OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

Feature Reduction in Time Series CNC Machine Data

Autonomous Multisensor Systems Group
Institute for Intelligent Cooperating Systems
Faculty of Computer Science
Otto von Guericke University, Magdeburg

Team Members: Debapriya, Shweta, Jayanthee

Supervisor: Hafez

16.09.2025

Autonomous Multisensor Systems

Contents

Why Feature Reduction Matters for CNC Data

Objective: To reduce high-dimensional CNC (Computer Numerical Control) Machine sensor data to a compact feature set to **retire redundant sensors and cut computational cost.**

Operational Costs

Smaller Models.
Fewer Sensors.

Redundancy

High compute Load.
Overlapping Channels.

Scalability

Compact Pipeline.
Simplified Deployment.

Real-time decisions

Lower Latency.
Linear Inputs.

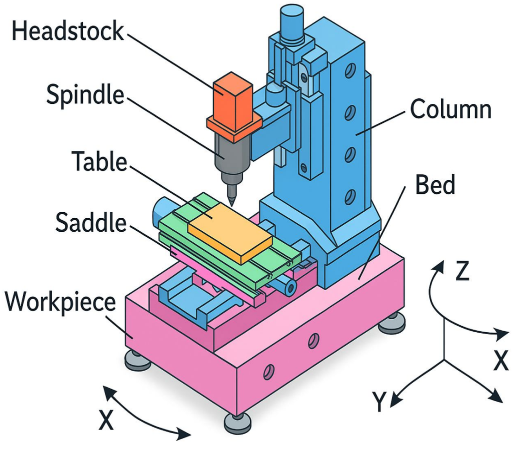


Fig 1: CNC Machine Overview [7]

https://www.researchgate.net/figure/Sketch-map-of-the-machine-tool-structure_fig1_334580700

Understanding Data

Dataset

Dataset Description

Size: (42,016, 52)

Data Type: Numerical, Multivariate time-series data

Components: Steel (S)

Features: LOAD|6, ENC_POS|1, DES_POS|1, TORQUE_FFW|1, CTRL_DIFF|2

Targets: CURRENT|6 (Spindle) , CURRENT|1 (X), CURRENT|2 (Y), CURRENT|3 (Z)

LOAD|1	LOAD|2	LOAD|3	LOAD|6	ENC_POS|1	ENC_POS|2	ENC_POS|6	CTRL_DIFF2|1	DES_POS|1	TORQUE_FFW|1	CURRENT|6	...
0.341797	6.47583	0.421143	0	22.200148	284.270207	181.730994	-0.00001	22.200139	0	-0.273438	...
0.341797	6.481934	0.415039	0	22.200139	284.270197	181.730994	0	22.200139	0	-0.273438	...
0.341797	6.481934	0.415039	0	22.200139	284.270197	181.730994	0	22.200139	0	-0.273438	...
...	...	...	...	...	...	...	...	...	...	...	...

Table 1: DMC_S_CP2 Dataset

Reducing High-Dimensional Data to Compressed Meaningful Features

Feature Reduction Workflow

Feature Reduction Workflow

1. Data Scaling and Splitting	2. Data Sequencing	3. Hyperparameter Configuration	4. Model Training	5. Output
StandardScaler	Sliding Windows	Latent_Dim = 25 or 14	Monitor Training Loss	Reduced Features
Training Set = 60%	Important to learn temporal features	Window Size = 60 or 120	Early Stopping	
Validation Set = 20%	Depends on the model			
Test Set = 20%				

Identifying the Best Architecture for Feature Selection

Architecture Selection

Architecture Selection

Autonomous Multisensor Systems

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

Principal Component Analysis

Linear
Variance-maximizing
Orthogonal
Components

Autoencoder

Non-Linear Neural
Encoder-Decoder
Network

LSTM+CNN Autoencoder

Hybrid
Temporal-Spatial
Encoder-Decoder
Network

PCA for Feature Reduction

Principal Component Analysis (PCA)

