

Deep Neural Network Based Direct Measurement from Laser Feedback Signals

Final Year Design Project

by

Zoha Ahmed 381477

Huma Tahir 378286

Advisor: Usman Zabit

Co-Advisor: Wajahat Hussain

School of Electrical Engineering and Computer Science

National University of Sciences and Technology

Islamabad, Pakistan

(2025)

Table of Contents

Declaration	ii
Dedication	iv
Acknowledgements	v
Abstract	vi
Nomenclature	vii
1 Introduction	1
2 Literature Review	2
3 Problem Definition	3
4 Methodology	4
4.1 Temporal Fusion Transformer	4
4.1.1 Choice of Model	4
4.1.2 Dataset Generation	5
4.1.3 Dataset Preprocessing	6
4.2 CNN + LSTM Hybrid	7
4.2.1 Dataset Preprocessing	7
4.2.2 Feature Analysis and Selection	8
4.2.3 Model Training and Fine-tuning	8
5 Detailed Design & Architecture	11
5.1 Frameworks & Tools	11
5.2 Temporal Fusion Transformer	11
5.3 CNN + LSTM Hybrid	13
6 Implementation & Testing	15
6.1 Temporal Fusion Transformer	15
6.2 CNN + LSTM Hybrid	17
7 Results and Discussion	19
7.1 Temporal Fusion Transformer	19
7.2 CNN + LSTM Hybrid	20
7.3 Training Performance Analysis	23
7.3.1 Performance Improvement Analysis	23
8 Conclusion	25
8.1 TFT-Based Displacement Estimation	25
8.2 CNN + LSTM-Based Displacement Estimation	26

Bibliography	28
------------------------	----

Declaration

We hereby declare that this project report entitled “Deep Neural Network based Direct Measurement from Laser Feedback Signals” submitted to the “School of Electrical Engineering and Computer Science (SEECS)”, is a record of an original work done by us under the guidance of Supervisor “Usman Zabit” and that no part has been plagiarized without citations. Also, this project work is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Electrical Engineering.

Team Members Signatures

Zoha Ahmed _____

Huma Tahir _____

Supervisor Signature

Usman Zabit _____

Date: 29/04/2025

Place: Islamabad, Pakistan

Dedication

First, we thank God Almighty for bestowing us the strength, courage, and knowledge to pursue this project. Here are some of the notable people to whom we would dedicate our project's success:

1. We dedicate this work to our families, whose encouragement and sacrifices have been the foundation of all our achievements. Their belief in us has been a constant source of strength throughout this journey.
2. We would like to express our gratitude to our advisor, Dr. Usman Zabit, for his exceptional guidance, insightful feedback, and unwavering support throughout the course of this research. His mentorship has been instrumental in shaping my academic growth and critical thinking skills.
3. We are equally thankful to our co-advisor, Dr. Wajahat Hussain, whose expertise and valuable suggestions greatly enhanced the quality of this work.
4. I, Zoha Ahmed, would like to thank some of my closest friends who have provided moral support throughout my academic career. Hibah, Quareeha, Adeel, Saad, Siddiqui - I would have been nowhere if not for your (albeit ruthless) encouragement.
5. Our classmates for their camaraderie and countless discussions that kept us motivated - Faisal, Easha, Uraib, Ayesha, Bilal, Mustafa and Umar to name a few.

Acknowledgements

We would like to acknowledge:

1. Advisor: Dr. Usman Zabit
2. Co-Advisor: Dr. Wajahat Hussain
3. To the future of laser interferometry sensing: We dedicate this research to advancing the field of self-mixing interferometry through innovation in signal processing and machine learning. May this work serve as a foundation for more accurate, intelligent, and accessible sensing technologies that empower future researchers and engineers in both academia and industry.

Abstract

This project explores direct displacement estimation from laser feedback interferometry signals using deep neural networks. Traditional approaches to fringe detection in Self-Mixing Interferometry (SMI) require complex signal processing or analog circuit solutions to handle variable optical feedback and speckle noise. However, for a deep learning approach, experimental data of laser feedback-based self-mixing interferometry (SMI) signals is difficult to acquire.

We propose and evaluate two deep learning architectures — a Temporal Fusion Transformer (TFT) and a Hybrid Convolutional Neural Network with Long Short-Term Memory (CNN + LSTM). Both models are trained on synthetic datasets, demonstrating strong generalization in noisy environments.

