Signal Fusion and Dimensionality Reduction for Classification and Anomaly Detection Tasks

by

© Aref Sayareh

A Thesis submitted to the

School of Graduate Studies

in partial fulfillment of the

requirements for the degree of

Master of Science

Supervisor: Dr. Vinicius P. da Fonseca

Department of Computer Science

Memorial University of Newfoundland

December 2024

St. John's Newfoundland

Abstract

This thesis explores sensor fusion for feature reduction applied to classification and anomaly detection tasks. We developed new signal fusion techniques applied to tactile sensor signals texture classification and dimensionality reduction for anomaly detection in time series data. The first part of this thesis introduces a novel approach dimensionality reduction of tactile signals for texture classification using principal component analysis (PCA) and reducing exploration time without compromising classifier accuracy. Various pipeline configurations demonstrated that a 3-second exploration combined with PCA-fused features achieved up to 98% classification accuracy with a significant reduction in feature input, yielding a reduction factor of 6750 times. This part of the work highlights PCA's advantage over alternative fusion techniques, offering both interpretability and dimensionality reduction that enhance classifier performance.

The second part of this thesis examines anomaly detection within thread line signals, comparing the efficacy of PCA-based fusion, averaging, and raw signal methods. Experimental results across three thread lines indicate that PCA-based fusion provides a balanced sensitivity to anomalies, offering a streamlined process by removing the need for individual signal threshold adjustments. This work contributes to advancements in classification and automated anomaly detection by showcasing the effectiveness of PCA-based signal fusion for feature reduction and robust anomaly detection.

Acknowledgments

Thanks to everyone who supported me on this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The rapid advancement of sensor technology has revolutionized data-driven applications, where vast streams of multidimensional data are generated from various domains, including robotics [1], healthcare [2], manufacturing [3], and finance [4]. Robust methods that combine information from multiple sources and reduce dimensional complexity are essential to effectively utilize these data for classification and anomaly detection. Signal fusion and dimensionality reduction are powerful techniques that address these challenges, allowing more accurate, efficient, and interpretable models [5].

Signal fusion involves integrating multiple sensor signals to capture complementary information, providing a more comprehensive representation of the underlying events. This is particularly important in environments where individual sensors might be insufficient due to noise, uncertainty, or limited scope. Using fusion, systems can enhance their robustness and accuracy, which is crucial for applications such as tactile robotic systems and fault detection in industrial processes.

Dimensionality reduction, on the other hand, reduces the number of features not only accelerates computation but also enhances the generalization of models by focusing on the most relevant attributes. Techniques such as Principal Component Analysis (PCA) and more advanced manifold learning methods allow significant data simplification while preserving essential information for classification and anomaly detection.

This thesis explores the synergy between signal fusion and dimensionality reduction to improve classification accuracy and anomaly detection reliability. This research aims to contribute to fields where the precise identification of patterns is critical by developing and evaluating techniques that combine these two processes. Applications investigated include sensor-based anomaly detection in production lines and texture classification using fused tactile signals. The results of this work could serve as a foundation for improved machine learning models in complex, sensor-driven environments where effective data fusion and dimensionality reduction are paramount.

1.2 Problem Statement

As technology advances rapidly, a single sensor can no longer meet the ever-growing functional demands of automation. Consequently, technology for fusing information from multiple sensing sources has begun to draw significant attention [6]. Sensors widely used in real-world applications collect various data types, such as temperature, pressure, motion, and light intensity. Each sensor contributes different information about the environment or system it observes. Sensor fusion techniques are utilized to improve the precision and reliability of the collected data. Sensor fusion combines data from multiple sensors to generate a more comprehensive understanding of the monitored conditions. By aggregating data from various sources, sensor fusion addresses the limitations of individual sensors, extracting the most pertinent and reliable information, thereby facilitating better decision-making and enhanced system performance.

In addition, sensors can be affected by various kinds of disturbances or unwanted signals, often referred to as 'noise.' These noises can alter the important characteristics and timing details of the signals that the sensors detect. As a result, this can cause incorrect interpretation or understanding of the data, which can lead to wrong conclusions or diagnostics based on that distorted information [7]. The applications of fusing and reducing dimension techniques filter out unwanted noise, enabling the system to concentrate on the critical features of the signals. This improves the clarity and accuracy of the data, which is vital for effective classification and anomaly detection. When several noisy signals are combined, the noise effect might be amplified, potentially causing greater inaccuracies. Thus, fusing signals and extracting most information not only improves the output of individual sensors but also contributes to a better overall classification and diagnosis.

Interpretability in machine learning models is one of the other challenges in signal fusion and denoising, especially in high-stakes fields such as healthcare and finance. In these domains, the decisions made by the models can have significant impacts on individuals and society. Therefore, understanding how and why these models make certain decisions is crucial to ensure that they are fair, transparent, and reliable.

1.3 Research Goals and Questions

This study seeks to integrate and fuse signals from various sensors to maximize information extraction while eliminating noise that leads to errors in classification fault diagnosis. The primary objectives for this research are as follows:

- Create a pipeline to integrate and reconstruct various signals with the highest information and lowest features. Evaluate and compare the pipeline with the baseline experiments.

- Extract the most important information from raw signals to detect anomalies.

Given these objectives, various Research Questions must be addressed:

- **Question 1:** Is it possible to enhance signal classification through signal fusion?

- **Question 2:** Can signal fusion improve the anomaly detection accuracy while reduce the number of singles?

1.4 Contributions

This thesis makes significant contributions to the field of signal processing and anomaly detection in two key areas.

- A primary contribution of this research is the analysis of signal fusion using Principal Component Analysis (PCA) for texture classification tasks performed by tactile robotic hands. By implementing PCA, this work addresses the balance between feature reduction and exploration efficiency, examining how reducing feature space impacts the exploration duration of the robotic hand while still maintaining effective texture recognition. This trade-off analysis provides insight into optimizing both classification performance and operational efficiency, which is critical for real-time robotic applications where quick and accurate classifications are essential.

- Another central contribution involves developing an unsupervised anomaly detection method applied to Instrumar Limited industrial fiber production data. Using a z-score approach, this study assesses the effectiveness of different fusion techniques, namely averaging fusion, PCA-based fusion, and raw signal evaluation, to detect anomalies in fiber properties. Through a detailed comparison of these methods, this work highlights the benefits of PCA in detecting subtle anomalies across all data streams and ensures consistent performance for all signals, reducing the manual threshold adjustments typically required in raw signal methods. This advancement addresses key challenges in industrial monitoring, providing a streamlined and adaptable approach to anomaly detection that can identify both known and previously unidentified patterns.

1.4.1 Thesis Outline

Chapter 2 presents a review of the literature on signal processing and signal fusion covering foundational studies and recent advances. This chapter establishes the research context for signal fusion in both tactile classification and anomaly detection, identifying existing challenges and gaps in current approaches. Chapter 3 focuses on signal fusion for texture classification using tactile robotic hands, emphasizing feature reduction through Principal Component Analysis (PCA). This chapter explores the trade-offs between the rate of feature reduction and the exploration duration of the robotic hand, demonstrating how PCA optimizes classification performance by efficiently managing the complexity of sensor data. Chapter 4 provides a comparative analysis of anomaly detection techniques applied to industrial fiber data from Instrumar Limited. Using an unsupervised z-score approach, this chapter evaluates the effectiveness of anomaly detection across raw signals, average fusion, and PCA fusion methods, illustrating the sensitivity of each approach to detect industrial anomalies.

Finally, Chapter 5 concludes with a summary of research findings, limitations, and proposed future directions to advance signal fusion techniques in tactile sensing and anomaly detection.

Chapter 2

Literature Review

In this section, we review various works in the field of signal fusion to explore and evaluate different methods used to combine multiple signals. Signal fusion is a critical process in fields like sensor networks, machine learning, and data integration, where information from multiple sources is fused to produce more accurate and reliable outcomes. The goal is to identify the most effective fusion techniques and their applicability across different domains. Various methods, from simple statistical approaches to advanced machine learning models, have been proposed to enhance the reliability, accuracy, and robustness of systems that use multiple signal inputs.

One of the key challenges in signal fusion is the trade-off between complexity and interpretability. While advanced methods like deep learning-based fusion models provide strong predictive abilities, they typically function as black boxes, complicating the understanding of the internal fusion processes. This lack of interpretability can be a major limitation in critical applications where understanding the reasoning behind predictions or classifications is essential. Therefore, in addition to exploring high-performing methods, we must place significant emphasis on identifying fusion techniques that offer a clear interpretative framework.

To address this, we will also review interpretable methods for signal fusion. These methods not only provide strong performance, but also offer insights into how different signals are combined, which is essential for applications like healthcare diagnostics, industrial monitoring, and decision-making systems. We aim to highlight fusion approaches that balance predictive power with transparency, which provides both accuracy and a clear understanding of the underlying mechanisms.

In image fusion algorithms, [8] proposed a method that uses the wavelet transform to enhance resolution. They compared different approaches to identify the most effective fusion strategy. This exploration is a valuable reference point for investigating signal fusion methodologies, particularly applying wavelet transformers to signals. In addition, [9] searched the domain of underwater acoustic signals, using variational mode decomposition and permutation entropy to extract frequency characteristics. Increasing the accuracy of classification capabilities in their work underscores the importance of innovative signal processing techniques for effective fusion.

Exploring synchronization strategies for data fusion in Advanced Driver Assistance Systems (ADAS), the work of [10] inspired our investigation of different window frames for signal fusion and dimensionality reduction. Their insights into synchronization techniques offer valuable considerations for our fusion methodology. In the Visual-Inertial Odometry (VIO) domain, [11] introduced an end-to-end selective sensor fusion framework, incorporating deep learning models post-signal fusion. This approach suggests integrating advanced machine learning techniques into our fusion methodology.

Meanwhile, [12] proposed a fault diagnosis method for deep learning and information fusion that demonstrated the conversion of signal data into images for classification with convolutional neural networks (CNN). Their approach offers different possibilities for signal representation and classification.

In [13], the authors presented a hybrid approach for multispectral image fusion. They combined spectral and spatial PCA (Principal Component Analysis) methods that inspired our methodology to consider PCA for signal fusion and dimensionality reduction. [14] proposed an image denoising fusion method based on fractional Fourier transform that shows its superiority over existing approaches. Their exploration of Fourier transform techniques suggests potential applications for signal fusion in our methodology. Furthermore, [15] utilized fuzzy logic for sensor fusion in a lathe monitoring system that demonstrates its potential for predictive maintenance applications. This approach highlights the power of fusion techniques in different domains. In addition to exploring fusion techniques, [16] employed canonical correlation analysis to improve system identification by fusing feature vectors extracted from EEG signals. Their focus on the statistical behavior of signals for feature extraction provides insight into potential strategies for our signal fusion approach.

In Body Sensor Networks (BSNs), the authors in [17] addressed the need for effective multisensor data fusion that offers insights into various fusion strategies. Their comprehensive overview provides valuable guidance for our fusion approach. Finally, the work shown in [18] explored tactile sensors in robots for texture categorization, achieving high classification accuracy with machine learning methods. Their work serves as a baseline for comparison, guiding our efforts to enhance classification accuracy through signal fusion techniques.

Chapter 3

Evaluating Signal Fusion in Robotic Texture Classification

3.1 Introduction

Robots in unstructured environments face significant challenges due to the variability and unpredictability of the surroundings. Unlike structured environments, which are controlled and consistent, unstructured environments present different types of objects, surfaces, and obstacles that robots must work with. These conditions can hinder the robot's ability to perform tasks reliably, as visual sensors alone often cannot provide sufficient information for accurate decision-making. For instance, robots may classify visually similar objects with completely different textures into the same class.

One way to improve the perception of the environment is by touching and physical engagement with objects [19]. This physical engagement helps to understand characteristics such as friction and roughness. In robotics, tactile sensors enable the perception of the environment in some challenging situations, e.g., under-reflection, cluttered environments, insufficient light conditions, and occlusion. Capacitive [20, 21] and magnetic [22, 23] sensors are used in robots to recognize the textures of objects or grab them. Lima et al. [18] used magnetic, angular rate, and gravity sensors in 9 degrees of freedom in addition to a barometer (to measure pressure) [24, 25] in an experiment to classify different textures. There were 12 different textures in which a tactile finger was moved around the texture in a square-like path 100 times. Figure 3.1 shows the path of the robotic finger and the textures. Each experiment took 12 seconds to touch a complete square path in [18]. Also, 10 different measurements were recorded in each step with distinct sensors. In Lima et al. [18], after collecting the data, they conducted an experiment that classified the textures based on the total $12s$ of a single signal among 10 signals. Their work did not investigate the use of sensor fusion to reduce the dimension of the feature vector. In a similar work, Sai et al. [26] analyzed two ways of feature extraction to achieve high-accuracy classification performance. Their first method is a statical-based preprocessing strategy that uses the signals' statical features. In contrast, the second method is a frequency-based preprocessing strategy with a Fast Fourier Transform. They also analyzed only one signal, and barometer signals had the highest accuracy in both studies. In the latter research, statical-based preprocessing performed better than the raw and frequency-based representation [27]. The authors of [28, 29] have employed similar strategies and data collected from the same sensing module used in this thesis, but with deep neural networks for grasping objects under positional uncertainty and estimating the pose of objects under grasp.

The fusion of tactile sensing has also been proven to be a valuable tool in surface approximation and reconstruction tasks. For instance, the flexibilization of end-effector approaches in surface approximation has been demonstrated using a bioinspired tactile sensing module, highlighting the advantages of adaptive tactile feedback [30]. Similarly, compliant tactile perception has been leveraged for haptic blind surface reconstruction, showcasing the ability to reconstruct surfaces with minimal visual guidance [31].