1. Simple and fast

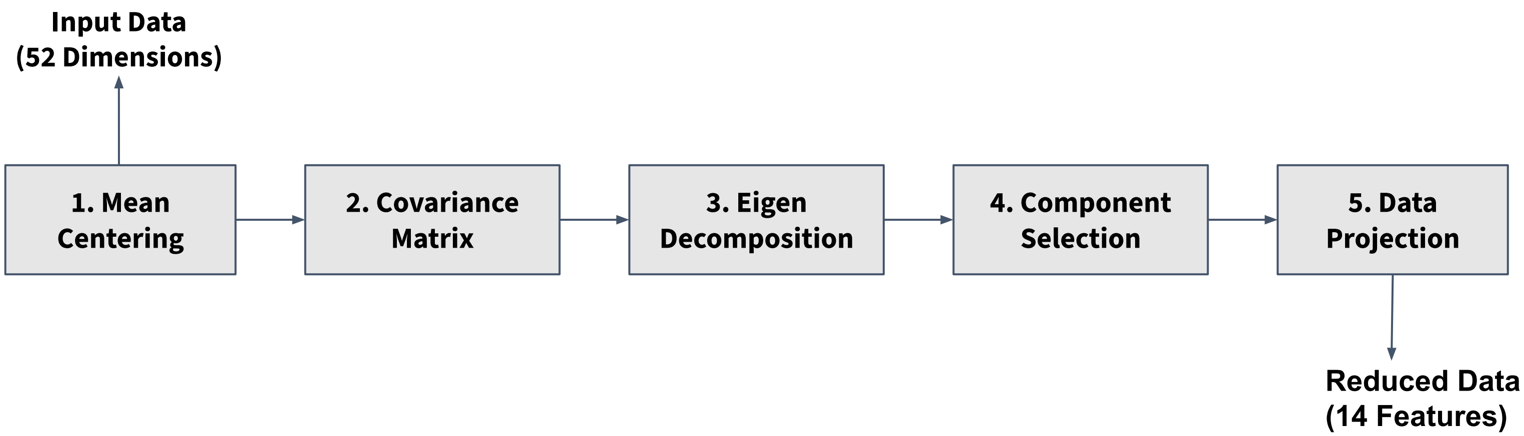
2. Deals with multicollinearity

3. Strong Compression

4. Interpretable

5. Reproducible

PCA Architecture



Hyperparameter	Value
Scaler	StandardScaler
Variance	0.98

Mean Centering: $X_{centered} = X - \bar{X}$

Covariance Matrix: $\Sigma = \frac{1}{n-1}\left(X_{centered}\right)^T X_{centered}$

Eigen Decomposition: $\Sigma v = \lambda v$

Projection: $Z = X_{centered} W$

Symbol	Description
X	Input data matrix
Σ	Covariance matrix
v, λ	Eigenvectors and eigenvalues
W	Projection matrix (top k eigenvectors)
Z	Projected data

Fig 2: PCA Architecture and Hyperparameters

Autoencoders for Feature Reduction

Autoencoder

Autoencoder for Feature Reduction

1. Non-Linear Machine dynamics

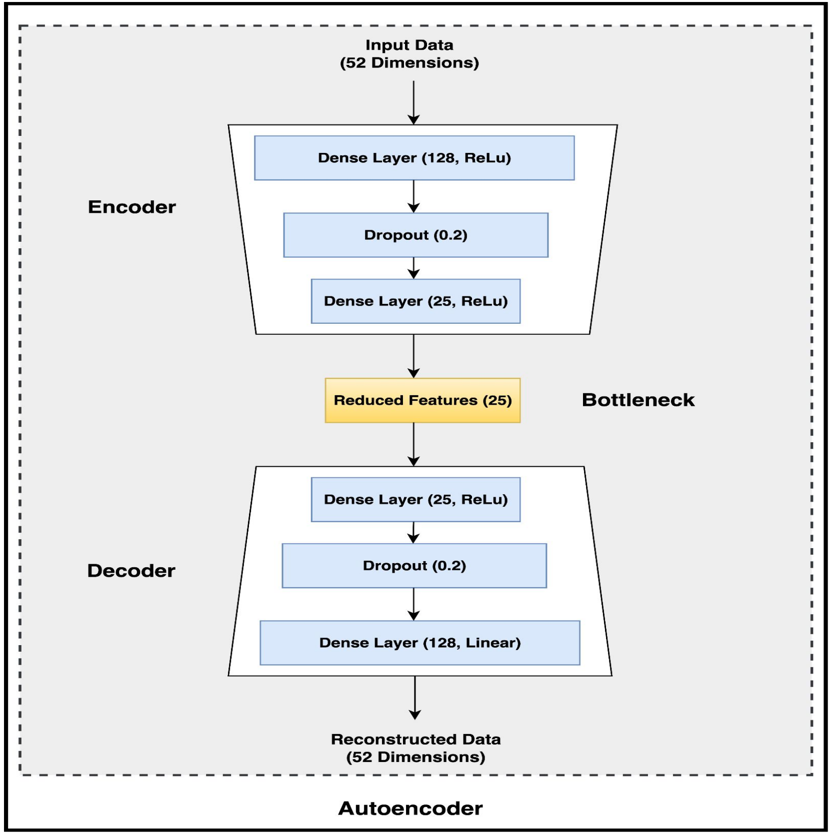
2. Task relevant compression

3. Time awareness

4. Denoising

5. Flexible

Autoencoder Architecture



Hyperparameters	Value
Scaling	StandardScaler()
Window Size	60
Latent Dim Size	25
Encoder Layers	2 Layers; ReLu Dropout = 0.2
Decoder Layers	2 Layers; ReLu Linear as Output layer Dropout = 0.2
Optimizer	Adam
Learning Rate	0.001
Training Strategy	Early Stopping; Patience = 10
Batch_size	64

