance when the values of a specific feature are randomly shuffled. This process breaks the association between the feature and the target variable, allowing assessment of the drop in predictive accuracy due to the feature's absence. It is particularly suitable for models like decision trees and ensemble methods, and offers a robust, interpretable approach, although at a higher computational cost [1].

Advantages:

- Simple to implement and interpret, reflecting real impact on model output [3].
- Compatible with any predictive model, including ensemble methods.
- Offers robust results when features are uncorrelated.

Disadvantages:

- Computationally expensive due to repeated model evaluations.
- Suffers from distortion in the presence of correlated features.
- Does not explicitly consider temporal dependencies in time-series data.

1.4.3 WINIT (Windowed Feature Importance in Time)

WINIT is a feature removal-based explainability method specifically designed for time-series data. It computes the importance of each observation by analyzing the impact on predictions over a temporal window. By aggregating effects over multiple time steps, WINIT effectively captures delayed influences and long-range dependencies that are typical in industrial processes. Its design makes it particularly suitable for applications where temporal causality is critical, such as in CNC energy analysis [6].

Advantages:

- Tailored for time-series scenarios, effectively capturing temporal causality [5].
- Aggregates impact across multiple time steps, providing richer insights.
- Suitable for scenarios where feature influence unfolds over time.

Disadvantages:

- Computationally intensive due to repeated window-based analysis.
- Requires careful tuning of window size and scoring function.
- Interpretation of results can be complex and less intuitive without visualization.

1.4.4 LIME (Local Interpretable Model-agnostic Explanations)

LIME is a local surrogate model approach that approximates the behavior of complex models by training interpretable models (like linear regressors or decision trees) on perturbed samples around a prediction instance. This enables the estimation of local feature importances in a model-agnostic fashion. For tabular data, LIME generates explanations by drawing samples from replacement distributions, making it flexible for use with a variety of model architectures including LSTM, RF, and XGBoost [2].

Advantages:

- Model-agnostic and flexible; applicable to any classifier or regressor [4].
- Intuitive and user-friendly, especially for individual predictions.
- Valuable for debugging and understanding model behavior in production.

Disadvantages:

- Provides only local explanations, which may not generalize globally.
- Can be unstable—different runs may yield different attributions.
- May perform poorly in high-dimensional or correlated feature spaces.
- Requires significant computational overhead for sampling and model fitting.

1.5 Overview of Dataset

This study uses four real-world time-series datasets collected from a DMC2 model CNC machine. Each dataset represents a unique scenario defined by the combination of workpiece material (Aluminum or Steel) and Control Program (CP1 or CP2). This diversity enables comprehensive evaluation of machine learning models and interpretability techniques.

1. DMC2_AL_CP1.csv

Purpose: Baseline scenario for aluminum machining using Control Program 1 (CP1).

Content: Time-series sensor data including LOAD, TORQUE, ENC_POS, CTRL_DIFF2, DES_POS, CURRENT, VELOCITY, etc.

Insights:

- Reflects consistent energy consumption in standard aluminum operations.
- Serves as a benchmark for model stability and feature interpretability.

2. DMC2_AL_CP2.csv

Purpose: Evaluates the impact of different machining instructions on energy patterns.

Differences from CP1: Variations in spindle speed, control logic, and operation timing.

Relevance:

- Enables comparison of feature importance techniques across programs with the same material.

3. DMC2_S_CP1.csv

Material Characteristics: Steel introduces greater resistance, load, and thermal variability.

Sensor Inputs: Similar to other datasets, but with higher variability in sensor readings.

Use Case:

- Highlights how material properties impact energy usage and model prediction accuracy.

4. DMC2_S_CP2.csv

Operational Complexity: Combines hard material (Steel) with a complex control program (CP2).

Analytical Value:

- Ideal for testing robustness of feature importance methods under dynamic conditions.

Common Dataset Characteristics

- **Time-Series Format:** All datasets include time-indexed sensor readings during machining.
- **Target Variable:** Energy consumption (or proxies like power/current) used in supervised learning.

2 Task Description and Project Goals

2.1 Task Description

The primary task of this project is to investigate and evaluate the effectiveness of selected feature importance techniques in modeling and interpreting energy consumption patterns of CNC machines. The focus lies on applying four prominent interpretability methods— Integrated Gradients, WinIT, LIME, Permutation Importance —across multiple real-world time-series datasets.

Each dataset represents distinct operational scenarios based on material type (Aluminum or Steel) and control program (CP1 or CP2), introducing variability in both process dynamics and energy profiles. The objective is to assess how these feature attribution methods perform under different machining contexts and levels of data correlation.

A key component of this task is the comparison of methods:

- Across different datasets to evaluate generalizability and robustness.
- With and without applying correlation-based feature filtering to observe sensitivity to multicollinearity.
- In terms of interpretability, computational cost, and practical relevance to CNC energy optimization.

Ultimately, the goal is to determine which feature importance technique provides the most reliable and actionable insights for CNC machining applications, particularly for supporting energy-aware decision-making in industrial settings.

2.2 Project Goals and Specific Objectives

The primary aim is to investigate which operational parameters most influence CNC machine energy consumption. This is achieved through implementing and comparing state-of-the-art interpretability techniques.

- Implement Integrated Gradients, WinIT, LIME and Permutation Importance on CNC datasets.
- Train machine learning models capable of learning temporal dependencies.
- Rank features by their impact on energy consumption patterns.
- Compare interpretability methods in terms of stability, accuracy, and usability.
- Propose actionable recommendations for energy optimization.

2.3 Broader Goals and Impact

Beyond technical evaluation, the project aspires to contribute to sustainable manufacturing practices. By identifying high-impact features driving energy use, the findings can:

- Inform energy-aware operation strategies.
- Guide development of more efficient CNC control algorithms.
- Contribute benchmark results to the academic field of interpretable time-series modeling.

3 Scientific/Technical Status at Project Start

The project is executed by a multidisciplinary team with expertise in machine learning, time-series analysis, and industrial process modeling. Each team member brings a unique set of skills that contribute to the successful execution of this research.

3.1 Team Expertise

Ranjith Mahesh has prior experience working with CNC machines during his undergraduate studies. Currently pursuing an MSc in Digital Engineering with a specialization in Machine Learning, he has completed coursework in machine learning and data mining. His background in CNC machine operations enables a deeper understanding of energy consumption patterns and the technical aspects of feature importance in predictive modeling.

Kavyashree Byalya Nanjegowda brings over 3 years of experience as a software engineer and is currently pursuing a Master's degree in Digital Engineering. She has a strong foundation in machine learning, demonstrated through both academic coursework (Python, Machine Learning, Data Mining) and hands-on project work. This background equips her with the skills necessary to contribute effectively to research, implementation, and analysis tasks in this project.

Nitin Bharadwaj Nataraj is also an MSc student in Digital Engineering, specializing in machine learning. He has worked on various machine learning projects, including applications of deep learning, statistical modeling, and time-series forecasting. His expertise in handling industrial datasets and optimizing models for real-world deployment strengthens the technical execution of this research.

3.2 Computational Resources

The team is well-equipped with both macOS and Windows-based computing environments. The computational resources available include:

- Two Windows-based systems powered by Intel Core i5 12th Gen H-series processors with 16GB RAM — capable of handling complex model training and deployment tasks.
- One macOS-based system: MacBook Air (2024) with Apple M3 chip, optimized for high-performance AI computations, including training and real-time inference.

3.3 Software Tools

- **Programming Language:** Python — using libraries such as TensorFlow and PyTorch.
- **Development Tools:** Visual Studio Code (IDE), GitLab (version control project collaboration).

- **Visualization Tools:** Streamlit — for creating interactive dashboards and explainability demos.

By leveraging the combined technical expertise of the team and the available software and hardware infrastructure, the research is conducted efficiently and with high reliability. This foundation ensures high-quality results in analyzing feature importance for CNC energy consumption prediction.