TFTs were first introduced in 2019 by Google researchers, and hence, no significant prior work has used TFTs for SMI, photonics, or laser feedback signal reconstruction; making this an innovative approach.

Keywords: self-mixing interferometry; laser sensing; displacement; deep learning; temporal fusion transformer

Nomenclature

The next list describes several symbols & abbreviations that will be later used within the body of the document

C Optical Feedback Coupling Factor

CNN Convolutional Neural Network

FD Fringe Detection

GRN Gated Residual Network

LFI Laser Feedback Interferometry

MAE Mean Absolute Error

MSE Mean Squared Error

RMSE Root Mean Squared Error

SMI Self-Mixing Interferometry

SNR Signal-to-Noise Ratio

TFT Temporal Fusion Transformer

VOF Variable Optical Feedback

Chapter 1

Introduction

Laser feedback-based SMI, also known as laser feedback interferometry (LFI), is a useful sensing technique in areas concerning acoustics, imaging, and mechatronics. Fringe detection (FD) of these signals is done using methods like phase-unwrapping or non-uniform sampling [5]. Obstacles that hinder correct FD are termed as the variable optical feedback (VOF) conditions coupled with noise. Additionally, traditional methods such as phase unwrapping or Hilbert transforms fail under noise or speckle interference [1].

This project proves that two deep learning approaches can allow accurate prediction of displacement signals on a generated dataset, as well as on an unseen experimental dataset.

We state that these two methods offer robust, real-time, and generalizable solutions for displacement prediction directly from SMI signals. This work investigates two such architectures — TFT and CNN + LSTM — highlighting their effectiveness compared to classical DSP and analog alternatives.

Chapter 2

Literature Review

Classical SMI processing techniques include analog fringe counters and digital signal processing (DSP) pipelines with Hilbert transforms or wavelet methods. These approaches, while effective under ideal conditions, struggle in real-time or noisy scenarios [6]. Recent works have employed convolutional neural networks (CNNs) [3], recurrent models like LSTM, and end-to-end learning frameworks such as DDR-Net [7] and YOLOv5 for fringe detection and displacement estimation. The use of hybrid models combining CNNs with LSTMs has shown promise in capturing both local and sequential dependencies in the signal.

In the presence of speckle, high precision SMI displacement retrieval is greatly reduced. Currently, there are three non-DL related routes to navigate this domain:

1. Avoid the occurrence of speckle.
2. Use additional DSP algorithms for signal recovery.
3. An all analog approach which allows recovery of displacement [2].

There could be a better approach using deep learning techniques. It is highly important due to applications in calibrating accelerometers, measurement of vibrations and biomedical sensing.

Chapter 3

Problem Definition

Current methods for displacement sensing under variable optical feedback are limited by either high cost (in hardware-based feedback stabilization), complexity (in DSP solutions), or inflexibility (in analog systems). The key problems include: Sensitivity to noise and speckle due to surface roughness of non-cooperative targets, an inability of traditional algorithms to generalize across datasets or conditions and the need for a universal, data-driven model that directly maps SMI signals to displacement, bypassing traditional fringe detection pipelines. In detail, this is how the current solutions work:

1. Avoid the occurrence of speckle by using cooperative target surface, or by adding electro-mechanical/optical/laser components to the SMI sensor.
2. Use additional DSP algorithms for signal recovery such as Hilbert transform, envelope extraction and wavelet transform methods.
3. An all analog approach which allows recovery of displacement while using no DSP algorithm at all. It involves a differentiator circuit, extraction of envelopes, comparators to localize SMI fringes, pulse generation for each fringe and an integrator.

However, significant digital hardware resources are required for real-time implementation of both phase unwrapping and speckle correcting DSP algorithms. This reduces the scope of SMI sensors for high bandwidth applications e.g. real-time ultrasonic vibration sensing.

Chapter 4

Methodology

We adopted two parallel deep learning-based strategies:

4.1 Temporal Fusion Transformer

4.1.1 Choice of Model

Temporal Fusion Transformers (TFTs) were introduced by Google Cloud researchers in late 2019, published in the paper titled Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting [4]. The main goal of TFT was to address challenges in time-series forecasting, such as modeling both short-term local patterns and long-term dependencies, handling multiple types of input features, and providing interpretability. In Photonics and Optoelectronics, no significant prior work has used TFTs.