Recent studies have also explored innovative approaches to enhance these capabilities. For example, tactile-enabled prosthetic fingers and feedback gloves have been employed to study object recognition through manipulation [32]. Additionally, early-phase tactile object recognition using underactuated robotic hands has demonstrated promising results in improving grasping efficiency [33]. Furthermore, fuzzy logic controllers have been applied for object manipulation using three-fingered robotic hands, showcasing the potential of soft computing techniques in robotics [34].

In Carvalho et al. [27] work, they used several numbers of data samples in different time instances. These data samples are called points, and they represented the data with 3- and 10-points and extracted different subsets of features in each point to compare and evaluate the different representations. The 3-point representation contains data in the initial, stable, and final stages of the experiment, while the 10-points are the sample data of uniformly distributed time between the initial and final stages of the experiment. They also applied statistical representation, which performed best among their strategies.

The work shown in [18, 35] explored tactile sensors in robots for texture categorization, achieving high classification accuracy with machine learning methods. Their work serves as a baseline for comparison, guiding our efforts to enhance classification accuracy through signal fusion techniques.

The main idea of this paper is to reduce the touch duration without significantly impacting the classification's accuracy. We reduced the touch exploration time while fusing the tactile signals to study the effects of the data fusion on the texture classification. We used the principal components as a data fusion approach, and we also reduced the touch duration and the features' dimensionality to construct the signals with fewer features than the main experiments. To analyze the signals in these time series experiments, we created data frames to apply different methods on them [36]. We explored different configurations, such as window size, to find the best data frame configuration and analyze the performance of our pipeline. Therefore, our work advances the research on robotics in several ways. Unlike previous studies focusing on prolonged touch durations, we investigate using signal fusion to reduce the touch exploration time while maintaining or enhancing classification accuracy.

We employ PCA and ICA (Independent Component Analysis) for signal fusion and dimensionality reduction, which have not been extensively explored in the context of tactile sensor data for texture classification. By experimenting with different window sizes and configurations for data frames, we aim to identify the most effective setup for signal analysis, potentially improving the overall efficiency and robustness of the classification process. Our study compares raw data representation and integrates these with data fusion techniques to highlight the best strategies for accurate texture classification. In summary, our research advances the research on tactile sensor data analysis by combining reduced touch durations with signal fusion techniques, leading to more efficient and robust robotic perception systems.

This work is organized as follows. Section 3.2 presents the materials and methodology employed in this study, such as the data collection and fusion techniques. Section 3.3 discusses the results and findings, providing a comprehensive analysis of the data. Finally, Section 3.4 offers a discussion on the implications of the findings, including potential limitations and future research directions.

3.2 Material and Methods

Figure 3.1 presents an overview of the pipeline used in this work. After preparing the dataset (step 1), we split it into train and test data (step 2). First, we extracted several samples with random starting points and durations shorter than the main experiments (step 3), and we used the train data set to build the fusion model (step

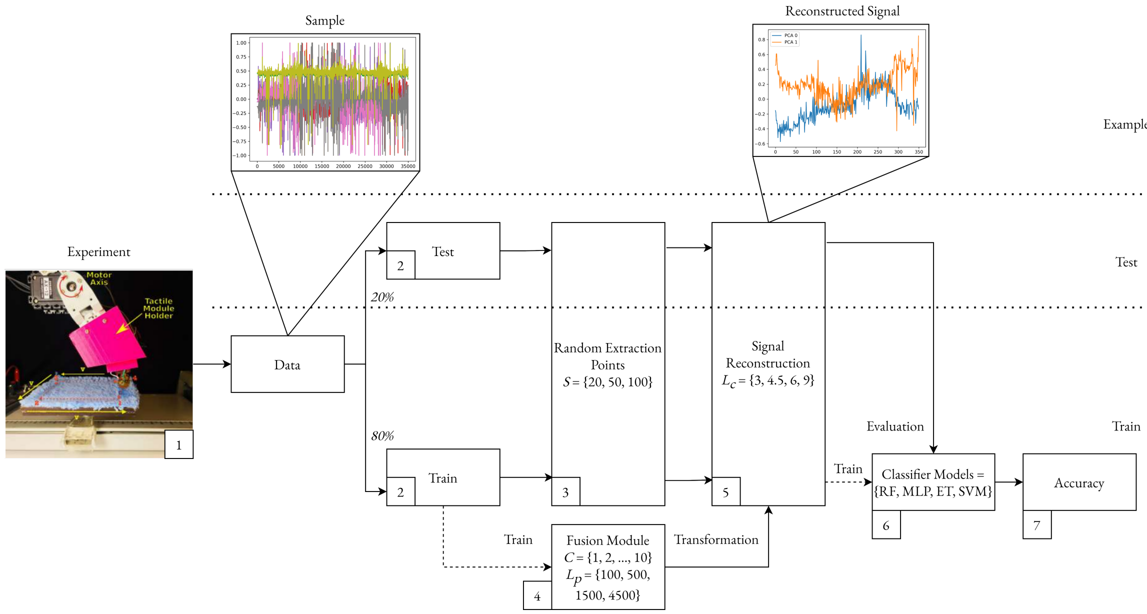

Figure 3.1: The pipeline for signal fusion and texture classification. Each number represents a step or module in our work. 1) Data collection with robotic hand with tactile sensing; 2) splitting data to train and test; 3) generate random extraction points; 4) training the fusion module 5) Reconstructing signals based on the fusion module and on the random extraction points; 6) training the texture classifiers with training reconstructed signals 7) evaluating the classifiers using testing reconstructed signals.

4). We picked the different lengths of data frames to reconstruct the signals using a trained fusion model (5). Finally, classifiers were trained and evaluated based on the new reconstructed signals (6). The following subsections detail each step.

3.2.1 Dataset preparation and Random Extraction

Our method was tested using the dataset from [37, 38] and used their method as a baseline. The dataset was generated by one robotic hand with tactile sensors that sensed 12 different textures. In each experiment, the robotic hand explored each texture for 12 seconds in a square path, and it recorded the IMU (Inertial Measurement Unit) and barometer signals. The experiments were repeated 100 times for each texture. Therefore, the dataset contains 10 signals, 9 made by the IMU, and 1 by the barometer sensor for each touch exploration. Because the IMU sensor produces multiple signals related to tactile movements, we decided to use the IMU signals and ignored the barometer data since it only contains one signal indicating the pressure of the tactile's tip on the texture, which is not related to movements. The data is collected with different speeds of exploration, i.e. 30mm/s, 35mm/s, and 40mm/s. We only used the 30mm/s dataset to compare and evaluate our method.

Each experiment is 12 seconds of touching and exploring the surface of the texture and recording the sensor data. The data in each experiment are saved in a file; therefore, we have 100 files for each texture. In step 2, to split the data, we picked 80 files for each texture for the train and the rest of the 20 files for the test. We used the train data to train the fusion module and the classifier, and test data was not used at all for the training phase.

In addition, the idea of this project was to reduce the exploration time while fusing the signals. Therefore, if we want to select a shorter duration of 12 seconds of an experiment as a new sample, we can pick it from different parts of each experiment. In other words, it means that we can have multiple sub-experiments (shorter duration of an experiment) in each original experiment. So, to create these sub-experiments from each experiment, in step 3, we generate random extraction points which indicate our sub-experiment starting point, and we pick the specific window frame from those starting points to apply our methods to them.

3.2.2 Signal Fusion

Our study extracts random samples from each experiment and uses Principal Component Analysis (PCA) and Independent Component Analysis (ICA) to fuse the signals within a specific time window. Both approaches transform the original signals into several user-defined new components, effectively reducing dimensionality while retaining essential information. Then, we reconstruct the signals from these components, aiming to preserve the critical features and improve the clarity and interpretability of the data for further analysis.

Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of a dataset through a linear transformation. By projecting the data onto a new set of orthogonal axes, PCA aims to capture the maximum variance with the fewest number of principal components, thereby simplifying the dataset while retaining its most significant features [39]. In contrast, Independent Component Analysis (ICA) takes a different approach to data decomposition. ICA attempts to unravel a multivariate signal into additive, statistically independent components. Unlike PCA, which focuses on variance maximization, ICA is designed to uncover hidden factors or sources in the data by leveraging the assumption of statistical independence among these sources [40].

We used non-overlapping windows on the signals to analyze them. A window frame is a small chunk of an experiment that is used in the fusion model for the transformation. The window frame can have different lengths, as the variable L_p is described below. The shape of one completed experiment is 36,000 measurements over 9 sensors (i.e., the shape is 36000 × 9), while the shape of a fusion window frame with a length of L_p that we used in our experiments is $(L_p \times 9)$ where $L_p \leq 36000$. We used the PCA and ICA models to fuse and reduce the number of features a window frame provides. Therefore, the fusion module transforms a $L_p \times 9$ data in a window frame into $(C \times 1)$, where C is the number of experiment-defined fusion components and $C \leq 9 \times L_p$.

The training dataset was divided into window frames with lengths of L_p and, after flattening them into a 1D array, used as samples to train the fusion module. Figure 3.2 shows the IMU signals of an experiment divided into a number of L_p windows (red windows). Each window contains a $(L_p, 9)$ data matrix that is flattened to $(1, L_p * 9)$ as a training sample and feeds to the fusion modules to train them. Also, Algorithm 1 describes the main loop on each experiment and divides them into L_p windows. After division, the flattened sample is saved in a list ($fusion\_sample$), which the fusion module will be trained with.

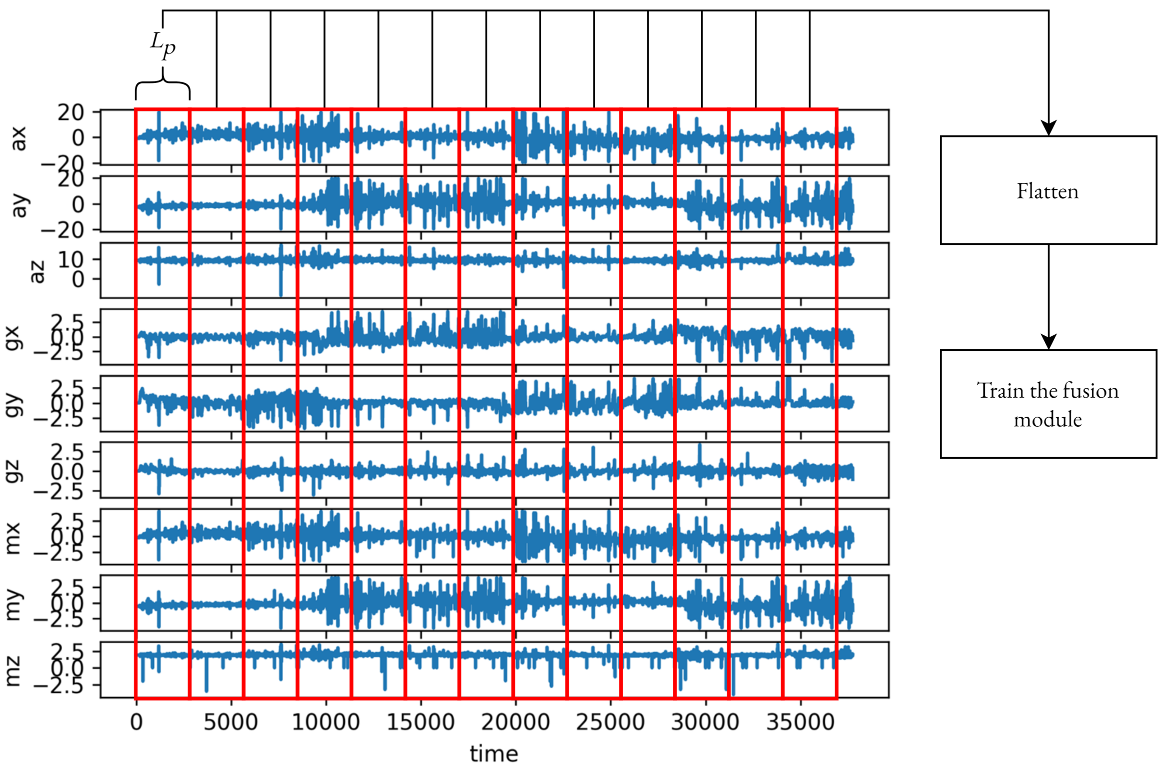

Figure 3.2: Splitting an experiment to non-overlap windows frames to train the fusion module (e.g. PCA)

Since the fusion window frame might be small, the resulting transformation might not have enough information for classifiers to perform accurately. For instance, L_p = 100 means 0.33ms, which is a very short exploration to determine the texture classes. In contrast, increasing L_p reduces the information within the first few components, which is not adequate for classification. Since we aimed to reduce the number of signals by fusing to increase the interpretability, we avoided adding more components

Algorithm 1: Fusion Training Algorithm

Initialize FusionModule: model ← PCA()/ICA()

Initialize sample list: pca_samples ← []

for each experiment in train_experiments **do**

  for $i = 0$ to $36000//L_p$ **do**

    Append sample to list:

    fusion_samples.append(experiment$[i * L_p : (i+1) * L_p]$)

  end for

end for

Fit model: model.fit(fusion_samples)

for large L_ps. Therefore, we decided to concatenate several transformations with small L_p and create a new window frame for classifier models. Variable N is defined to demonstrate the number of the fusion window frames required to be concatenated for classifiers. After N windows are concatenated, the resulting shape is (N, C), which is our new sample to use for classification. In addition, the length of the classifier window frame is demonstrated by L_c. We can consider that the L_c of the main data set is 36000, whereas our $L_c = N \cdot L_p$ (L_c : The length of the reconstructed signal:

Also, the L_c values can be expressed in seconds by dividing it by 3000, where the last equation expressed the L_c value with the number of samples that the window frame contains.

In order to create a new data set with a shorter duration than the main one ($12s$), we generated a number of sub-experiments from each experiment of the main dataset. The length of these sub-experiments is L_c where $L_c < 36000$, and the number of sub-experiments that are created from one experiment is denoted by S. The beginning of the sub-experiments' window frame is chosen randomly in the experiments (Fig 3.1 Module 3). If S_i denotes the beginning of the window frame for the ith sub-experiment, the boundary of this random variable is $0 \leq S_i < 36000 - L_c$. S_i must have a L_c distance to the end of the experiments to make new sub-experiments from the end of the experiments without any overflowing problems.