4 Literature Foundations and Methodological Insights

The methodological design of this project is grounded in the evolving field of explainable machine learning, particularly in the domain of feature importance and model interpretability for time-series data. With increasing adoption of complex black-box models in industrial applications, ensuring their transparency and explainability has become a crucial research imperative. The theoretical and practical frameworks guiding this research are drawn from seminal and contemporary works in the field, providing a robust basis for model interpretability—especially in temporal prediction tasks such as CNC machine energy consumption forecasting.

A pivotal contribution to the field is the work by Covert et al. [2], who proposed a unified removal-based explanation framework that integrates a wide range of feature attribution techniques under a common theoretical umbrella. Their framework categorizes explanation methods along three fundamental dimensions:

- How features are removed from the model.
- What aspect of model behavior is analyzed.
- How the influence of each feature is summarized.

This classification helped formalize and unify methods such as LIME [5], Integrated Gradients [7], and Permutation Importance, which were previously seen as disparate. The concept of removal-based explanations—assessing the change in model behavior upon feature perturbation or exclusion—is especially relevant for evaluating model sensitivity to operational parameters in CNC machines. The structured comparison of explanation strategies provided by Covert et al. forms a critical lens through which our project evaluates the suitability and robustness of different interpretability techniques.

In parallel, the research by Leung et al. [3] introduced a novel advancement specifically tailored for time-series applications: the Windowed Feature Importance in Time (WinIT) method. Unlike traditional methods that assume feature observations are temporally independent, WinIT explicitly captures temporal dependencies—a characteristic intrinsic to time-series data but often overlooked in standard explainability frameworks. By evaluating the impact of a feature observation over a moving temporal window, WinIT accounts for both delayed effects

and correlated feature interactions over time. This is particularly significant in CNC energy consumption modeling, where changes in parameters such as torque or load may not result in immediate energy consumption changes but may influence downstream predictions.

Furthermore, WinIT leverages feature removal mechanisms combined with divergence-based scoring metrics such as prediction difference, Kullback–Leibler divergence (KL), and Jensen–Shannon divergence (JS), offering a more granular and temporally-aware representation of feature importance. These innovations align well with the structure of CNC datasets, where each row corresponds to sequential observations from sensors and control logs, and where interpreting longitudinal influence is critical for actionable insights.

The adoption of these frameworks in this project allows for a methodologically rigorous approach to interpretability, ensuring that the feature attribution results are not only statistically sound but also industrially meaningful. Moreover, these methods provide a comparative basis to assess not only which features are important, but also how and when they are important in a process that evolves over time—a key consideration in predictive maintenance and energy optimization in manufacturing.

In essence, the combined influence of Covert et al.’s structured taxonomy and Leung et al.’s temporal modeling innovations informs both our experimental design and the evaluation criteria, enabling a comprehensive and contextualized feature importance analysis suited for real-world industrial systems.

5 Project Plan

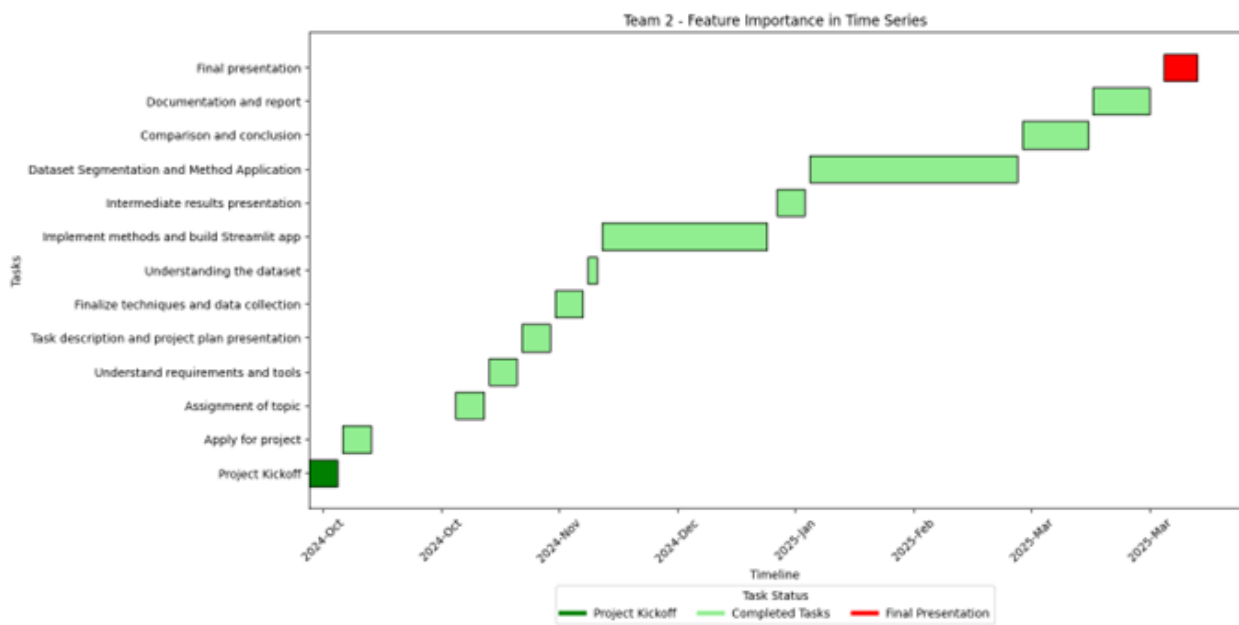


Figure 2: Project plan Gantt Chart

The project followed a well-defined and structured timeline, illustrated in the Gantt chart above. The project was initiated in October 2024 and concluded with the final presentation in March 2025. The Gantt chart outlines all major tasks, color-coded by their status, with the final presentation marked in red, the project kickoff in dark green, and all completed tasks in light green.

This timeline reflects a sequential yet iterative workflow, allowing the team to build incrementally upon earlier stages of progress. The project execution included exploratory analysis, technical implementation, evaluation, and documentation, ensuring a balanced distribution of workload across months.

5.1 Key Tasks and Workflow Breakdown

- **Project Kickoff and Team Formation (October 2024):**

The project began with the kickoff phase where team responsibilities were assigned, and the project scope was defined.

- **Initial Preparation and Requirement Analysis (October 2024):**

This phase included applying for the project, understanding requirements, identifying tools, and assigning the research topic. This foundational work ensured a clear understanding of project goals and deliverables.

- **Task Description and Project Plan Presentation (October 2024):**

The team formally presented the project plan, outlining techniques, timelines, and toolkits to be used. This presentation served as an early checkpoint to align academic and technical objectives.

- **Technique Finalization and Dataset Preparation (Late October – Early November 2024):**

The first major milestone was the finalization of feature importance techniques, including Integrated Gradients, WinIT, LIME and Permutation Importance. Simultaneously, dataset preprocessing was performed by filtering relevant attributes (e.g., LOAD, TORQUE, ENC_POS, etc.) and preparing them for time series modeling.

- **Feature Importance Techniques Implementation (November – December 2024):**

The implementation phase began with training LSTM models and integrating interpretability techniques. The team individually implemented each method and evaluated performance. This phase marks another milestone: achieving initial results for each selected technique, which provided a baseline for comparison.

- **Intermediate Project Presentation (January 2025):**

A key checkpoint was the intermediate results presentation, where the team shared initial findings, demonstrated the Streamlit application, and gathered feedback from supervisors. This milestone helped reflect on progress and identify areas needing improvement.

- **Refinement and Reiteration (January – February 2025):**

Based on feedback and performance gaps, methods that produced unsatisfactory results were revisited. This iterative phase involved redoing experiments, refining model parameters, and reevaluating interpretability outputs to ensure robust conclusions. This agile approach enhanced the quality and credibility of the results.

- **Documentation and Finalization (March 2025):**

The final phase focused on compiling documentation, structuring the report, integrating visualizations, and preparing for the final presentation. The final report and presentation preparation phase ensured that all technical findings were communicated clearly, both in written and visual formats.