Fig 3: Autoencoder Architecture and Hyperparameters [2]

LSTM + CNN - Autoencoder for Feature Reduction

LSTM + CNN - Autoencoder

LSTM+CNN Autoencoder

1. Captures Temporal Dynamics via LSTM

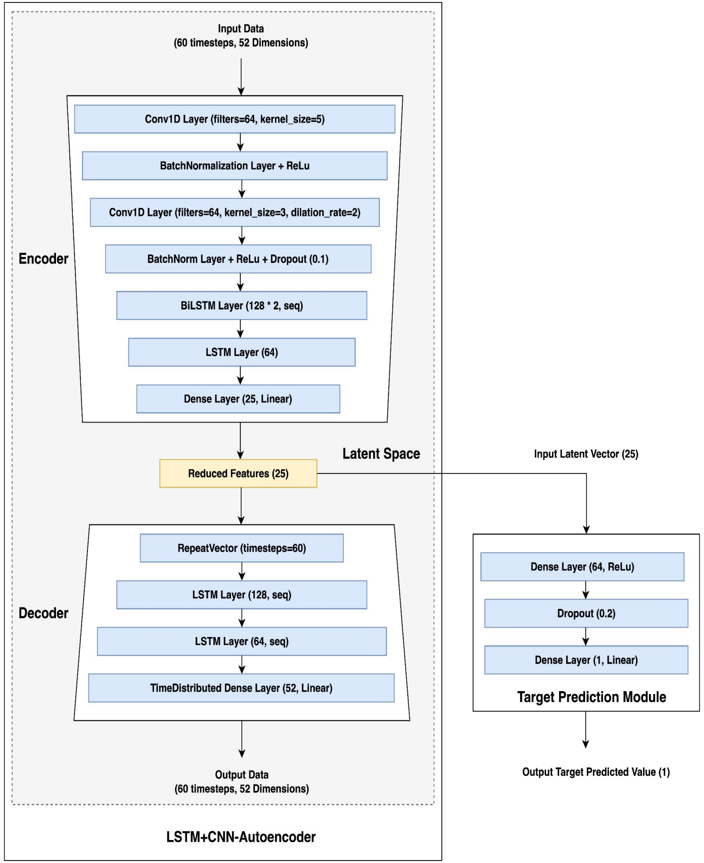
2. Extraction Local Features through CNN

3. Temporal-Spatial Fusion with Hybrid LSTM+CNN

4. Noise Robustness using CNN filters + LSTM smoothing

5. Better Generalization to Complex Machine States

LSTM+CNN Autoencoder Architecture



Hyperparameters	Value
Scaling	StandardScaler()
Window Size	60 (Autoencoder), 10 (Predictor)
Latent Dim Size	25
Encoder Layers	2 Conv1D; ReLu, 2 LSTM (BiLSTM, LSTM), 1 Dense, Dropout(0.1)
Decoder Layers	2 LSTM, 1 RepeatVector (60 timesteps) TimeDistributed Dense; Linear
Regression Layers	2 Dense Layers; ReLu; Linear, Dropout (0.2)
CNN Filters, Kernels, Dilation_rate	Filters = 64, Kernel size 5 and 3, Dilation = 2
Optimizer	Adam
Loss Function	**MSE (Reconstruction) + α * MSE (Predictor)** α = 0.5
Learning Rate	0.001
Training Strategy	Early Stopping; Patience = 10
Batch_size	64

Fig 4: Supervised LSTM+CNN Autoencoder and Hyperparameters [1]

Assessing the Quality of Reduced Features

Evaluation

Evaluation Workflow

1. Data Input	2. Data Splitting	3. Data Sequencing	4. LSTM Training	5. Evaluation Results
Reduced Features Sets 14 (PCA) 25 (Autoencoders)	Training Set = 60% Validation Set = 20% Test Set = 20%	Sliding Windows Important to learn Temporal Features	Hyperparameter Configuration Monitor Training Loss EarlyStopping	Metrics used : MAE, RMSE, R^2 **Higher R^2** = Reduced Features **effectively capture underlying data patterns**