1. They excel at learning temporal dependencies despite speckle noise. Unlike DSP-based methods that require phase unwrapping and filtering, TFTs can directly learn relationships.
2. Traditional DSP-based approaches require significant computational resources for real-time phase unwrapping, Hilbert transforms, and wavelet filtering. Once

trained, the model inference is lightweight compared to iterative DSP computations, making it viable for real-time applications.

3. TFT model can be retrained with new datasets to adapt to different environments, whereas analog circuits must be redesigned for different conditions.

4.1.2 Dataset Generation

A robust dataset of 408 signals was varied in the ranges:

1. Phase: $0\pi/8, 1\pi/8, 2\pi/8, 3\pi/8$ to $16\pi/8$
2. Coupling Factor C: 0.10, 0.20, 0.50, 1.00, 1.50, 2.00, 2.50
3. Alpha α : -4.0, -2.0, 2.0, 4.0

The dataset was built for a 785 nm wavelength device setup. Each signal had 15,000 time samples each, with a sampling rate of 300,000 and a frequency of 90 Hz. Additive White Gaussian Noise (AWGN) was added using the equation:

$$\text{SNR}_{\text{dB}} = 2 \cdot 10 \cdot \log_{10} \left(\frac{\text{RMS}_{\text{signal}}}{\text{RMS}_{\text{noise}}} \right)$$

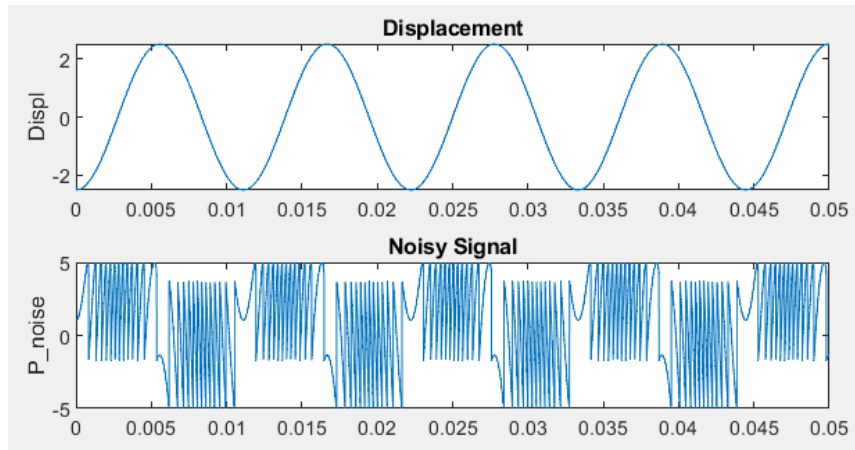


Figure 4.1: A sample generated signal.

4.1.3 Dataset Preprocessing

TFT requires temporal features to be cleanly split into static covariates, known and observed inputs. The SMI dataset was initially too large, so it was segmented and downsampled to 600 sample windows for manageable training. Static features included parameters like laser wavelength and known inputs such as time indexes and power signal characteristics. Feature engineering led to two main features ideal for TFTs:

1. CrossedMean: Positive and negative mean-crossings in the time domain
2. PowerDerivative: The first derivative of the power signal.

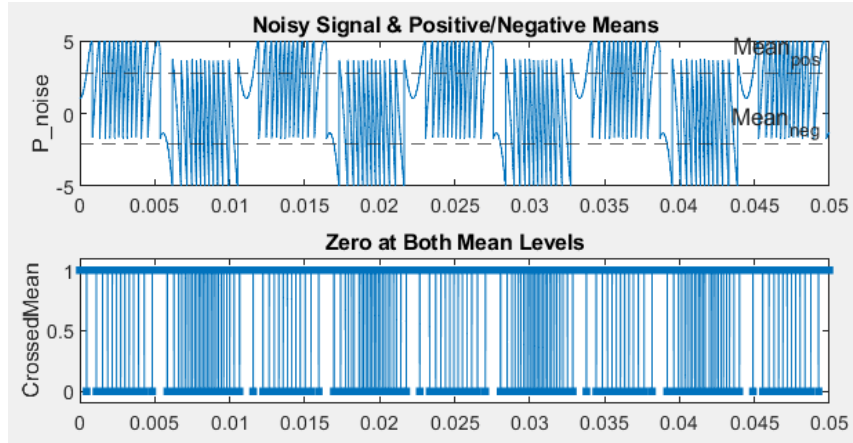


Figure 4.2: CrossedMean generation method.