Milestone	Description	Timeline
Finalization of Techniques	Selection of LIME, IG, Permutation, and WinIT	Oct–Nov 2024
Achieving Initial Results	Successful implementation and evaluation of methods	Nov–Dec 2024
Intermediate Results Presentation	Midpoint progress checkpoint	Dec 2024
Method Reiteration	Redoing underperforming techniques	Jan–Feb 2025
Documentation and Final Presentation	Report writing, application polish, and final delivery	Mar 2025

Table 1: Key project milestones and timelines

6 Project Execution and Scientific/Technical Results

To fulfill our objective of identifying the most influential factors affecting energy consumption in CNC operations, we adopted a structured methodology combining interpretable machine learning techniques with model evaluation. This allowed us to not only make accurate predictions but also understand the reasoning behind them.

We successfully implemented four categories of feature importance methods:

- Gradient-Based: Integrated Gradients (IG)
- Feature Removal-Based: Permutation Importance (PI)
- Model-Based Saliency: Windowed Feature Importance in Time (WinIT)
- Other Approaches: Local Interpretable Model-agnostic Explanations (LIME)

Our approach involved data preprocessing, techniques implementation, results evaluation and comparison to derive meaningful insights.

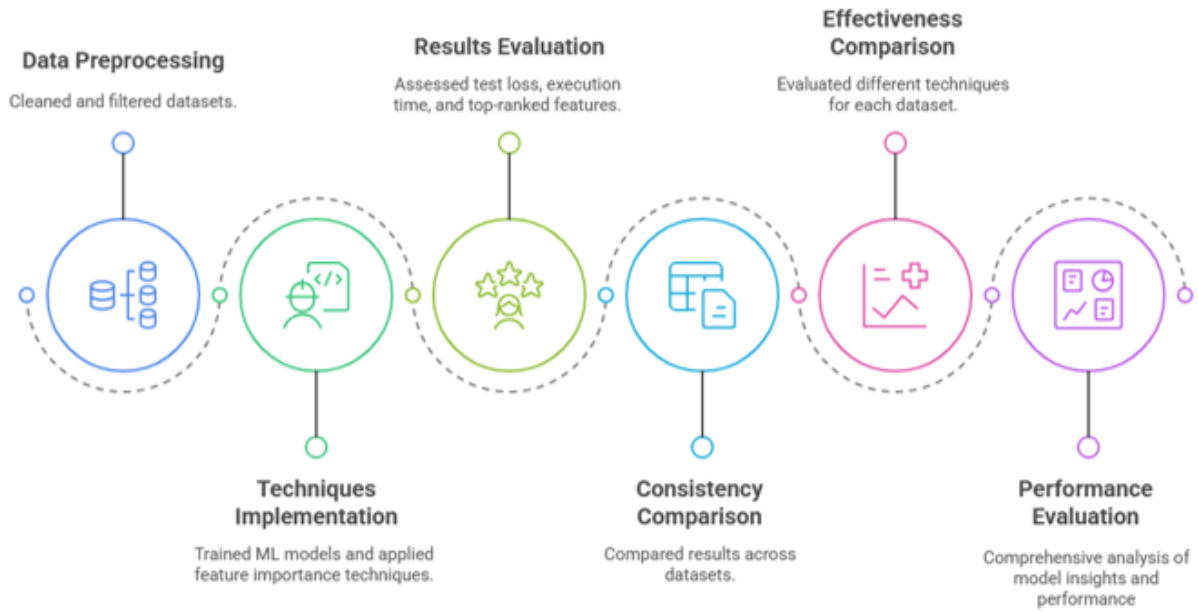


Figure 3: Overview of Proposed Methodology

6.1 Dataset Understanding and Preprocessing

The first phase involved analyzing and preparing the CNC machine datasets to ensure reliable and interpretable model training.

6.1.1 Initial Cleanup and Feature Filtering

To ensure the dataset was clean, relevant, and optimized for analysis, several preprocessing steps were performed. First, irrelevant or constant columns such as `CYCLE`, `A_DBD|0`, and unused `POWER` channels were identified and removed. These columns did not contribute meaningful variation to the dataset and could potentially introduce noise or redundancy in the analysis. Next, a total of 28 columns containing missing values (NaNs) were dropped. Since these columns had substantial missing data, retaining them could have negatively affected model performance or required complex imputation strategies that were unnecessary given the dataset's overall size.

Additionally, features associated with inactive axes, specifically those labeled with `|4` and `|5`, were excluded. These axes were not in use during the data collection process and thus did not contain any actionable information. After this filtering process, 52 relevant features were retained. These selected features corresponded to active axes—X, Y, Z—and the spindle, which were central to the operation and analysis of the system. This careful selection of features helped streamline the dataset and ensured that only important features were carried forward for further processing and model development.

6.1.2 Target Selection

For the purpose of modeling and analysis, four target variables were selected from the dataset: `CURRENT|1`, `CURRENT|2`, `CURRENT|3`, and `CURRENT|6`. These specific features represent the electric current consumption associated with each of the active axes (X, Y, Z) and the spindle, respectively. They were chosen because they serve as effective proxies for the energy usage or power draw of the system components during operation. Monitoring and predicting these current values is crucial for understanding the machine's energy behavior, enabling the development of models aimed at optimizing performance, detecting anomalies, or improving energy efficiency.

6.1.3 Feature Scaling

To bring all input features onto a common scale and facilitate effective model training, feature scaling techniques were applied. The input features were standardized using `StandardScaler`, which transforms each feature to have zero mean and unit variance. This helps improve model convergence and performance, particularly for algorithms sensitive to feature magnitudes.

Additionally, the target variables—`CURRENT|1`, `CURRENT|2`, `CURRENT|3`, and `CURRENT|6`—were normalized using `MinMaxScaler` to map their values within a fixed range, typically between 0 and 1. This normalization ensured that the output variables remained bounded and comparable, which is especially useful for regression tasks.

6.1.4 Correlation-Based Filtering

To assess the impact of feature redundancy on model performance, two parallel approaches were employed. In the first approach, correlation-based filtering was applied to eliminate highly correlated features. Specifically, features exhibiting a Pearson correlation coefficient greater than 0.9 with any other feature were removed. This helped reduce multicollinearity and ensured that redundant information did not bias the model or lead to overfitting.

In contrast, the second approach retained all features without applying any correlation filtering. This was done to preserve the full range of feature interactions and assess whether complex relationships between correlated variables could contribute positively to the model's predictive performance.

The figures below represent the correlation heatmaps for the four different datasets used in the study. These visualizations help in identifying pairs of features with high correlation (absolute value greater than 0.9), which were considered for removal in the correlation-based filtering process.