Algorithm 2: Sub-experiment Extraction Algorithm

for each experiment in experiments **do**
    // Initialize random extraction points:
    random_extraction_points $\leftarrow$ random.randint$(0, \text{experiment_length} - L_c, S)$
    // Initialize sub-experiment:
    sub_experiment $\leftarrow$ [experiment$[i : i + L_c]$ for i in random_extraction_points]
    // Append sub-experiment to sub-experiments:
    sub_experiments += sub_experiment
end for

After determining random extraction points, the sub-experiments were divided into new matrices for further processing. The algorithm 2 is a function for specifying the random extraction point and the division of sub-experiments. In the next phase, we transformed each sub-experiment with the fusion module. Therefore, the sub-experiments' shapes changed from $(L_c, 9)$ to (N, C). To do so, the sub-experiments were divided into N parts with the shape of $(L_p, 9)$. Then, each part was fused to components $(1, C)$ and concatenated with other parts to form a sample for our new dataset.

Figure 3.3 shows the transformation and reconstruction of one complete experiment. The experiment is divided into L_p window frames to transform them using PCA with $C = 2$ and use the components for the fused signals. Each window frame is transformed separately, and the new signals are called reconstructed signals. Here $L_p = 100; L_c = 35000; N = 350$. Also, Algorithm 3 shows the steps for transforming each sub-experiment and making new reconstructed signals for our new dataset.

We extracted sub-experiments and reconstructed the signals in both train and test experiments. The reconstructed train data are used to train the classifiers, and the test data is used to evaluate and compare them. The fusion module and the classifiers do not have access to the test dataset in the training phase.

3.2.3 Classification & Evaluation

After the reconstruction phase, two data sets were created, one for training and the other for evaluating the classifier models. The classifier models that were used in this project are:

- Support Vector Machine (SVM) [41]

- Random Forest Classifier (RF) [42]

Algorithm 3: Sub-experiment Transformation Algorithm

```
// Initialize new dataset:
new_dataset ← []
for each sub_experiment in sub_experiments do
    // Reshape sub_experiment:
    sub_experiment.shape ← (N, L_p)
    // Initialize new sample:
    new_sample ← np.zeros((N, C))
    for i = 0 to N do
        // Transform sub_experiment:
        new_sample[i] ← fuse(sub_experiment[i])
    end for
    // Append new sample to new dataset:
    new_dataset.append(new_sample)
end for
```

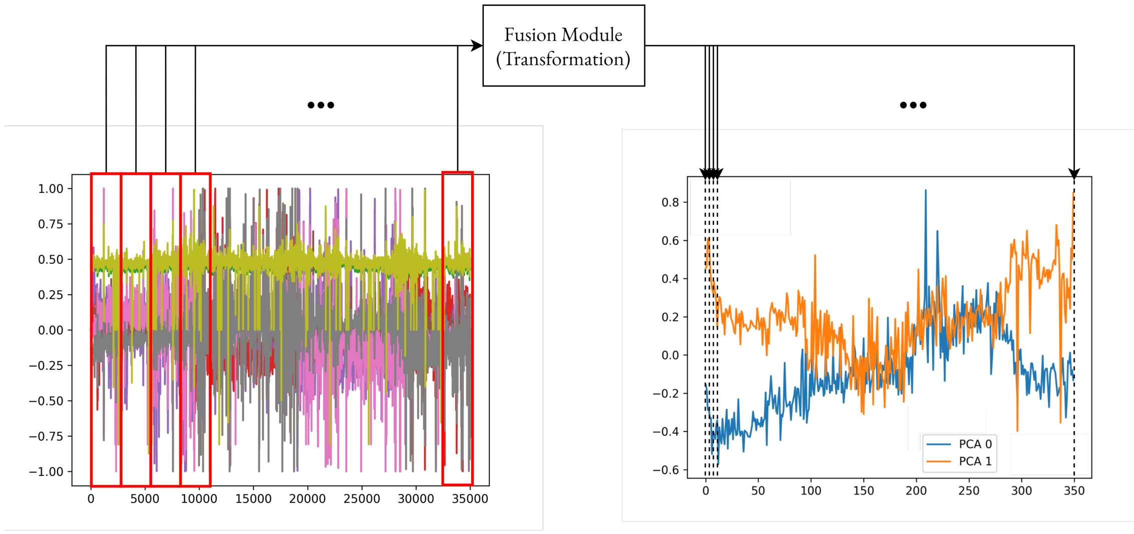

Figure 3.3: Reconstructing and fusing signals of an experiment. Split data to window frames of size L_p and transform it using a fusion module, such as PCA. Then concatenate the components of each transformation to reconstruct a new signal with lower dimensions.

- Multi-Layer Perceptron (MLP) [43]
- Extra Tree Classifier (ET) [44]

These models are imported from Scikit-Learn package [45] in Python with their default configurations.

The main goal of this project was to find the best variables to fuse the signals and classify the textures with less duration than 12s while the accuracy stays high.

3.2.4 The Pipeline

The variables (N, C, L_c, L_p, M) have different values. Each set of values of these variables is called the *pipeline configuration*. As Equation 3.1 shows, the pipeline must test and evaluate 2560 different pipeline configurations. Figure 3.1 and Table 3.1 show the values of the variables.

Variable	**Variable Name**	**Values**
FusionMethod	-	PCA, ICA, NO-FUSION
SamplePerFile	S	~~10~~, ~~20~~, 50, 100
Seed	-	20 different seeds (check Equation 3.4)
Model	M	MLP, ET, RF, SVM
FusionWindowSize	L_p	~~100~~, 500, 1500, ~~4500~~
SampleWindowSize	L_c	3, 4.5, 6, 9
Components	C	~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, 7, 8, 9, 10

Table 3.1: Pipeline configuration variables and their values

$$\begin{aligned}|\text{pipeline configuration}| &= |N| \cdot |C| \cdot |L_p| \cdot |L_c| \cdot |M| && (3.1)\\ &= 4 * 10 * 4 * 4 * 4 && (3.2)\\ &= 2560 && (3.3)\end{aligned}$$

To ensure that the evaluations were accurate, we ran each pipeline configuration with 20 random seeds. Therefore, the total runs were $20 * 2560$, which is 51200. We used the digits of the π value as the seed number. The π digits are divided into a consecutive sequence with a length of 4 for our seed numbers.

$$random\_seeds = \{3141, 5926, 5358, 9793, ...\} \tag{3.4}$$

With each pipeline configuration, when the random seed changed, the following modules ran again to evaluate the model again:

1. Random Extraction Point
2. Signal Reconstruction
3. Classifier
   - Training
   - Evaluation

So, the pipeline first generates a new data set with item 1 & 2 w.r.t values (N, C, L_c, L_p) and the new seed number. Then, it trains the classifier model and evaluates it with the test data set. Since we had 51200 configurations, we tried to multi-process the modules to make it faster. For items 1 & 2, the pipeline creates new signals in different processes for each experiment instead of reconstructing one experiment per time. Also, classifiers in the Scikit-Learn package support multi-processing, except for SVM. Therefore, for the SVM model, we first create several data sets with different pipeline configurations, and then we train the SVMs with those data sets in different processes.

In addition to multi-processing for increasing the performance, we found that some pipeline configurations are a subset of other pipeline configurations. In one experiment, if the $\max(C)$ and the $\max(N)$ are extracted and saved in a matrix, we can use them for other pipeline configurations with the same (L_p, L_C) but different (C, N). The matrix that contains the train data set is a 5-dimensional matrix, and the dimensions are explained in Equation below:

$$
\begin{aligned}
x\_dataset.shape &= (|\mathrm{textures}|, |\mathrm{experiments}|, \max(\mathrm{N}), \mathrm{L_p}, \max(\mathrm{C})) && (3.5)\\
y\_dataset.shape &= (|\mathrm{textures}|, |\mathrm{experiments}|, \max(\mathrm{N})) && (3.6)
\end{aligned}
$$

when we make a data set with $N = 100$ and $C = 10$, the data with any other N and C will be available for training and evaluation. After separating data with a specific pipeline configuration, the x data and y (the labels) are in 5 and 3 dimensions, respectively. The data in x are changed to 2-dimensions with shape of $(|\mathrm{textures}| * |\mathrm{experiments}| * \mathrm{N_i}, \mathrm{L_{p_i}} * \mathrm{C_i})$ and y to $(|\mathrm{textures}| * |\mathrm{experiments}| * \mathrm{N_i}, 1)$

3.3 Results

These evaluations aim to determine the best configuration for this classification. The ideal configuration is to have the highest accuracy with the lowest number of features. In order to compare the feature reduction between different configurations, we defined a formula to check the proportion of features before and after the fusion module. We called this reduction factor, defined as 3.7. In this equation, increasing the size of the fusion window or decreasing the number of fusion components will increase the value of the reduction factor. The range of the reduction factor with our defined pipeline configurations is within $[90, 40500]$.

$$ReductionFactor = \frac{Original\ number\ of\ features}{Reduced\ number\ of\ features} = \frac{L_p \cdot N \cdot 9}{C \cdot N} = \frac{L_p \cdot 9}{C} \tag{3.7}$$

Table 3.2 shows the best configurations with the highest accuracy. The exploration time of the tactile robot in both configurations was 9 seconds, and the best models for the classifications were SVC and RF, while the PCA module was used for the fusion. Note that several pipeline configurations had achieved 100% accuracy, so we sorted them based on the reduction factor to find the highest feature reduction among the configurations. In addition, Figure 3.4 compares the classifier models with different fusion methods. As it shows, the RF classifier has performed as the best classifier in all three methods, while the PCA was the best fusion method in comparison to the ICA and no-fusion methods.

Also, we are interested in reducing the exploration time of the robot, in which

Rank	FusionMethod	Model	L_c	Accuracy	ReductionFactor
1	PCA	SVC	9	100.00	6750.00
2	PCA	RF	9	100.00	3375.00

Table 3.2: The highest accuracy pipeline configurations

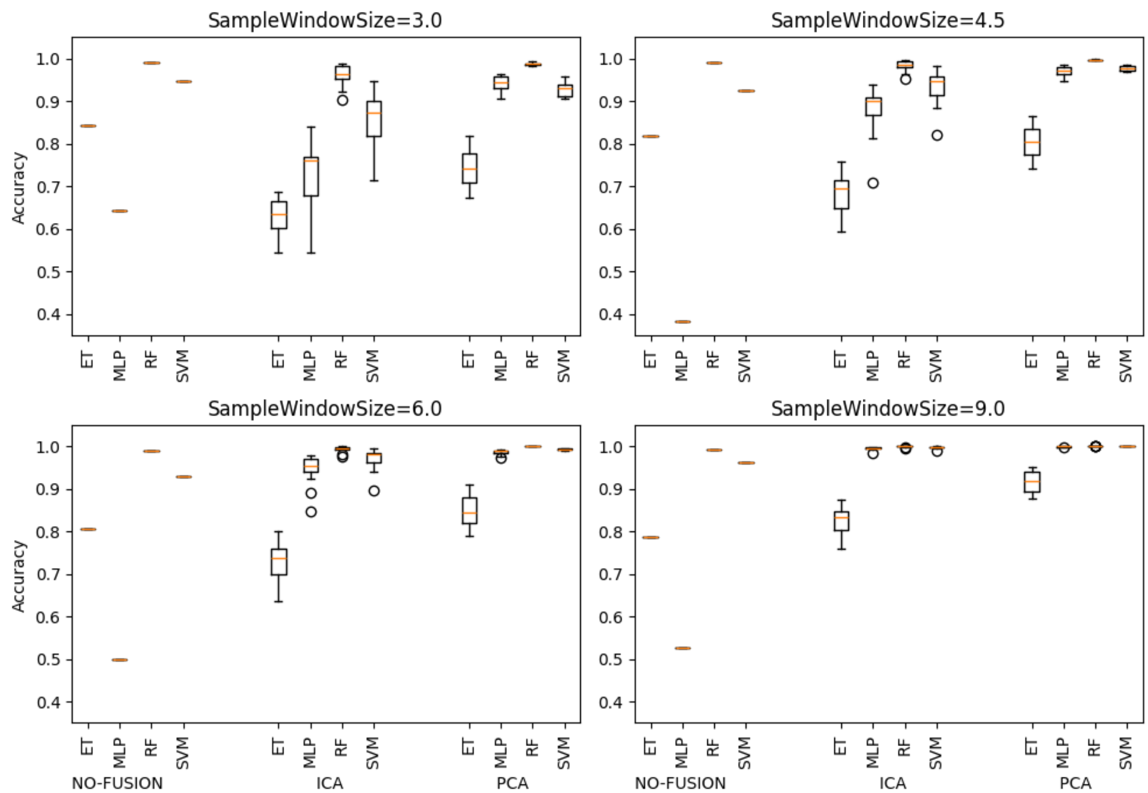

Figure 3.4: The accuracy of all of the pipeline configurations separated by the fusion module, classifier, and the sample window size (L_c).

Table 3.3 shows the evaluation of different configurations with 3 second exploration. As Table 3.3 presents, the reduction factor of these configurations is not high enough as the maximum value of this variable is 40500. Therefore, we checked the configurations with higher accuracy than 95% and sorted them out based on their reduction factor. As Table 3.4 shows, the pipeline could achieve the value of 6750.0 for the reduction factor while the accuracy is close to 98%.

