applied sciences

MDPI

Article

Unsupervised Feature Space Analysis for Robust Motor Fault Diagnosis Under Varying Operating Conditions

Ubada El Joulani [1,*], Tatiana Kalganova [1] and Stanislas Pamela [2]

[1] Department of Electronic and Electrical Engineering, Brunel University of London, Kingston Lane, Uxbridge UB8 3TE, UK; tatiana.kalganova@brunel.ac.uk

[2] United Kingdom Atomic Energy Authority, RACE, Culham Campus, B1, Abingdon OX14 3DB, UK; stanislas.pamela@ukaea.uk

* Correspondence: ubada.eljoulani@brunel.ac.uk

Abstract

Reliable fault diagnosis of induction motors from current signals is critical for preventing failures in industrial systems. However, deep learning models often exhibit performance degradation when the torque load and other operating conditions change. Although a lot of research has been completed on supervised fault classification using current signals, the investigation of the behaviour of these datasets for unsupervised learning has not been done. This study quantifies and analyses the "shadowing effect" of operational variability, demonstrating that a baseline 1D-CNN achieving 100% accuracy under static 0 Nm loads drops to 53.19% accuracy when subjected to 4 Nm load in the KAIST dataset using a stator current. Similar trends were validated using the Paderborn University (PU) bearing dataset. Using 1D-CNN feature extraction followed by Principal Component Analysis (PCA), t-SNE, and hierarchical clustering, we show that standard linear mitigation strategies, such as removing high-variance principal components, are ineffective because fault and load features are deeply entangled. Hierarchical clustering analysis confirms that the feature space is organised by load dominance, with the primary tree split consistently occurring by torque load rather than fault type. Crucially, we identify that internal geometric metrics, such as "spread" and "diameter", correlate with external purity metrics like the proposed "Dominance Score". The findings establish a quantitative basis for developing unsupervised, load-invariant diagnostic models that utilise geometric stopping criteria to isolate fault clusters without using ground-truth labels.

Keywords: condition monitoring; fault diagnosis; contextual features; artificial intelligence; clustering

Check for updates

Academic Editor: Douglas O'Shaughnessy

Received: 18 December 2025
Revised: 28 January 2026
Accepted: 6 February 2026
Published: 11 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Rotating machinery, particularly induction motors, serves as a critical component in many industrial applications. Induction motors are considered indispensable components in mechanical equipment, providing power for the machine to work [1]. The reliability of these motors is tied to operational efficiency and safety, with unexpected failures leading to costs and downtime and therefore production losses and safety hazards as well. The implementation of strategies for fault detection on rotary machines is crucial for the reliability and safety of modern industrial systems [2]. In fact, bearing faults alone account for 40–45% of motor failures, rotor faults for 8–10%, and stator faults for 30–40%, collectively representing the dominant failure modes in industrial environments [1,3]. Consequently,

https://doi.org/10.3390/app16041780

condition monitoring systems have become crucial in modern predictive maintenance. Early detection of faults enables interventions before faults evolve into critical failures, thereby reducing repair costs and minimising production losses and enhancing overall system reliability [2–4].

Motor signal datasets are fundamental for the development of AI-driven condition monitoring systems. These datasets provide the raw material for training and testing the AI models for condition monitoring. Existing datasets capture faults at various operating frequencies, such as the 60 Hz induction motor dataset by [5,6], another that operates at 50 Hz as in [7,8], or the mixed 100 Hz (at 1500 rpm) and 60 Hz (at 900 rpm) experiments by [9]. Recently, the Order Domain Transformer (ODT) has been proposed as a normalisation technique that enables fusion of datasets with different frequencies, hence extending the existing datasets in the normalised format and improving the overall performance of neural network (NN) models due to wider variability of signals introduced [10]. Some resources, like the KAIST dataset [9,11] extend the scope of datasets by capturing signals at different speed and load conditions relying on fixed, segmented load profiles. However, industrial applications operate under continuous variable speed and irregular load changes and are influenced by stochastic factors that are difficult to reproduce [9]. For instance, the dataset by [9] is restricted to two discrete torque settings of 0.1 Nm and 0.7 Nm, while the KAIST dataset [11] utilises only three static load conditions (0, 2 and 4 Nm). Similarly, databases for rotor fault detection often rely on stepped load increments, such as the 0.5 Nm to 4 Nm range used to identify broken rotor bars [6]. These discrete snapshots represent a narrow slice of real-world motor behaviour and do not represent real operating conditions with vast variations in speed and load. In this work, the analysis uses mechanical fault modes. Specifically, the KAIST motor dataset provides bearing inner/outer race faults, shaft misalignment and rotor unbalance, while the Paderborn University dataset provides rolling element bearing damage. This work focuses on understanding how varying mechanical torque load influences neural network performance for mechanical fault diagnosis.

When a motor's load shifts dynamically, the resulting signal deviation can easily be misidentified as a defect by an anomaly detector [4,12]. In this case, the data shift must be fully understood for the consequent task of fault classification within the dynamic operating environment. For instance, ref. [13] used Topological Data Analysis (TDA) for fault classification under no-load conditions, while ref. [14] extended TDA to quantify eccentricity faults under varying load conditions ranging from 0 to 3.5 Nm. Ref. [2] employed FPCA and FDM to classify faults under varying load levels of 0%, 20%, 40%, 60%, and 80%. In [15], the authors successfully differentiated between distinct faults, such as inner race, outer race and ball faults, using an autoencoder with a feed-forward neural network using loads of 0, 1, 2 and 3 horsepower (hp). Ref. [16] utilised a CNN to diagnose faults under loads of 0, 1, 2 and 3 hp as well. Ref. [1] utilised pre-trained unsupervised GANs to fine-tune discriminators across five fault categories, namely healthy motor, bending rotor, defective bearing, phase missing, and stator winding shorted, where varying rotating speeds (100–1800 rpm) were used as operating conditions. Similarly, in the broader context of complex industrial systems, ref. [17] proposed a digital twin-assisted framework to enhance diagnosis reliability by fusing virtual and real data, utilising a convolutional neural network-gated recurrent unit model, while ref. [18] also developed an intelligent diagnosis method for an electric-hydraulic control system using residual analysis for extracting strong features. Additionally, probabilistic approaches such as the probabilistic neural network (PNN) have been applied to classify bearing fault types under varying load conditions [19]. Although many efforts have been made to utilise fault classification with a discrete, stepped load range, limited efforts have been made to understand the influence of load variations on unsupervised fault classification.

Current signals are commonly leveraged due to their simple and low-cost implementation, as sensors are part of motor control systems [12,14]. This placement on control panels facilitates practical industrial deployment by bypassing the difficulties of accessing machinery in challenging environments [2]. Consequently, various approaches have been developed to utilise this modality: ref. [13] utilised time-domain stator current signals directly, while [2] employed a two-phase current alongside line voltages. Others, such as [12], transformed current signals into two-dimensional images for analysis, and research by [10,20] has further validated the use of current data alone for robust fault diagnosis.

The fundamental principle of Motor Current Signal Analysis (MCSA) is that fault conditions manifest as characteristic spectral changes in the current signature [2,3]. Traditional diagnostic approaches rely on converting these raw signals into signatures using techniques such as Fast Fourier Transform (FFT) [2] or Recurrence Plots (RPs) [12]. While effective, these methods require extensive domain expertise to manually engineer features and identify relevant fault harmonics, which often shift depending on motor specifications and operating conditions [2]. Instead of using signal processing techniques, researchers have utilised raw unprocessed time-domain signals as model inputs, using deep learning to automatically learn representative features. For instance, ref. [21] demonstrated the efficacy of a compact adaptive 1D-CNN. Similarly, ref. [16] proposed a deep CNN framework which can automatically extract robust features from raw signals, and in [22], the authors introduced a model combining 1D and 2D CNNs to process raw signals directly. This eliminates the need for complex preprocessing and allows for automated feature extraction using deep learning models.

Simultaneously, the reliance on supervised learning poses a practical barrier to deployment. Although supervised models, ranging from Decision Trees and SVMs [13] to CNNs as in [16,21], achieve high accuracy, they depend on availability of abundant labelled fault data. In industrial environments, collecting labelled data for every possible failure mode is costly and time-consuming [1,2]. While semi-supervised approaches have been explored in [1,15], they still require some degree of labelling. Consequently, the use of unsupervised learning allows the labelling bottleneck to be bypassed by uncovering patterns and clusters within the abundant raw operational data available, without requiring manual annotation for every operating condition [2].

Despite the promise of unsupervised classification on raw current signals, its effectiveness is compromised by operational variability, specifically the fluctuating load characteristics. Analysis of public datasets, such as the KAIST motor dataset [11] and Paderborn University (PU) dataset [9], reveals that models trained under static conditions struggle to generalise when under varying loads. For instance, studies focusing on a single load condition often report high diagnostic accuracy [23]. However, when data from multiple load conditions are integrated, performance drops significantly, as shown in [24,25]. This degradation occurs because load variations introduce dominant variance in the raw signal that unsupervised models struggle to distinguish from fault signatures. While recent attempts using domain adaptation have been made [20], the adequacy of dimensionality reduction techniques, such as Principal Component Analysis (PCA), in this context needs to be explored. It remains to be investigated whether removing top principal components based on high variance can effectively eliminate these dominant contextual variables or if dominant features driven by load changes will obscure the subtle features of faults, a phenomenon defined as the shadowing effect by [26].

A more targeted approach is required to understand how these contextual variables structure the feature space and influence the performance of the neural network model. Hierarchical clustering has emerged as a powerful tool for analysing complex and multi-scale feature relationships by organising data into nested structures that preserve the local context. For instance, ref. [27] demonstrated the utility of this approach on marine machinery to identify and label distinct anomaly patterns. Building on these advancements, this paper proposes the application of hierarchical clustering to systematically identify and trace how operational conditions (such as multiple torque loads) propagate through cluster levels in the feature space.

This paper makes the following contributions:

- Systematically characterises how operational variability (multiple torque loads) manifests as contextual features within hierarchical cluster structures.
- Identifies and evaluates hierarchical clustering metrics to capture contextual influence at different scales, enabling context-aware feature assessment beyond conventional variance-based approaches.
- Demonstrates that hierarchical cluster analysis can guide the design of robust unsupervised fault classification systems for raw time-domain current data under varying operational conditions.
- Provides practical validation on the KAIST motor dataset [11] and PU motor dataset [9], quantifying the impact of contextual features on 1D-CNN performance and confirming that hierarchical metrics inform more targeted feature selection and model robustness than variance-based methods.

The remainder of this paper is organised as follows: Section 2 introduces the technical background for the dimensionality reduction and clustering tools used in this work, including Principal Component Analysis (PCA), t-Distributed Stochastic Neighbour Embedding (t-SNE) and hierarchical clustering. Section 3 then details the methodology, covering the datasets, preprocessing pipeline, 1D-CNN feature extractor and feature space analysis procedures. Section 4 presents the experimental results, while Section 5 discusses the findings in the context of unsupervised, load-invariant fault diagnosis. Section 6 concludes the paper and outlines directions for future work.

2. Theoretical Background

This section provides the mathematical and conceptual foundations for the unsupervised techniques later applied in the methodology to analyse the feature space extracted by the 1D-CNN. To investigate the structural organisation of high-dimensional fault signatures and the potential "shadowing effect" of operational variability, both linear and non-linear dimensionality reduction methods are employed. Principal Component Analysis (PCA) is used to assess class separability through global variance, while t-Distributed Stochastic Neighbour Embedding (t-SNE) provides a granular view of local manifold structures. Unsupervised k-means clustering and hierarchical clustering are established as the primary tools for quantifying cluster purity and identifying natural groupings without the use of labels. These methods allow for an evaluation of how torque load conditions structure the feature space relative to mechanical fault types.

2.1. PCA

Principal Component Analysis (PCA) is a linear dimensionality reduction technique used in this study for both visualisation and diagnostics. It transforms the feature space into a new coordinate system of orthogonal principal components, ordered by the amount of data variance they capture.

For visualisation, plotting the first two or three principal components (PCs) provides a low-dimensional view of the feature space, allowing for an initial assessment of how the data points are grouped. More importantly, PCA is used as a diagnostic tool as well, investigating what has been defined as the "shadowing" effect of contextual features.

2.2. *t-SNE*

t-Distributed Stochastic Neighbour Embedding (t-SNE) is also used for visualisation. It is a non-linear technique that is effective at revealing the underlying local structure of data. It maps the high-dimensional feature vectors into a 2D space, providing a clear visual map of how individual data points and classes cluster together. This helps in quantitively assessing the separability of the fault classes and provides a different perspective on the influence of operational load conditions.

2.3. *Clustering*

Unsupervised clustering aims to group high-dimensional features based on natural geometric proximity without pre-defined labels. In this study, K-means is utilised as the primary algorithm, employing the Euclidean distance to minimise the within-cluster sum of squares. To assess the structure of the resulting feature space, metrics are categorised into internal and external validation.

Internal vs. external metrics:

- Internal validation: This is an unsupervised method that does not require labels. It evaluates clusters based on their geometric properties, like shape and spread. For hierarchical analysis, split quality metrics like the Silhouette Score and Davies–Bouldin Index are employed to measure inter-cluster separation and intra-cluster cohesion.
- External validation: This uses ground-truth labels to evaluate the purity of these clusters, providing a baseline to determine if geometric groupings correspond to true fault classes. This mutimetric approach allows for a quantitative assessment of how torque loads shadow the fault features.

The first part of the Table 1 presents the similarity and distance measures, which are not validation metrics but rather the mathematical rulers used by clustering algorithms to determine how "close" or "similar" two data points are. The choice of measure can fundamentally change how clusters are formed. In our case, K-means uses the Euclidean distance.

Table 1. Summary of similarity and distance measures and cluster validation metrics.

Metric	Type	Requires Child Cluster	Validation	Ranges, Ideal	Description
Euclidean Distance [28]	Similarity measure	No	Distance	[0, ∞], lower	Standard point-to-point geometric distance, used for cluster assignment
Diameter (proposed)	Cluster quality	No	Internal	[0, ∞], lower	Largest distance between two points in a cluster
Spread [29]	Cluster quality	No	Internal	[0, ∞], lower	Average standard deviation across cluster features; scatter of points
Centroid Cohesion (WCSS)	Cluster quality	No	Internal	[0, ∞], lower	Total squared distance from points to centroid
Dominance Score (proposed)	Cluster quality	No	External	[0, 1], higher	Proportion of dominant label in a cluster
Silhouette Score [30]	Cluster & Split quality	Yes	Internal	[−1, 1], higher	Measures cohesion and separation by comparing intra-cluster and nearest inter-cluster distances

Table 1. *Cont.*

Metric	Type	Requires Child Cluster	Validation	Ranges, Ideal	Description
Calinski–Harabasz Index [31]	Split quality	Yes	Internal	[0, ∞], higher	Ratio of between-cluster dispersion to within-cluster dispersion
Davies–Bouldin Index [32]	Split quality	Yes	Internal	[0, ∞], lower	Ratio of within-cluster spread to separation between clusters
Dunn's Index [33]	Split quality	Yes	Internal	[0, ∞], higher	Minimum inter-cluster distance divided by maximum intra-cluster diameter
V-measure [34]	Split quality	Yes	External	[0, 1], higher; 0 is also good.	Harmonic mean of homogeneity and completeness for clustering vs. classes

The second part of the table presents the validation metrics used to score the quality of the resulting clusters. These are critical for interpreting the output and are divided into internal and external validation metrics, as mentioned before. The suite of metrics allows us to assess whether the clusters are geometrically good, using internal metrics, and/or whether the clusters are correct, using external metrics. This distinction allows us to find the relationship between internal validation and external validation, which is key in understanding the impact of contextual features, such as the load torque.

In this analysis, not all the validation metrics have been used, as some gave different results compared to the majority of the other metrics, such as the "Mahalanobis distance" metric. Also, more focus is given to internal validation metrics than to external ones; therefore, others were excluded as well, such as the "Rand Index" and "Jaccard Index".

3. Methodology

Building on the theoretical background in Section 2, the proposed methodology consists of extracting features from a raw motor current signal and analysing the resulting feature space. The process begins with selection of appropriate datasets that contain variations in both mechanical fault types and operating conditions, such as the torque load. A one-dimensional convolutional neural network (1D-CNN) is employed to learn high-dimensional representation of the data. Finally, a series of preprocessing methods is applied, along with PCA and t-SNE for visualisation of the features. Consequently, clustering is employed, using K-means, which is used in supervised and unsupervised tasks. Finally, hierarchical clustering is explored using K-means, in order to understand the structure, relationship and separability of the data.

3.1. Datasets

In this study, two publicly available datasets known for their use in condition monitoring, with varying operating conditions, are utilised: the KAIST motor dataset [11] and PU (Paderborn University) motor dataset [9].

The KAIST dataset covers most of the major mechanical faults, more specifically, bearing damage (inner: BPFI; outer: BPFO), shaft misalignment, and rotor unbalance. The data was collected under different load conditions (0 Nm, 2 Nm, 4 Nm), from a 3 hp, 4-pole AC motor driven at 380 V, 60 Hz at a rated speed of 1770 rpm, sampled at a frequency of

25.6 kHz. However, in the dataset, the bearing outer (BPFO) class has not been used as the files were corrupt and not possible to use. Furthermore, after analysing the Technical Data Management Streaming (TDMS) format files for the current data, we noticed that the 0 Nm Normal file has 7,682,458 data points, whereas the 2 Nm and 4 Nm Normal files have 3,072,983 data points. The 0 Nm dataset should have had the same number of data points as the 2 Nm and 4 Nm datasets; as the sampling frequency is 25.6 kHz and measurement time is 120 s, around 3,072,000 data points are expected, which matches with the 2 Nm and 4 Nm Normal files. Therefore, the data has been used with the above considerations. The specifications and files used are summarised in Table 2.