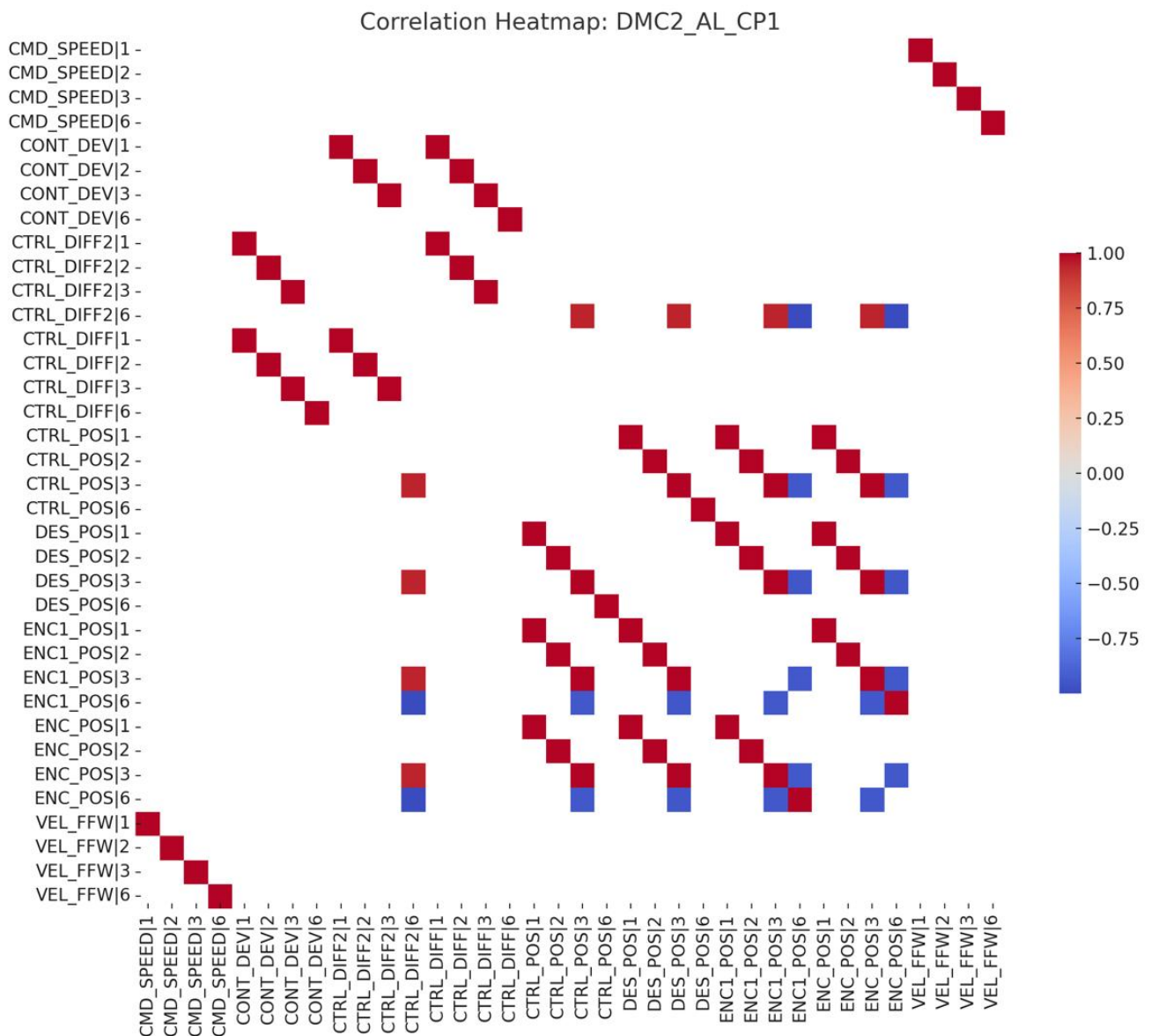


Figure 4: Correlation Heatmap for DMC2_AL_CP1 Dataset

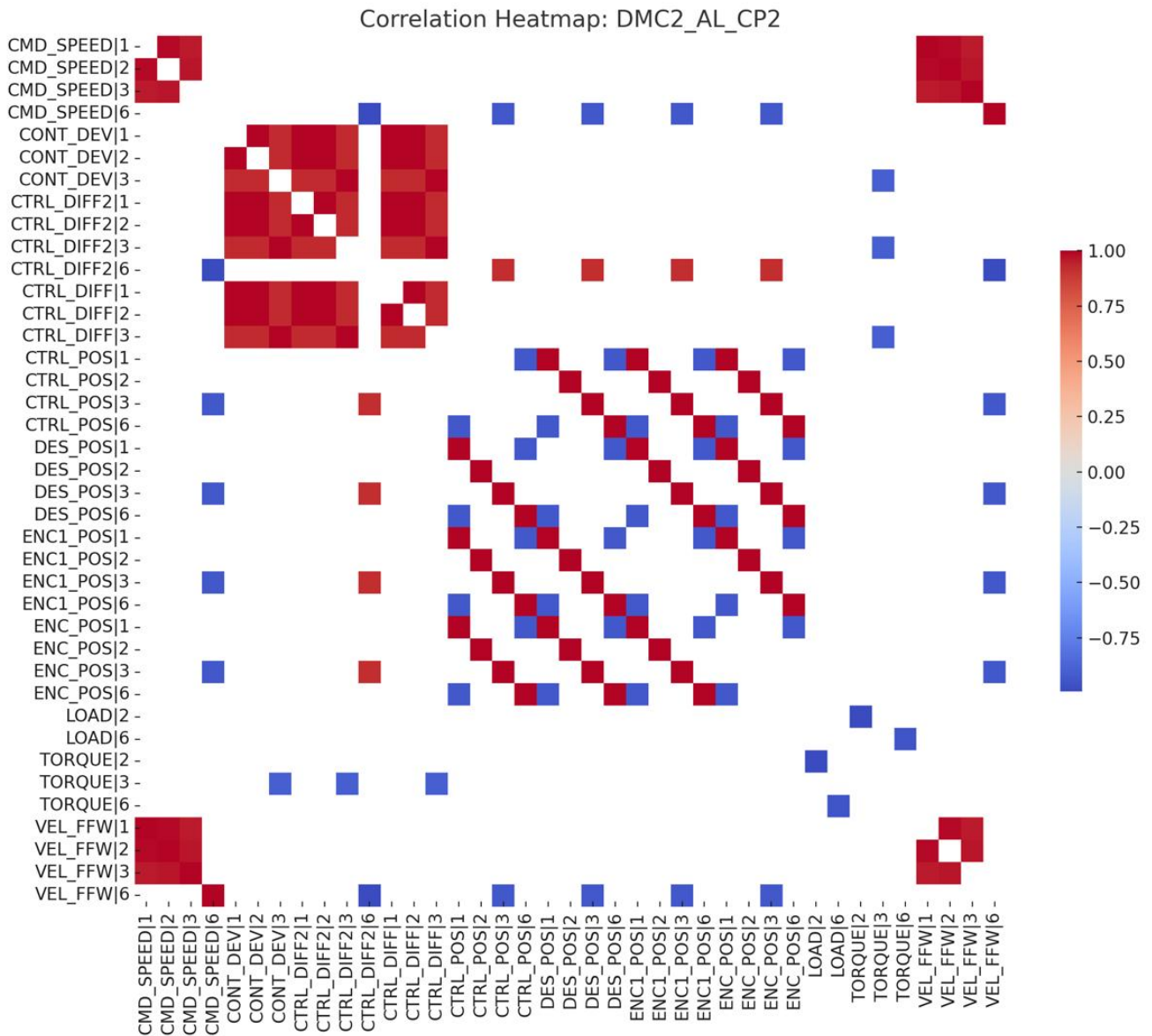


Figure 5: Correlation Heatmap for DMC2_AL_CP2 Dataset

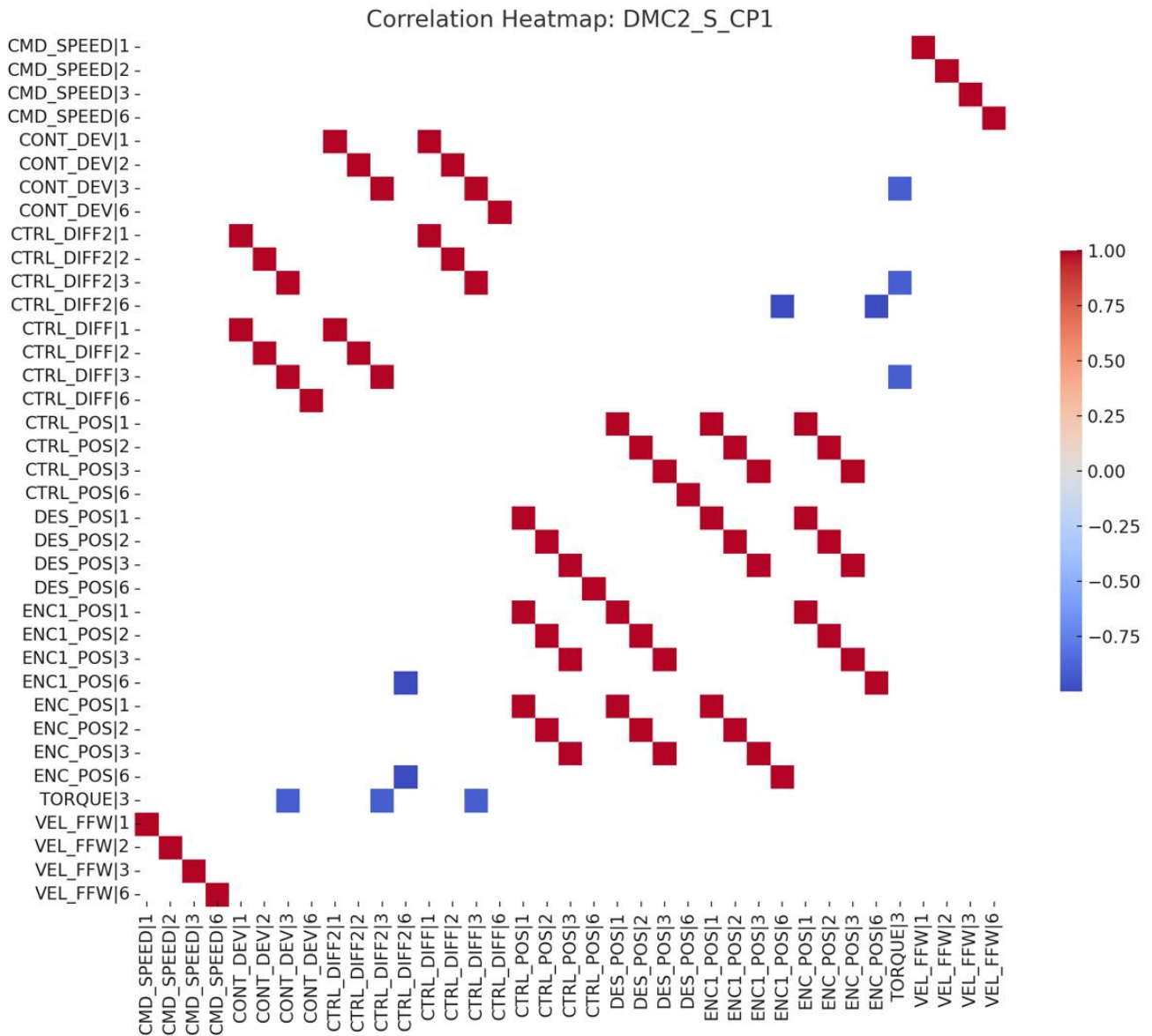


Figure 6: Correlation Heatmap for DMC2_S_CP1 Dataset

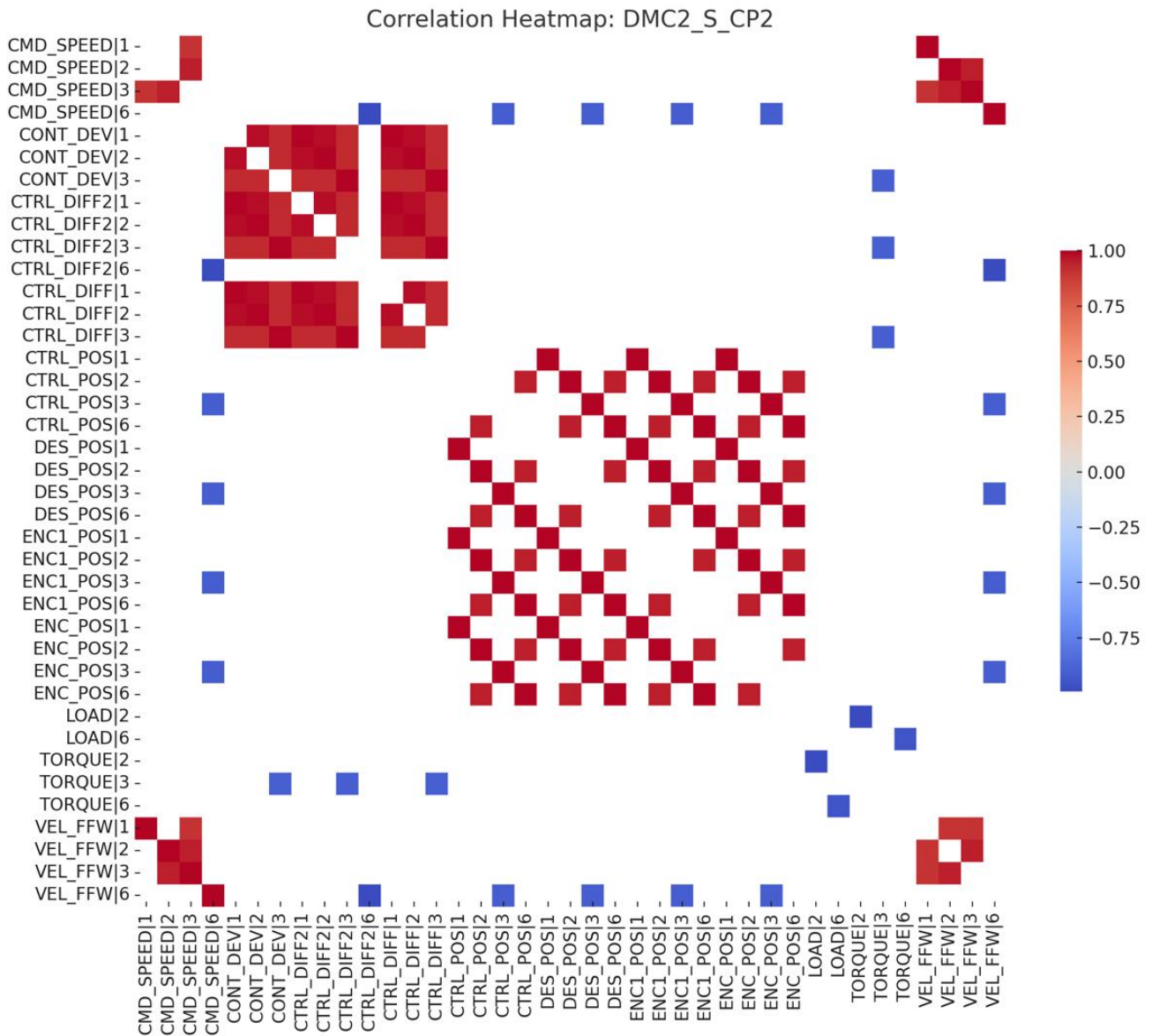


Figure 7: Correlation Heatmap for DMC2_S_CP2 Dataset

6.2 Implementation of Feature Importance Techniques

The selected feature importance techniques were implemented independently and applied across all datasets under two preprocessing settings: with and without correlation filtering to evaluate their performance in identifying critical features influencing CNC energy consumption. Each technique was executed through structured Python functions, which handled all stages—from data loading and model training to computation of feature importance and result storage.

For this project, models were selected based on the nature and requirements of each feature importance technique. Integrated Gradients (IG) were applied on Feedforward Neural Networks (FNN) and Long Short-Term Memory (LSTM) networks, since these models are differentiable and well-suited for capturing non-linear patterns—especially useful when working with time-series data.

For WINIT, which is specifically designed for temporal feature attribution in time series, we applied it on both LSTM (to leverage its sequential memory capabilities) and XGBoost (to explore how WINIT performs on strong, non-recurrent models). These combinations helped us explore the interpretability potential of each method across diverse model types.

LIME, being a model-agnostic local approximation method, was tested on LSTM, Random Forest (RF), and XGBoost (XGB) to evaluate how well it interprets both sequential and non-sequential models.

Permutation Importance (PI) was used with Decision Trees (DT) and Random Forests, as it is inherently compatible with tree-based models and requires no gradient computation.

In addition, visualizations were generated using horizontal bar plots to support intuitive comparison and interpretation of feature rankings.

All techniques were implemented following a unified structure to ensure fair and reproducible comparisons:

- Encapsulated functions for each method with preprocessing, training, and evaluation steps.
- Fixed hyperparameters across methods.
- Standardized result formats (CSV + plots).

6.3 Execution Methodology

For each feature importance technique implemented in this project, the dataset was loaded and preprocessed according to one of the two prepared versions (with or without correlation filtering), allowing us to assess the influence of preprocessing on interpretability outcomes.