Our Results

Evaluation Results

Results - Principal Component Analysis

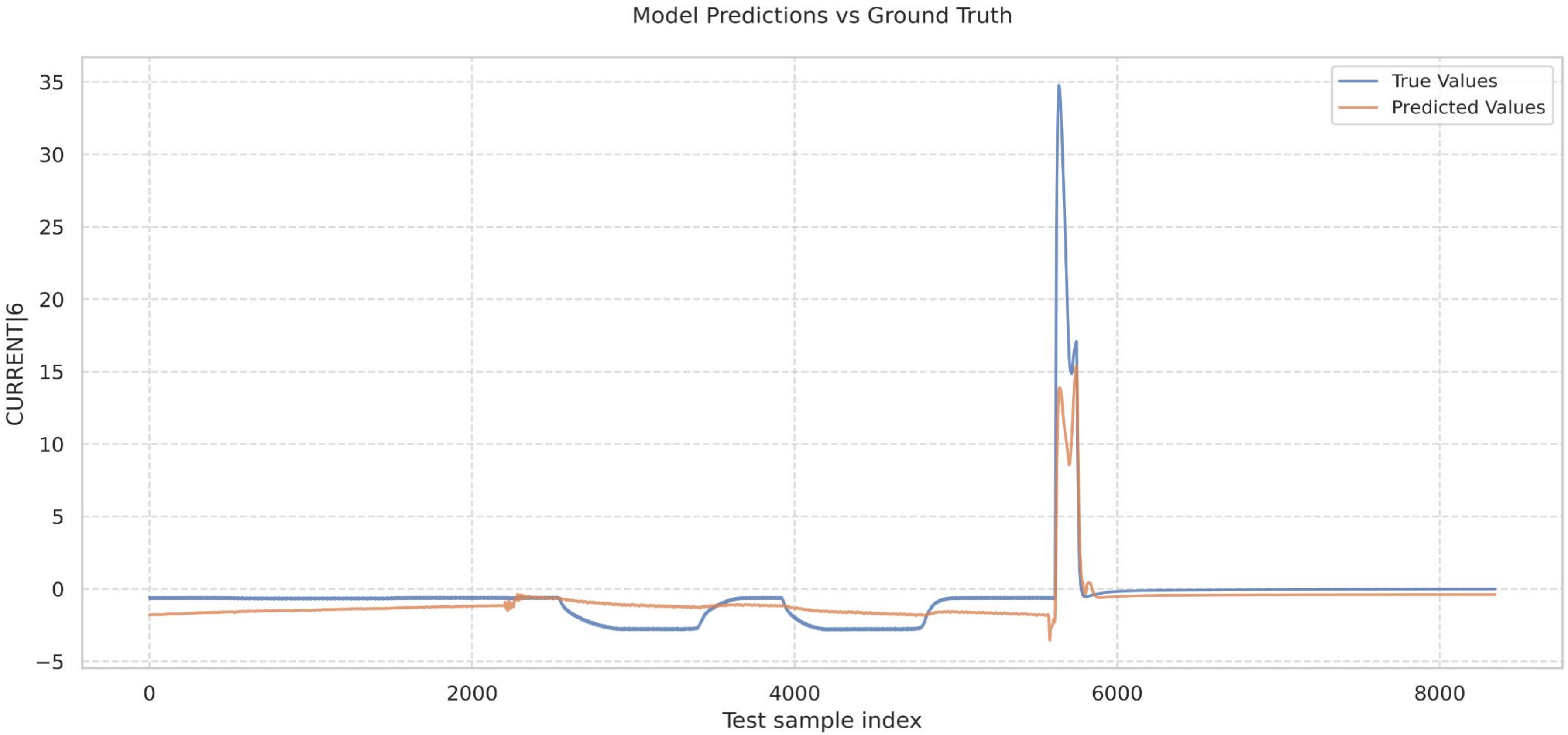


Fig 5: Model Prediction Performance on PCA-Reduced Features

Results - Autoencoder

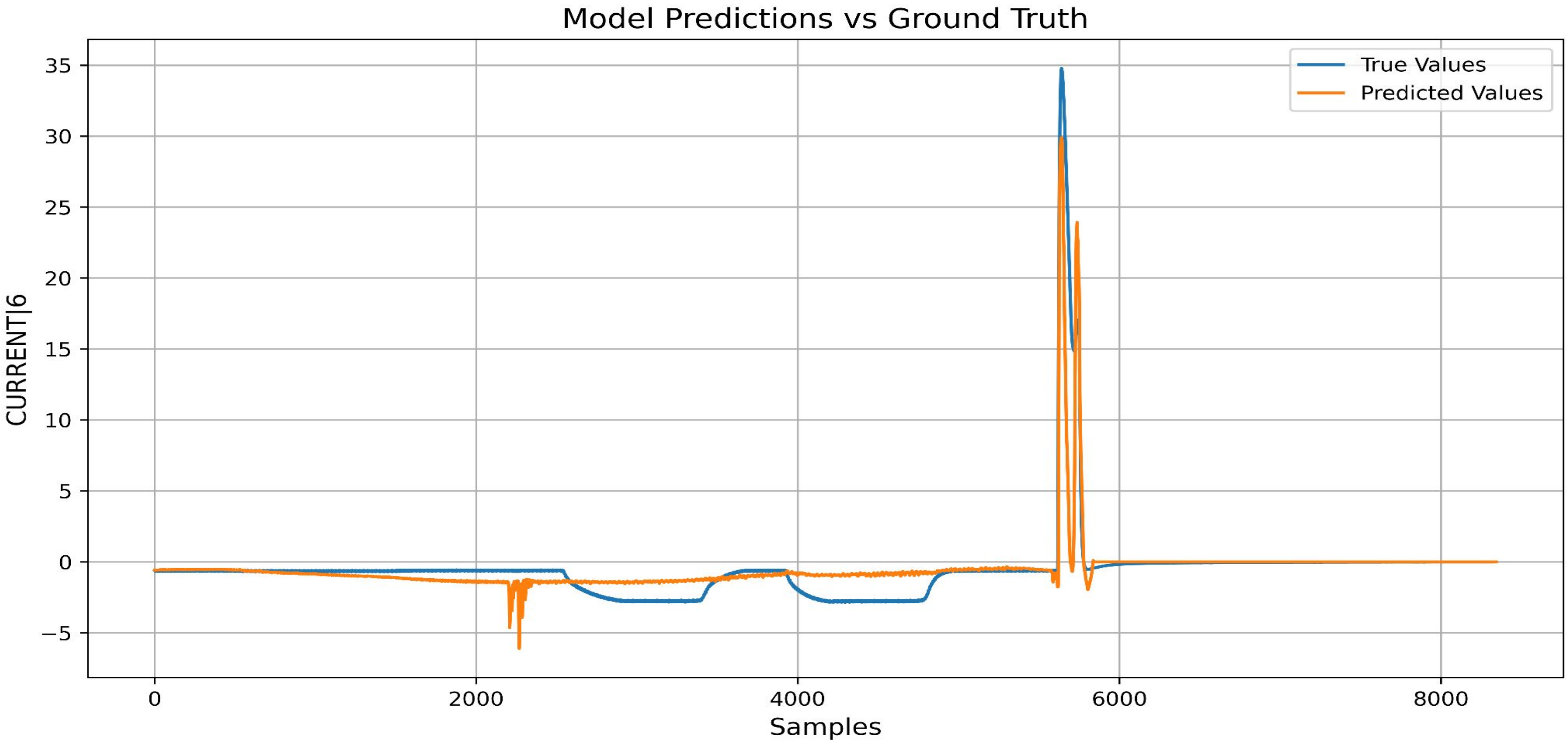


Fig 5: Model Prediction Performance on Autoencoder-Reduced Features

Results - LSTM+CNN-Autoencoder