Table 2. KAIST motor dataset operating conditions [11].

Sampling Rate (kHz)	Length (s)	Fault Types	Fault Severity	Load (Nm)	File Name
25.6	120	Normal	n/a	0 2 4	0Nm_Normal 2Nm_Normal 4Nm_Normal
	60	Unbalance	583 mg	0 2 4	0Nm_Unbalance_0583mg 2Nm_Unbalance_0583mg 4Nm_Unbalance_0583mg
			1169 mg	0 2 4	0Nm_Unbalance_1169mg 2Nm_Unbalance_1169mg 4Nm_Unbalance_1169mg
			1751 mg	0 2 4	0Nm_Unbalance_1751mg 2Nm_Unbalance_1751mg 4Nm_Unbalance_1751mg
			2239 mg	0 2 4	0Nm_Unbalance_2239mg 2Nm_Unbalance_2239mg 4Nm_Unbalance_2239mg
			3318 mg	0 2 4	0Nm_Unbalance_3318mg 2Nm_Unbalance_3318mg 4Nm_Unbalance_3318mg
		Misalignment	0.1 mm	0 2 4	0Nm_Misalign_01 2Nm_Misalign_01 4Nm_Misalign_01
			0.3 mm	0 2 4	0Nm_Misalign_03 2Nm_Misalign_03 4Nm_Misalign_03
			0.5 mm	0 2 4	0Nm_Misalign_05 2Nm_Misalign_05 4Nm_Misalign_05
		Bearing Inner	0.3 mm	0 2 4	0Nm_BPFI_03 2Nm_BPFI_03 4Nm_BPFI_03
			1.0 mm	0 2 4	0Nm_BPFI_10 2Nm_BPFI_10 4Nm_BPFI_10
			3.0 mm	0 2 4	0Nm_BPFI_30 2Nm_BPFI_30 4Nm_BPFI_30

The Paderborn University (PU) dataset is a widely used benchmark for data-driven classification tasks in condition monitoring. Unlike the KAIST dataset, the PU dataset uses a Permanent Magnet Synchronous Motor (PMSM), which does not have motor slip (the difference between synchronous speed and rotor speed). The motor has a power rating of 425 W, a nominal torque of 1.35 Nm and a nominal speed of 3000 rpm, with a pole pair number of $p = 4$. The current signal was captured at a high sampling frequency of 64 kHz. The PU dataset used in this work contains exclusively mechanical bearing damage, such as healthy bearings and bearings with inner-ring, outer-ring and inner/outer-ring damage.

The data has four classes, specifically the Healthy class, InnerRing damage class, OuterRing damage class, and OuterInnerRing damage class. The same as the KAIST dataset, this one has multiple operating conditions as well. These are summarised in Table 3. We will be using operating conditions No. 0 and 2, which have different load torques.

Table 3. PU dataset operating conditions [9].

No.	Rotational Speed (rpm)	Load Torque (Nm)	Radial Force (N)	Name of Setting
0	1500	0.7	1000	N15_M07_F10
1	900	0.7	1000	N09_M07_F10
2	1500	0.1	1000	N15_M01_F10
3	1500	0.7	400	N15_M07_F04

3.2. Preprocessing

In order for the data to be trained, the raw motor current data undergoes a series of preprocessing steps to ensure it is in a suitable format for the model. This involves segmenting the continuous signal using a sliding window, normalising the data, and splitting it into training and testing sets.

The raw time-series signal is first divided into segments of 5120 samples each, which based on the literature review is the size chosen usually. Each of the segments is then standardised using Z-score normalisation. The process ensures that every segment has a mean of 0 and a standard deviation of 1, which helps stabilise and accelerate the training process.

In order to avoid data leakage, where segments from the same file appear in training and testing sets, the splitting is done at the file level. It is a more robust method, where the files are split into 80/20% groups before any segmentation occurs. This ensures that the model is tested on data not seen before, providing a more realistic assessment of its generalisation capability. It is important to note that in the case of the "Normal" class, the splitting has been done at the segment level, as the number of "Normal" files is 3 and therefore would not be ideal for splitting at the file level. The segment distribution is presented in Table 4.

Table 4. Segment distribution for KAIST and PU datasets.

	Files per Class	File Distribution After Splitting (80/20)		Segment Distribution After Splitting (80/20)	
		Training	Testing	Training	Testing
		KAIST dataset classes			
0 (Bearing)	9	7	2	2100	600
1 (Misalignment)	9	7	2	4200	1200
2 (Normal)	3	2	1	2100	600
3 (Unbalance)	15	12	3	7200	1800

Table 4. *Cont.*

	Files per Class	File Distribution After Splitting (80/20)		Segment Distribution After Splitting (80/20)	
		Training	Testing	Training	Testing
		PU dataset classes			
0 (Healthy)	120	96	24	4806	1200
1 (Outer Ring)	240	192	48	9628	2406
2 (OuterInner Ring)	60	48	12	2399	601
3 (Inner Ring)	220	176	44	8811	2201

As the classes have different numbers of files and therefore are imbalanced, a Weighted Cross-Entropy loss function is used during training, where the model pays more attention to under-represented faults and does not become biased towards the majority class.

3.3. Model

As shown in the literature review, there are multiple models that can be used; however, the 1D-CNN has shown great results and good extraction of the features needed for the analysis. Therefore, this model is employed, and it is one-dimensional, as it is a current time-series measurement. In this analysis, this model is not only used for classification output, but as mentioned, it serves as a powerful unsupervised feature extractor. The raw current signals are segmented into sliding windows of 5120 data points in length. These segments are then fed into the network, where they propagate through a series of four convolutional layers designed to capture high-frequency signatures associated with mechanical faults. The architecture then expands the feature depth from 32 to 256 filters, with a kernel size of 5 and then 3. The stride and padding are at default values, therefore 1 for stride and 0 for padding. Each convolution layer is followed by batch normalisation, ReLU activation and max pooling. The model is trained using the Adam optimiser with a learning rate of 0.0001, using a batch size of 64 over a maximum of 200 epochs, with early stopping to prevent overfitting. The architecture is shown in Figure 1.

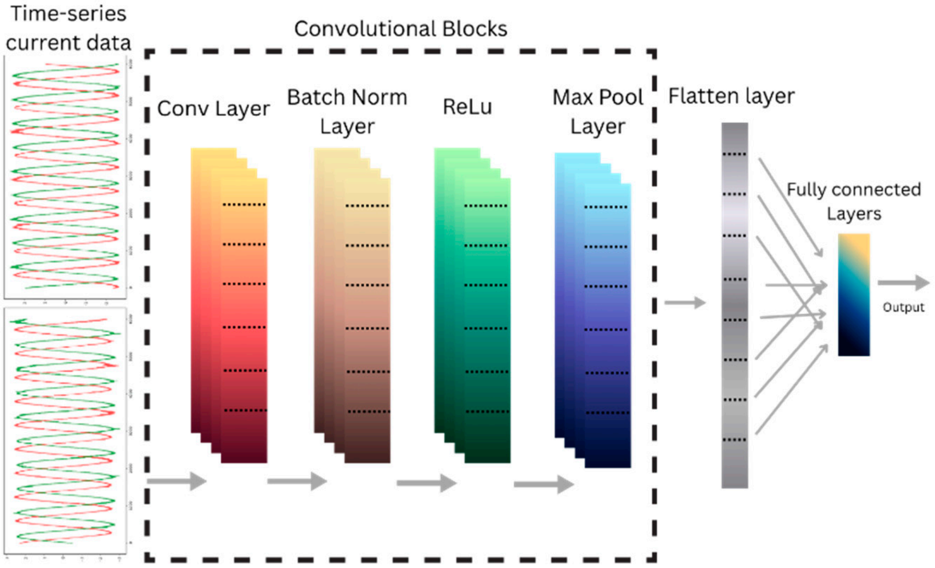


Figure 1. 1D-CNN architecture used.

The parameters of the 1D-CNN for both datasets utilised are detailed in Table 5 below.

Table 5. 1D-CNN parameters for KAIST and PU datasets.

Parameter	KAIST	PU
Segment length	5120	5120
conv layer 1	Filters: 32; kernel: 5	Filters: 32; kernel: 5
conv layer 2	Filters: 64; kernel: 5	Filters: 64; kernel: 5
conv layer 3	Filters: 128; kernel: 5	Filters: 128; kernel: 5
conv layer 4	Filters: 256; kernel: 3	Filters: 256; kernel: 3
Pooling	MaxPool1D (2)	MaxPool1D (2)
Dense layer 1	Flatten 256	Flatten 256
Dense layer 2	256 (num classes)	256 (num classes)
Dropout	0.5	0.5
Optimiser	Adam	Adam
Learning rate	0.0001	0.0001
Weight decay	1×10^{-4}	1×10^{-4}
Batch size	64	64
Epochs	200	200

3.4. Feature Space Analysis Techniques

Once the features are extracted by the 1D-CNN, a suite of unsupervised analysis techniques is applied to the resulting learned feature space. The purpose of this analysis is to visualise the high-dimensional data, understand the structure and assess the influence of contextual operating conditions on how the model organises the data. For this purpose, Principal Component Analysis (PCA) is used, along with t-SNE, and consequently clustering and hierarchical clustering are applied to these features using K-means.

3.4.1. Removal of Principal Components

In the literature, different authors have defined the quality of the principal components using different metrics. However, the standard one is the explained variance from the PCs. The hypothesis is that, as these components are ordered by the most variance, it is likely that the top PCs capture the dominant features, which, as shown before, are the load conditions.

The approach here consists of multiple experiments, summarised also in Figure 2:

1. Removing multiple top PCs from the full feature space.
2. Removing multiple top PCs from a variance-reduced (80%) space.
3. Removing a single top PC from the full feature space.
4. Removing a single top PC from the variance-reduced (80%) space.

The first approach is therefore the cumulative removal of the top five principal components from the full feature space. The second approach reduces the component space to retain 80% variance, which corresponds to less features and dimensions, and then explores removing the top variance components. The third and fourth approaches test if a single principal component (e.g., just PC0 or PC1) is singularly responsible for the load dominance.

Thus, the third approach consists of removing only a single principal component from the full dimension space, and in the fourth approach, a single top PC is removed but from a reduced dimension space (80% variance).

3.4.2. Clustering

In order to quantitatively and qualitatively assess the structure of the learned feature space, unsupervised clustering algorithms are employed. The goal is to determine if the data points corresponding to different fault types form distinct, natural groups and see how these are affected by the operating conditions. There are multiple clustering algorithms such as DBSCAN, HDBSCAN, Agglomerative Clustering and others [35]; however, the well-known K-means is well-suited for this analysis, as it can work with an undefined number of clusters or by specifying the number of clusters. The latter will be used for hierarchical clustering, to build a tree-like diagram of nested clusters, known as a dendrogram. It provides a powerful visualisation of the data's structure without needing a pre-specified number of clusters. This is important for exploring the relationship between different fault classes and operating conditions and understanding separability at various levels of granularity. In this way, it is possible to analyse if the data points group first by the operating load and then by fault type, which would quantitatively confirm the dominance of the contextual feature. The quality of these clusters is then assessed using a series of metrics. Some of these metrics are taken from the literature review, and others are calculated and created, such as the diameter and the Dominance Score. These are summarised in Table 1. In this analysis, not all the validation metrics have been used, as some gave different results compared to the majority of the other metrics, such as the "Mahalanobis distance" metric. Also, more focus is given to internal validation metrics than to external ones; therefore, others were excluded as well, such as the "Rand Index" and "Jaccard Index".

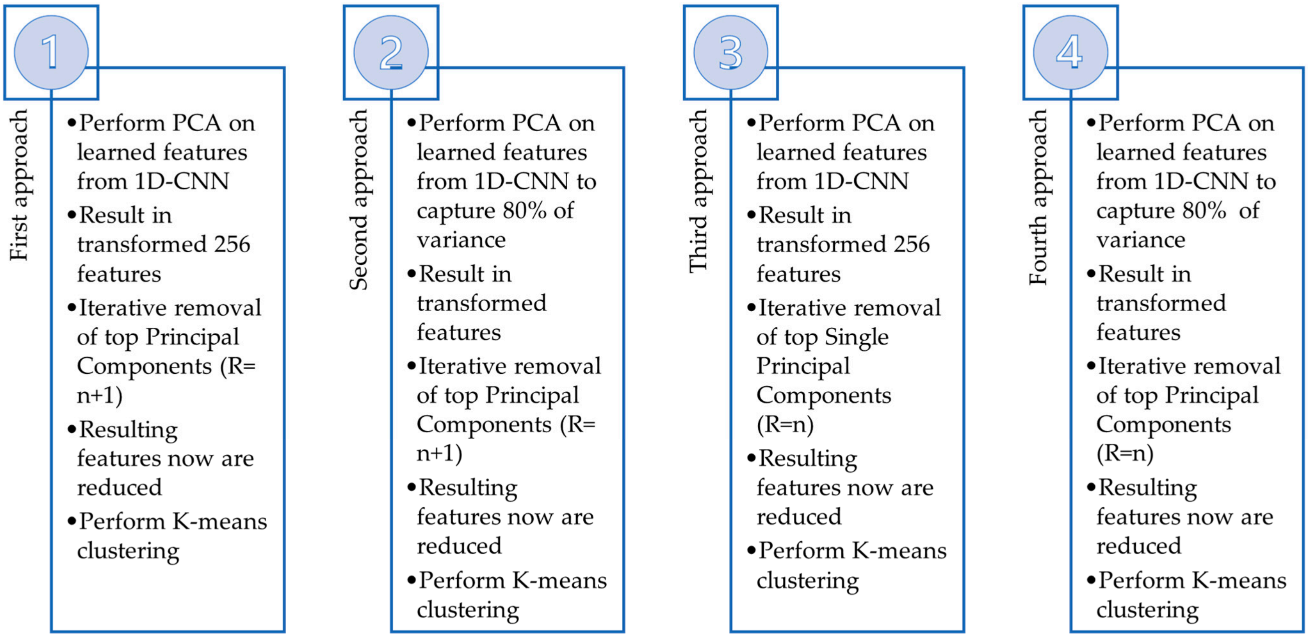

1
First approach
•Perform PCA on learned features from 1D-CNN
•Result in transformed 256 features
•Iterative removal of top Principal Components (R= n+1)
•Resulting features now are reduced
•Perform K-means clustering
2
Second approach
•Perform PCA on learned features from 1D-CNN to capture 80% of variance
•Result in transformed features
•Iterative removal of top Principal Components (R= n+1)
•Resulting features now are reduced
•Perform K-means clustering
3
Third approach
•Perform PCA on learned features from 1D-CNN
•Result in transformed 256 features
•Iterative removal of top Single Principal Components (R=n)
•Resulting features now are reduced
•Perform K-means clustering
4
Fourth approach
•Perform PCA on learned features from 1D-CNN to capture 80% of variance
•Result in transformed features
•Iterative removal of top Principal Components (R=n)
•Resulting features now are reduced
•Perform K-means clustering

Figure 2. Process of removal of principal components to remove load dominance from fault features: removing multiple top PCs from full feature space (1), removing multiple top PCs from a variance-reduced space (2), removing a single top PC from full feature space (3), and removing a single top PC from a variance-reduced space (4).

3.5. Experimental Environment

All deep learning models and clustering algorithms were implement using Python 3.10.16, PyTorch 2.5.1+ cuda 12.4, numpy 2.1.3, pandas 2.2.3, scikit-learn 1.6.0, scipy 1.14.1, matplotlib 3.9.3 and seaborn 0.13.2. The hardware configuration consisted of an AMD Ryzen 9 5900X 12-Core processor, 64 GB of RAM, and an NVIDIA GeForce RTX 3090 GPU.

4. Results

This section reports the experimental results obtained using the proposed methodology. First, we evaluate the baseline supervised 1D-CNN performance under different torque loads on both the KAIST and PU datasets. We then analyse the learned feature space using PCA and t-SNE visualisations, followed by clustering and hierarchical analysis. Finally, we examine the effect of different principal component removal strategies on the classification performance and feature space structure.

4.1. Experiment A: 1D-CNN on Multiple Torque Loads

The model is trained and tested on both the KAIST motor dataset and PU motor dataset using raw time-series current signals. The 1D-CNN architecture employed in this study is designed as a standardised baseline feature extractor rather than a highly optimised domain-invariant classifier. Therefore, it is important to note that the objective of this study is not to develop an architecture that overfits these datasets but to use a representative general-purpose architecture, as shown in the Introduction, to expose the intrinsic behaviour of deep features under load shifts. The classification degradation observed in this study should be interpreted as a characteristic of a standard supervised learning paradigm when facing distribution shifts, and we utilised a fixed simplified architecture to ensure that the resulting feature space distortion is a product of the domain shift, rather than the result of complex architectural regularisation. The quantification of the impact of these varying conditions is also the focus of this work, establishing a benchmark before proceeding with feature space analysis.