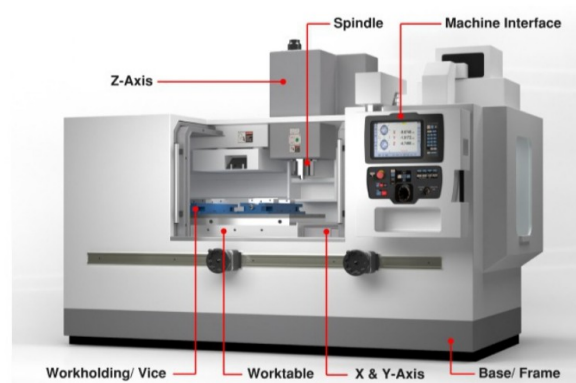
Models were trained using consistent hyperparameters (e.g., number of epochs, hidden layers, and learning rate) to ensure fair comparison across techniques and configurations.

Feature importance scores were then computed based on the respective method used—such as gradient-based attribution, permutation-based ranking, or local surrogate modeling. After execution, for each configuration, the following outputs were recorded and saved in CSV format for further analysis:

- **Test Loss:** Representing the prediction error of the trained model.
- **Execution Time:** Captured during each run to assess computational cost.
- **Top Features:** The 10 most influential features identified by each method.

6.4 Comparison and Analysis of Results

Feature Importance in Time Series For Energy Consumption in CNC Machine



 Select a Technique to Learn More

- ☒ Integrated Gradients
- ☐ WINit
- ☐ LIME
- ☐ Permutation Importance

About Integrated Gradients

Integrated Gradients is a gradient-based attribution method designed for neural networks. It quantifies feature importance by integrating the gradients of a model's prediction with respect to input features, computed along a straight-line path from a baseline input to the actual input. This approach satisfies axioms like sensitivity and implementation invariance. It is theoretically grounded and suitable for complex models such as LSTMs and FNNs, particularly useful in capturing subtle interactions in data, especially in time series.

 Start Analysis



<https://feature-importance-time-series.streamlit.app>

Figure 8: Home Page of Streamlit Application

To facilitate easy and interactive comparison of results across datasets, models, and methods, we developed a dedicated web application using **Streamlit**. This centralized dashboard enables users to explore:

- Individual top feature rankings for each technique variant.
- Visual comparisons across different feature importance techniques,
- Model-specific insights and interpretability performance.

The application provides a user-friendly interface for visualizing horizontal bar plots and interpreting CSV-based results generated from the experiments. It serves as an effective tool for both quick insights and in-depth analysis. The Streamlit app can be accessed at:

<https://feature-importance-time-series.streamlit.app/>

We compared the performance of the four feature importance techniques using three key evaluation metrics:

- **Test Loss** — Measures the predictive performance of the underlying model.
- **Execution Time** — Reflects the computational efficiency of each method.
- **Interpretability** — Assesses the clarity, consistency, and relevance of the features identified as important.

6.4.1 Technique-wise Best Performers

Integrated Gradients (IG)

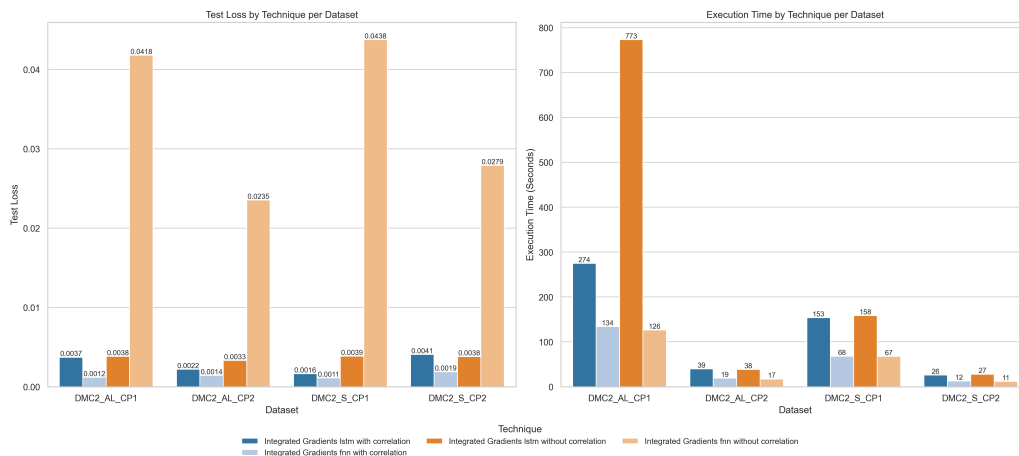


Figure 9: Test Loss, Execution Time per Dataset for IG

Integrated Gradients was evaluated using both FNN and LSTM models, with and without correlation-based preprocessing. Among all variants, FNN with correlation filtering emerged as the best-performing configuration, achieving the lowest test loss consistently across all datasets and maintaining a lower execution time than LSTM variants.

In addition to performance metrics, we also analyzed the top-ranked features generated by each configuration. We observed that the features identified by FNN with correlation were highly consistent with those found in more complex (but slower) LSTM models — especially in terms of physical relevance (e.g., `LOAD`, `TORQUE`, `ENC_POS`). This similarity further justified selecting the FNN model, as it offered comparable interpretability at a significantly reduced computational cost.

On the other hand, LSTM with correlation provided slightly better accuracy in some CP1 cases, but required more time and resources.

Key Insight:

- **Best trade-off (Accuracy , Speed and Interpretability):** FNN with Correlation
- **Most accurate for CP1:** LSTM with Correlation
- **Conclusion:** When interpretability and efficiency are both priorities, IG with FNN and correlation provides the most balanced solution.

WINIT

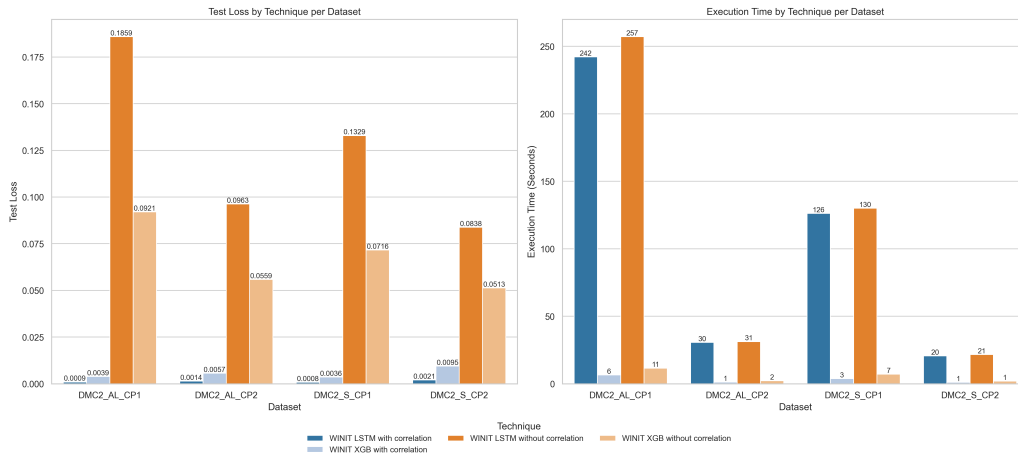


Figure 10: Test Loss, Execution Time per Dataset for WinIT

WINIT is designed specifically for time-series applications and uses a window-based relevance propagation mechanism to attribute importance to input features over time. We evaluated two model configurations with this technique: LSTM with correlation filtering and XGBoost with correlation filtering, along with additional unfiltered variations for comparative insight.

The LSTM with correlation configuration demonstrated exceptionally low test loss on the CP1 datasets (e.g., 0.0009 on DMC2_AL_CP1 and 0.0008 on DMC2_S_CP1), confirming WINIT's

ability to accurately capture temporal dependencies in sequential models. While the execution time was slightly higher than that of XGBoost (due to LSTM’s recurrent structure), it remained within acceptable limits, especially given the accuracy gains.