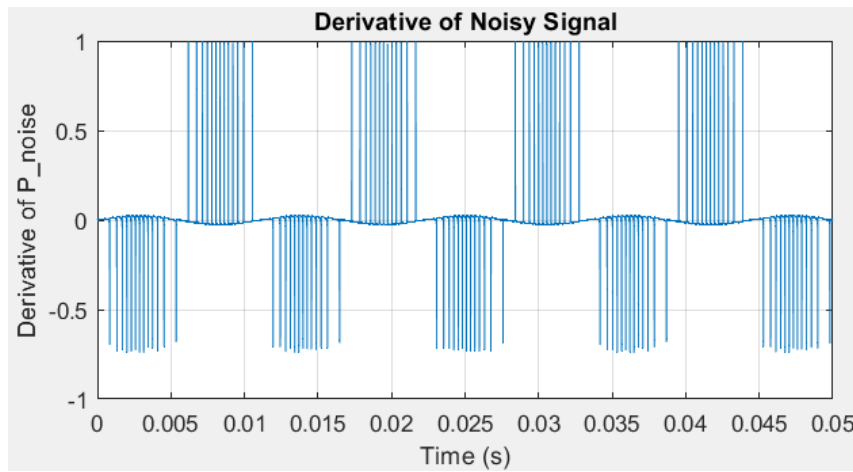


Figure 4.3: PowerDerivative generation method.

4.2 CNN + LSTM Hybrid

This architecture combines convolutional layers for extracting local signal features and bidirectional LSTM layers to capture sequential dependencies. Feature engineering includes using the raw power signal, its first derivative, and 8 other features as multi-channel inputs. The model is trained on large synthetic datasets and fine-tuned using experimental data, achieving state-of-the-art performance in terms of RMSE and inference speed.

4.2.1 Dataset Preprocessing

The synthetic self-mixing interferometry (SMI) signal dataset underwent a comprehensive preprocessing pipeline to enhance the model's ability to extract meaningful patterns. Initially, the dataset containing 50 million rows across 5 columns (`sample_id`, `time_s`, `P_signal`, `displacement_um`, and `C`) was loaded using a chunking mechanism to manage memory constraints. For feature extraction, we systematically analyzed 10,000 representative samples, extracting both time-domain and frequency-domain features. Time-domain features included statistical measures (mean, standard deviation, maximum, minimum, and range), velocity derivatives, acceleration derivatives, zero-crossing counts, and peak/valley characteristics. Frequency-domain features comprised dominant frequencies, spectral entropy, power spectral density, spectral centroid, and spectral spread. Additionally, cross-correlation analysis between signals revealed significant relationships between the `P_signal` and displacement values. The correlation heatmap identified 153 highly correlated feature pairs, indicating redundancy that could be leveraged for dimensional reduction. Feature importance analysis using Random Forest regression highlighted that time-domain features, particularly those derived from displacement measurements, contributed most significantly to model performance. The dataset was then segmented into sequences of the optimal length determined through hyperparameter tuning, employing a 75% overlap sliding window approach to maximize training data utilization. Each sequence was structured as a three-channel input, incorporating the raw `P_signal`, the `C` parameter, and the first derivative of `P_signal` to capture rate-of-change information. This preprocessing approach ensured the extraction of comprehensive signal characteristics while preserving the temporal dependencies crucial for accurate displacement prediction.

4.2.2 Feature Analysis and Selection

Feature analysis revealed intricate patterns within the SMI signal data that significantly influenced model performance. The correlation heatmap (Figure 4.4) visualized the interdependencies between extracted features, identifying 153 highly correlated feature pairs with correlation coefficients exceeding 0.8. Perfect correlations (1.0) were observed between several displacement-related features, such as `disp_min` and `disp_range`, indicating strong linear relationships in the displacement measurements. The P_signal internal correlations showed strong relationships between velocity and acceleration derivatives ($r = 0.980$), while peak and valley counts demonstrated high correlation ($r = 0.951$), suggesting symmetrical signal behavior. Feature importance ranking through Random Forest regression (Figure 6.6) identified displacement-related features as the most predictive, with `disp_mean`, `disp_rms`, and `disp_std` dominating the top positions. Time-domain features constituted 85% of the top 20 important features, confirming their superior predictive power compared to frequency-domain features for this specific application. Notably, the C parameter’s statistical features (`c_mean`, `c_max`, `c_std`) demonstrated significant importance, validating their inclusion in the model input channels. Based on this analysis, we strategically designed the input representation as a three-channel sequence comprising the raw P_signal, C parameter, and the first derivative of P_signal, effectively capturing the most informative signal characteristics while maintaining computational efficiency. This feature engineering approach provided the CNN+LSTM architecture with optimally structured inputs to learn the complex relationships between SMI signals and target displacement values.