4.1.1. KAIST Motor Dataset

The results for the KAIST motor dataset, presented in Table 6, reveal a degradation in classification performance as the torque increases. At 0 Nm load, the model achieves 100% accuracy and an F1 score of 1 for all fault classes, indicating that under stable operating conditions the 1D-CNN can reliably distinguish between fault features. At 2 Nm load, the performance drops to an average accuracy of 91.73%, with a marked decrease in the performance of the Normal class, which is affected both by the higher load and larger number of data points in the 0 Nm Normal file noted previously in the dataset description. Performance for the other classes also decreases, suggesting that the change in load already impacts the separability of specific fault classes. At 4 Nm load, the performance degrades even more to an average accuracy of 53.19%. At this load, the model is highly confused. For all loads combined, when the model is trained and tested on the full dataset mixing all three loads, the average accuracy is 87.97%, which is higher than the 4 Nm only case but still notably lower than for 0 Nm and 2 Nm alone, and the higher standard deviation indicates less table training across runs. While this is higher than the 4 Nm only case, it is significantly worse than the 0 Nm and 2 Nm cases. The high standard deviation also suggests that training is unstable, as the model struggles to find consistent patterns across the conflicting load conditions. In terms of computational cost, training on the 4 Nm data requires slightly less time than for 0 Nm and 2 Nm, and this is reflected in the corresponding CO_2 emission estimates.

Table 6. KAIST motor dataset 1D-CNN results.

Load	Class	Precision	Recall	F1	Avg Accuracy	Avg Macro F1	Avg Weighted F1	Time (s)	CO_2 (kg)
0	BPFI	1.0000 (±0.0000)	1.0000 (±0.0000)	1.0000 (±0.0000)	100.00% (±0.00%)	1.0000 (±0.0000)	1.0000 (±0.0000)	250.05 ± 12.11	0.0067 ± 0.0002
	Misalignment	1.0000 (±0.0000)	1.0000 (±0.0000)	1.0000 (±0.0000)					
	Normal	1.0000 (±0.0000)	1.0000 (±0.0000)	1.0000 (±0.0000)					
	Unbalance	1.0000 (±0.0000)	1.0000 (±0.0000)	1.0000 (±0.0000)					

Table 6. *Cont.*

Load	Class	Precision	Recall	F1	Avg Accuracy	Avg Macro F1	Avg Weighted F1	Time (s)	CO_2 (kg)
2	BPFI	1.0000 ± 0.0000	1.0000 (±0.0000)	1.0000 ± 0.0000	91.73% ± 0.71%	0.7284 ± 0.0045	0.8890 ± 0.0033	259.40 ± 10.28	0.0065 ± 0.0001
	Misalignment	1.0000 (±0.0000)	1.0000 ± 0.0000	1.0000 ± 0.0000					
	Normal	0.0563 ± 0.0594	0.0100 ± 0.0122	0.0165 ± 0.0198					
	Unbalance	0.8311 ± 0.0026	0.9747 ± 0.0205	0.8971 ± 0.0100					
4	BPFI	0.9822 ± 0.0201	0.9733 ± 0.0281	0.9775 ± 0.0184	53.19% ± 2.56%	0.5129 ± 0.0243	0.4929 ± 0.0273	224.12 ± 8.10	0.0060 ± 0.0001
	Misalignment	0.2520 ± 0.0950	0.0857 ± 0.0443	0.1263 ± 0.0600					
	Normal	0.2070 ± 0.0543	0.5467 ± 0.1744	0.2903 ± 0.0556					
	Unbalance	0.5887 ± 0.1213	0.7543 ± 0.0515	0.6577 ± 0.0898					
ALL	BPFI	0.8579 ± 0.2842	0.9223 ± 0.1512	0.8420 ± 0.2163	87.97% ± 14.62%	0.8901 ± 0.1341	0.8809 ± 0.1416	765.51 ± 50.32	0.0198 ± 0.0007
	Misalignment	0.9626 ± 0.0717	0.7445 ± 0.2967	0.8006 ± 0.2443					
	Normal	0.9780 ± 0.0439	1.0000 ± 0.0000	0.9884 ± 0.0232					
	Unbalance	0.9396 ± 0.0897	0.9356 ± 0.1183	0.9295 ± 0.0735					

4.1.2. PU Motor Dataset

The PU dataset here is not used for comparison with the KAIST motor dataset but to assess whether similar load-related effects are observed. The results presented in Table 7 show a moderate overall performance, with the 1D-CNN achieving 78.58% accuracy on M01 (0.1 Nm load) and 75.96% accuracy on M07 (0.7 Nm load). When data from both load settings are combined, the average accuracy drops to 71.49%.

Table 7. PU motor dataset 1D-CNN results.

Load	Class	Precision	Recall	F1	Avg Accuracy	Avg Macro F1	Avg Weighted F1	Time (s)	CO_2 (kg)
M01	Healthy	0.779 (±0.0621)	0.6882 (±0.1081)	0.7216 (±0.0537)	78.58% (±1.12%)	0.8071 (±0.0134)	0.7829 (±0.0122)	453.41 (±88.23)	0.0135 (±0.0026)
	OuterRing	0.7704 (±0.045)	0.7639 (±0.0884)	0.7615 (±0.0292)					
	InnerRing	0.7928 (±0.0779)	0.8132 (±0.0986)	0.7932 (±0.0206)					
	OuterInnerRing	0.9395 (±0.048)	0.9671 (±0.0388)	0.9522 (±0.0329)					
M07	Healthy	0.7327 (±0.1011)	0.756 (±0.1763)	0.7169 (±0.0757)	75.96% (±1.07%)	0.7811 (±0.0149)	0.7559 (±0.0153)	510.60 (±84.23)	0.0153 (±0.0025)
	OuterRing	0.7482 (±0.0255)	0.6598 (±0.0465)	0.6997 (±0.0207)					
	InnerRing	0.7762 (±0.0847)	0.8374 (±0.0937)	0.7959 (±0.018)					
	OuterInnerRing	0.9545 (±0.0456)	0.8818 (±0.0966)	0.9119 (±0.0414)					

Table 7. *Cont.*

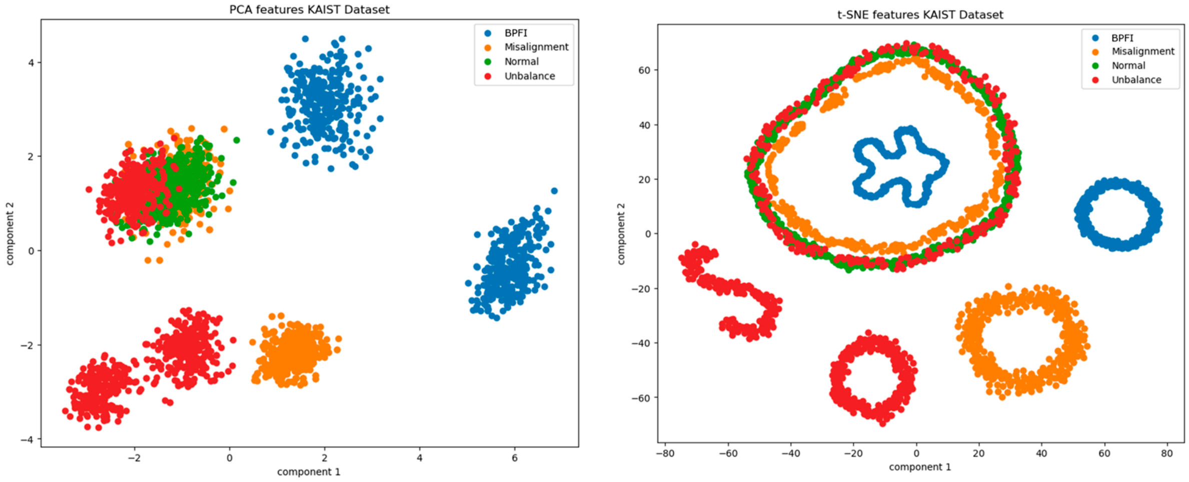
Load	Class	Precision	Recall	F1	Avg Accuracy	Avg Macro F1	Avg Weighted F1	Time (s)	CO_2 (kg)
Both	Healthy	0.7966 (±0.0827)	0.6438 (±0.0803)	0.7044 (±0.032)	71.49% (±1.12%)	0.7465 (±0.0149)	0.7123 (±0.0113)	895.10 (±233.91)	0.0267 (±0.0070)
	OuterRing	0.7273 (±0.0207)	0.5924 (±0.0167)	0.6526 (±0.0122)					
	InnerRing	0.6455 (±0.0371)	0.8463 (±0.0379)	0.7309 (±0.0197)					
	OuterInnerRing	0.9371 (±0.0408)	0.8649 (±0.0646)	0.898 (±0.0415)					

As with the KAIST dataset, combining multiple operating conditions degrades the classifier's performance and makes generalisation across loads more challenging. In terms of the training time, M07 took, on average, longer to train than M01, which is also reflected in higher estimated CO_2 emissions.

4.2. Experiment B: Feature Visualisation Using PCA and t-SNE

After analysing the model classification accuracy, the focus shifts to a direct analysis of the learned feature space extracted from the penultimate dense layer of the 1D-CNN. By employing PCA and t-SNE, the high-dimensional feature space is projected into two dimensions and three dimensions for visualisation. The objective is to visually inspect how the model organises the data and to identify the influence of both the fault classes and the operating conditions.

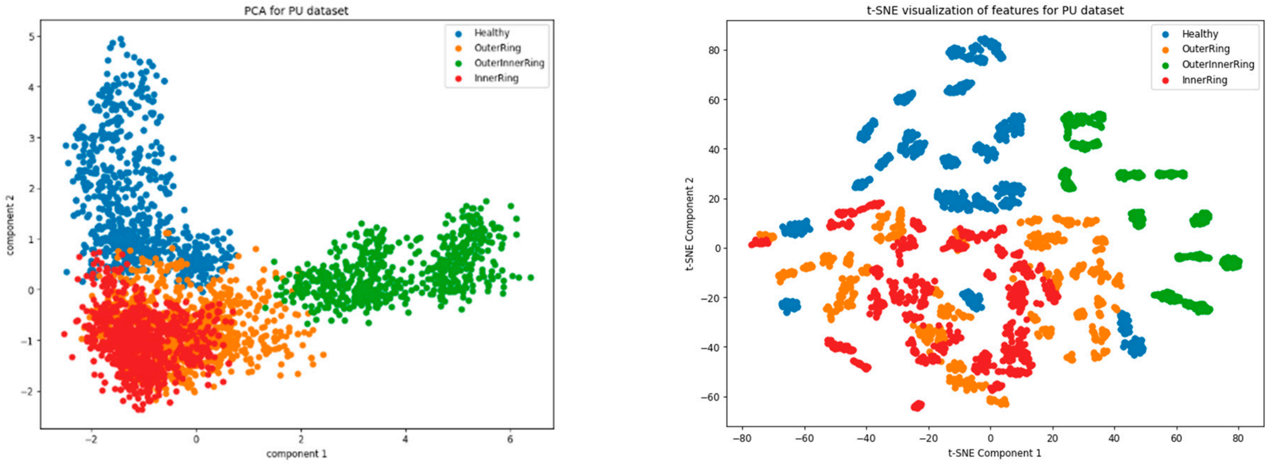
For the KAIST dataset, the visualisations reveal a well-structured feature space where the fault classes are clearly distinct, as shown in Figure 3. The PCA plot reveals that the learned features are separable to a large degree. The BPFI class forms a highly distinct cluster, well-separated from the others. The other classes also form their groupings, though they are situated more closely together and there is some overlapping space where there is a mixture of common features from the Unbalance, Normal and Misalignment classes. With regard to t-SNE, it reveals a separation of the classes, forming some shapes in the feature space. The circle shows, as in the PCA plot, the common features from some classes. This indicates that the 1D-CNN successfully learned non-linear features that distinguish between the different fault types.



(**a**) (**b**)

Figure 3. Damage class features for all loads, visualised using PCA and t-SNE for KAIST motor dataset: (**a**) PCA visualisation and (**b**) t-SNE visualisation.

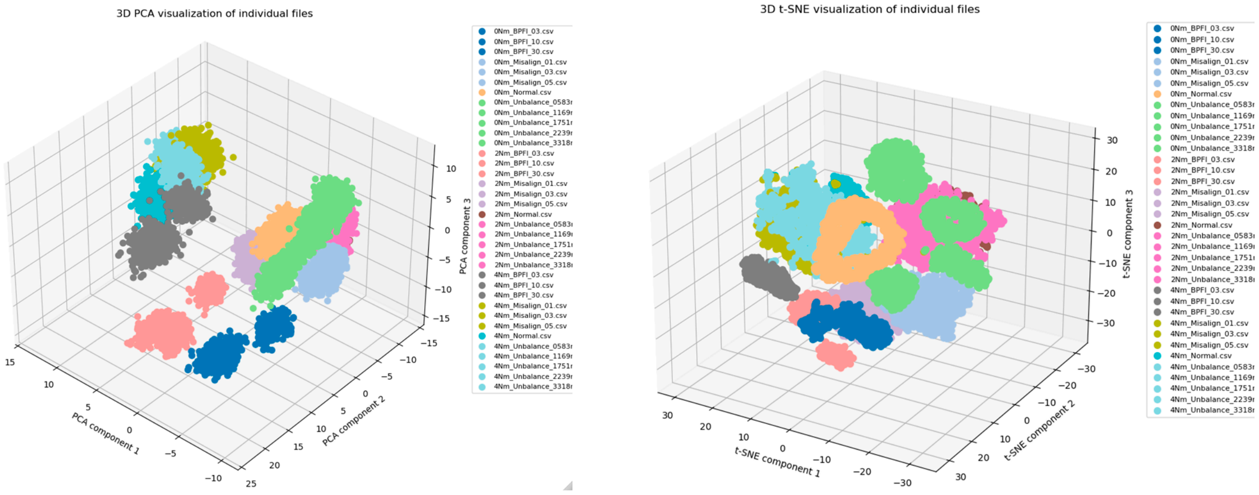
In contrast, the feature space for the PU dataset shown in Figure 4 shows more overlapping and less clear separation between the fault classes. The PCA plot reveals considerable mixing of the fault classes. Although t-SNE improves the visualisation by forming more defined local clusters, significant overlap between the classes remains, and the clusters are less distinctly separated than in KAIST.



(**a**) (**b**)

Figure 4. Damage class features for all loads, visualised using PCA and t-SNE for PU motor dataset: (**a**) PCA visualisation and (**b**) t-SNE visualisation.

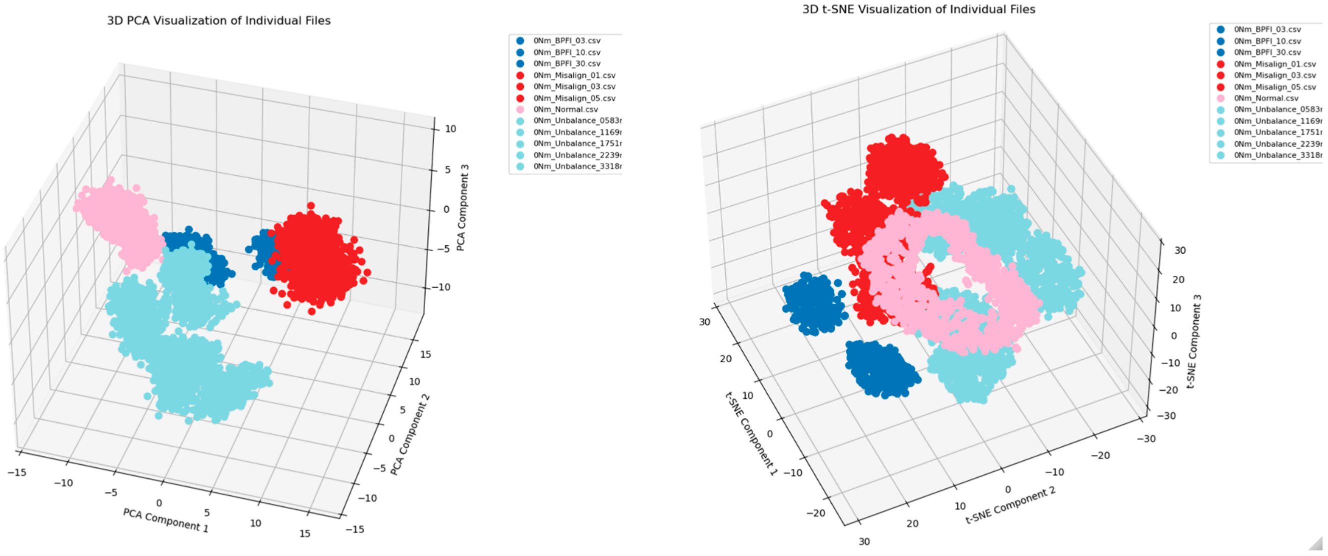
To gain a more granular understanding of the feature space, 3D visualisations were generated with the learned features labelled by the individual data files, as visualised in Figure 5. The 3D t-SNE plots reveal a clear pattern, where data points originating from the same source file form tight and distinct clusters. This shows that the features learned by the 1D-CNN are consistent. The same trend is observed in the PCA plot, but due to PCA being a linear technique, the clusters are less compact and show some overlap. A key observation here is in relation to measurements at a 4 Nm load. Looking at the PCA plot, in the top-left section of the graph, there is a lot of overlap of features from different fault classes but at the same load torque. This indicates that at a higher load, the features behave differently and less distinctively.