Rank	FusionMethod	Model	L_c	Accuracy	ReductionFactor
1	PCA	RF	3	99.23	500.00
2	PCA	RF	3	99.22	450.00
3	PCA	RF	3	99.21	562.50
7	ICA	RF	3	98.90	562.50
8	ICA	RF	3	98.90	500.00
9	ICA	RF	3	98.85	450.00
10	PCA	RF	3	98.80	2250.00

Table 3.3: Highest accuracy of the pipeline configurations where $L_c = 3$

In addition, Figure 3.5 presents the average accuracy of different configurations with a specific L_c and a reduction factor. The ideal configuration is the one with the highest reduction factor and lowest touch duration. However, to achieve both, the accuracy will drop. In addition, there is a growing trend in all of the plots while the reduction factor is increasing and it is less than 300. The reason for that is the classifier models are not tuned, and the parameters are set by default. Since there

Rank	FusionMethod	Model	L_c	Accuracy	ReductionFactor
1	PCA	RF	3	97.94	6750.00
5	PCA	MLP	3	95.22	6750.00
6	PCA	RF	3	97.94	4500.00
10	PCA	MLP	3	95.03	4500.00

Table 3.4: Highest reduction factors of pipeline configurations where $L\_c = 3$ and the accuracy > 95%.

are so many features when we have a low reduction factor, when we increase it, we expect the accuracy to increase. Therefore, based on the application, we can choose different configurations to have a high reduction factor or low-touch exploration or have them both in a medium range.

3.3.1 Feature Importance

Since the RF classifier and the PCA models are both interpretable, we can find out which parts of the signal are important for the classifier while we have the PCA transformation as the intermediate phase. The RF models in the Scikit-learn package have the feature importance property, showing each feature's importance in the feature vector after the train. Because we used the reconstructed signal as the input of the RF models, the feature importance matrix of the classifier gives us the importance of the components of each fusion window after the transformation. In

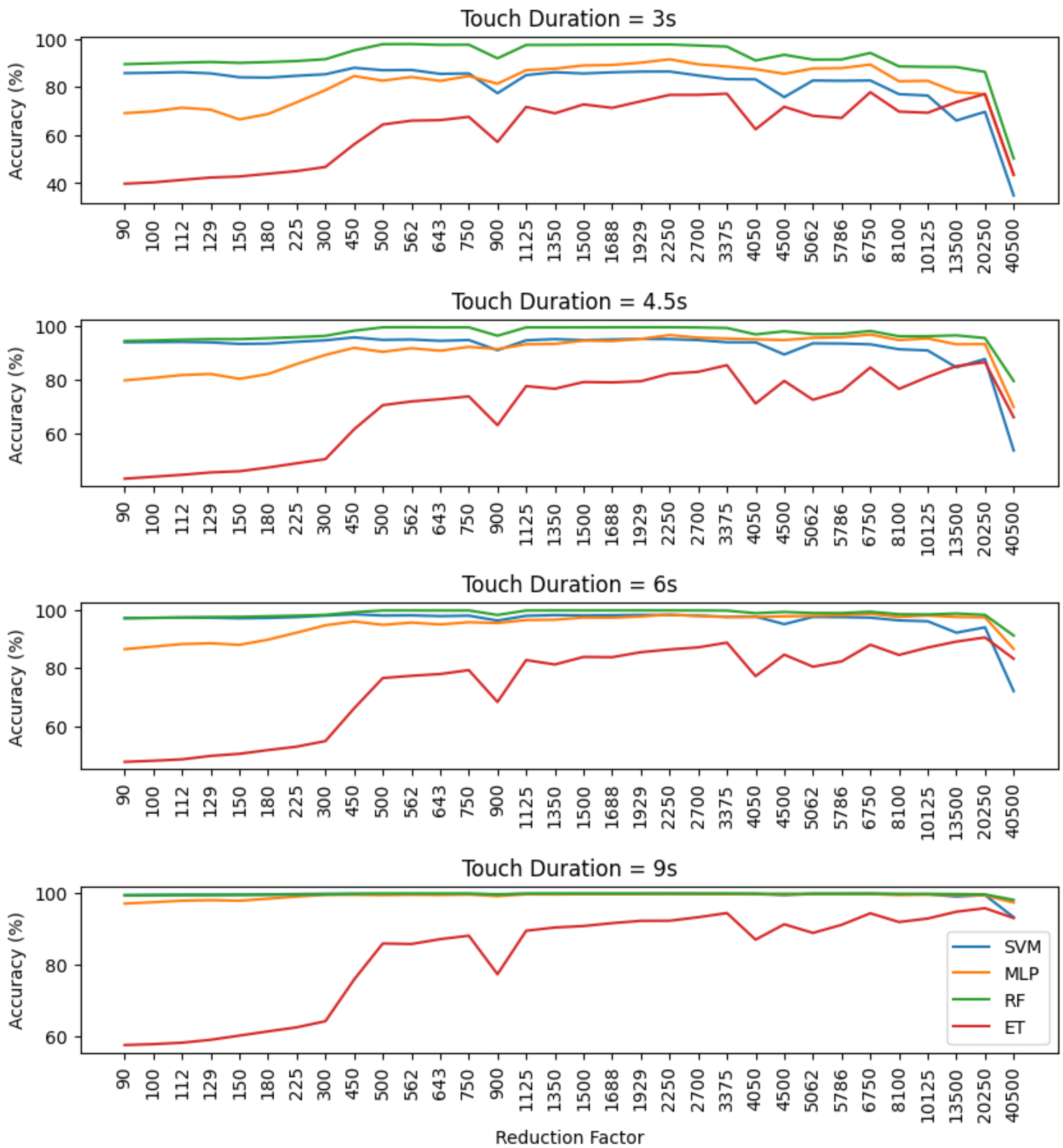

Figure 3.5: The average accuracy of pipeline configurations based on the reduction factors and separated by the texture classifier models and touch duration.

addition, PCA models have the eigenvalue matrix, which shows the importance of each feature for each component. Based on this matrix, we can understand which signal and measurement were more important for the PCA's component in a fusion window.

In order to find the importance of the signals for the classifier (Fig 3.6), first, we checked the importance of the reconstructed signal for the classifier (RF model). Then, based on the eigenvalues of the PCA and the importance of each fusion window for the classifier, the importance of each signal and its measurements were calculated. As Fig 3.6 shows, the beginning and the end of the touch duration were more important than other parts of the touch duration. Also, based on the eigenvalues and the classifier's feature importance matrix, the beginning and end of signals were mostly important to the classifier.

3.4 Discussion

The main objective of this research was to fuse the signals and maintain the most information in the reconstructed signals. Signal fusion not only reduces the number of signals but also makes smaller feature vectors for the classifier, which is more comprehensible. In addition to that, we reduced the touch duration of the tactile robotic hand while the classifiers predicted the texture accurately. We reduced the touch duration from 12 seconds to only 3 seconds, where we had the 98% accuracy and the feature input was reduced 6750 times with respect to the original dimensions. Also, PCA showed a better resutls in compare to ICA, since ICA extract independent features from the source, while the PCA combines the singlas in to high information

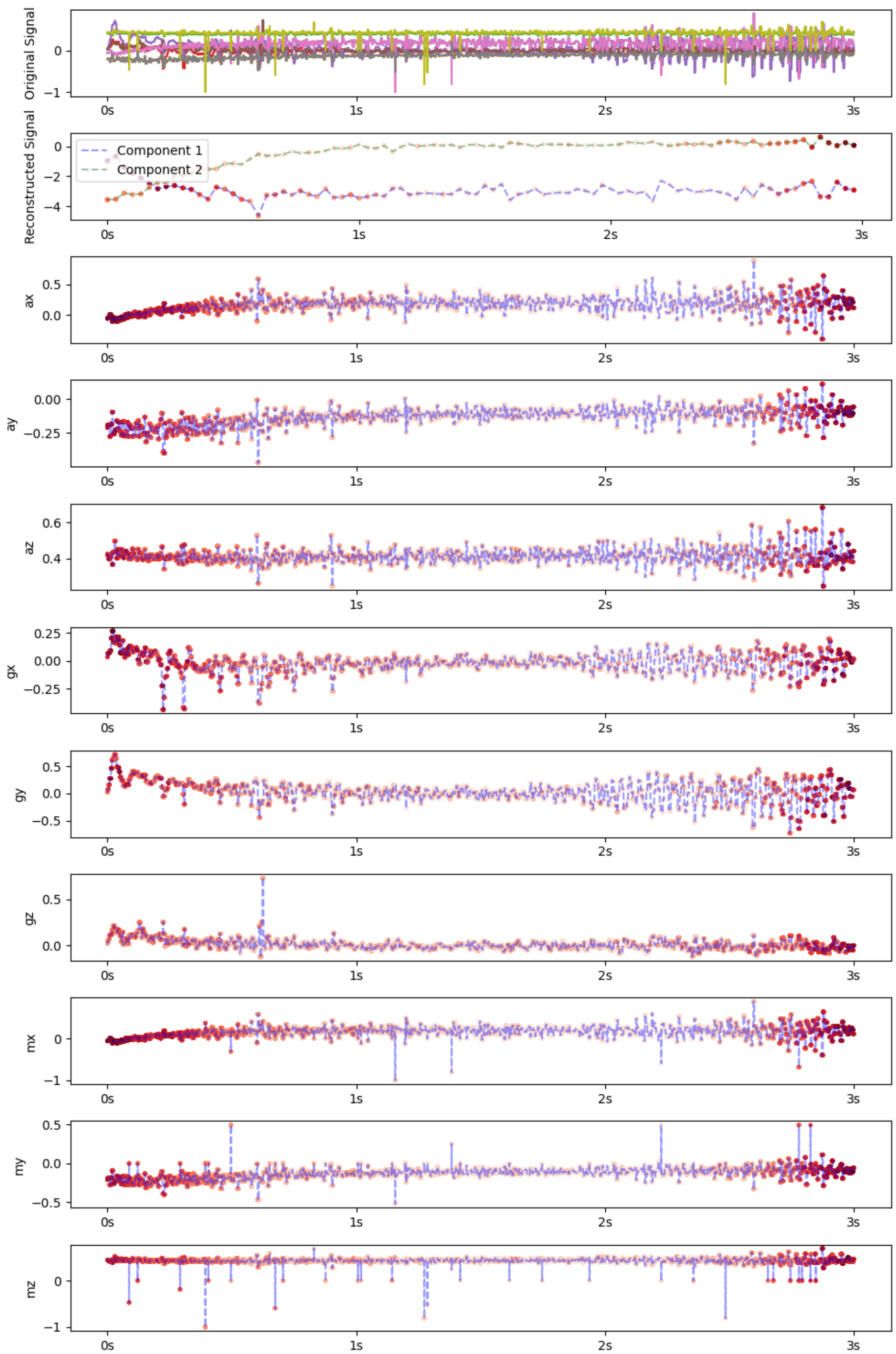

Figure 3.6: The importance of the reconstructed signals and the raw signals. The red areas show the higher importance of that part of the signals.

data points. Based on the results, we find out that it is not necessary to have a long touch duration for the texture classification, but by pre-processing the signals and extracting the high amount of information in fewer features (components), the classifiers work efficiently. However, the dataset only follows one pattern for each texture. If the tactile hand changed the movement direction with a higher frequency, we could extract more information in less time. In the last works [18], they classified the texture with about 100% accuracy, while they used the data of all 12 seconds. Also, they only analyzed one signal for the classification while we used 9 signals and fused them.

In conclusion, our findings suggest that using all of the signals' information and extracting the most information in fewer features can improve the accuracy while we reduce the feature input dimensionality. This method also enables us to reduce the touch duration for classifying the textures.

Chapter 4

Unsupervised Anomaly Detection in Fiber Production Using Signal Fusion

4.1 Introduction

The production of synthetic fibers involves the creation of materials derived from polymers sourced from petroleum or natural gas. Fundamentally, the procedure entails extruding a molten polymer through a small opening, which solidifies into fibrous filaments via cooling methods, followed by intricate post-processing to satisfy particular properties desired by end users. Depending on the manufactured product, several adaptations and customizations to this general procedure are applied. Maintaining product quality, eradicating defective fibers, and minimizing production downtime are crucial for industry stakeholders. Fiber defects can lead to expensive customer claims; thus, minimizing these claims is vital to sustained success within this market.

To support fiber manufacturers in achieving their objectives, Instrumar Limited has created a polymer fiber monitoring system [46], known as the Instrumar Fiber System (IFS). This system employs electromagnetic sensors engineered by Instrumar Limited to observe the properties of polymer fibers during production. Each sensor records the fiber's physical characteristics and produces a continuous time-series data set. The data is subsequently evaluated to identify patterns indicative of physical defects in the fiber or production process. Currently, IFS depends on manually identifying data patterns that indicate physical problems, followed by searching for these patterns within the data. As this approach requires extensive manual labor and is prone to errors, implementing an automated data evaluation and fault detection procedure will improve efficiency and reduce costs.

Detecting and categorizing faults within time-series data captured by sensors in the manufacturing of synthetic fibers presents distinct challenges. Primarily, most of the dataset reflects standard operational conditions, with only a small portion indicative of defective items that require detection. In addition, each type of defect has a unique signature. Furthermore, while some data anomalies remain unclassified, they frequently do not cause defects and are, therefore, irrelevant to customers.

Effective anomaly detection is critical for ensuring product consistency and minimizing operational disruptions. To address this need, this chapter explores various methods for detecting anomalies in sensor data collected from thread lines.

The primary signals analyzed (i.e., Magnitude, Phase, Node Quality, and Node Count) each capture unique aspects of thread-line behavior, allowing for a comprehensive view of operational stability. However, due to these signals' high-dimensional and noisy nature, traditional anomaly detection methods may struggle to differentiate between genuine faults and minor fluctuations. This chapter presents a systematic approach to anomaly detection, leveraging statistical analysis, fusion techniques, and dimensional reduction to improve detection accuracy.

We utilize a non-overlapping sliding window method to segment the signal data and apply three distinct detection techniques: a rolling median approach on individual signals, averaging across signals within each window, and Principal Component Analysis (PCA) to reduce dimensionality. This study compares these methods to identify the most effective technique for real-time anomaly detection in fiber production lines. The chapter is structured to discuss the dataset and signal processing techniques first, followed by detailed explanations of each anomaly detection method, and concludes with a performance evaluation of these methods.