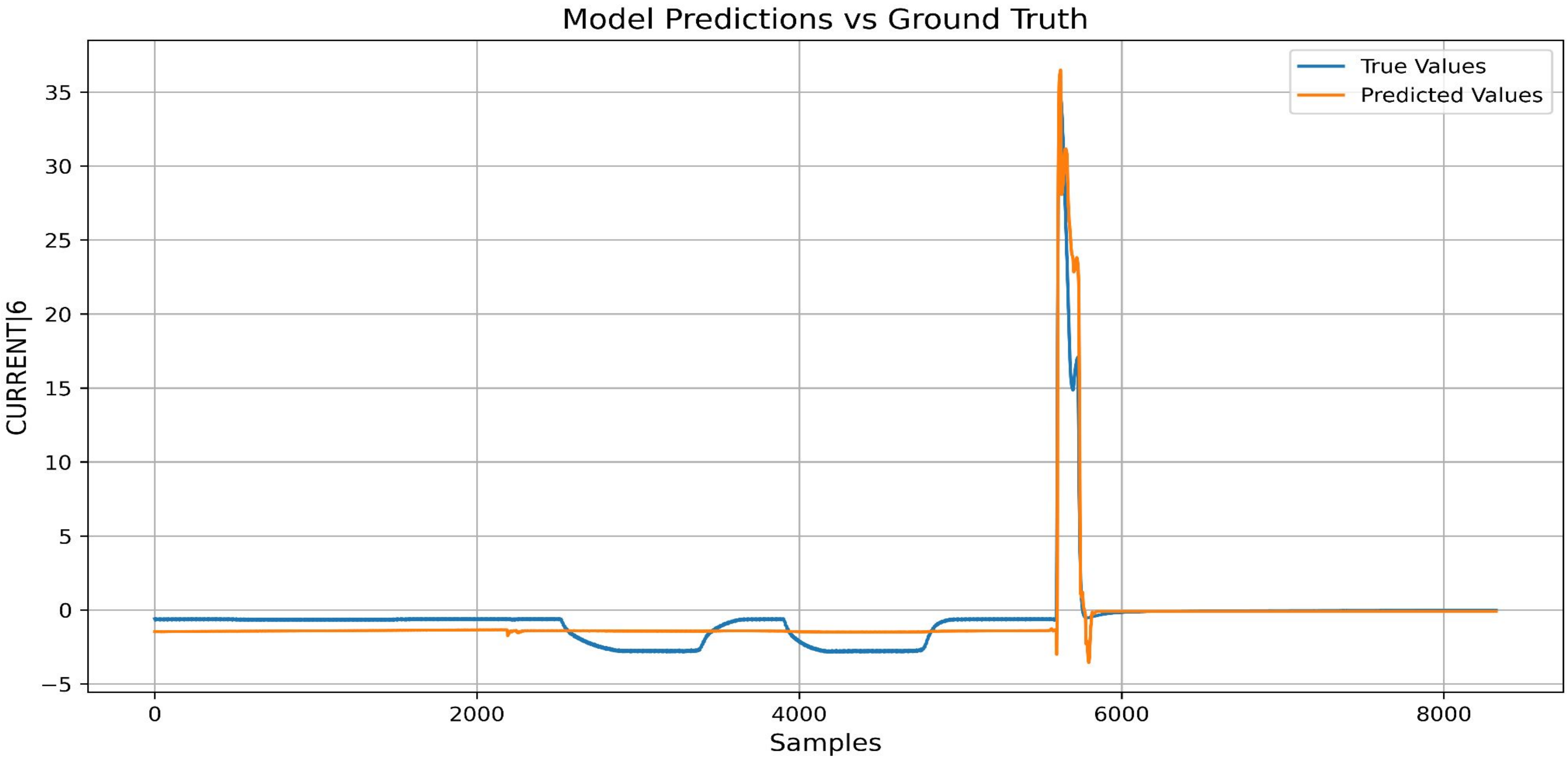


Fig 5: Model Prediction Performance on Autoencoder-Reduced Features

Result Summary

Model	Num. of Features	MAE	RMSE	R^2	Training Time (sec)
Original Dataset	52	0.59	1.79	**0.70**	200.30
PCA	**14**	1.15	1.82	0.68	51.41
Autoencoder	25	1.22	2.37	0.70	110.60
LSTM+CNN-Autoencoder	25	0.86	2.38	**0.85**	121.95