(**a**) (**b**)

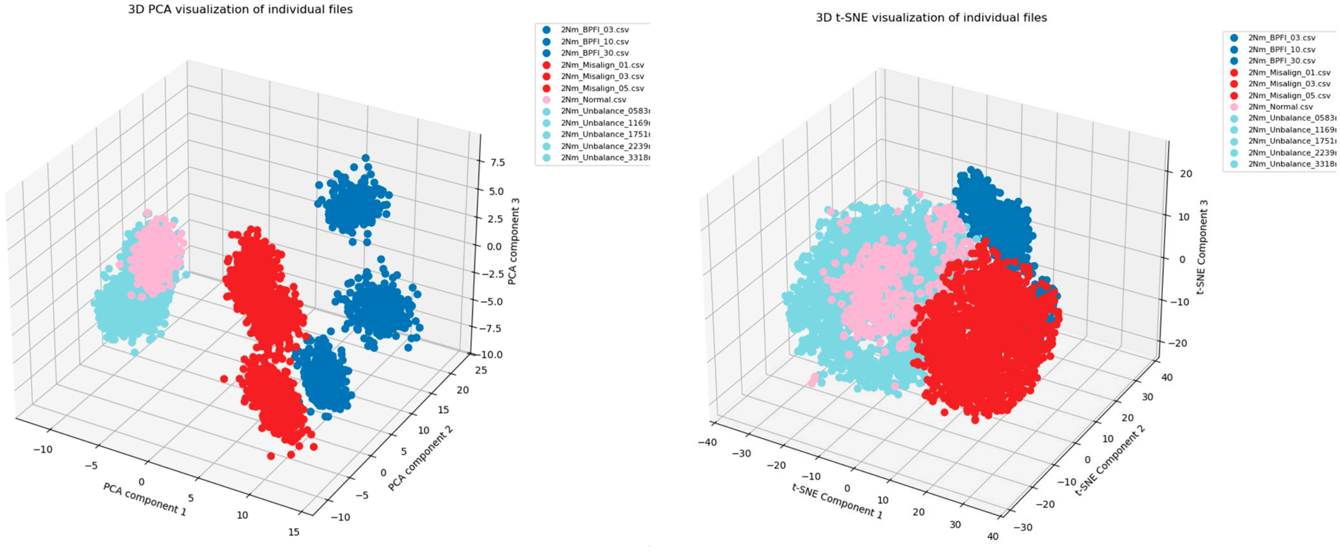
Figure 5. Individual file features, visualised using PCA and t-SNE for KAIST motor dataset: (**a**) PCA visualisation and (**b**) t-SNE visualisation.

To isolate the effect of torque load, the feature space is replotted for each load condition separately, as shown in Figures 6–8. At 0 Nm, the PCA and t-SNE plots show distinct and well-separated groupings for each fault type, with tight, dense clusters. At 2 Nm, the BPFI class remains clearly distinct, but the clusters for the other classes move closer together, and their boundaries become less sharp. At 4 Nm, the PCA plot shows a dense region where points from different fault classes largely overlap, and the corresponding t-SNE visualisation exhibits a mixed distribution with no clear local class-wise structure.



(**a**) (**b**)

Figure 6. Visualisation of 0 Nm load features using PCA and t-SNE for KAIST motor dataset: (**a**) PCA visualisation and (**b**) t-SNE visualisation.

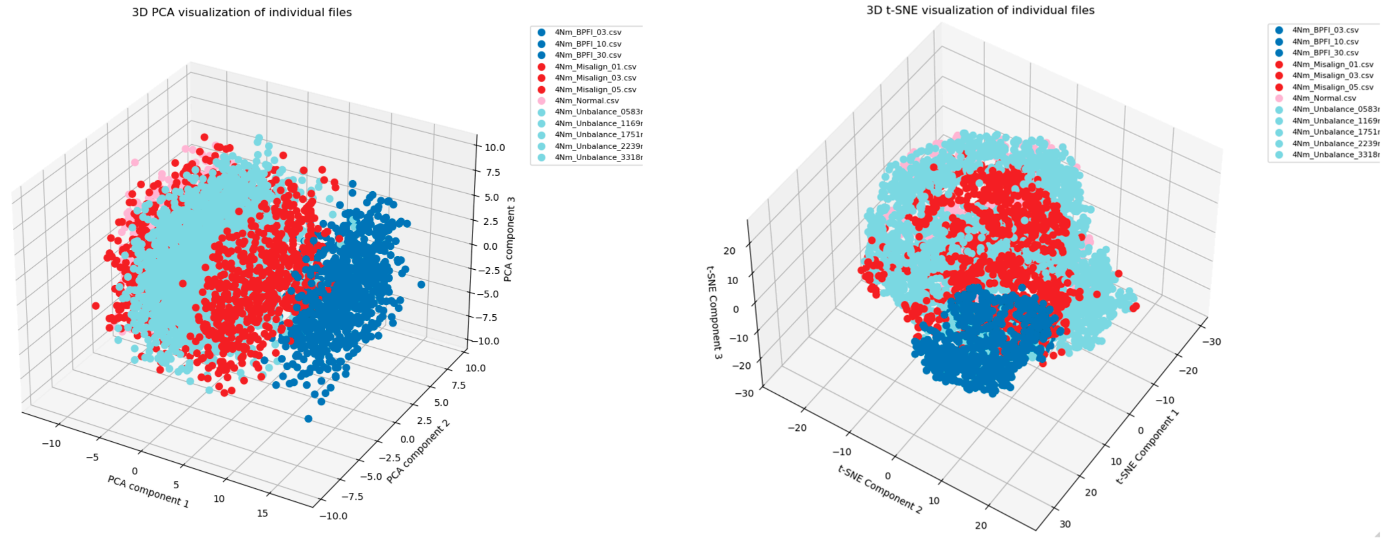


(**a**) (**b**)

Figure 7. Visualisation of 2 Nm load features using PCA and t-SNE for KAIST motor dataset: (**a**) PCA visualisation and (**b**) t-SNE visualisation.

4.3. Experiment C: Removal of Principal Components

To move from visual inspection to a quantitative analysis, the composition of the clusters formed by unsupervised algorithms is examined. K-means is first applied to the learned feature space to group the data based on geometric similarity without labels, and ground-truth labels are used to determine purity of the resulting clusters. Figure 9 shows the resulting cluster composition using stacked bar charts for one of the principal component removal experiments, where K-means identifies six clusters although the data contain four damage classes. Some clusters contain more mixed damage labels, such as in Cluster 6, while others are relatively pure, motivating a more detailed tabular analysis of how principal component removal affects the dominance of contextual features such as the torque load.



(**a**) (**b**)

Figure 8. Visualisation of 4 Nm load features using PCA and t-SNE for KAIST motor dataset: (**a**) PCA visualisation and (**b**) t-SNE visualisation.

This experiment evaluates the effect of removing the top principal components (PCs) from the learned feature space as a potential mitigation strategy. After each removal step, K-means clustering is performed, and a set of purity metrics is computed to assess the separation of the fault classes. For each cluster, the variance in the percentages of the four fault classes is calculated, followed by standardised variance where 100 corresponds to a pure cluster and 0 to a fully mixed cluster. The average of these standardised variances across clusters is used as a summary purity measure for each iteration.

In the first approach shown in Table 8, the baseline (k = 0) has an average score of 73.77. Upon removing just the top PC (k = 1), the score plummets to 37, and further removal of additional components leads to continued degradation, although some individual clusters remain pure. A similar patter appears in the second approach, presented in Table 9, where the average score drops from 59.36 at baseline to 37 after removing the top PC, with subsequent removal further lowering cluster purity. Tables 8–11 include colouring from red to green which respectively indicates lower scores and higher scores.

In the third approach, presented in Table 10, removing only PC1 decreases the average score from 73.77 to 25.87, and removing PC2 gives a score of 15.14. Across this approach, the purity scores do not improve, and the resulting clusters become increasingly mixed in terms of fault labels. In the fourth approach, shown in Table 11, applied in the reduced space, removing PC1 and PC2 again lowers cluster purity. Removing PC3 produces 10 clusters, many of which achieve relatively high purity scores, but the number of clusters no longer corresponds to the expected four fault classes.

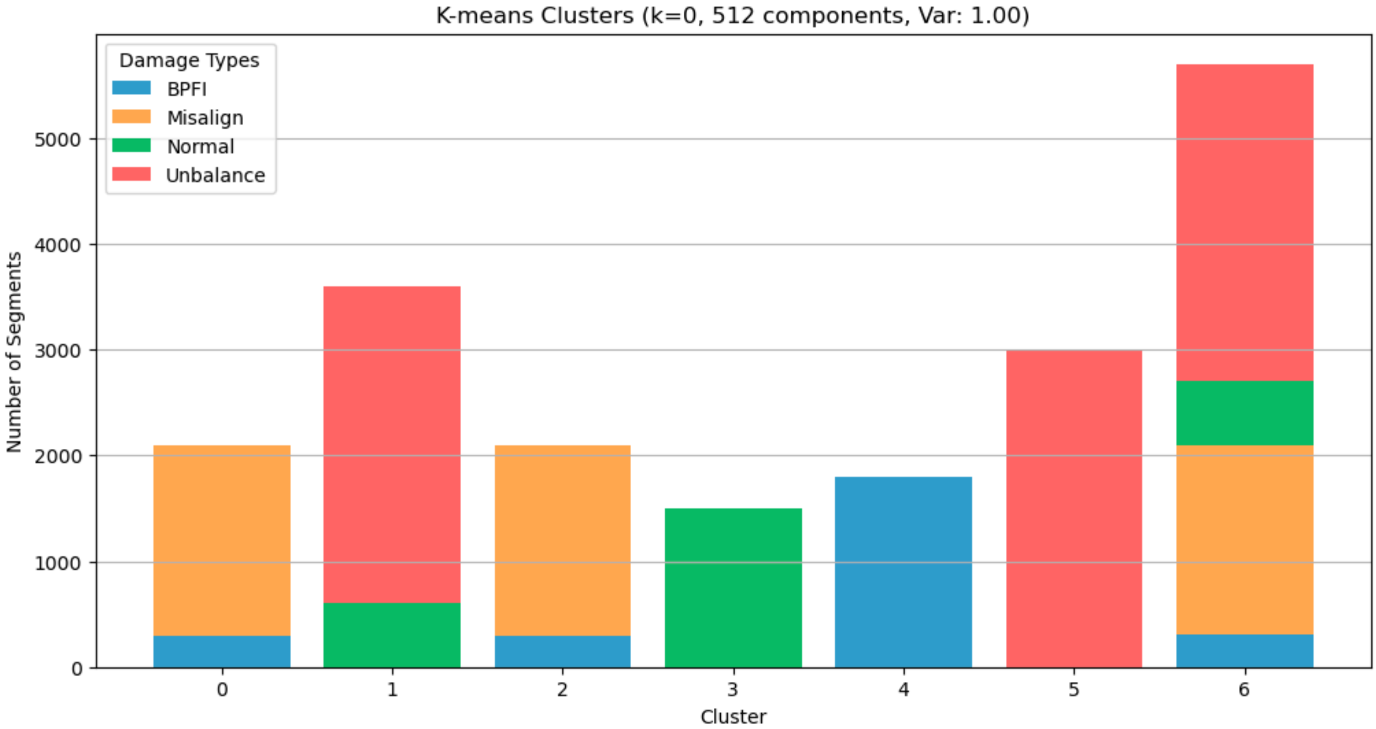


Figure 9. Cluster composition using optimal number of clusters from K-means.

Table 8. First approach: removal of top principal components. The table shows percentages of segment distribution across different damage classes for each iteration when removing k number of principal components, along with metrics that better visualise the separation of segments, such as the average standardised variance.

Removing Top Pcs and Evaluating Clusters (K-Means)												
First Approach: Removing from Full PCs 512												
Iteration	Cluster	BPFI	Normal	Misalign	Unbalance	Max	Tot segments	std	Inv std	Variance	Standardised Variance	Average
k = 0, variance: 1	0	14.29	0	85.71	0	85.71	2100	35.53	64.47	1262.60	67.34	73.77
	1	0	16.67	0	83.33	83.33	3600	34.36	65.64	1180.44	62.96	
	2	14.29	0	85.71	0	85.71	2100	35.53	64.47	1262.60	67.34	
	3	0	100	0	0	100	1500	43.30	56.70	1875.00	100.00	
	4	100	0	0	0	100	1799	43.30	56.70	1875.00	100.00	
	5	0	0	0	100	100	3000	43.30	56.70	1875.00	100.00	
	6	5.28	10.52	31.57	52.62	52.62	5701	18.74	81.26	351.14	18.73	
k = 1, variance: 0.89	0	49.74	0.55	49.71	0	49.74	3619	24.73	75.27	611.36	32.61	37.00
	1	0	11.01	20.55	68.45	68.45	8766	26.12	73.88	682.11	36.38	
	2	0	36.37	0	63.63	63.63	4715	26.79	73.21	717.89	38.29	
	3	33.33	0	66.67	0	66.67	2700	27.64	72.36	763.94	40.74	
k = 2, variance: 0.79	0	33.32	0	66.64	0.04	66.64	2701	27.62	72.38	762.78	40.68	31.05
	1	0	33.33	0	66.67	66.67	4503	27.64	72.36	763.94	40.74	
	2	14.29	9.52	28.58	47.61	47.61	12596	14.82	85.18	219.59	11.71	
k = 3, variance: 0.73	0	0	100	0	0	100	752	43.30	56.70	1875.00	100.00	52.11
	1	13.91	9.87	9.15	67.07	67.07	6383	24.36	75.64	593.25	31.64	
	2	13.12	8.22	34.49	44.18	44.18	6860	14.84	85.16	220.16	11.74	
	3	0	100	0	0	100	748	43.30	56.70	1875.00	100.00	
	4	18.03	0.12	48.47	33.38	48.47	5057	17.95	82.05	322.17	17.18	
k = 4, variance: 0.67	0	19.08	26.02	8.37	46.53	46.53	7255	13.93	86.07	194.05	10.35	12.65
	1	10.49	6.47	38.21	44.83	44.83	12545	16.75	83.25	280.41	14.96	
k = 5, variance: 0.65	0	7.93	15.8	30.93	45.34	45.34	9440	14.36	85.64	206.23	11.00	9.76
	1	18.83	11.67	23.94	45.56	45.56	10360	12.65	87.35	159.90	8.53	

Table 9. Second approach: removal of top principal components. The table shows percentages of segment distribution across different damage classes for each iteration when removing k number of principal components, along with metrics that better visualise the separation of segments, such as the average standardised variance.

Removing Top Pcs and Evaluating These Clusters (K-Means)												
Second Approach: Removing After Reducing to 80% Variance												
Iteration	Cluster	BPFI	Normal	Misalign	Unbalance	Max	Tot segments	std	Inv std	Variance	Standardised Variance	Average
k = 0, variance: 0.80	0	0	16.67	0	83.33	83.33	3600	34.36	65.64	1180.44	62.96	59.36
	1	33.33	0	66.67	0	66.67	2700	27.64	72.36	763.94	40.74	
	2	0	100	0	0	100	1500	43.30	56.70	1875.00	100.00	
	3	14.29	9.52	28.57	47.62	47.62	6300	14.82	85.18	219.69	11.72	
	4	33.33	0	66.67	0	66.67	2700	27.64	72.36	763.94	40.74	
	5	0	0	0	100	100	3000	43.30	56.70	1875.00	100.00	
k = 1, variance: 0.69	0	0	36.4	0	63.6	63.6	4717	26.79	73.21	717.48	38.27	37.00
	1	0	10.98	20.55	68.47	68.47	8763	26.13	73.87	682.75	36.41	
	2	33.33	0	66.67	0	66.67	2700	27.64	72.36	763.94	40.74	
	3	49.72	0.58	49.69	0	49.72	3620	24.71	75.29	610.50	32.56	
Kk= 2, variance: 0.59	0	33.32	0	66.64	0.04	66.64	2701	27.62	72.38	762.78	40.68	31.05
	1	14.29	9.52	28.58	47.61	47.61	12596	14.82	85.18	219.59	11.71	
	2	0	33.33	0	66.67	66.67	4503	27.64	72.36	763.94	40.74	
k = 3, variance: 0.52	0	0	100	0	0	100	784	43.30	56.70	1875.00	100.00	52.10
	1	18.01	0.12	48.45	33.41	48.45	5046	17.95	82.05	322.13	17.18	
	2	13.12	8.22	34.51	44.15	44.15	6862	14.83	85.17	219.97	11.73	
	3	0	100	0	0	100	716	43.30	56.70	1875.00	100.00	
	4	13.94	9.86	9.18	67.02	67.02	6392	24.33	75.67	591.87	31.57	
k = 4, variance: 0.47	0	19.09	26.02	8.37	46.52	46.52	7251	13.93	86.07	193.91	10.34	12.65
	1	10.49	6.48	38.19	44.84	44.84	12549	16.74	83.26	280.28	14.95	
k = 5, variance: 0.45	0	7.98	15.65	31.13	45.25	45.25	9415	14.36	85.64	206.19	11.00	9.77
	1	18.77	11.82	23.77	45.64	45.64	10385	12.65	87.35	160.01	8.53	

Table 10. Third approach: removal of top single principal component. The table shows percentages of segment distribution across different damage classes for each iteration when removing k number of principal components, along with metrics that better visualise the separation of segments, such as the average standardised variance.