4.2 Material and Methods

This section presents a methodology for detecting anomalies in industrial fiber production lines using time series data from Instrumar's fiber sensors. These sensors

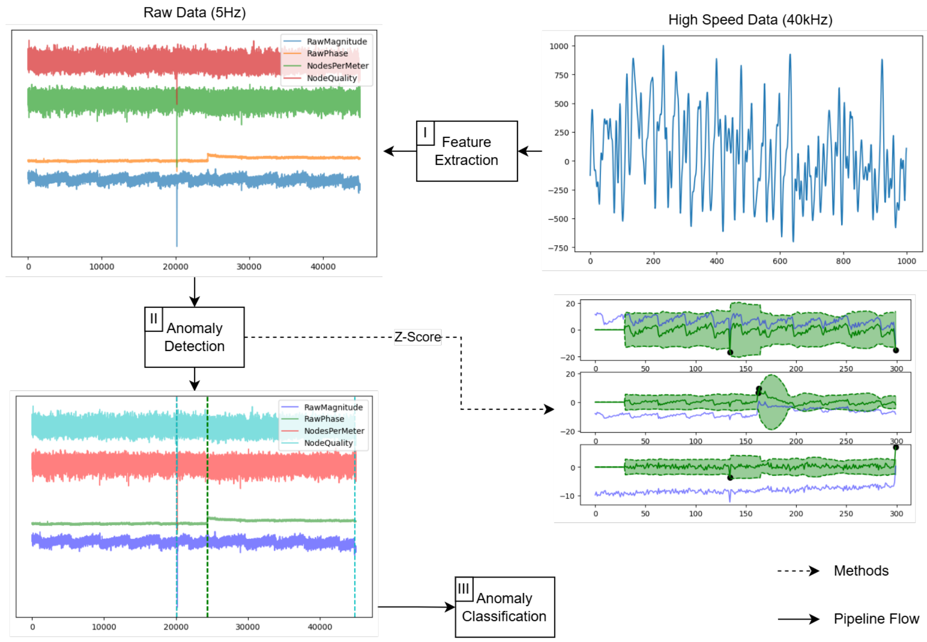

Figure 4.1: An overview of the pipeline that Instrumar uses for fault detection and classification

measure fiber characteristics, including magnitude, phase, node quality, and node count, which are processed every 200 ms to monitor production conditions. Given the complexity of this data, three distinct anomaly detection techniques are applied. Figure 4.1 illustrates the workflow of the Instrumar new fault detection and classification system and the methods utilized in this study.

The first technique utilizes a rolling median in the raw signal approach, independently detecting anomalies by calculating z-scores over 30-second intervals for each signal component. The second approach averages the four signals in each window, producing a single fused value that reflects collective behavior. Anomalies are flagged based on significant deviations from this average. Finally, Principal Component Analysis (PCA) is used to reduce the dimensionality of the signals, transforming them into two or three principal components, which are then analyzed for outliers using z-scores [47].

Each method operates on non-overlapping 30-second windows, using a threshold z-score of 5 to identify anomalous events. This study aims to compare the methods for accurately detecting production anomalies, improving fault classification in fiber manufacturing, and reducing reliance on manual, heuristic-based detection methods.

4.2.1 Dataset and Signals

In this section, we explore the critical basis of our research, utilizing actual factory data provided by our industrial collaborator, Instrumar Limited. The data set in the form of a time series used in this investigation is sourced from Instrumar's exclusive sensors, depicted in Figure 4.1, which are specially designed to oversee the production of industrial fibers.

The Instrumar Fiber Sensor Unit (SU), positioned on a manufacturing line, assesses the electrical impedance of the fiber as it moves through an electromagnetic field. Essentially, the sensor responds to the electrical characteristics and geometry of the fiber. The sensor captures the electrical impedance response signal between 20 kHz and 40 kHz (High-Speed Data in Figure 4.1). Subsequently, this information is processed by Instrumar's Sensor Processing Unit (SPU, Figure 4.1 i), which computes and delivers four fiber properties every 200 ms: magnitude, phase, node quality, and node count (Raw Signals in Figure 4.1). The magnitude is associated with the denier or the fiber's density. Interruptions in the electrical response signal caused by nodes in the fiber contribute to a lower reading. The frequency of these interruptions represents the node count, whereas their amplitude indicates node quality. The phase indicates the time delay in the response signal, being sensitive to conductivity, which relates to the quantity of finish applied to the fiber.

The fiber properties mentioned above are assessed every 200ms, and this data is used to identify fiber defects and deduce the process anomalies that might have led to these defects. These properties function as the input for Instrumar's existing system and are crucial resources for advancing our research on fault classification in an industrial setting.

Instrumar's current data analysis system associates data with established patterns characteristic of fiber defects or physical occurrences. Instrumar engineers recognize these data patterns and develop heuristic algorithms to detect them, which are called custom fault alarms. This method is typically labor-intensive and prone to errors. Moreover, configuring the parameters for these algorithms poses challenges. They frequently require adjustments to reduce false positives or capture overlooked events based on customer input. Additionally, they fail to recognize large data fluctuations from previously unknown or unidentified patterns, which might result in non-compliant products being incorrectly classified as usual, potentially leading to expensive customer claims and significant financial losses.

This system uses a more robust two-stage approach. Initially, the data undergo unsupervised algorithms to detect anomalies (Figure 4.1 ii). The main goal of this stage is to only detect the the anomalies. After that, the anomalies will be classified in the further stages. Also in some cases, the products will be graded based on the number of anomalies detected in the product. The low graded ones will be recycled while the high graded products will be used in a sensitive situation. These anomalies are then classified into specific faults using a multiclass classification algorithm (Figure 4.1 iii), which are reported to the customer for further action [48]. Direct processing of raw time-series data through these algorithms is not feasible. Instead, data are segmented into time windows and features are extracted for each fiber property within those windows. The extracted features include the mean, median, standard deviation, variance, chi-square, 25th and 75th percentiles, minimum and maximum values, kurtosis, skewness, and stability (defined as the current fiber property level over the average of the last 24 hours), which is 12 features per fiber property. With four fiber properties, this results in 48 features in total. Together, these features form a data instance for each window.

Our primary goal in this research is to enhance anomaly detection accuracy through signal fusion, leveraging the unique characteristics of each fiber property captured by Instrumar's sensors. To achieve this, we have compared and evaluated two fusion methods, each aimed at more effectively identifying anomalies by integrating signals. Additionally, we have thoroughly assessed the sensitivity of these methods to each signal, ensuring that our approach optimally utilizes the distinct information provided by each fiber property. Our work aims to address limitations in current systems by improving the detection of anomalies. It will ultimately provide a more robust and reliable solution for fault classification in industrial fiber production.

4.2.2 Anomaly Detection

Sensor data from each thread line in fiber production lines detect anomalies that may indicate faults or irregularities in the manufacturing process. Each signal reflects different characteristics of the thread line's behavior, providing a comprehensive view of its operational state. Anomalies in any of these signals could indicate potential problems in the thread line, which requires the development of effective detection methods.

A non-overlapping sliding window approach was used to segment the signal data into manageable intervals for anomaly detection. Each window consists of 150 data points, corresponding to a 30-second span of signal readings. This window length was selected to capture short-term variations in the signal while maintaining computational efficiency. Within each window, statistical measures such as mean, standard deviation, and z-scores ($z = \frac{x-\mu}{\sigma}$) were calculated to assess deviations that could indicate anomalous behavior.

We employed three different methods for anomaly detection, each with a distinct approach to processing the signal data. The first method used raw signals with a rolling median calculation, focusing on individual signal components without any fusion, allowing direct anomaly detection within each component (Figure 4.1 ii). The second method involved averaging the four signal components within each window, producing a representative value per window to identify the overall deviations (Figure 4.1 ii). The third method applied Principal Component Analysis (PCA) to the signals, reducing them to three primary components and capturing the main variations in the signals before analyzing for anomalies (Figure 4.1 ii). Each method used the sliding 30-second window to isolate significant deviations based on z-scores[47], which were calculated to provide a consistent threshold in all techniques. An anomaly was identified if the calculated z-score surpassed a threshold of 5, a value chosen to capture only the most significant deviations and reduce false positives.

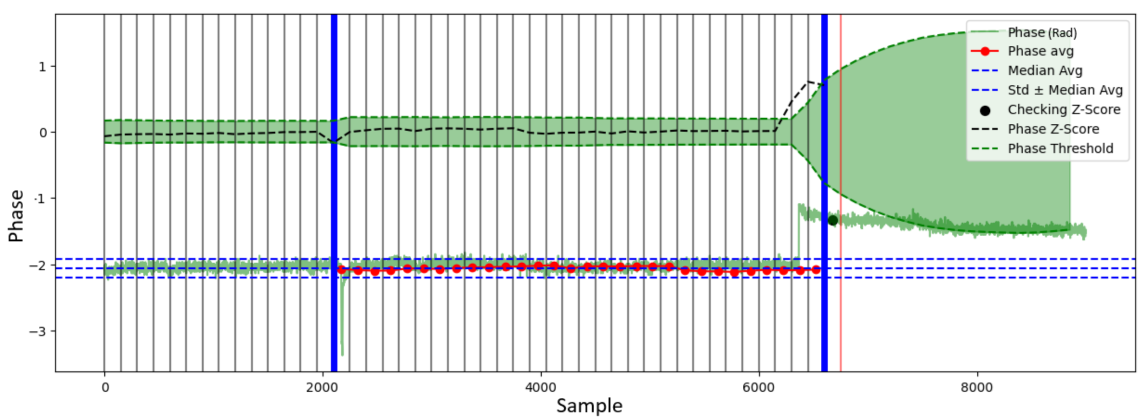

Figure 4.2: A demonstration of how rolling median works for Phase signal with the rolling median window size of 30.

Based on a rolling median calculation, the first method isolates anomalies by examining the median trend within each signal component over time. Specifically, the median of each signal component was calculated across the previous 30 windows, followed by computing the standard deviation of these average values across the same 30 windows. Figure 4.2 illustrates the procedure for calculating the z-score for each window. In this Figure, the rightmost window z-score value (black dot) is derived from the prior 30 windows (located between the two vertical blue lines). We define μ as the median of the averages of these 30 windows (red dots) and σ as the standard deviation of these averages (dashed blue lines). Subsequently, the z-score (dashed black line) for the new window is determined with x representing the new window's average, and this method is repeated for all 4 signals. This approach enables an adaptive threshold that reflects local changes in each signal component. If the z-score exceeded the threshold of 5 (green area in Figure 4.2), the data point was marked as an anomaly. Using the median as the central metric, this method minimizes the impact of outliers and captures the overall trend in each signal component. Figure 4.3 shows the anomalies captured by this method.

The second approach used an averaging-based fusion of the four signal components within each window. Here, the four signals, Magnitude, Amplitude, Phase, and Node Count, were fused by calculating their average over the 150 data points in each window. This produced a single representative value for each window that captures the overall behavior of the signals during that period.

Figure 4.4 illustrates the average fusion technique. Initially, the signals are seg-

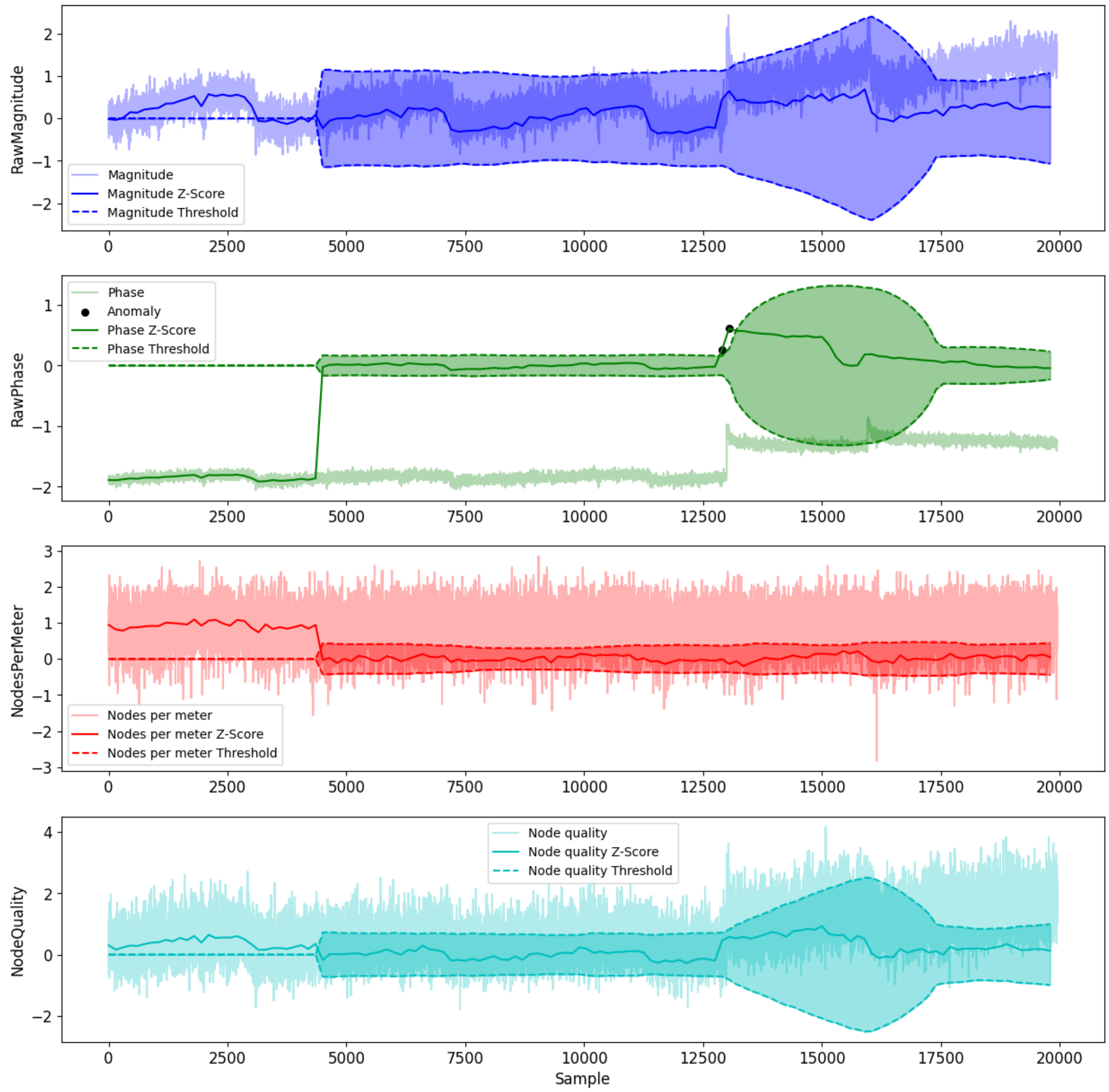

Figure 4.3: The detected Anomalies in 4 signals using rolling median algorithm

mented into 30-second windows, and then the mean of all four signals in each window is computed. Once all the windows are fused, their $150 * 4$ feature size is reduced to a single point. Connecting these points creates a fused signal using the averaging method.