From Black-Box to Transparent Insights

Interpretability

Interpretability Results - PCA

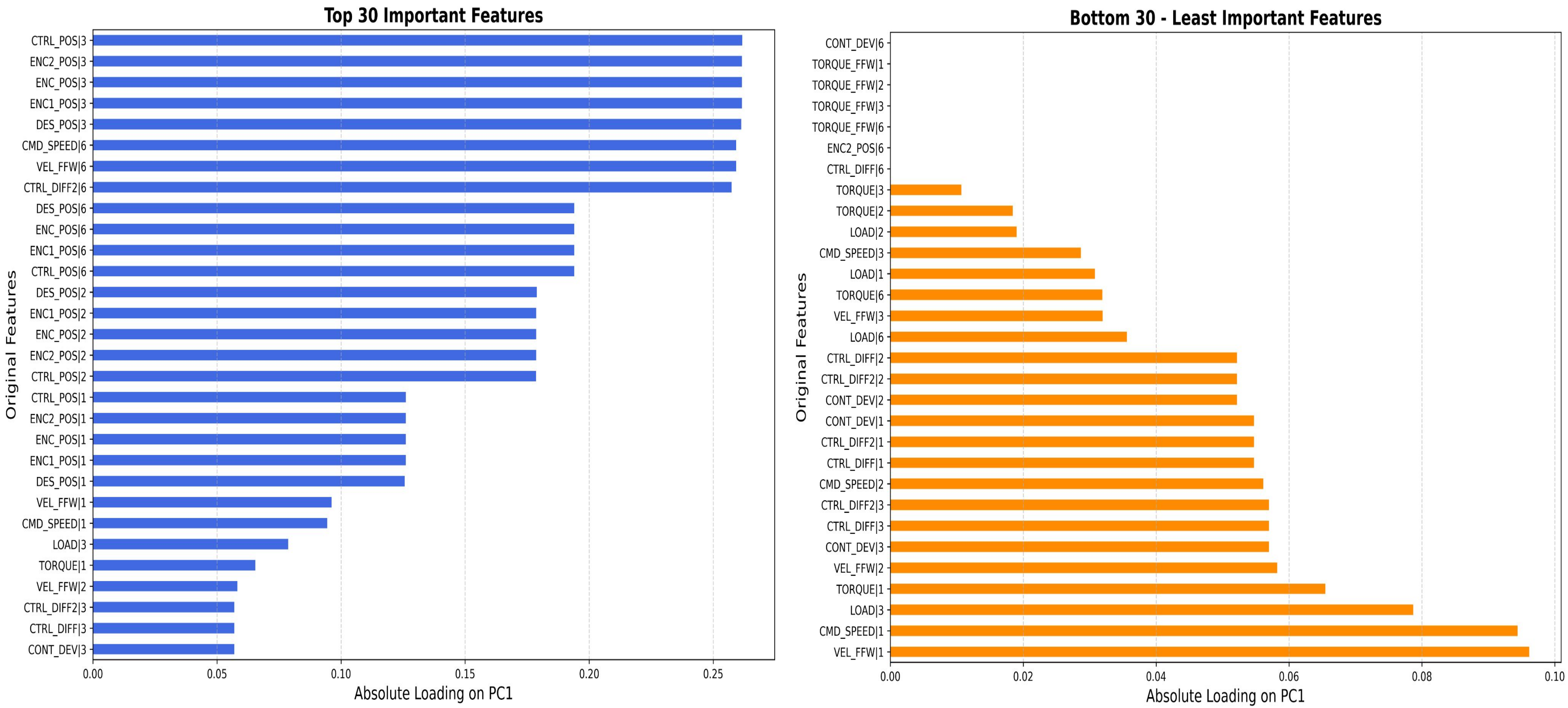


Fig 7: PCA Loadings for Feature Importance

SHapley Additive exPlanations (SHAP) Workflow

1. Select Explanation Target	2. Prepare Data	3. Initialize Explainer	4. Compute SHAP Values	5. Aggregate SHAP Values
Target: Latent Dimension (25) Explain **which Original Features (52)** influenced most.	