Removing Single Pcs and Evaluating These Clusters (K-Means)												
Third Approach: Removing from Full PCs 512												
Iteration	Cluster	BPFI	Normal	Misalign	Unbalance	Max	Tot segments	std	Inv std	Variance	Standardised Variance	Average
PC0, variance: 1	0	14.29	0	85.71	0	85.71	2100	35.533	64.467	1262.60	67.34	73.77
	1	0	16.67	0	83.33	83.33	3600	34.358	65.642	1180.44	62.96	
	2	14.29	0	85.71	0	85.71	2100	35.533	64.467	1262.60	67.34	
	3	0	100	0	0	100	1500	43.301	56.699	1875.00	100.00	
	4	100	0	0	0	100	1799	43.301	56.699	1875.00	100.00	
	5	0	0	0	100	100	3000	43.301	56.699	1875.00	100.00	
	6	5.28	10.52	31.57	52.62	52.62	5701	18.739	81.261	351.14	18.73	
PC1, variance: 0.9	0	17.99	0.00	45.43	36.58	45.43	3324	17.504	82.496	306.41	16.34	25.87
	1	0.13	25.06	8.21	66.60	66.6	7183	25.647	74.353	657.75	35.08	
	2	49.87	50.13	0.00	0.00	50.13	3591	25.000	75.000	625.01	33.33	
	3	5.29	31.57	10.52	52.61	52.61	5702	18.733	81.267	350.91	18.72	
PC2, variance: 0.92	0	2.66	17.72	20.64	59	59	10174	20.779	79.221	431.77	23.03	15.14
	1	25.23	37.37	6.23	31.17	37.37	9626	11.656	88.344	135.86	7.25	
PC3, variance: 0.94	0	27.03	36.49	6.08	30.4	36.49	9867	11.438	88.562	130.82	6.98	16.32
	1	0.33	18.12	21.14	60.41	60.41	9933	21.936	78.064	481.18	25.66	
PC4, variance: 0.96	0	99.75	0	0.17	0.08	99.75	2367	43.157	56.843	1862.52	99.33	39.27
	1	0.44	28.45	23.71	47.39	47.39	6326	16.719	83.281	279.52	14.91	
	2	5.28	31.6	10.46	52.66	52.66	5697	18.768	81.232	352.23	18.79	
	3	0.18	33.27	11.09	55.45	55.45	5410	21.243	78.757	451.28	24.07	
PC5, variance: 0.97	0	99.87	0	0.13	0	99.87	2349	43.226	56.774	1868.51	99.65	39.38
	1	0.06	33.05	11.02	55.08	55.08	5447	21.136	78.864	446.72	23.83	
	2	0	28.57	23.81	47.62	47.62	6300	16.962	83.038	287.71	15.34	
	3	5.3	31.56	10.47	52.59	52.59	5704	18.731	81.269	350.86	18.71	

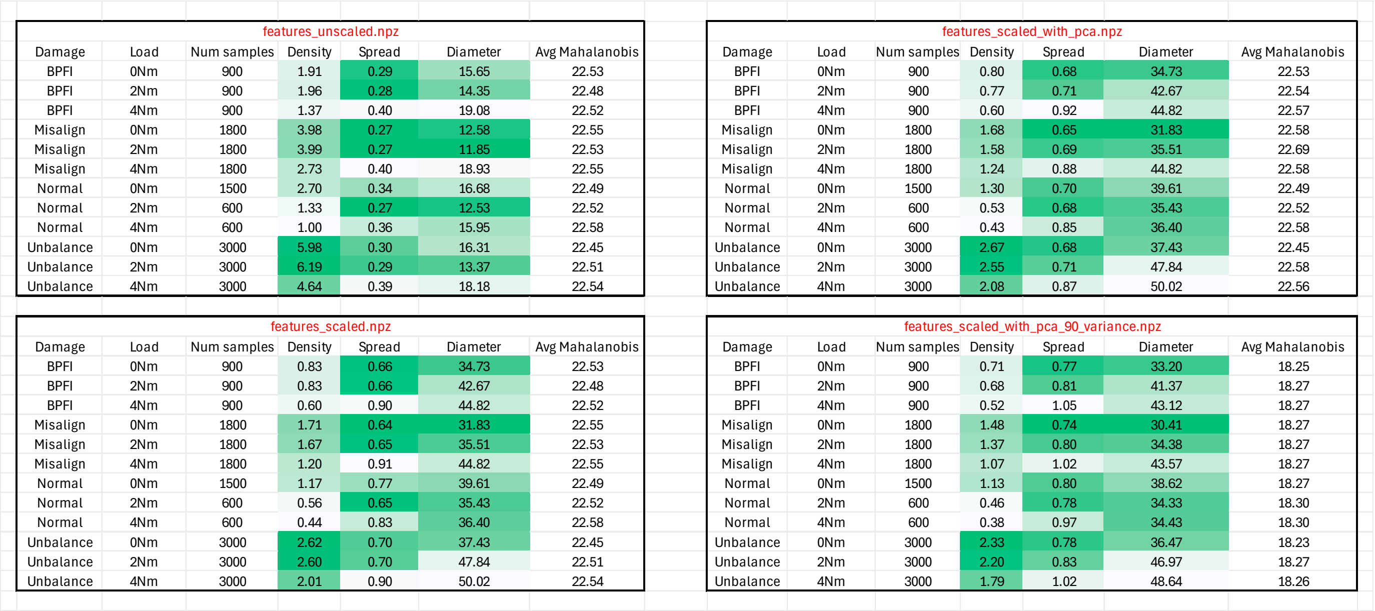
Table 11. Fourth approach: removal of top single principal component. The table shows percentages of segment distribution across different damage classes for each iteration when removing k number of principal components, along with metrics that better visualise the separation of segments, such as the average standardised variance.

Removing Single Pcs and Evaluating These Clusters (K-Means)												
Fourth Approach: Removing from Reduced 80% Variance												
Iteration	Cluster	BPFI	Normal	Misalign	Unbalance	Max	Tot segments	std	Inv std	Variance	Standardised Variance	Average
k = 0, variance: 1	0	0	16.67	0	83.33	83.33	3600	34.358	65.642	1180.44	62.96	59.36
	1	33.33	0	66.67	0	66.67	2700	27.640	72.360	763.94	40.74	
	2	0	100	0	0	100	1500	43.301	56.699	1875.00	100.00	
	3	14.29	9.52	28.57	47.62	47.62	6300	14.822	85.178	219.69	11.72	
	4	33.33	0	66.67	0	66.67	2700	27.640	72.360	763.94	40.74	
	5	0	0	0	100	100	3000	43.301	56.699	1875.00	100.00	
PC1, variance: 0.71	0	17.99	0	45.43	36.58	1510	3324	17.504	82.49556	306.41	16.34	25.87
	1	0.13	25.06	8.21	66.6	4784	7183	25.647	74.35344	657.75	35.08	
	2	49.87	50.13	0	0	1800	3591	25.000	74.99983	625.01	33.33	
	3	5.3	31.57	10.52	52.61	3000	5702	18.730	81.27009	350.81	18.71	
PC2, variance: 0.72	0	25.26	37.36	6.23	31.15	3597	9629	11.651	88.34911	135.74	7.24	15.14
	1	2.64	17.73	20.65	59	6000	10,171	20.783	79.21692	431.94	23.04	
PC3, variance: 0.74	0	7.26	92.74	0	0	1800	1941	39.222	60.77815	1538.35	82.05	68.32
	1	0	0	16.67	83.33	3000	3600	34.358	65.6424	1180.44	62.96	
	2	14.07	85.88	0	0.05	1800	2096	35.614	64.3863	1268.34	67.64	
	3	0	0	0	100	3000	3000	43.301	56.69873	1875.00	100.00	
	4	7.98	32.32	3.05	56.65	1469	2593	21.370	78.62953	456.70	24.36	
	5	0	0	100	0	711	711	43.301	56.69873	1875.00	100.00	
	6	9.17	33.04	2.92	54.86	1501	2736	20.581	79.4186	423.59	22.59	
	7	54.11	5.1	39.45	1.34	605	1118	22.428	77.57234	503.00	26.83	
	8	0	0	100	0	789	789	43.301	56.69873	1875.00	100.00	
	9	98.77	0.08	0	1.15	1201	1216	42.594	57.40645	1814.21	96.76	
PC4, variance: 0.77	0	0.2	33.27	11.09	55.44	3000	5411	21.234	78.7661	450.88	24.05	39.26
	1	0.46	28.45	23.71	47.38	2998	6327	16.708	83.2919	279.16	14.89	
	2	5.28	31.6	10.46	52.66	3000	5697	18.768	81.23217	352.23	18.79	
	3	99.75	0	0.17	0.08	2359	2365	43.157	56.84303	1862.52	99.33	
PC5, variance: 0.78	0	0	28.57	23.81	47.62	3000	6300	16.962	83.03809	287.71	15.34	39.24
	1	0.9	33.03	11.01	55.06	3000	5449	20.884	79.11571	436.15	23.26	
	2	5.37	31.56	10.47	52.6	3000	5703	18.717	81.28335	350.31	18.68	
	3	99.87	0	0.13	0	2345	2348	43.226	56.77375	1868.51	99.65	

4.4. Experiment D: Hierarchical Clustering

To gain a deeper understanding of the learned feature space, hierarchical clustering is applied in addition to PCA and K-means. Four preprocessing pipelines are first evaluated to identify a suitable feature representation: features unscaled, features scaled, features scaled with PCA, and features scaled with PCA retaining 90% variance. The ground-truth calculations of the 12 classes (damage + load) are presented in Figure 10. The table includes shades of green colouring, where lighter green indicates lower score and darker green indicates higher scores.

The ground-truth analysis revealed that the features scaled with the PCA pipeline provided the most stable and interpretable baseline with distinctions in metrics like "spread" and "diameter". This preprocessing configuration is therefore selected for the full hierarchical clustering analysis. An example dendrogram for KAIST is shown in Figure A2 in Appendix A, which plots the structure of the hierarchical tree. The very first split at the top of the tree separates the data into two main branches: one containing all the 4 Nm data and the other containing all the 0 Nm and 2 Nm data. This shows that the algorithm perceives 0 Nm and 2 Nm as being more similar to each other and different from 4 Nm. Within the 0/2 Nm branch, the subclusters corresponding to different damage types are relatively clear and distinct. In contrast, the 4 Nm branch is shown as more complex, and the damage classes are not distinct, apart from the case of BPFI.



features_unscaled.npz

Damage	Load	Num samples	Density	Spread	Diameter	Avg Mahalanobis
BPFI	0Nm	900	1.91	0.29	15.65	22.53
BPFI	2Nm	900	1.96	0.28	14.35	22.48
BPFI	4Nm	900	1.37	0.40	19.08	22.52
Misalign	0Nm	1800	3.98	0.27	12.58	22.55
Misalign	2Nm	1800	3.99	0.27	11.85	22.53
Misalign	4Nm	1800	2.73	0.40	18.93	22.55
Normal	0Nm	1500	2.70	0.34	16.68	22.49
Normal	2Nm	600	1.33	0.27	12.53	22.52
Normal	4Nm	600	1.00	0.36	15.95	22.58
Unbalance	0Nm	3000	5.98	0.30	16.31	22.45
Unbalance	2Nm	3000	6.19	0.29	13.37	22.51
Unbalance	4Nm	3000	4.64	0.39	18.18	22.54

features_scaled_with_pca.npz

Damage	Load	Num samples	Density	Spread	Diameter	Avg Mahalanobis
BPFI	0Nm	900	0.80	0.68	34.73	22.53
BPFI	2Nm	900	0.77	0.71	42.67	22.54
BPFI	4Nm	900	0.60	0.92	44.82	22.57
Misalign	0Nm	1800	1.68	0.65	31.83	22.58
Misalign	2Nm	1800	1.58	0.69	35.51	22.69
Misalign	4Nm	1800	1.24	0.88	44.82	22.58
Normal	0Nm	1500	1.30	0.70	39.61	22.49
Normal	2Nm	600	0.53	0.68	35.43	22.52
Normal	4Nm	600	0.43	0.85	36.40	22.58
Unbalance	0Nm	3000	2.67	0.68	37.43	22.45
Unbalance	2Nm	3000	2.55	0.71	47.84	22.58
Unbalance	4Nm	3000	2.08	0.87	50.02	22.56

features_scaled.npz

Damage	Load	Num samples	Density	Spread	Diameter	Avg Mahalanobis
BPFI	0Nm	900	0.83	0.66	34.73	22.53
BPFI	2Nm	900	0.83	0.66	42.67	22.48
BPFI	4Nm	900	0.60	0.90	44.82	22.52
Misalign	0Nm	1800	1.71	0.64	31.83	22.55
Misalign	2Nm	1800	1.67	0.65	35.51	22.53
Misalign	4Nm	1800	1.20	0.91	44.82	22.55
Normal	0Nm	1500	1.17	0.77	39.61	22.49
Normal	2Nm	600	0.56	0.65	35.43	22.52
Normal	4Nm	600	0.44	0.83	36.40	22.58
Unbalance	0Nm	3000	2.62	0.70	37.43	22.45
Unbalance	2Nm	3000	2.60	0.70	47.84	22.51
Unbalance	4Nm	3000	2.01	0.90	50.02	22.54

features_scaled_with_pca_90_variance.npz

Damage	Load	Num samples	Density	Spread	Diameter	Avg Mahalanobis
BPFI	0Nm	900	0.71	0.77	33.20	18.25
BPFI	2Nm	900	0.68	0.81	41.37	18.27
BPFI	4Nm	900	0.52	1.05	43.12	18.27
Misalign	0Nm	1800	1.48	0.74	30.41	18.27
Misalign	2Nm	1800	1.37	0.80	34.38	18.27
Misalign	4Nm	1800	1.07	1.02	43.57	18.27
Normal	0Nm	1500	1.13	0.80	38.62	18.27
Normal	2Nm	600	0.46	0.78	34.33	18.30
Normal	4Nm	600	0.38	0.97	34.43	18.30
Unbalance	0Nm	3000	2.33	0.78	36.47	18.23
Unbalance	2Nm	3000	2.20	0.83	46.97	18.27
Unbalance	4Nm	3000	1.79	1.02	48.64	18.26

Figure 10. Ground-truth metric calculations for each class.

The numerical results in Figure A1 in Appendix A provide a quantitative view of the clustering behaviour. At level 1, the tree split coincides with the 4 Nm load separation, confirming that load is the dominant organising factor in the feature space. External metrics based on ground-truth labels, such as the proposed Dominance Score and V-measure, increase with depth and reach higher values by level 4–5, indicating that many clusters at deeper levels achieve high purity with respect to damage and load labels.

Figures 11 and 12 show boxplots of split quality and cluster quality metrics across hierarchical levels for both the KAIST and PU datasets. For the Dominance Score, the median and interquartile range generally improve from level 0 to level 5, confirming that cluster purity increases as the tree deepens. Internal geometric metrics such as spread and diameter show a consistent downward trend with increasing depth, reflecting tighter and more compact clusters. The Silhouette Score for KAIST improves up to level 2 and then decreases, while the Davies–Bouldin Index shows a corresponding increase, indicating that additional splits beyond level 2 sometimes over-partition already-pure clusters and degrade the average split quality.

Figures 13 and 14 summarise damage and load composition across hierarchical levels. At level 0–1, clusters are highly mixed in terms of damage types, whereas from level 2–5, the bars become progressively purer, showing that damage-specific clusters are increasingly isolated for both datasets. For KAIST, load composition plots show that load separation is largely achieved by level 2, whereas for PU, deeper levels are needed before most clusters become load-specific. At level 5, PU still retains clusters with mixed loads, suggesting that load-related features are even stronger in PU than in KAIST, consistent with the closer torque values of 0.1 Nm and 0.7 Nm compared to 0, 2 and 4 Nm in KAIST.