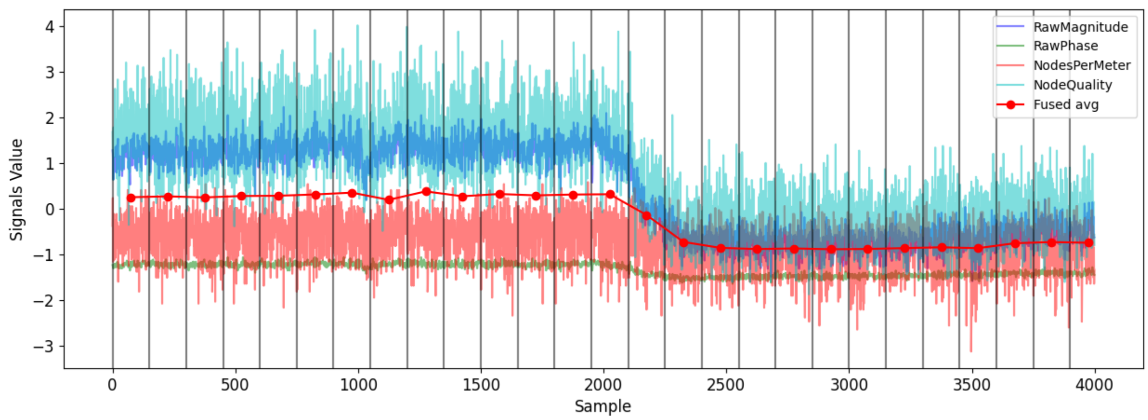

Figure 4.4: Fusing 4 signals using their average in non overlapping 150 window size

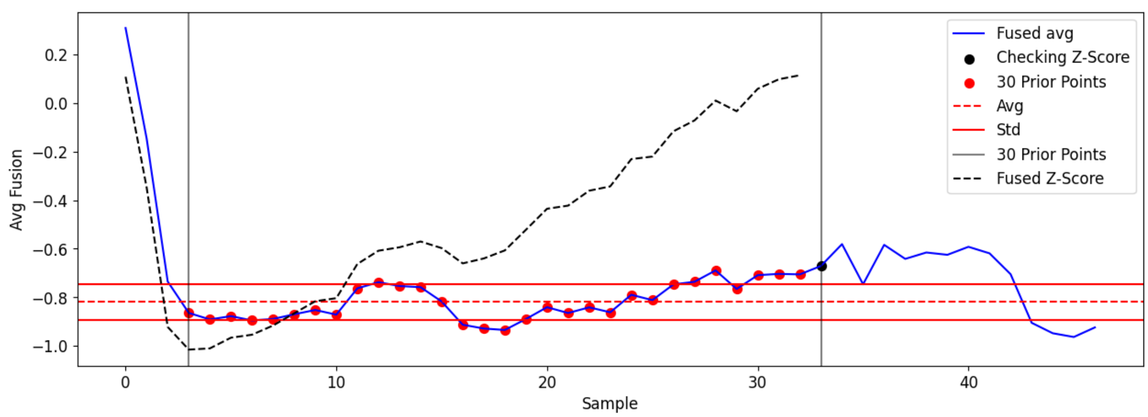

Figure 4.5: Finding Z-score for each fused point based using last 30 points

For anomaly detection, the average of the fused values and the standard deviation of these points were computed across the previous 30 points. Figure 4.5 shows an example of calculating the z-score for the black dot in the fused signal. As the figure demonstrates, the mean and standard deviation of 30 points before the black point are calculated as μ and σ of the z-score formula. A z-score that exceeded the threshold of 5 indicated an anomaly, suggesting a deviation in the fused signal's behavior from typical patterns. Using an average-based fusion, this method reflects the collective behavior of the four signals while simplifying the data's dimensionality.

Figure 4.6 presents the anomalies identified using the z-score applied to the averaged fused signal. It is important to observe that this figure adopts the same time interval as the first method (rolling median z-score). Certain anomalies remain undetected by this approach.

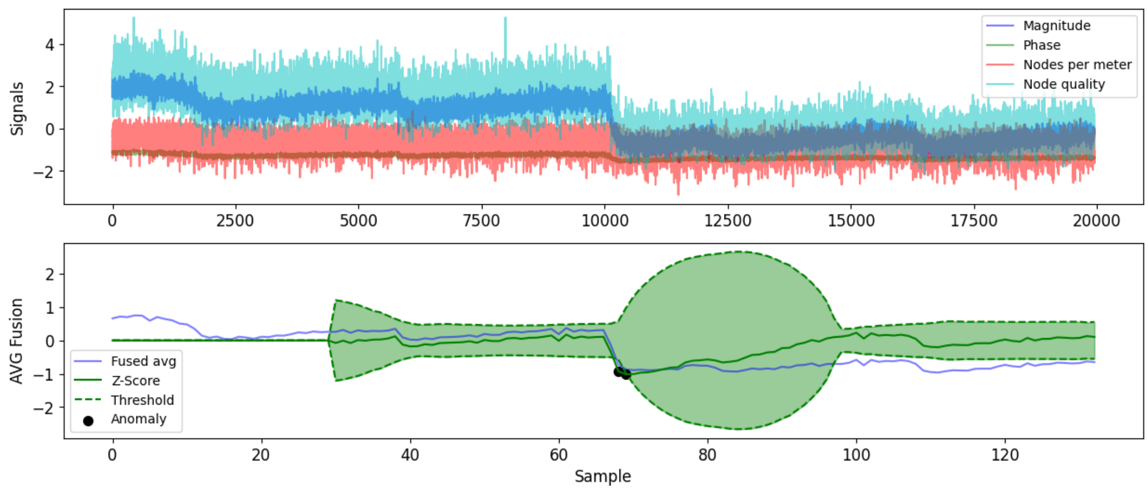

Figure 4.6: In the first figure, raw signals are displayed within a specific time range, while the second plot presents the fused signal derived from these raw signals, along with the detected anomalies using this fused signal.

The third and final method utilized Principal Component Analysis (PCA) for signal fusion within each window, transforming the four signals into a lower-dimensional space. Specifically, PCA was applied to reduce the 150×4 data points in each window to three principal components, capturing the most significant signal variance (Figure

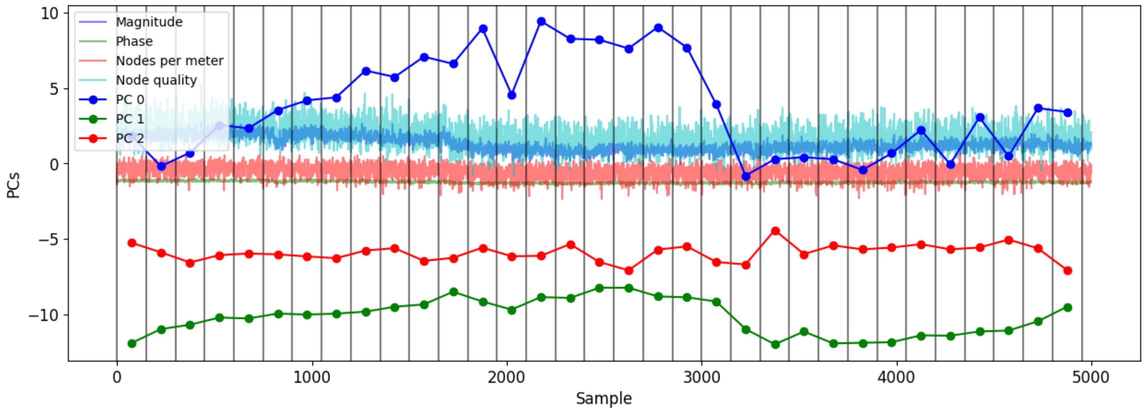

Figure 4.7: The PCA transformation of the signals using 3 components and 150 window size.

4.7). The transformed data in the principal component space was then analyzed for anomalies. As with the previous methods, the average and standard deviation of the PCA-transformed values were computed across the last 30 windows, along with their.

The z-score for each transformed data point was calculated based on this average and standard deviation, with anomalies flagged when the z-score surpassed 5 (Figure 4.8). This method provides an alternative fusion technique, extracting a high percentage of information and emphasizing primary variations across the signals, most likely to indicate abnormal events.

As previously stated, this dataset comprises 4 thread lines. To ensure effective PCA performance within this context, we trained the PCA using one thread line and applied both PCA transformation and z-score testing across the remaining thread lines. Figures 4.9 and 4.10 illustrate the anomalies identified by applying z-score to the PCA-fused signal. These figures demonstrate that anomalies were detected using

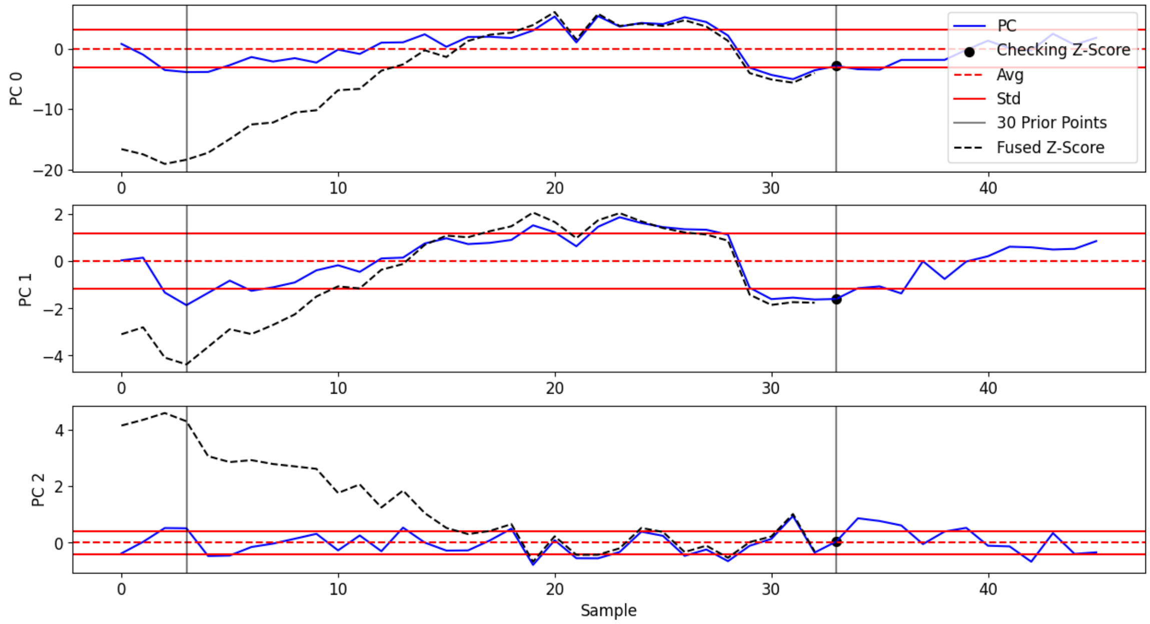

Figure 4.8: Finding z-score of each point and PC separately

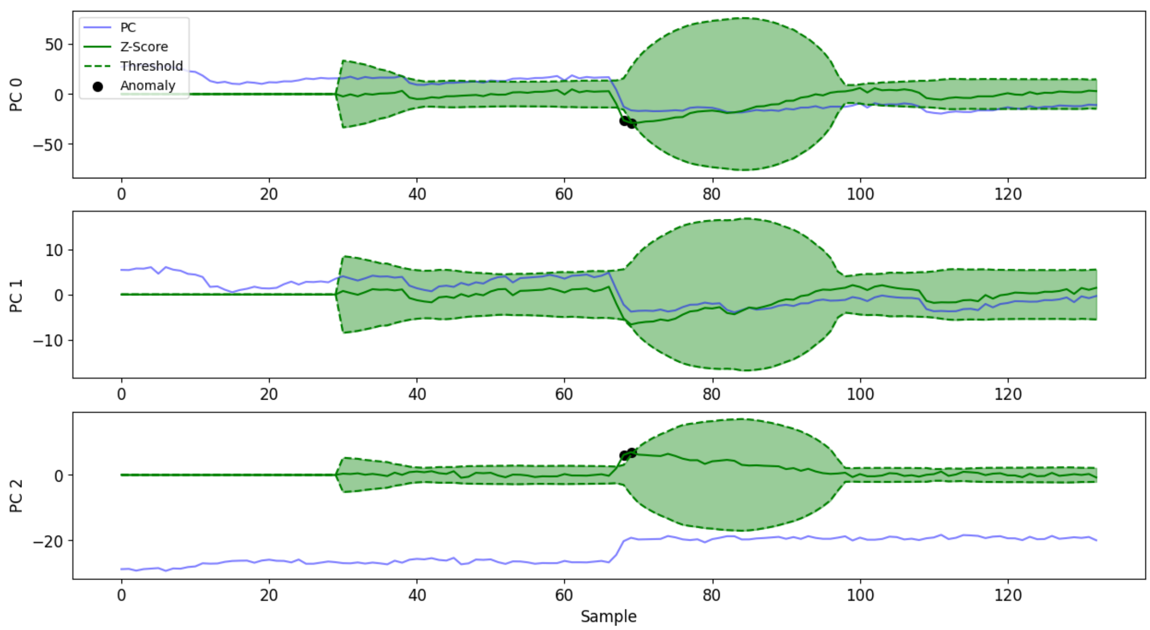

Figure 4.9: Anomalies detected in each fused component

the raw signal method (first method).