On the other hand, XGBoost with correlation delivered moderate test loss but much faster execution, making it a practical alternative where computational efficiency is prioritized. Interestingly, even though XGBoost lacks a recurrent structure, WINIT was still able to extract reasonable attributions when applied to this model, showing the technique’s adaptability.

We also analyzed the top features produced by each WINIT configuration. Across both LSTM and XGBoost variants, **TORQUE**, **LOAD**, and **CTRL_DIFF** series features repeatedly appeared among the top-ranked attributes. This consistency further validated the reliability of WINIT’s attributions across different architectures and preprocessing setups.

Key Insights:

- **Best accuracy (CP1):** WINIT using LSTM with Correlation
- **Best speed-efficiency trade-off:** WINIT using XGBoost with Correlation
- **Interpretability:** Both variants consistently highlighted similar influential features, suggesting robust temporal relevance detection.

LIME

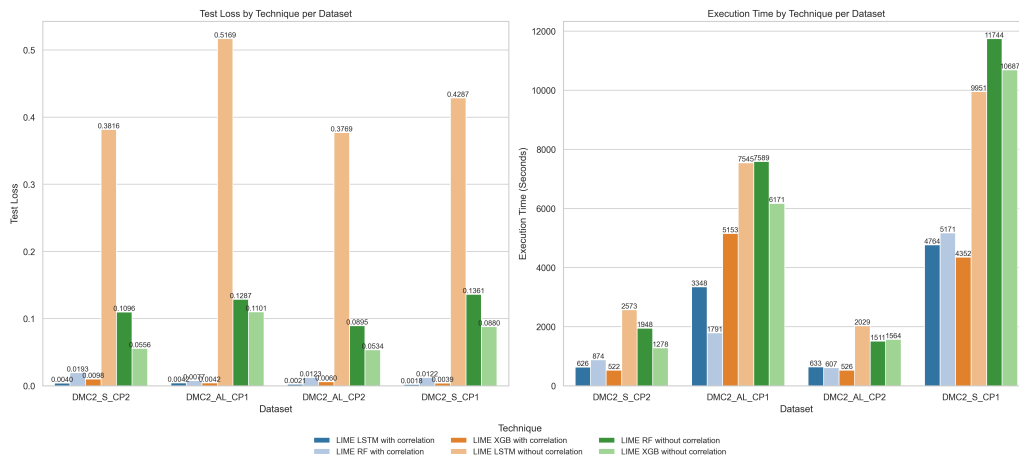


Figure 11: Test Loss, Execution Time per Dataset for LIME

LIME is a model-agnostic technique that explains predictions by learning an interpretable model locally around each prediction. We applied LIME across three model types—LSTM, Random Forest (RF), and XGBoost (XGB)—with both correlation-filtered and unfiltered datasets to evaluate its flexibility and performance.

From our evaluations, XGBoost with correlation filtering emerged as the most balanced configuration, offering good accuracy, lower execution time compared to LSTM, and consistent

feature interpretability. It consistently highlighted important features like TORQUE, LOAD, and CTRL_DIFF, aligning well with domain expectations.

LSTM-based LIME configurations yielded competitive accuracy, particularly when correlation filtering was applied. However, they required moderate to high execution times on larger datasets. Despite this, the interpretability remained reliable, with consistent top features across datasets. Removing correlation filtering generally led to increased test loss and reduced generalization performance.

RF-based LIME results were stable but slightly less accurate than XGB. However, RF models offered interpretability with relatively faster execution, making them a strong middle-ground option.

Across all LIME variations, top features remained consistent—dominated by TORQUE, LOAD, and CTRL_DIFF—supporting the credibility of the attributions.

Key Insights:

- **Best performance trade-off:** XGBoost with Correlation
- **Highest interpretability (but slower):** LSTM with Correlation
- **Observation:** Feature rankings remained consistent across models, reinforcing LIME’s reliability.

Permutation Importance (PI)

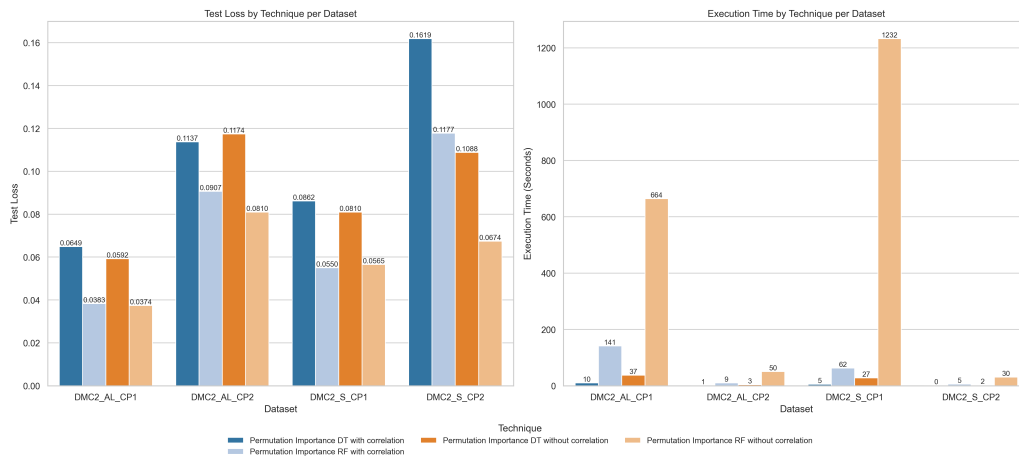


Figure 12: Test Loss, Execution Time per Dataset for PI

Permutation Importance (PI) was applied using Decision Trees (DT) and Random Forests (RF), under both correlation-filtered and unfiltered feature sets. This technique assesses the significance of each feature by measuring the drop in model performance when a feature’s values are randomly shuffled.

Across all datasets, Random Forest with correlation filtering provided the most balanced performance. It achieved nearly the same test loss as the unfiltered counterpart but with significantly lower execution time.

Feature rankings were consistent across RF configurations. Key features such as TORQUE|6, LOAD|2, and CMD_SPEED|6 frequently appeared in top ranks, reinforcing PI's reliability and interpretability.

Decision Trees, though less accurate than RF, offered much faster computation, especially when correlation filtering was applied, making DT-based PI suitable for lightweight evaluations on simpler datasets.

Key Insights:

- **Best performance trade-off:** Random Forest with Correlation
- **Fastest variant:** Decision Tree with Correlation
- **Observation:** TORQUE and LOAD consistently ranked among top features, validating their predictive relevance

6.4.2 Technique Suitability by Dataset Type

After a thorough comparison of all techniques across the four datasets, the following observations were made regarding their relative strengths:

CP1 Datasets (DMC2_AL_CP1 & DMC2_S_CP1)

- **WINIT (LSTM with Correlation)** yielded the lowest test loss, making it the most appropriate choice when high accuracy is the primary objective.
- **Integrated Gradients (FNN with Correlation)** delivered slightly higher test loss but significantly faster execution, making it a strong contender when balancing speed and interpretability.

CP2 Datasets (DMC2_AL_CP2 & DMC2_S_CP2)

- **Integrated Gradients (FNN with Correlation)** emerged as the top-performing method, offering the best combination of accuracy and efficiency.
- It consistently achieved the lowest test losses while maintaining fast computation across both CP2 datasets.

Insights Across Datasets

- For tasks demanding high precision on CP1 datasets, **WINIT (LSTM)** is the most effective.
- For scalable or time-sensitive applications on CP2 datasets, **IG (FNN)** is preferable due to its strong performance and low computational cost.
- **Top 10 Most Influential Features Affecting CNC Energy Consumption (Consistent Across All Variations):**
TORQUE|1, LOAD|6, TORQUE|6, LOAD|1, LOAD|3, TORQUE|3, TORQUE|2, CTRL_DIFF2|6, LOAD|2, TORQUE_FFW|1

6.5 Team Contribution and Collaboration

This project was carried out through effective collaboration among all three team members — **Kavyashree, Ranjith, and Nitin**—with a clear division of responsibilities and shared ownership of critical phases.