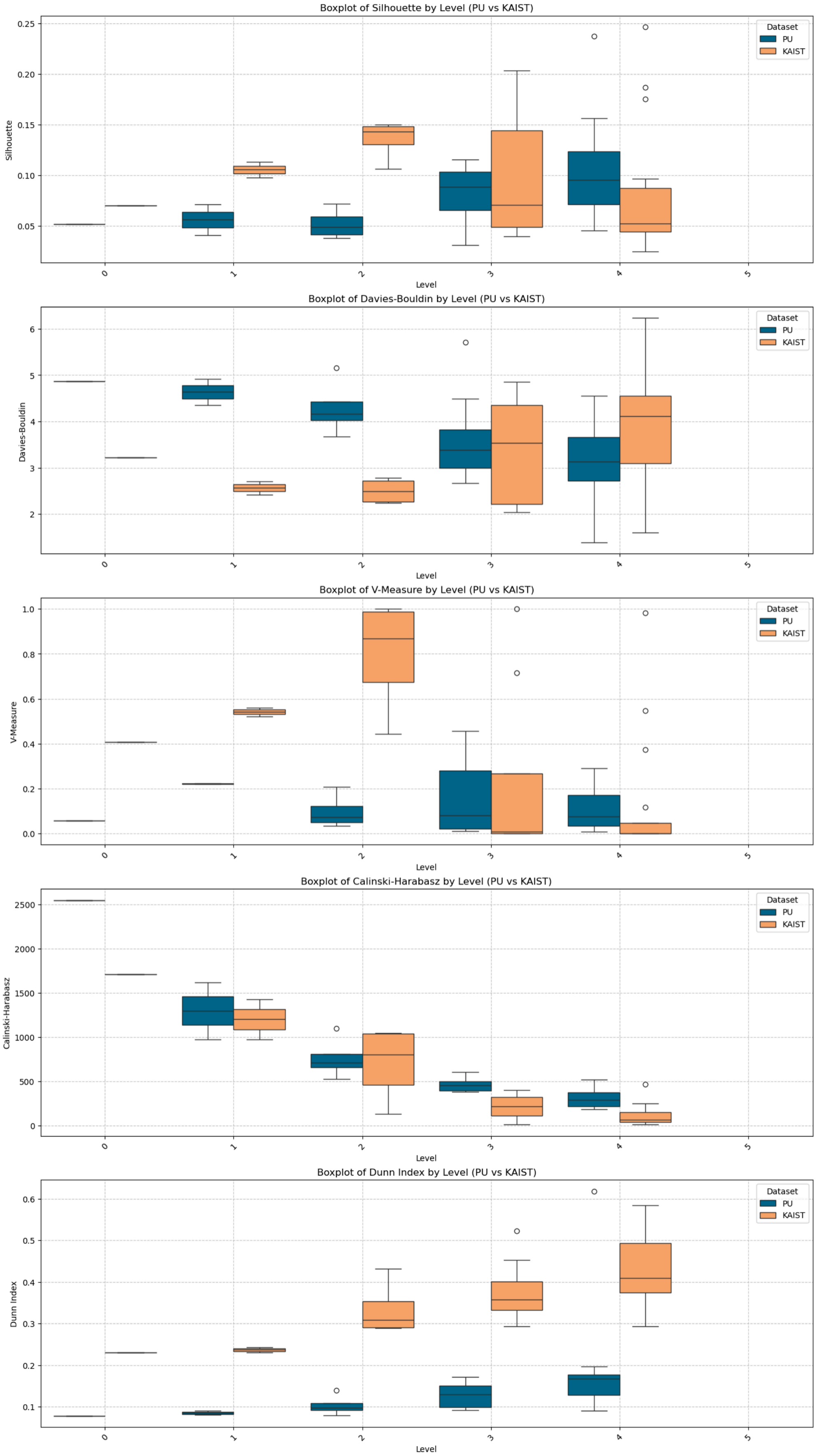


Figure 11. Hierarchical clustering boxplot graph for split quality metrics represented for KAIST and PU datasets, which shows the trend of the metrics as the level of clustering gets deeper.

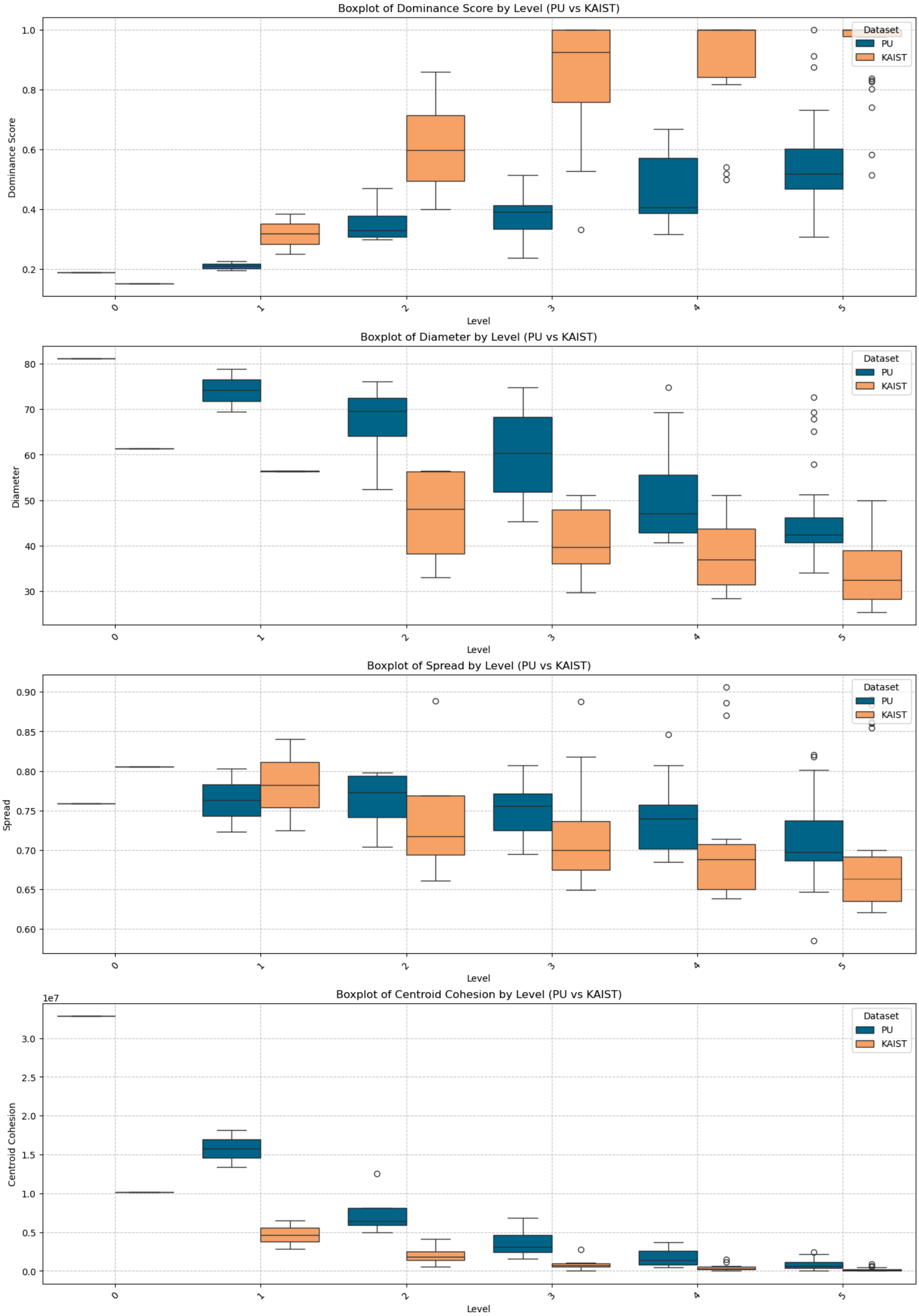


Figure 12. Hierarchical clustering boxplot graph for cluster (cohesion) quality metrics for KAIST and PU datasets, which show the trend of the metrics as the level of clustering gets deeper.

https://doi.org/10.3390/app16041780

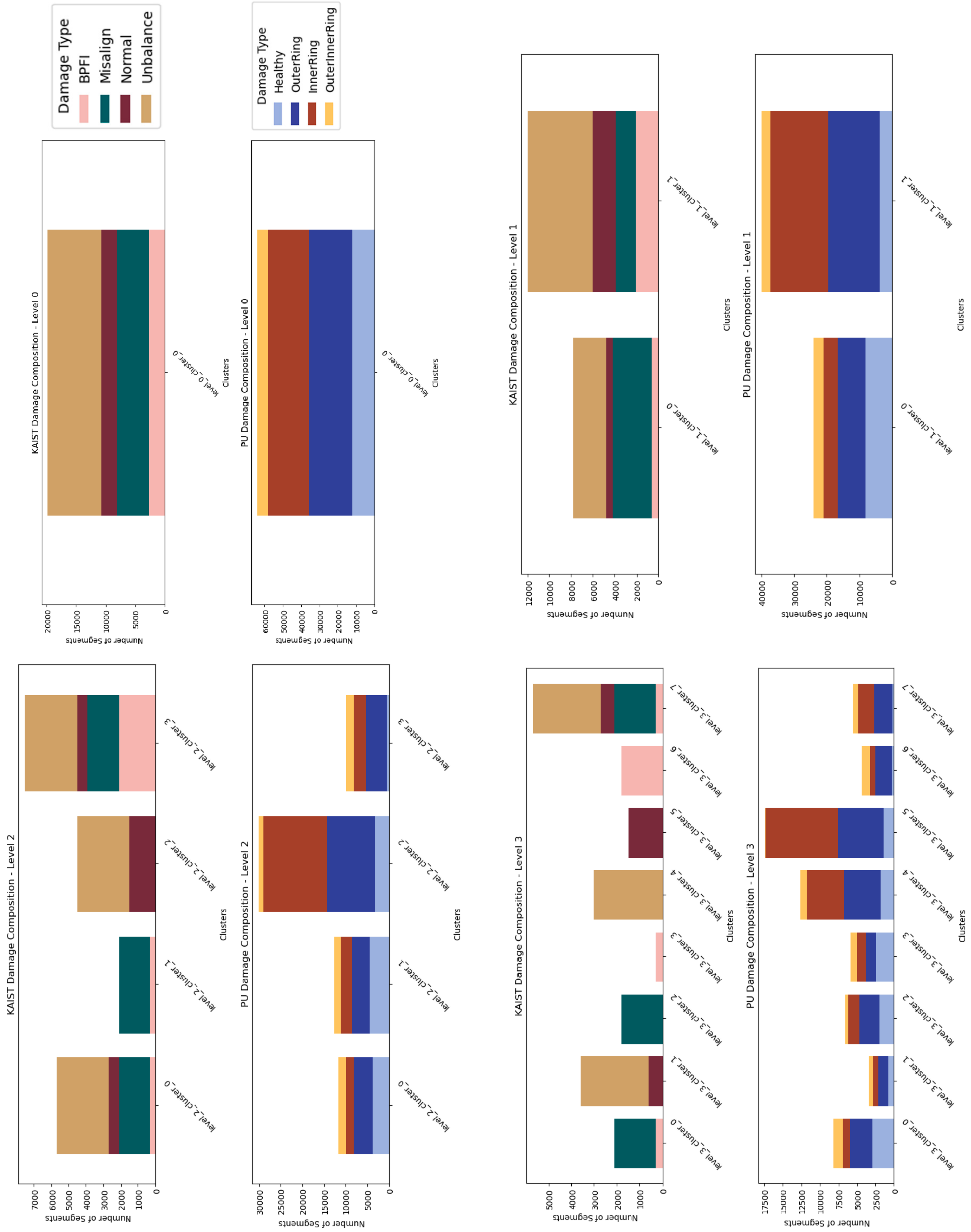


Figure 13. *Cont.*

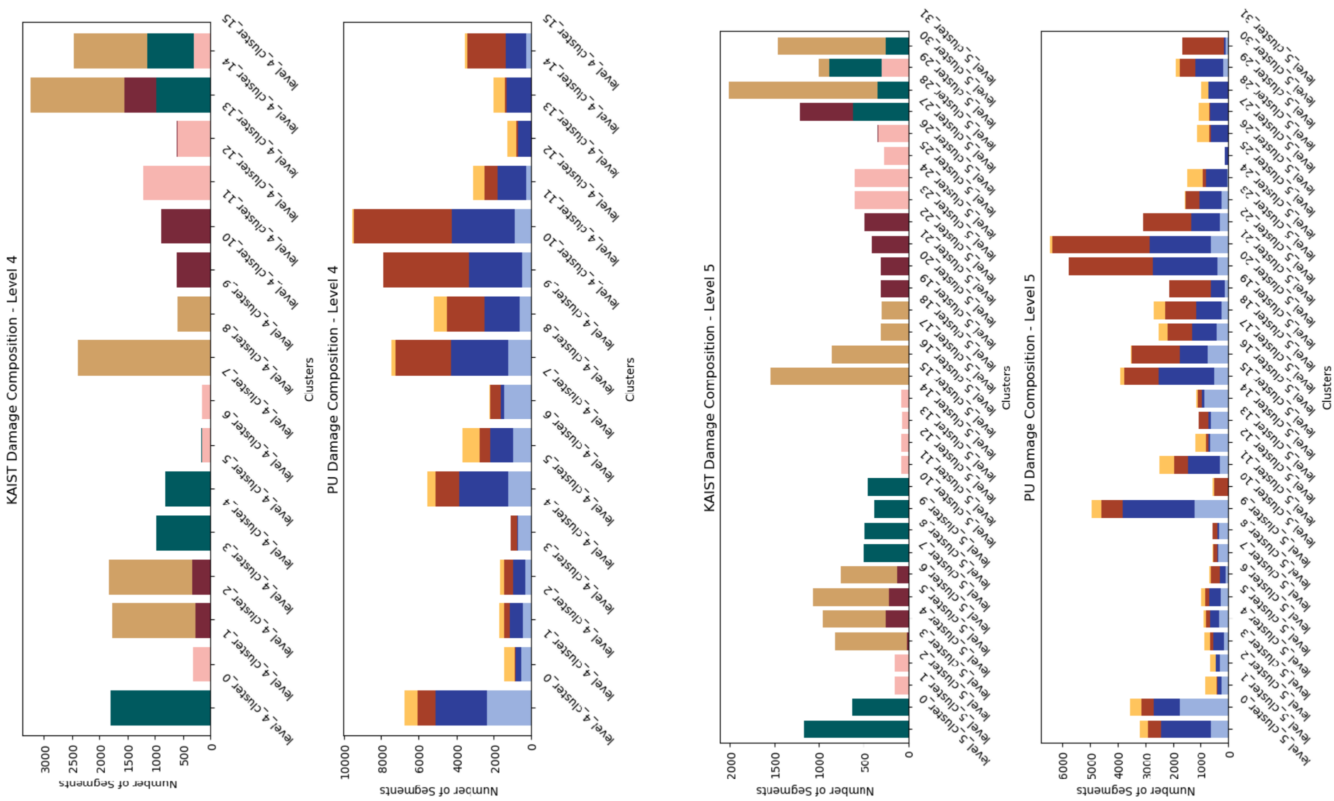


Figure 13. Hierarchical clustering of damage composition for KAIST and PU datasets.