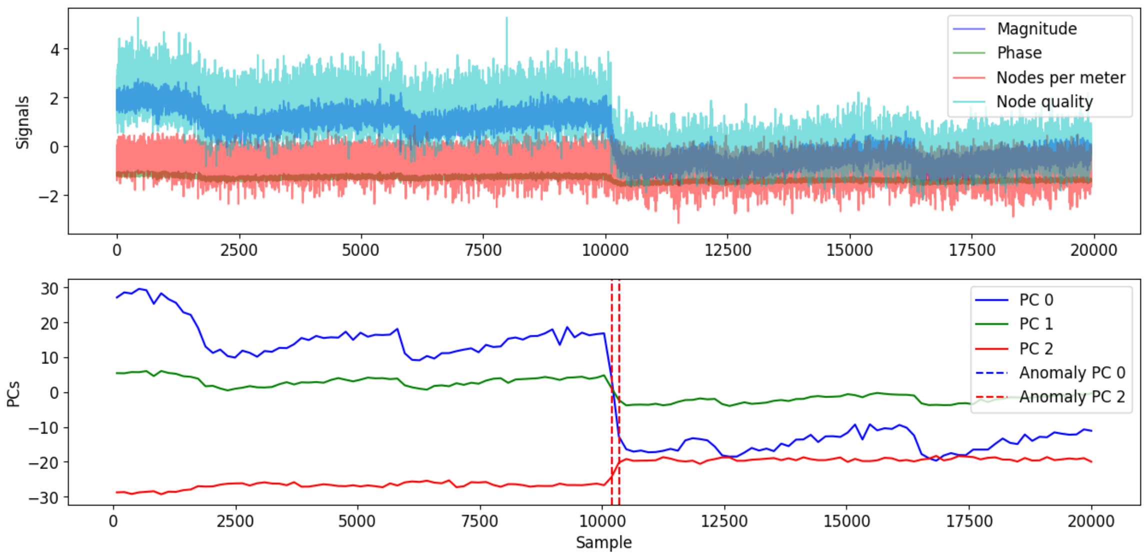

Figure 4.10: Anomalies detected in fused signal in respect to the raw signals

4.3 Results

The anomaly detection methods were tested on two variations of data configurations, each designed to evaluate the impact of specific signal combinations on detection accuracy. The first configuration included the four signal components: Magnitude, Node Quality, Phase, and Node Count. This complete dataset comprehensively represented thread line behavior, capturing all aspects of signal fluctuations across each component. The second configuration limited the analysis to only the Node Quality and Node Count signals. This reduced dataset focused on a subset of signals that are more directly correlated with structural integrity, allowing an examination of whether fewer targeted signals could still yield effective anomaly detection results. These two data configurations enabled a comparative analysis of how signal combinations influence anomaly detection capabilities across thread lines.

Figure 4.11: Fused signals with different PCA components in two data configuration

Principal Component Analysis (PCA) was employed as one of the fusion techniques, transforming the signals into their principal components. We examined the fused signals for each principal component to determine the optimal number of PCA components for each data configuration. Figure 4.11 presents these signals for each component individually. The graphs indicate that the first three components of the first data configuration and the first two components of the second data configuration highlight anomalies and signal trends. Furthermore, Figure 4.12 reveals that the numbers of components in each configuration capture approximately 60% of the information across both configurations. This selection process allowed for a dimensional reduction that preserved essential signal characteristics while minimizing redundancy, facilitating effective anomaly detection with minimal data loss.

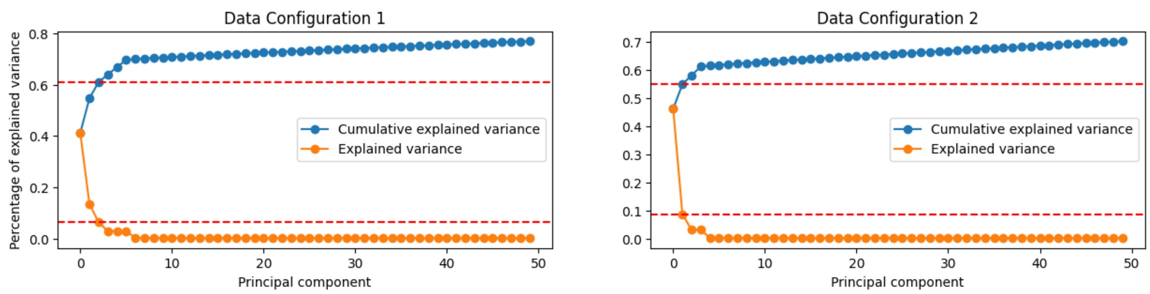

Figure 4.12: Scree plot of the PCA components

The experiments were carried out on three separate thread lines on which PCA was not trained, with anomaly detection results recorded for each method and each data configuration. Table 4.1 and the bar graph in Figure 4.13 demonstrate these records. The rolling median, PCA-based fusion, and averaging fusion methods were applied for both data types, and the number of anomalies detected by each method was logged. Across the three thread lines, notable differences emerged in the anomaly counts reported by each method. Also, the raw signal method occasionally flagged anomalies in regions where PCA-based fusion did not. The results for the averaging fusion method were less consistent, with a lower overall detection rate across both data types. This inconsistency suggests that averaging may obscure subtle variations in the signal, leading to missed anomalies.

Thread Line #	Conf #	Methods		
		Raw	AVG	PCA
2	1	1090	103	421
	2	301	146	290
3	1	1983	390	932
	2	782	402	633
4	1	2170	440	999
	2	773	417	614
Sum	1	5243	933	2352
	2	1856	965	1537

Table 4.1: Anomaly Counts by Detection Method and Configuration

In our analysis, as illustrated in Figure 4.13, we observed a significant inconsistency between data configurations 1 and 2 when using the raw signal method. This high contrast highlights the sensitivity of the raw signal approach to variations in signals in the data set. However, this substantial performance gap is notably reduced when the PCA method is applied. The PCA technique effectively smooths out the

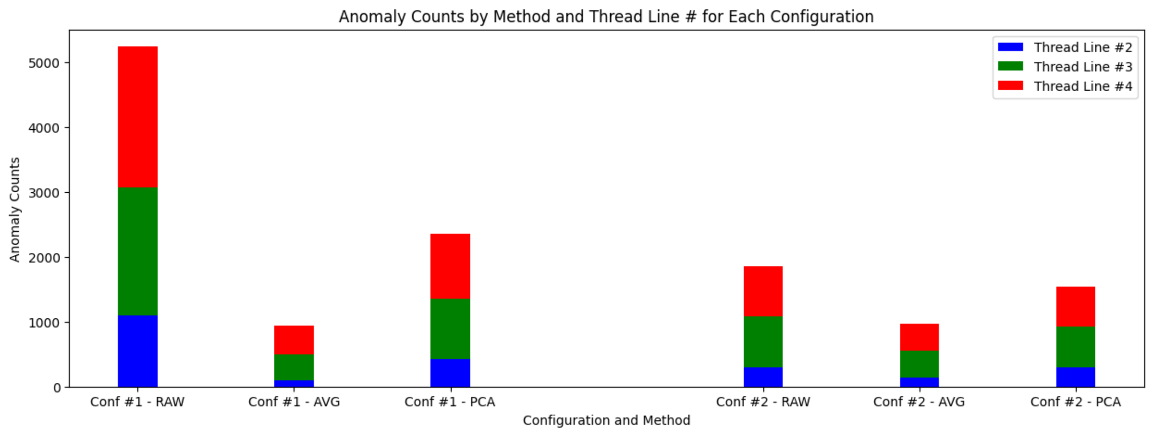

Figure 4.13: Number of anomalies detected in different data configuration and methods

differences, indicating a more robust handling of the underlying data structure, since there is a complex co-relation between signals.

We examined each signal individually to investigate further the anomalies detected by the raw signals. This analysis allowed us to pinpoint specific anomalies. Subsequently, we reconstructed signals within a defined range surrounding each anomaly identified by the PCA method. This reconstruction process was instrumental in determining which individual signals contributed to the anomalies detected in the fused signals.

Figure 4.14 further emphasizes our findings: most anomalies detected through raw signals originated from the phase signal, which exhibited increased sensitivity to this measurement. In contrast, the PCA fusion approach showed a more balanced sensitivity across all signals, indicating a more uniform capability in anomaly detection.

Another critical distinction between the two methods lies in the anomaly detection

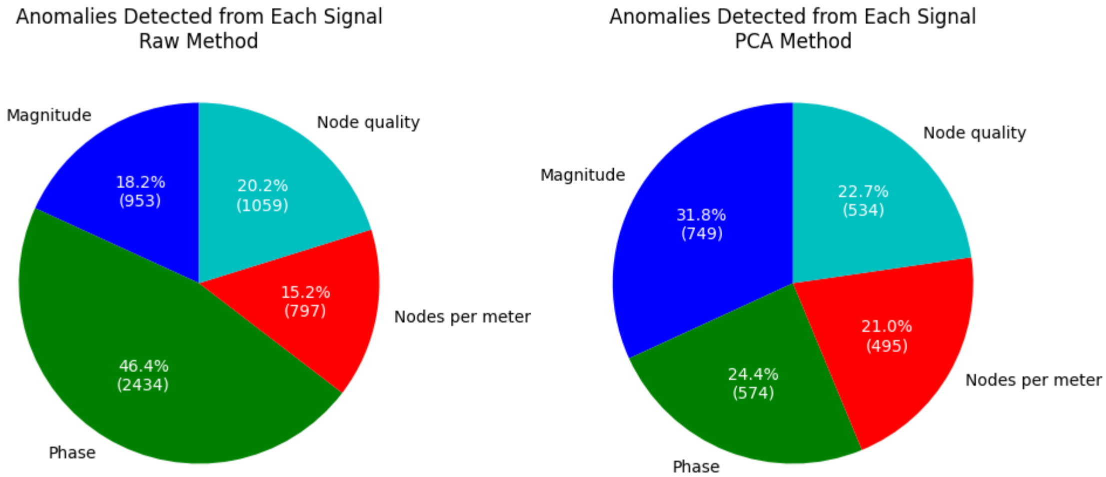

Figure 4.14: Comparison between signal sensitivity in raw signal and PCA fusion method

process. When using the raw signals method, the threshold for each signal had to be manually adjusted due to the inherent noise and ramp variations present in the individual signals. This manual adjustment added complexity to the analysis and introduced potential inconsistencies. However, the PCA method alleviates this concern by extracting a comprehensive representation of the information contained within all signals. Applying a z-score on the aggregated data eliminates the need for threshold modifications, leading to a more streamlined and efficient anomaly detection process.

In addition, the results of the averaging-based fusion method revealed limitations in its effectiveness for anomaly detection. Unlike the PCA-based method, which retains the most informative variance in the data, averaging tends to smooth out individual signal differences. This averaging effect was particularly evident in the reduced detection rate observed for both data configurations, as the averaging reduced the influence of significant deviations within specific signal components. Consequently, the averaging-based fusion method demonstrated a lower sensitivity to anomalies, frequently failing to detect instances identified by either the raw signal or PCA-based methods. This outcome indicates that simple averaging is inadequate for capturing the complex, multivariate patterns present in the thread line data, as it cannot isolate significant deviations that occur across multiple dimensions.

4.4 Conclusions and Discussion

The results of this study underscore the impact of signal fusion techniques on anomaly detection in complex multivariate data, particularly within the context of thread line monitoring. Comparative analysis of raw signal processing, PCA-based fusion, and averaging techniques has provided valuable information on each approach's strengths and limitations.

Despite its straightforward nature, the raw signal technique demonstrated significant noise and fluctuation sensitivity, requiring manual threshold calibrations to efficiently detect anomalies. This limitation highlights the challenge of dealing with high-dimensional, noisy data when relying on unprocessed signals. Although the raw signal method occasionally captured unique anomalies that PCA fusion did not, it lacked the overall robustness and consistency of the PCA method. This inconsistency likely results from the varying sensitivities across individual signals, making it difficult to standardize anomaly detection thresholds without introducing biases.

On the contrary, PCA-based fusion demonstrated superior performance by reducing dimensionality and capturing essential variations between multiple signals. By extracting principal components that retain approximately 60% of the variance of the data, PCA effectively highlighted meaningful trends and anomalies with fewer components, offering both computational efficiency and interpretability. This technique smoothed out extraneous fluctuations and focused on significant deviations, allowing a more uniform detection capability across different signal components. Moreover, the automated nature of the z-score thresholding in PCA fusion removed the need for manual threshold tuning, streamlining the detection process. These findings suggest that PCA's dimensional reduction and feature extraction capabilities are well-suited for anomaly detection applications where multivariate data integration and interpretability are crucial.

The fusion technique based on averaging missed a significant number of anomalies identified in two other methods. This indicates that averaging for fusing signals is not effective. The tendency of averaging to smooth out variations reduced its sensitivity, as it obscured subtle deviations within individual signal components. The findings confirm that while averaging may be computationally less intensive, it fails to capture the nuances necessary for reliable anomaly detection in thread line monitoring.