6.5.1 Literature Survey and Problem Understanding

All members collaboratively conducted an in-depth literature review to explore the landscape of feature importance techniques and their applicability to time-series modeling in CNC environments. This effort shaped the direction of methodology and model selection.

6.5.2 Individual Implementation Responsibilities

- **Kavyashree** implemented the Integrated Gradients (IG) technique across various models and preprocessing strategies.
- **Ranjith** handled the implementation of Permutation Importance (PI) using Decision Trees and Random Forests.
- **Nitin** was responsible for LIME implementation, covering LSTM, Random Forest, and XGBoost variants.

The WINIT method, being specialized for time-series attribution, was implemented jointly by all three members. This required collaboration in interpreting the method, configuring it for both LSTM and XGBoost models, and validating results across datasets.

6.5.3 Evaluation and Comparative Analysis

The final phases of the project—including comparison, interpretation, and conclusion writing—were executed collaboratively. All members participated in analyzing test loss, execution time, and top-ranked features to derive meaningful and actionable insights.

This collaborative and well-coordinated approach ensured technical depth and consistency throughout the implementation, evaluation, and reporting stages of the project.

7 Comparison of Project Plan and Execution

7.1 Challenges and Deviations from Initial Plan

Initially, our project execution followed the proposed plan closely, including early phases such as topic assignment, dataset understanding, and the implementation of feature importance techniques. During the intermediate phase, we successfully implemented all intended techniques and obtained results using LSTM models.

However, at that stage, we had not yet incorporated correlation-based filtering in our preprocessing strategy. While the technical execution was complete, we were not fully satisfied with the interpretability and relevance of the resulting feature rankings.

This prompted a critical reflection on our approach and revealed a methodological gap — the absence of correlation filtering, which could affect the quality, clarity, and accuracy of feature attribution. Additionally, our initial evaluation was limited to LSTM models, which constrained the scope of insights and did not account for the varying performance of techniques across different model architectures.

7.2 Adaptations Made During Project Execution

In response to these insights, we made a strategic shift in our methodology. The following key changes were introduced:

- **Correlation-Based Filtering:** We introduced correlation filtering into the data preprocessing phase to remove highly correlated features and reduce redundancy, thereby enhancing model stability and interpretability.
- **Expanded Model Evaluation:** Rather than evaluating all techniques only on LSTM models, we tested each interpretability method across diverse model types — including FNN, Random Forest (RF), XGBoost (XGB), and Decision Trees (DT) — to better understand technique-model interactions.
- **Re-execution of Experiments:** We reran all experiments using the updated methodology to ensure consistency, fair comparison, and enhanced result quality.

These adjustments resulted in more robust and interpretable outcomes, significantly improving the scientific rigor of the project. The refined methodology enabled us to make more informed comparisons and draw clearer conclusions regarding the most suitable combinations of feature importance techniques and model types for each dataset.

Overall, this deviation from the original plan turned out to be a valuable learning experience. It reinforced the importance of adaptability in research and ultimately increased the impact, quality, and reliability of our final results.

8 Usability and Limitations of Results

8.1 Practical Applications of the Results

The outcomes of this project have significant value in the context of CNC machine operations and industrial energy optimization. The following areas highlight how these results can be applied in real-world scenarios:

- **Energy Optimization:** The project identifies key machine parameters—such as torque, load, and control deviation—that strongly influence energy consumption. This enables industries to fine-tune operational settings and reduce energy costs.
- **Targeted Process Improvements:** Understanding which features contribute most to energy usage allows engineers to focus on high-impact process enhancements, including workflow redesign and CNC program optimization.
- **Data-Driven Decision Making:** The feature rankings support evidence-based decisions for parameter tuning, resource planning, and preventive maintenance strategies.
- **Explainable AI Integration:** The interpretability techniques used (e.g., LIME, IG, WINIT) can be embedded into industrial AI systems to enhance transparency and build trust among engineers and technical managers.
- **Scalability of Methodology:** While based on specific datasets, the overall approach to feature attribution and model comparison can be adapted and applied to other CNC machines and manufacturing environments.

8.2 Constraints and Limitations

Despite their usefulness, the results of this project come with several limitations that affect their real-world deployment:

- **Real-Time Integration Challenges:** Some techniques, particularly LIME and WINIT, are computationally intensive and may require optimization for real-time industrial use.

- **Limited Transferability:** Although the methodology is scalable, the specific feature importance rankings are machine- and dataset-dependent. Applying them directly to other machines may not yield equivalent benefits.
- **Interpretability for Non-Technical Users:** While techniques like LIME and IG enhance model transparency, interpreting their outputs still requires technical expertise. Communicating these insights to non-technical stakeholders remains a challenge.
- **Static Dataset Limitation:** The study relies on pre-recorded datasets, which may not capture dynamic factors such as machine wear, tool degradation, or seasonal variation—factors that can affect long-term applicability.
- **Limited Control Over Some Features:** Even though important features have been identified, not all are directly controllable in real operations. Internal machine states or low-level sensor outputs may not be adjustable by operators.

9 Conclusions

In this project, we implemented and evaluated multiple feature importance techniques using a variety of machine learning models, including FNN, LSTM, XGBoost, and Random Forest. Our goal was to assess how different interpretability methods perform in the context of time-series forecasting for industrial energy analytics. By experimenting with various preprocessing strategies and model architectures, we gained valuable insights into the trade-offs between accuracy, execution time, and interpretability. The following subsections summarize the key lessons learned and outline promising directions for future work.

9.1 Lessons Learned

Throughout the course of this project, we gained practical experience in applying and analyzing feature importance techniques within real-world industrial contexts. Working with diverse models and interpretability methods allowed us to uncover several meaningful insights that extend beyond just technical performance. Key takeaways from our work include:

- **Feature Importance Techniques Vary in Strengths:** Different techniques excel in different aspects—some prioritize accuracy, others focus on speed or interpretability—highlighting the importance of selecting methods based on application context.
- **Preprocessing Matters:** Data preprocessing, particularly correlation filtering, significantly influenced model performance and the clarity of feature importance rankings.
- **Model Architecture Impacts Interpretability:** The choice of models (e.g., FNN vs. LSTM vs. XGBoost) played a critical role in the effectiveness of interpretability methods. While FNNs were efficient, LSTMs yielded richer temporal insights.

- **Collaborative Analysis Enhances Outcomes:** Collaboration among team members improved result consistency and deepened collective understanding of model interpretability.
- **Practical Considerations Are Crucial:** Real-world applicability is affected by factors such as computational efficiency, real-time feasibility, and how easily results can be interpreted by end users.

9.2 Future Work

While our study provided meaningful comparative insights, several areas remain open for further development:

- **Real-Time Implementation:** Optimize current methods or develop lightweight variants to enable real-time industrial deployment.
- **Dynamic Feature Analysis:** Extend the study to streaming data to analyze how feature importance evolves over time and under changing machine states.
- **Cross-Machine Generalization:** Evaluate the transferability of our methodology across different CNC machines and operational environments to explore universality.
- **Hybrid Interpretability Models:** Investigate combining methods (e.g., blending LIME and IG) to leverage their complementary strengths.
- **Feedback Loop Systems:** Develop closed-loop control systems where feature importance insights continuously inform and adjust machine parameters in real-time.

In conclusion, this project not only enhanced our technical understanding of feature importance in time-series forecasting but also emphasized the value of interpretability in making machine learning truly applicable to industrial decision-making.

References

- [1] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [2] Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.
- [3] Kin Kwan Leung, Clayton Rooke, Jonathan Smith, Saba Zuberi, and Maksims Volkovs. Temporal dependencies in feature importance for time series prediction. In *International Conference on Learning Representations (ICLR)*, 2023.
- [4] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

- [5] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.
- [6] Clayton Rooke, Jonathan Smith, Kin Kwan Leung, Maksims Volkovs, and Saba Zuberi. Winit: Temporal windowing for feature importance in time series. In *International Conference on Learning Representations (ICLR)*, 2023.
- [7] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328, 2017.