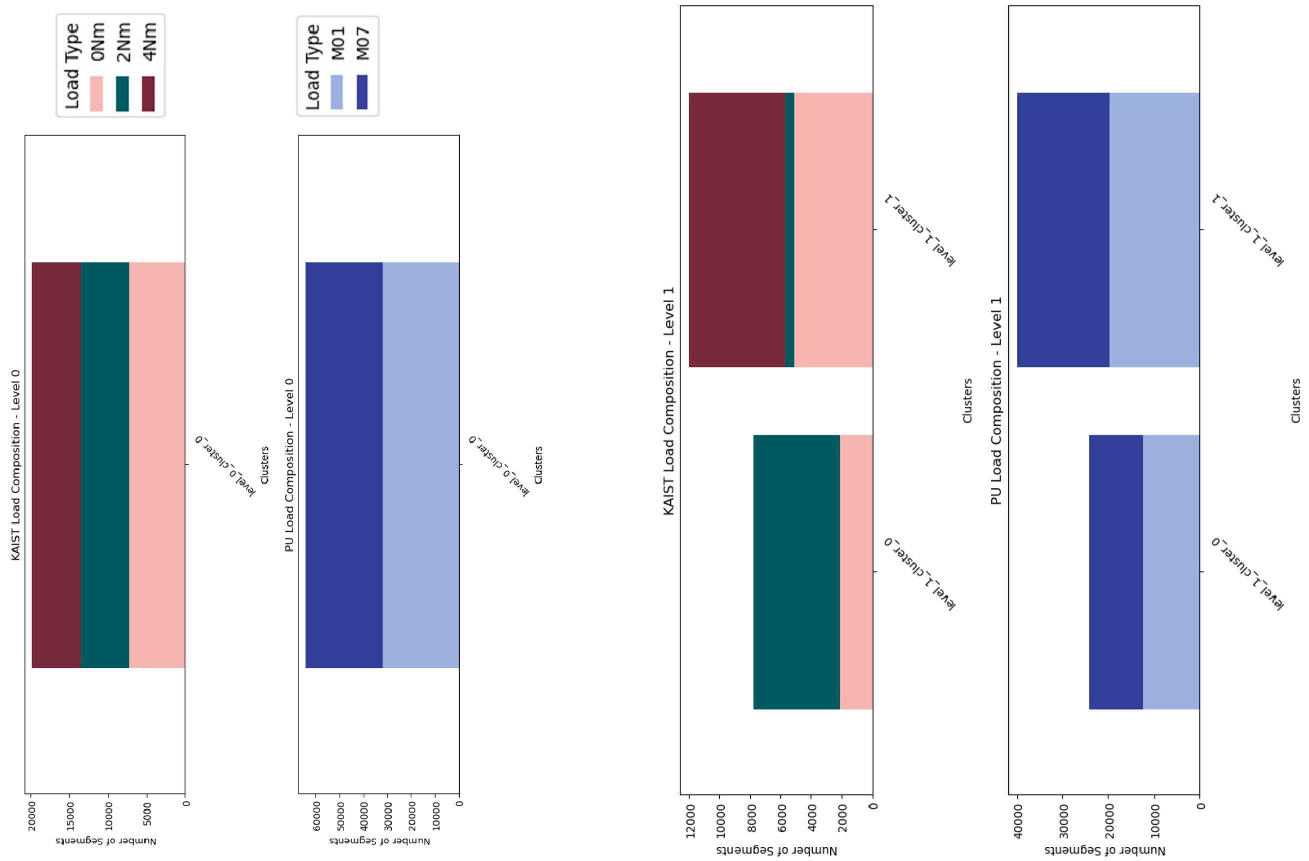


Figure 14. *Cont.*

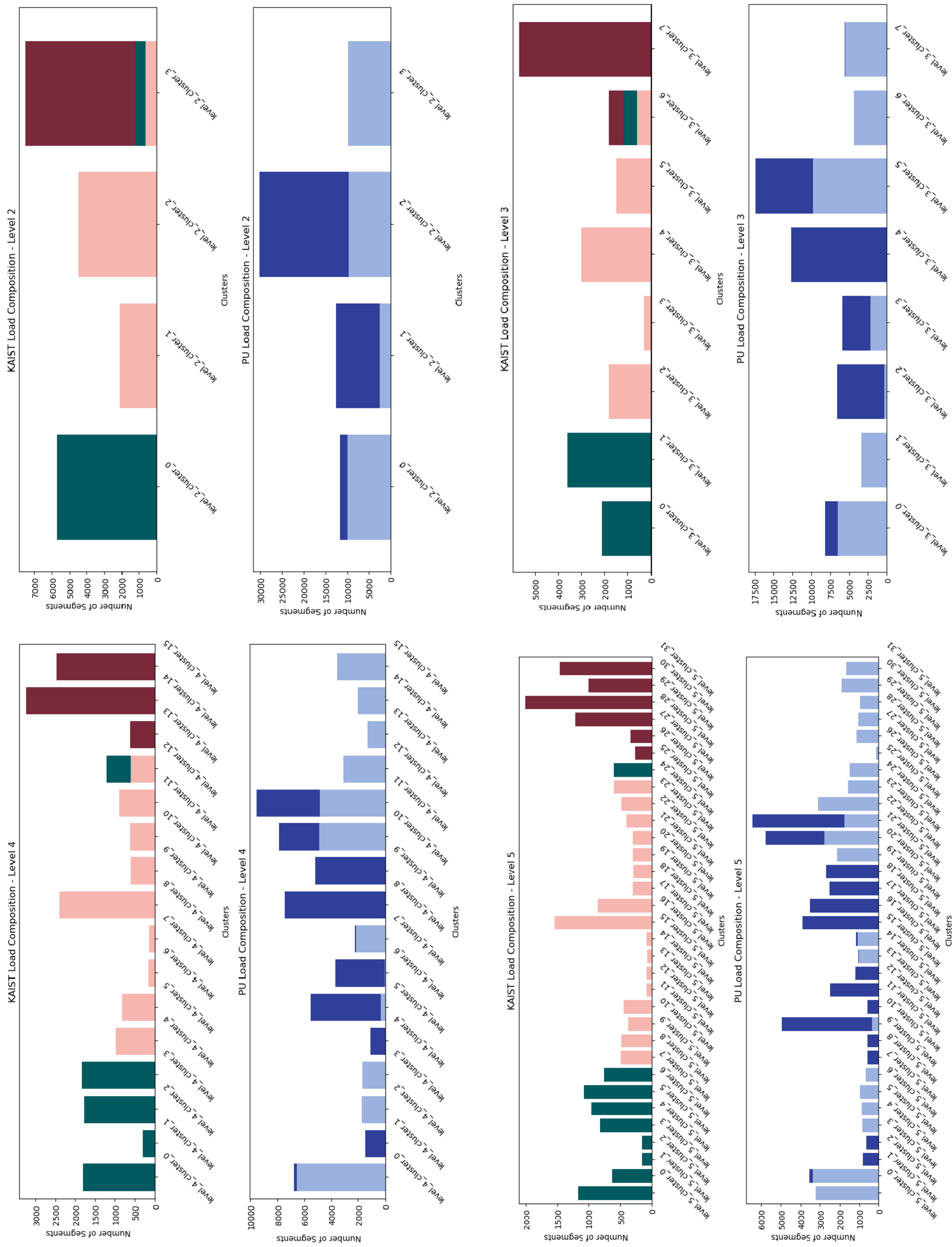


Figure 14. Hierarchical clustering of load composition for KAIST and PU datasets.

5. Discussion

The experimental results collectively characterise the shadowing effect of operational variability on automated fault diagnosis, linking baseline performance degradation, the limitations of linear mitigation strategies, and the structural insights obtained from hierarchical clustering. Across the KAIST and PU datasets, the torque load emerges as a dominant contextual feature that reshapes the learned feature space and reduces the separability of fault classes.

In the supervised baseline (Experiment A), classification performance on KAIST remains high at a single load but degrades as torque increases and when data from multiple loads are combined. This indicates that, as the load increases, the decision boundaries learned by the 1D-CNN become increasingly misaligned with the underlying fault classes and instead align with the load-driven variability. The PU results show the same behaviour; therefore, performance under a single load condition is consistently higher than when data from different loads are combined, showing that the interference of contextual features is not specific to a single dataset or machine type and motivating a deeper unsupervised feature space analysis.

The feature space visualisation (Experiment B) provides qualitative evidence of this shadowing effect. For KAIST, the progression from clear, class-wise separation at 0 Nm to extensive overlap at 4 Nm shows that at higher loads, points from different fault classes cluster first by load and then by damage type. This behaviour indicates that load-driven features reshape the representation learned by the 1D-CNN and progressively reduces the distinctiveness of fault-specific features. In PU, the greatest overall overlap between classes further illustrates how bearing fault features can be masked when contextual variation is strong relative to fault signatures.

These observations motivate the move from visual inspection to a quantitative analysis using unsupervised clustering (Experiment C). By applying K-means without labels and then examining the cluster composition with ground-truth fault and load labels, cluster purity can be measured, and the influence of contextual features on the feature space can be assessed. The principal component removal experiment provides a critical negative result: removing high-variance principal components does not improve the separation of fault classes and sometimes worsens it. The dominant contextual influence of torque load is not found in the first few PCs; instead, load-related features and fault-related features are mixed across different components. Therefore, the top PCs, despite having higher variance, also carry essential discriminative information needed for separation of fault classes, and simple linear component removal strategies are ineffective for isolating contextual features such as the torque load.

The hierarchical clustering analysis (Experiment D) further clarifies how contextual features and fault-related features are organised. For KAIST, the first split of the hierarchical tree separates 4 Nm and 0/2 Nm, confirming that the torque load is the dominant organising factor in the feature space, while the increasing Dominance Score and V-measure with depth show that fault class information, although initially masked, remains present and can be recovered in deeper clusters. Internal geometric metrics such as spread and diameter follow trends that mirror the behaviour of the Dominance Score, suggesting that they can serve as unsupervised proxies for cluster purity when labels are not available. Among external metrics, the Dominance Score and V-measure provide the clearest indication that splits are meaningful with respect to damage and load labels, whereas internal metrics are more relevant for practical unsupervised settings. Therefore, diameter and spread are the most informative for internal cluster quality, while Silhouette is useful for split quality but can deteriorate once pure clusters are over-partitioned. Low Silhouette and high

Davies–Bouldin values at some levels indicate that many fault groups remain geometrically close and overlapping, which aligns with the previously observed shadowing effect.

The behaviour of Silhouette and V-measure for KAIST, improving up to level 2 and then declining, highlights the need for principled stopping criteria in hierarchical clustering. Once clusters have become relatively pure, further splitting tends to partition single damage classes into multiple subclusters, reducing the average split quality scores without improving class separation. This behaviour, together with the widening of boxplots at deeper levels, reflects the difficulty of separating load and damage features without explicit guidance. The analysis suggests that internal metrics derived from hierarchical clustering, such as spread, could support mitigation strategies for contextual feature dominance by defining purer and load-aware clusters before training classifiers. These clusters could then be used as a refined training set or pseudo-labels to improve load-invariant damage classification, and future work could explore alternative feature engineering strategies and clustering algorithms to further enhance separation.

6. Conclusions

This research quantitatively characterised the shadowing effect of contextual features on data-driven mechanical fault diagnosis. We demonstrated that varying operating conditions, such as the torque load, are not just background noise but act as dominant structural features that distort the learned feature space from a 1D-CNN. While the standard classifier achieved high accuracy under static conditions, its performance degraded significantly when subject to varying loads, caused by overlapping feature projections at higher torques.

Our analysis confirmed that linear mitigation strategies, such as removing high-variance principal components, are ineffective because fault features and contextual load features are deeply entangled. Instead, we validated the use of hierarchical clustering as a robust analytical method. By applying this technique to the KAIST and PU motor datasets, we demonstrated that the feature space is organised hierarchically, where the model prioritises load separation before fault classification. Furthermore, we identified the internal cluster metrics, such as spread and diameter, which serve as unsupervised proxies for cluster purity, aligning with ground-truth external metrics like the Dominance Score.

One of the limitations of this study is that the analysis was done in an offline setting, where the computational costs may present challenges for real-time edge deployment without further optimisation. Furthermore, while the method was validated on two distinct datasets, the study focused on induction motors and PMSMs and mechanical damage. The generalisability to other rotating machinery types remains to be tested.

Building on the unsupervised metrics identified in this work, future research will focus on developing an intelligent preprocessing pipeline. We aim to utilise the identified internal metrics as automatic stopping criteria for hierarchical clustering. This would enable the extraction of load-invariant, pure clusters from raw operational data, facilitating the training of robust fault diagnosis models without utilising the labelling of operating conditions.

Author Contributions: Conceptualisation, U.E.J. and T.K.; methodology, U.E.J.; software, U.E.J.; validation, U.E.J.; formal analysis, U.E.J. and T.K.; investigation, U.E.J.; resources, U.E.J.; data curation, U.E.J.; writing—original draft preparation, U.E.J.; writing—review and editing, U.E.J., T.K. and S.P.; visualisation, U.E.J.; supervision, T.K. and S.P.; project administration, T.K. and S.P.; funding acquisition, T.K. and S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported via the UKAEA Fusion Opportunities in Skills, Training, Education and Research (FOSTER) program.

Institutional Review Board Statement: Not applicable.

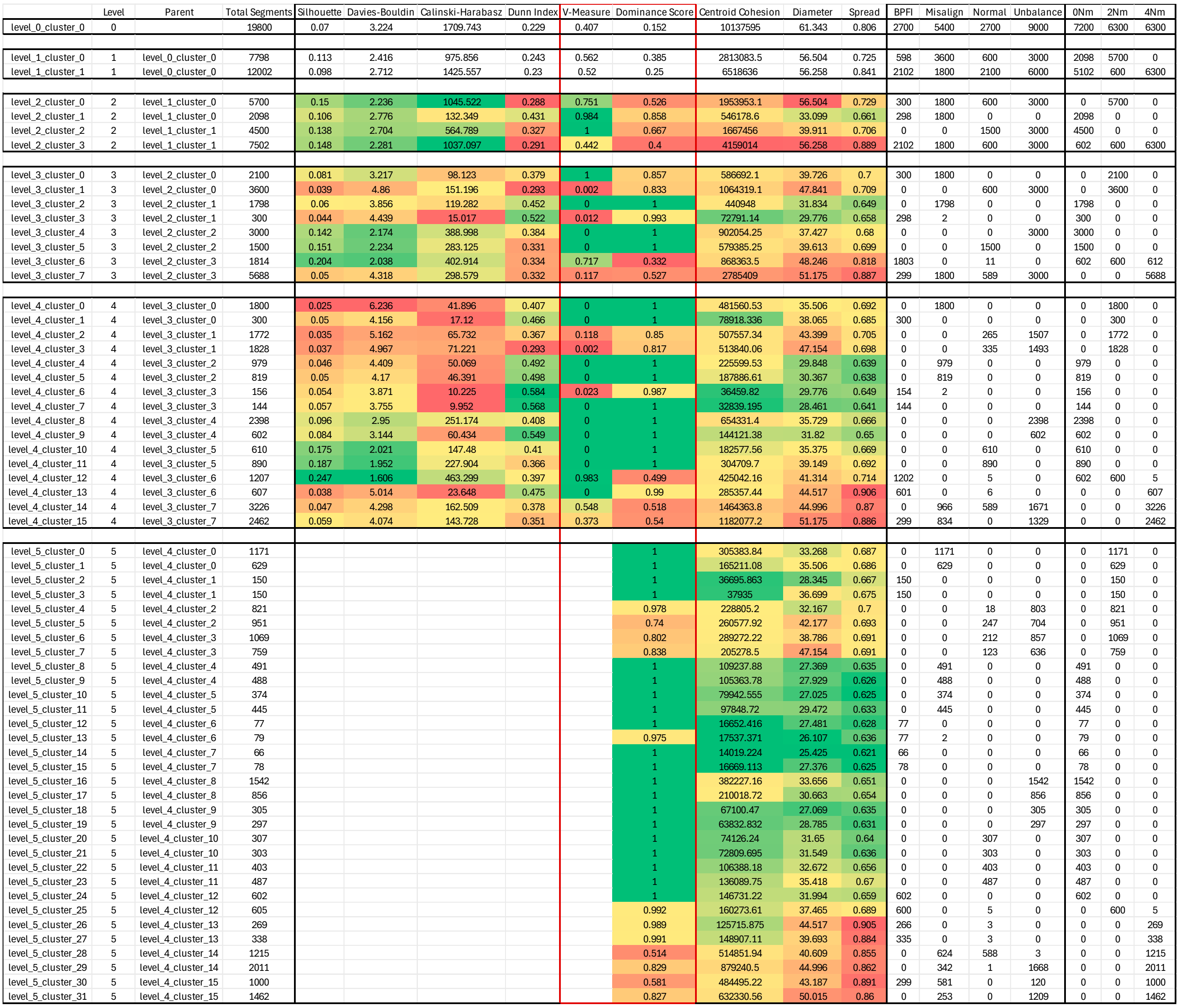
Informed Consent Statement: Not applicable.

Data Availability Statement: The open-source datasets used in this study may be found at the following links: KAIST motor dataset: https://data.mendeley.com/datasets/ztmf3m7h5x/6 (accessed on 28 October 2024). PU motor dataset: https://mb.uni-paderborn.de/en/kat/research/bearing-datacenter/data-sets-and-download (accessed on 9 February 2025).

Acknowledgments: During the preparation of this manuscript/study, the authors used Microsoft Copilot based on GPT-5.1 and Google Gemini 3 Pro for the purposes of proofreading. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A