This study demonstrates that PCA-based fusion offers a balanced approach to anomaly detection in multivariate sensor data, achieving an optimal trade-off between sensitivity, consistency, and ease of use. The success of this method reinforces PCA's utility not only in dimensionality reduction but also as a robust tool for anomaly detection, capable of handling high-dimensional data without excessive noise sensitivity. Future work could further explore adaptive thresholding mechanisms or hybrid models that integrate PCA with complementary techniques to enhance anomaly detection in diverse sensor-driven applications. In addition to that, it is important to analyze the significant of the anomalies detected in the phase signals in raw method which is dominant in anomaly detection.

Chapter 5

Conclusion and Future Works

In this thesis, we investigated the use of signal fusion techniques for anomaly detection and texture classification in complex multivariate datasets, particularly in the context of sensor data from fiber monitoring and tactile robotic hands. Our research spanned two main areas: developing effective methods for identifying anomalies in multicomponent sensor signals, and exploring signal fusion approaches to enhance texture classification accuracy in tactile robotics.

In our anomaly detection research, we analyzed three primary techniques: raw signal analysis, principal component analysis (PCA)-based fusion, and averaging-based fusion. Each approach yielded distinct insights:

The raw signal analysis method captured unique anomalies within individual signals but was prone to noise and required manual thresholding adjustments, which introduced inconsistency and reduced practicality. PCA-based fusion was most effective in capturing key variances across signals and demonstrated robustness against noise, reducing the need for manual tuning. This method balanced anomaly detection sensitivity with dimensionality reduction, enhancing interpretability and detection accuracy. Averaging-based fusion simplified the signal data but resulted in a reduced sensitivity to subtle anomalies, which limited its effectiveness in identifying complex, multivariate deviations in the data. For texture classification in tactile robotics, we examined the efficacy of signal fusion to improve classification accuracy and interpretability. Our approach aimed to leverage fused signals to capture rich textural patterns, which individual signals alone could not effectively represent. This study demonstrated that signal fusion could produce significant improvements in classification accuracy and robustness, underscoring its potential to inform real-world applications in robotic sensing. tactile.

One of the challenges identified in anomaly detection is the lack if true labels for anomalies, which are essential for evaluating and comparing the performance of different detection methods. The lack of labeled data makes it harder to quantify the accuracy and robustness of the proposed techniques and can hinder the development of more effective anomaly detection systems. In the context of texture classification, a key limitation lies in the static exploration path (square path) used by the tactile robotic hand. This fixed trajectory may limit the richness of the data collected. By introducing dynamic or more frequent changes in the direction of exploration, it may be possible to capture additional information about the texture, potentially enhancing classification accuracy and enabling a more comprehensive understanding of surface properties.

Building on the findings of both chapters, several areas are recommended for future research and development. Enhancing thresholding mechanisms through adaptive methods could improve anomaly detection accuracy, reducing reliance on fixed z-scores and improving adaptability to dynamic signal variations. Additionally, exploring hybrid models that integrate PCA with machine learning approaches, such as clustering or deep learning-based anomaly detection, could capture a wider range of anomaly patterns. Implementing real-time, low-latency anomaly detection systems is essential for operational environments like thread line monitoring. Efficient, optimized algorithms for PCA or other dimensionality reduction methods could enable live anomaly feedback with immediate corrective actions, enhancing the system's value in industrial applications.

Applying interpretable deep learning approaches, such as convolutional neural networks (CNNs) and transformers, to fused tactile signals could enhance texture classification accuracy and provide a richer understanding of textural characteristics in robotic applications. Such models would allow for fine-grained classification that adapts to complex sensory patterns, capturing subtle textural variations more effectively.

Expanding these signal fusion techniques to a broader set of tactile robotics applications, such as object recognition or material identification, would validate the generalizability of our approach. Testing in diverse robotic environments with different sensor configurations would provide insight into the scalability and robustness of signal fusion methods.

This thesis has demonstrated that signal fusion, when applied thoughtfully, can significantly enhance both anomaly detection and texture classification in multivariate sensor data. As robotic sensing and industrial monitoring continue to demand higher precision, the findings presented here lay the groundwork for more adaptive, interpretable, and scalable solutions in sensor-based anomaly detection and classification tasks.

Bibliography

[1] Karthik Allam. Data-driven dynamics: Unraveling the potential of smart robotics in the age of big data. *EPH-International Journal of Applied Science*, 9(2):18–22, 2023.

[2] Qiong Cai, Hao Wang, Zhenmin Li, and Xiao Liu. A survey on multimodal data-driven smart healthcare systems: approaches and applications. *IEEE Access*, 7:133583–133599, 2019.

[3] Fei Tao, Qinglin Qi, Ang Liu, and Andrew Kusiak. Data-driven smart manufacturing. *Journal of Manufacturing Systems*, 48:157–169, 2018.

[4] RAHUL SINGH Gautam and VENKATA MRUDULA Bhimavarapu. Data driven decision making: Application in finance. *Iconic Research and Engineering Journals*, 5(12):52–56, 2022.

[5] Neta Rabin, Maayan Kahlon, Sarit Malayev, and Anat Ratnovsky. Classification of human hand movements based on emg signals using nonlinear dimensional-

ity reduction and data fusion techniques. *Expert Systems with Applications*, 149:113281, 2020.

[6] Xiaobin Xu, Lei Zhang, Jian Yang, Chenfei Cao, Wen Wang, Yingying Ran, Zhiying Tan, and Minzhou Luo. A review of multi-sensor fusion slam systems based on 3d lidar. *Remote Sensing*, 14(12):2835, 2022.

[7] Shubhojeet Chatterjee, Rini Smita Thakur, Ram Narayan Yadav, Lalita Gupta, and Deepak Kumar Raghuvanshi. Review of noise removal techniques in ecg signals. *IET Signal Processing*, 14(9):569–590, 2020.

[8] P Mirajkar Pradnya and D Ruikar Sachin. Wavelet based image fusion techniques. In *2013 International Conference on Intelligent Systems and Signal Processing (ISSP)*, pages 77–81. IEEE, 2013.

[9] Yuxing Li, Xiao Chen, Jing Yu, and Xiaohui Yang. A fusion frequency feature extraction method for underwater acoustic signal based on variational mode decomposition, duffing chaotic oscillator and a kind of permutation entropy. *Electronics*, 8(1):61, 2019.

[10] Nico Kaempchen and Klaus Dietmayer. Data synchronization strategies for multi-sensor fusion. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, volume 85, pages 1–9, 2003.

[11] Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. Selective sensor fusion for neural visual-inertial

odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10542–10551, 2019.

[12] Duy Tang Hoang and Hee Jun Kang. A motor current signal-based bearing fault diagnosis using deep learning and information fusion. *IEEE Transactions on Instrumentation and Measurement*, 69(6):3325–3333, 2019.

[13] Hamid Reza Shahdoosti and Hassan Ghassemian. Combining the spectral pca and spatial pca fusion methods by an optimal filter. *Information Fusion*, 27:150–160, 2016.

[14] Lihong Chang, Xiangchu Feng, Xiaoping Li, and Rui Zhang. A fusion estimation method based on fractional fourier transform. *Digital Signal Processing*, 59:66–75, 2016.

[15] Mustafa Kuntoğlu and Hacı Sağlam. Investigation of signal behaviors for sensor fusion with tool condition monitoring system in turning. *Measurement*, 173:108582, 2021.

[16] Pinki Kumari and Abhishek Vaish. Feature-level fusion of mental task's brain signal for an efficient identification system. *Neural Computing and Applications*, 27:659–669, 2016.

[17] Raffaele Gravina, Parastoo Alinia, Hassan Ghasemzadeh, and Giancarlo Fortino. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Information Fusion*, 35:68–80, 2017.

[18] Bruno Monteiro Rocha Lima, Thiago Eustaquio Alves de Oliveira, and Vinicius Prado da Fonseca. Classification of textures using a tactile-enabled finger in dynamic exploration tasks. In *2021 IEEE Sensors*, pages 1–4. IEEE, 2021.

[19] Shan Luo, Joao Bimbo, Ravinder Dahiya, and Hongbin Liu. Robotic tactile perception of object properties: A review. *Mechatronics*, 48:54–67, 2017.

[20] T. Taunyazov, H.F. Koh, Y. Wu, C. Cai, and H. Soh. Towards effective tactile identification of textures using a hybrid touch approach. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4269–4275. IEEE, 2019.

[21] A. Pagoli, F. Chapelle, J.A. Corrales-Ramon, Y. Mezouar, and Y. Lapusta. Large-area and low-cost force/tactile capacitive sensor for soft robotic applications. *Sensors*, 22(11):4083, 2022.

[22] Y. Yan, Y. Shen, C. Song, and J. Pan. Tactile super-resolution model for soft magnetic skin. *IEEE Robotics and Automation Letters*, 7(2):2589–2596, 2022.

[23] R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta. Reskin: versatile, replaceable, lasting tactile skins. *arXiv preprint arXiv:2111.00071*, 2021.

[24] Thiago Eustaquio Alves De Oliveira, Ana-Maria Cretu, and Emil M Petriu. Multimodal bio-inspired tactile sensing module. *IEEE Sensors Journal*, 17(11):3231–3243, 2017.

[25] Thiago Eustaquio Alves de Oliveira and Vinicius Prado da Fonseca. Bioin-tacto: A compliant multi-modal tactile sensing module for robotic tasks. *HardwareX*, 16:e00478, 2023.

[26] V Naga Sai Siddhartha Danyamraju, Tahsin Ahmed Prottoy, SM Shahriar Jobayer, and Vinicius Prado da Fonseca. Comparing data representation techniques for tactile sensing in classification tasks. In *2023 IEEE SENSORS*, pages 1–4. IEEE, 2023.

[27] Humberto Navarro de Carvalho, Lucas Pontes Castro, Thaís G. Do Rego, Telmo M. Silva Filho, Yuri de A. M. Barbosa, Leonardo Vidal Batista, Amilcar Soares, and Vinicius Prado Da Fonseca. Evaluating data representations for object recognition during pick-and-place manipulation tasks. In *2022 IEEE International Systems Conference (SysCon)*, pages 1–6, 2022.

[28] Viral Rasik Galaiya, Mohammed Asfour, Thiago Eustaquio Alves de Oliveira, Xianta Jiang, and Vinicius Prado da Fonseca. Exploring tactile temporal features for object pose estimation during robotic manipulation. *Sensors*, 23(9):4535, 2023.

[29] Viral Rasik Galaiya, Thiago Eustaquio Alves De Oliveira, Xianta Jiang, and Vinicius Prado Da Fonseca. Grasp approach under positional uncertainty using compliant tactile sensing modules and reinforcement learning. In *2024 IEEE*

Canadian Conference on Electrical and Computer Engineering (CCECE), pages 424–428. IEEE, 2024.

[30] Thiago Eustaquio Alves de Oliveira, Vinicius Prado da Fonseca, Bruno Monteiro Rocha Lima, Ana-Maria Cretu, and M Petriu. End-effector approach flexibilization in a surface approximation task using a bioinspired tactile sensing module. In *2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pages 1–6. IEEE, 2019.

[31] Laurent YE Ramos Cheret, Vinicius Prado Da Fonseca, and Thiago E Alves de Oliveira. Leveraging compliant tactile perception for haptic blind surface reconstruction. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17139–17145. IEEE, 2024.

[32] Vinicius Prado da Fonseca, Xianta Jiang, Emil M Petriu, and Thiago Eustaquio Alves de Oliveira. Tactile object recognition in early phases of grasping using underactuated robotic hands. *Intelligent Service Robotics*, 15(4):513–525, 2022.

[33] Vinicius Prado da Fonseca, Daniel John Kucherhan, Thiago E Alves de Oliveira, Da Zhi, and Emil M Petriu. Fuzzy controlled object manipulation using a three-fingered robotic hand. In *2017 Annual IEEE international systems conference (SysCon)*, pages 1–6. IEEE, 2017.

[34] Daniel J Kucherhan, Miriam Goubran, Vinicius P da Fonseca, Thiago E Alves de Oliveira, Emil M Petriu, and Voicu Groza. Object recognition through manipulation using tactile enabled prosthetic fingers and feedback glove-experimental study. In *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–6. IEEE, 2018.

[35] Bruno Monteiro Rocha Lima, Vinicius Prado da Fonseca, Thiago Eustaquio Alves de Oliveira, Qi Zhu, and Emil M Petriu. Dynamic tactile exploration for texture classification using a miniaturized multi-modal tactile sensor and machine learning. In *2020 IEEE International Systems Conference (SysCon)*, pages 1–7. IEEE, 2020.

[36] N. Kaempchen and K. Dietmayer. Data synchronization strategies for multi-sensor fusion. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, volume 85, pages 1–9. IEEE, 2003.

[37] Bruno Monteiro Rocha Lima, Thiago Eustaquio Alves de Oliveira, and Vinicius Prado da Fonseca. Multimodal Tactile Texture Dataset, 2023.

[38] Bruno Monteiro Rocha Lima, Venkata Naga Sai Siddhartha Danyamraju, Thiago Eustaquio Alves de Oliveira, and Vinicius Prado da Fonseca. A multimodal tactile dataset for dynamic texture classification. *Data in Brief*, 50:109590, 2023.

[39] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal*

of Science, 2(11):559–572, 1901.

[40] Aapo Hyvarinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[41] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[42] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[43] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[44] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63:3–42, 2006.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[46] M Chan. New online fiber sensor technology unlocks value in fiber manufacturing. *International Fiber Journal*, 2000.

[47] Max Schneckenburger, Sven Höfler, Luis Garcia, Rui Almeida, and Rainer Börret. Material removal predictions in the robot glass polishing process using machine learning. *SN Applied Sciences*, 4(1):33, 2022.

[48] Shemonto Das, Vimal Simha, Joshua Swamidas, Amilcar Soares, and Vinicius Prado Da Fonseca. Unbalanced fault classification using active learning in synthetic fiber manufacturing process. In *2024 IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2024.