**Training Subset** as Baseline - 300 Windows **Test Windows** as explain set. (200 Windows) Window_Size = 60	**GradientExplainer** Inputs: Encoder and Baseline (Training subset)	Run GradientExplainer on Test Subset **Output** : SHAP Tensor Contribution of **Each Feature, at Each Timestep to Each Latent Dim** **Shape:** (Test Windows = 200, Timesteps = 60, Original Features = 52, Latent Dim = 25)	Average Absolute SHAP One Score per Feature (52 total)

Interpretability for Autoencoder

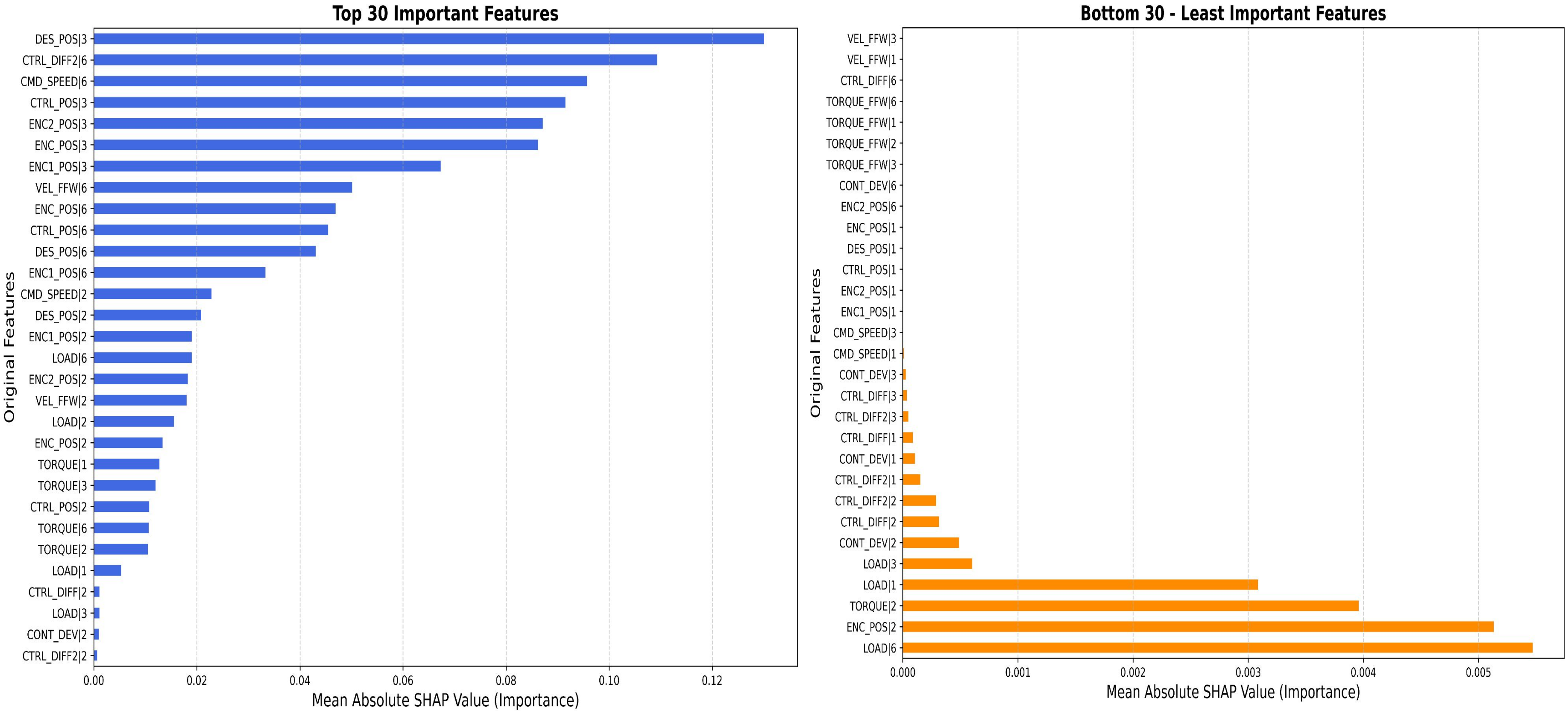
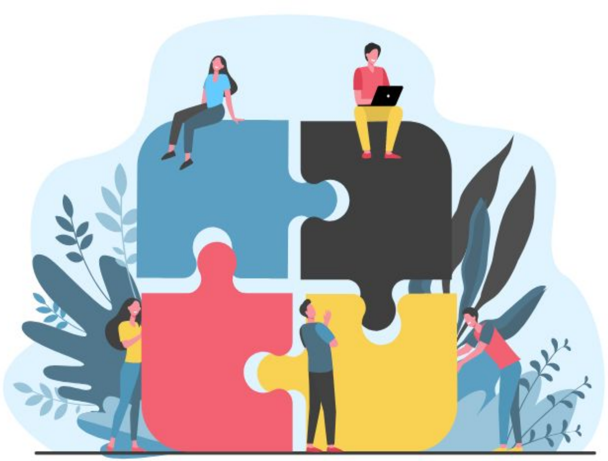