	Level	Parent	Total Segments	Silhouette	Davies-Bouldin	Calinski-Harabasz	Dunn Index	V-Measure	Dominance Score	Centroid Cohesion	Diameter	Spread	BPFI	Misalign	Normal	Unbalance	0Nm	2Nm	4Nm
level_0_cluster_0	0		19800	0.07	3.224	1709.743	0.229	0.407	0.152	10137595	61.343	0.806	2700	5400	2700	9000	7200	6300	6300
level_1_cluster_0	1	level_0_cluster_0	7798	0.113	2.416	975.856	0.243	0.562	0.385	2813083.5	56.504	0.725	598	3600	600	3000	2098	5700	0
level_1_cluster_1	1	level_0_cluster_0	12002	0.098	2.712	1425.557	0.23	0.52	0.25	6518636	56.258	0.841	2102	1800	2100	6000	5102	600	6300
level_2_cluster_0	2	level_1_cluster_0	5700	0.15	2.236	1045.522	0.288	0.751	0.526	1953953.1	56.504	0.729	300	1800	600	3000	0	5700	0
level_2_cluster_1	2	level_1_cluster_0	2098	0.106	2.776	132.349	0.431	0.984	0.858	546178.6	33.099	0.661	298	1800	0	0	2098	0	0
level_2_cluster_2	2	level_1_cluster_1	4500	0.138	2.704	564.789	0.327	1	0.667	1667456	39.911	0.706	0	0	1500	3000	4500	0	0
level_2_cluster_3	2	level_1_cluster_1	7502	0.148	2.281	1037.097	0.291	0.442	0.4	4159014	56.258	0.889	2102	1800	600	3000	602	600	6300
level_3_cluster_0	3	level_2_cluster_0	2100	0.081	3.217	98.123	0.379	1	0.857	586692.1	39.726	0.7	300	1800	0	0	0	2100	0
level_3_cluster_1	3	level_2_cluster_0	3600	0.039	4.86	151.196	0.293	0.002	0.833	1064319.1	47.841	0.709	0	0	600	3000	0	3600	0
level_3_cluster_2	3	level_2_cluster_1	1798	0.06	3.856	119.282	0.452	0	1	440948	31.834	0.649	0	1798	0	0	1798	0	0
level_3_cluster_3	3	level_2_cluster_1	300	0.044	4.439	15.017	0.522	0.012	0.993	72791.14	29.776	0.658	298	2	0	0	300	0	0
level_3_cluster_4	3	level_2_cluster_2	3000	0.142	2.174	388.998	0.384	0	1	902054.25	37.427	0.68	0	0	0	3000	3000	0	0
level_3_cluster_5	3	level_2_cluster_2	1500	0.151	2.234	283.125	0.331	0	1	579385.25	39.613	0.699	0	0	1500	0	1500	0	0
level_3_cluster_6	3	level_2_cluster_3	1814	0.204	2.038	402.914	0.334	0.717	0.332	868363.5	48.246	0.818	1803	0	11	0	602	600	612
level_3_cluster_7	3	level_2_cluster_3	5688	0.05	4.318	298.579	0.332	0.117	0.527	2785409	51.175	0.887	299	1800	589	3000	0	0	5688
level_4_cluster_0	4	level_3_cluster_0	1800	0.025	6.236	41.896	0.407	0	1	481560.53	35.506	0.692	0	1800	0	0	0	1800	0
level_4_cluster_1	4	level_3_cluster_0	300	0.05	4.156	17.12	0.466	0	1	78918.336	38.065	0.685	300	0	0	0	0	300	0
level_4_cluster_2	4	level_3_cluster_1	1772	0.035	5.162	65.732	0.367	0.118	0.85	507557.34	43.399	0.705	0	0	265	1507	0	1772	0
level_4_cluster_3	4	level_3_cluster_1	1828	0.037	4.967	71.221	0.293	0.002	0.817	513840.06	47.154	0.698	0	0	335	1493	0	1828	0
level_4_cluster_4	4	level_3_cluster_2	979	0.046	4.409	50.069	0.492	0	1	225599.53	29.848	0.639	0	979	0	0	979	0	0
level_4_cluster_5	4	level_3_cluster_2	819	0.05	4.17	46.391	0.498	0	1	187886.61	30.367	0.638	0	819	0	0	819	0	0
level_4_cluster_6	4	level_3_cluster_3	156	0.054	3.871	10.225	0.584	0.023	0.987	36459.82	29.776	0.649	154	2	0	0	156	0	0
level_4_cluster_7	4	level_3_cluster_3	144	0.057	3.755	9.952	0.568	0	1	32839.195	28.461	0.641	144	0	0	0	144	0	0
level_4_cluster_8	4	level_3_cluster_4	2398	0.096	2.95	251.174	0.408	0	1	654331.4	35.729	0.666	0	0	0	2398	2398	0	0
level_4_cluster_9	4	level_3_cluster_4	602	0.084	3.144	60.434	0.549	0	1	144121.38	31.82	0.65	0	0	0	602	602	0	0
level_4_cluster_10	4	level_3_cluster_5	610	0.175	2.021	147.48	0.41	0	1	182577.56	35.375	0.669	0	0	610	0	610	0	0
level_4_cluster_11	4	level_3_cluster_5	890	0.187	1.952	227.904	0.366	0	1	304709.7	39.149	0.692	0	0	890	0	890	0	0
level_4_cluster_12	4	level_3_cluster_6	1207	0.247	1.606	463.299	0.397	0.983	0.499	425042.16	41.314	0.714	1202	0	5	0	602	600	5
level_4_cluster_13	4	level_3_cluster_6	607	0.038	5.014	23.648	0.475	0	0.99	285357.44	44.517	0.906	601	0	6	0	0	0	607
level_4_cluster_14	4	level_3_cluster_7	3226	0.047	4.298	162.509	0.378	0.548	0.518	1464363.8	44.996	0.87	0	966	589	1671	0	0	3226
level_4_cluster_15	4	level_3_cluster_7	2462	0.059	4.074	143.728	0.351	0.373	0.54	1182077.2	51.175	0.886	299	834	0	1329	0	0	2462
level_5_cluster_0	5	level_4_cluster_0	1171						1	305383.84	33.268	0.687	0	1171	0	0	0	1171	0
level_5_cluster_1	5	level_4_cluster_0	629						1	165211.08	35.506	0.686	0	629	0	0	0	629	0
level_5_cluster_2	5	level_4_cluster_1	150						1	36695.863	28.345	0.667	150	0	0	0	0	150	0
level_5_cluster_3	5	level_4_cluster_1	150						1	37935	36.699	0.675	150	0	0	0	0	150	0
level_5_cluster_4	5	level_4_cluster_2	821						0.978	228805.2	32.167	0.7	0	0	18	803	0	821	0
level_5_cluster_5	5	level_4_cluster_2	951						0.74	260577.92	42.177	0.693	0	0	247	704	0	951	0
level_5_cluster_6	5	level_4_cluster_3	1069						0.802	289272.22	38.786	0.691	0	0	212	857	0	1069	0
level_5_cluster_7	5	level_4_cluster_3	759						0.838	205278.5	47.154	0.691	0	0	123	636	0	759	0
level_5_cluster_8	5	level_4_cluster_4	491						1	109237.88	27.369	0.635	0	491	0	0	491	0	0
level_5_cluster_9	5	level_4_cluster_4	488						1	105363.78	27.929	0.626	0	488	0	0	488	0	0
level_5_cluster_10	5	level_4_cluster_5	374						1	79942.555	27.025	0.625	0	374	0	0	374	0	0
level_5_cluster_11	5	level_4_cluster_5	445						1	97848.72	29.472	0.633	0	445	0	0	445	0	0
level_5_cluster_12	5	level_4_cluster_6	77						1	16652.416	27.481	0.628	77	0	0	0	77	0	0
level_5_cluster_13	5	level_4_cluster_6	79						0.975	17537.371	26.107	0.636	77	2	0	0	79	0	0
level_5_cluster_14	5	level_4_cluster_7	66						1	14019.224	25.425	0.621	66	0	0	0	66	0	0
level_5_cluster_15	5	level_4_cluster_7	78						1	16669.113	27.376	0.625	78	0	0	0	78	0	0
level_5_cluster_16	5	level_4_cluster_8	1542						1	382227.16	33.656	0.651	0	0	0	1542	1542	0	0
level_5_cluster_17	5	level_4_cluster_8	856						1	210018.72	30.663	0.654	0	0	0	856	856	0	0
level_5_cluster_18	5	level_4_cluster_9	305						1	67100.47	27.069	0.635	0	0	0	305	305	0	0
level_5_cluster_19	5	level_4_cluster_9	297						1	63832.832	28.785	0.631	0	0	0	297	297	0	0
level_5_cluster_20	5	level_4_cluster_10	307						1	74126.24	31.65	0.64	0	0	307	0	307	0	0
level_5_cluster_21	5	level_4_cluster_10	303						1	72809.695	31.549	0.636	0	0	303	0	303	0	0
level_5_cluster_22	5	level_4_cluster_11	403						1	106388.18	32.672	0.656	0	0	403	0	403	0	0
level_5_cluster_23	5	level_4_cluster_11	487						1	136089.75	35.418	0.67	0	0	487	0	487	0	0
level_5_cluster_24	5	level_4_cluster_12	602						1	146731.22	31.994	0.659	602	0	0	0	602	0	0
level_5_cluster_25	5	level_4_cluster_12	605						0.992	160273.61	37.465	0.689	600	0	5	0	0	600	5
level_5_cluster_26	5	level_4_cluster_13	269						0.989	125715.875	44.517	0.905	266	0	3	0	0	0	269
level_5_cluster_27	5	level_4_cluster_13	338						0.991	148907.11	39.693	0.884	335	0	3	0	0	0	338
level_5_cluster_28	5	level_4_cluster_14	1215						0.514	514851.94	40.609	0.855	0	624	588	3	0	0	1215
level_5_cluster_29	5	level_4_cluster_14	2011						0.829	879240.5	44.996	0.862	0	342	1	1668	0	0	2011
level_5_cluster_30	5	level_4_cluster_15	1000						0.581	484495.22	43.187	0.891	299	581	0	120	0	0	1000
level_5_cluster_31	5	level_4_cluster_15	1462						0.827	632330.56	50.015	0.86	0	253	0	1209	0	0	1462

Figure A1. KAIST hierarchical clustering metrics for scaled features with PCA. Colour from red to green which respectively indicates lower scores and higher scores.

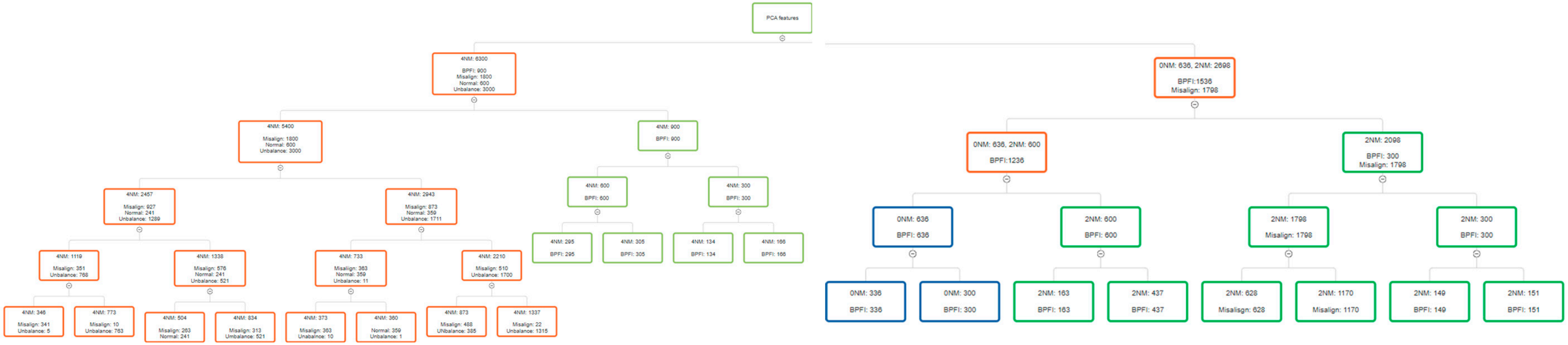


Figure A2. Example dendrogram of hierarchical clustering on KAIST dataset.

References

1. Chen, X.; Chen, Z. Cross-domain fault diagnosis of induction motor based on an improved unsupervised generative adversarial network and fine-tuning under limited labeled data. *Measurement* **2025**, *249*, 116988. [CrossRef]
2. Barroso, M.; Bossio, J.M.; Alaíz, C.M.; Fernández, Á. Fault Detection in Induction Motors using Functional Dimensionality Reduction Methods. *arXiv* **2023**, arXiv:2306.09365. Available online: https://arxiv.org/pdf/2306.09365 (accessed on 4 November 2025).
3. Gundewar, S.K.; Kane, P.V. Condition Monitoring and Fault Diagnosis of Induction Motor. *J. Vib. Eng. Technol.* **2020**, *9*, 643–674. [CrossRef]
4. Visentin, A.; Dalla, M.; Provan-Bessell, B.; O'sullivan, B. Unsupervised Induction Motor Anomaly Detection Using a Deep Convolutional Autoencoder Based on Multi-Sensor Data Fusion. In Proceedings of the 2025 IEEE International Conference on Smart Computing, SMARTCOMP 2025, Cork, Ireland, 16–19 June 2025; pp. 194–201. [CrossRef]
5. Samiullah, M.; Ali, H.; Zahoor, S.; Ali, A. Fault Diagnosis on Induction Motor using Machine Learning and Signal Processing. *arXiv* **2024**, arXiv:2401.15417. Available online: http://arxiv.org/abs/2401.15417 (accessed on 28 October 2024).
6. Treml, A.E.; Flauzino, R.A.; Suetake, M.; Ravazzoli, N.A. Experimental Database for Detecting and Diagnosing Rotor Broken Bar in Three-Phase Induction. *IEEE Dataport* **2020**. [CrossRef]
7. Piechocki, M.; Pajchrowski, T.; Kraft, M.; Wolkiewicz, M.; Ewert, P. Unraveling Induction Motor State through Thermal Imaging and Edge Processing: A Step towards Explainable Fault Diagnosis. *Eksploat. I Niezawodn.—Maint. Reliab.* **2023**, *25*. [CrossRef]
8. Sun, Z.; Machlev, R.; Wang, Q.; Belikov, J.; Levron, Y.; Baimel, D. A public data-set for synchronous motor electrical faults diagnosis with CNN and LSTM reference classifiers. *Energy AI* **2023**, *14*, 100274. [CrossRef]
9. Lessmeier, C.; Kimotho, J.K.; Zimmer, D.; Sextro, W. Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification. *PHM Soc. Eur. Conf.* **2016**, *3*. [CrossRef]
10. Elhalwagy, A.; Kalganova, T. Heterogeneous Induction Motor Current Dataset Fusion for Efficient Generalised MCSA-Based Fault Classification. In Proceedings of the 2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Abu Dhabi, United Arab Emirates, 14–17 November 2023; pp. 576–581. [CrossRef]
11. Jung, W.; Kim, S.H.; Yun, S.H.; Bae, J.; Park, Y.H. Vibration, acoustic, temperature, and motor current dataset of rotating machine under varying operating conditions for fault diagnosis. *Data Brief* **2023**, *48*, 109049. [CrossRef]
12. Jung, W.; Yun, S.-H.; Lim, Y.-S.; Cheong, S.; Bae, J.; Park, Y.-H. Fault Diagnosis of Inter-turn Short Circuit in Permanent Magnet Synchronous Motors with Current Signal Imaging and Unsupervised Learning. *arXiv* **2022**, arXiv:2206.07651. Available online: https://arxiv.org/pdf/2206.07651 (accessed on 4 November 2025).
13. Wang, B. Induction Motor Fault Classification with Topological Data Analysis. In Proceedings of the 2024 IEEE Energy Conversion Congress and Exposition, ECCE 2024—Proceedings, Phoenix, AZ, USA, 20–24 October 2024; pp. 5381–5386. [CrossRef]
14. Wang, B.; Lin, C.; Inoue, H.; Kanemaru, M. Induction Motor Eccentricity Fault Detection and Quantification Using Topological Data Analysis. *IEEE Access* **2024**, *12*, 37891–37902. [CrossRef]
15. Amarbayasgalan, T.; Ryu, K.H. Unsupervised Feature-Construction-Based Motor Fault Diagnosis. *Sensors* **2024**, *24*, 2978. [CrossRef]
16. Xia, M.; Li, T.; Xu, L.; Liu, L.; De Silva, C.W. Fault Diagnosis for Rotating Machinery Using Multiple Sensors and Convolutional Neural Networks. *IEEE ASME Trans. Mechatron.* **2018**, *23*, 101–110. [CrossRef]
17. Yang, J.; Cai, B.; Kong, X.; Shao, X.; Wang, B.; Yu, Y.; Gao, L.; Yang, C.; Liu, Y. A digital twin-assisted intelligent fault diagnosis method for hydraulic systems. *J. Ind. Inf. Integr.* **2024**, *42*, 100725. [CrossRef]
18. Kong, X.; Cai, B.; Yu, Y.; Yang, J.; Wang, B.; Liu, Z.; Shao, X.; Yang, C. Intelligent diagnosis method for early faults of electric-hydraulic control system based on residual analysis. *Reliab. Eng. Syst. Saf.* **2025**, *261*, 111142. [CrossRef]
19. Hadi Salih, I.; Babu Loganathan, G. Induction Motor Fault Monitoring and Fault Classification Using Deep Learning Probabilistic Neural Network. *Solid State Technol.* **2020**, *63*, 2196–2213. Available online: https://solidstatetechnology.us/index.php/JSST/article/view/2846 (accessed on 10 November 2025).
20. Wang, X.; Liu, Z.; Dai, M.; Li, W.; Tang, J. Time-shift denoising Combined with DWT-Enhanced Condition Domain Adaptation for Motor Bearing Fault Diagnosis via Current Signals. *IEEE Sensors J.* **2024**, *24*, 35019–35035. [CrossRef]
21. Eren, L.; Ince, T.; Kiranyaz, S. A Generic Intelligent Bearing Fault Diagnosis System Using Compact Adaptive 1D CNN Classifier. *J. Signal Process. Syst.* **2018**, *91*, 179–189. [CrossRef]
22. Sonmez, E.; Kacar, S.; Uzun, S. A new deep learning model combining CNN for engine fault diagnosis. *J. Braz. Soc. Mech. Sci. Eng.* **2023**, *45*, 644. [CrossRef]
23. Liu, H.; Liu, Z.; Luo, Z.; Chen, J.; Liu, H.; Zhang, Y. Contrastive Learning Based Fault Diagnosis of Motor with Multi-Modal Feature Fusion. In Proceedings of the 2023 China Automation Congress, CAC 2023, Chongqing, China, 17–19 November 2023; pp. 2486–2491. [CrossRef]

24. Usman, M.; Komatsu, T.; Htun, K.M.; Liu, Z.; Beck, A. Benchmarking Sensor Modalities with Few-shot Domain Adaptation for Cross-Domain Fault Diagnosis. In Proceedings of the 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE), Bari, Italy, 28 August–1 September 2024; pp. 3219–3224. [CrossRef]
25. Hou, W.; Xiao, G.; Liu, X.; Jiang, L.; Jing, W. 1D-DCTN: 1-D Deformable Convolutional Transformer Network for Multi-Signal Fault Diagnosis. In Proceedings of the 2024 43rd Chinese Control Conference (CCC), Kunming, China, 28–31 July 2024; pp. 5050–5057. [CrossRef]
26. Dobriban, E.; Owen, A.B. Deterministic Parallel Analysis: An Improved Method for Selecting Factors and Principal Components. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2019**, *81*, 163–183. [CrossRef]
27. Velasco-Gallego, C.; Lazakis, I.; Cubo-Mateo, N. Development of a Hierarchical Clustering Method for Anomaly Identification and Labelling of Marine Machinery Data. *J. Mar. Sci. Eng.* **2024**, *12*, 1792. [CrossRef]
28. Ran, X.; Xi, Y.; Lu, Y.; Wang, X.; Lu, Z. Comprehensive survey on hierarchical clustering algorithms and the recent developments. *Artif. Intell. Rev.* **2023**, *56*, 8219–8264. [CrossRef]
29. Byerly, A.; Kalganova, T. Class Density and Dataset Quality in High-Dimensional, Unstructured Data. *arXiv* **2022**, arXiv:2202.03856. [CrossRef]
30. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
31. Caliñski, T.; Harabasz, J. A Dendrite Method Foe Cluster Analysis. *Commun. Stat.* **1974**, *3*, 1–27. [CrossRef]
32. Davies, D.L.; Bouldin, D.W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [CrossRef]
33. Dunn, J.C. Well-Separated Clusters and Optimal Fuzzy Partitions. *J. Cybern.* **1974**, *4*, 95–104. [CrossRef]
34. Hirschberg, J.B.; Rosenberg, A. V-measure: A conditional entropy-based external cluster evaluation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2007; pp. 410–420. [CrossRef]
35. Mahnoor; Shafi, I.; Chaudhry, M.; Montero, E.C.; Alvarado, E.S.; Diez, I.d.l.T.; Samad, A.; Ashraf, I. A Review of Approaches for Rapid Data Clustering: Challenges, Opportunities, and Future Directions. *IEEE Access* **2024**, *12*, 138086–138120. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.