Fig 7: SHAP Scores for Feature Importance

Key Findings and Takeaways

Conclusion

Conclusion



1. **Achieved effective dimensionality reduction** while **preserving predictive performance** of original features.

2. **Supervised LSTM+CNN Autoencoder** delivered the highest **R^2 (~0.80)** and **reduced training time by ~40%.**

3. Identified **key contributing** and **redundant** features. enabling potential **cost savings** in sensor deployment and computation.
    a. Important Features: **CTRL_POS|3, Encoder_POS|3**
    b. Least Contributing Features: **TORQUE_FFW|1, TORQUE_FFW|2**

4. **Demonstrated that Interpretable Feature Reduction** is possible in Black-Box Autoencoders using Explainable AI.

Opportunities for Further Development and Improvement

Future Work

Future Improvement of our project

1. To Explore Advanced Dimensionality Reduction Architectures for **smoother Latent Spaces and Better Generalization**. For Ex:
    a. Variational/Transformer-based Autoencoders
    b. MultiTask Elastic Net
    c. Hyperparameter Tuning for better RMSE values.

2. **Enhance Model Interpretability** by validating Feature Importance across multiple XAI techniques

3. Target-level Explainability: Use SHAP on CURRENT|6 Regressor to **Rank Original Features/Sensors driving Energy (Current) usage**

References

1. Xingchen Ran, Weidong Chen, Blaise Yvert, Shaomin Zhang, *A hybrid autoencoder framework of dimensionality reduction for brain-computer interface decoding*, Computers in Biology and Medicine, Volume 148, 2022, 105871. ISSN 0010-4825. https://doi.org/10.1016/j.compbiomed.2022.105871
2. M. Ashraf *et al*., "A Survey on Dimensionality Reduction Techniques for Time-Series Data," in *IEEE Access*, vol. 11, pp. 42909-42923, 2023, doi: 10.1109/ACCESS.2023.3269693. https://ieeexplore.ieee.org/document/10107391
3. Wan, X., Li, H., Zhang, L. *et al.* "Dimensionality reduction for multivariate time-series data mining". *J Supercomput* 78, 9862–9878 (2022). https://doi.org/10.1007/s11227-021-04303-4
4. W. Todo, B. Laurent, J.-M. Loubes, and M. Selmani, "Dimension Reduction for Time Series with Variational AutoEncoders", arXiv:2204.11060, Apr. 2022. https://doi.org/10.48550/arXiv.2204.11060
5. Ruisheng Ran, Tianyu Gao, Bin Fang, "Transformer-based dimensionality reduction", 15 Oct 2022. https://arxiv.org/abs/2210.08288
6. Xingchen Ran, Weidong Chen, Blaise Yvert, Shaomin Zhang, "A hybrid autoencoder framework of dimensionality reduction for brain-computer interface decoding", Computers in Biology and Medicine, Volume 148, 2022,105871, ISSN 0010-4825, https://doi.org/10.1016/j.compbiomed.2022.105871.
7. https://www.researchgate.net/figure/Sketch-map-of-the-machine-tool-structure_fig1_334580700
8. https://deepai.org/machine-learning-glossary-and-terms/feature-reduction
9. https://medium.com/aimonks/a-comparative-analysis-of-dimensionality-reduction-techniques-in-machine-learning-3389c5103348
10. https://medium.com/data-science/dimensionality-reduction-for-data-visualization-pca-vs-tsne-vs-umap-be4aa7b1cb29
11. https://www.geeksforgeeks.org/dimensionality-reduction/

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

Feature Reduction in Time Series CNC Machine Data

Thank You For Your Attention!

Autonomous Multisensor Systems