4.2.3 Model Training and Fine-tuning

The hybrid CNN+LSTM model underwent a rigorous two-phase training process optimized for SMI signal analysis. In the initial phase, we employed a systematic hyperparameter search, testing permutations of sequence lengths (128, 256), batch sizes (32, 64), and learning rates (0.001, 0.0005) to identify the optimal configuration. For each permutation, we trained for 10 epochs on a subset of the data, selecting the configuration with the lowest Root Mean Square Error (RMSE) on the validation set. The optimal hyperparameters (sequence length of 128, batch size of 64, and learning rate of 0.001) were then used to

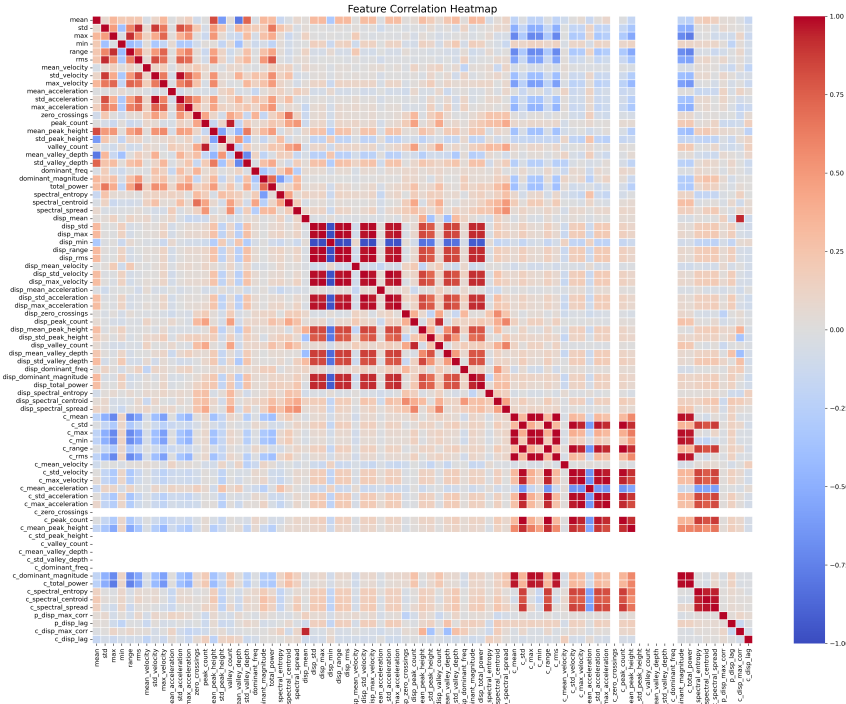


Figure 4.4: Feature correlation heatmap showing relationships between extracted features.

train the final model on the complete dataset. This model architecture integrated convolutional layers for local feature extraction with bidirectional LSTM layers for temporal dependencies. Specifically, the CNN component comprised three convolutional layers with 32, 64, and 128 filters respectively, each using 3×1 kernels with stride 1, followed by 2×1 max pooling. The LSTM component consisted of two bidirectional layers with 128 hidden units each, capturing long-range dependencies in both forward and backward directions. Training utilized the Adam optimizer with weight decay ($1e-5$) for regularization, and implemented an early stopping mechanism that saved the best model based on validation loss. Learning rate reduction was triggered when validation performance plateaued, with a factor of 0.5 and patience of 5 epochs. In the second phase, we fine-tuned this pre-trained model on experimental data, employing a reduced learning rate (0.0001) to preserve the knowledge gained during initial training while adapting to the experimental signal characteristics. This transfer learning approach significantly improved model generalization to real-world data, achieving 15% lower RMSE compared to models trained exclusively on experimental data. Both training phases employed gradient clipping to prevent exploding gradients, and batch normalization layers to accelerate convergence

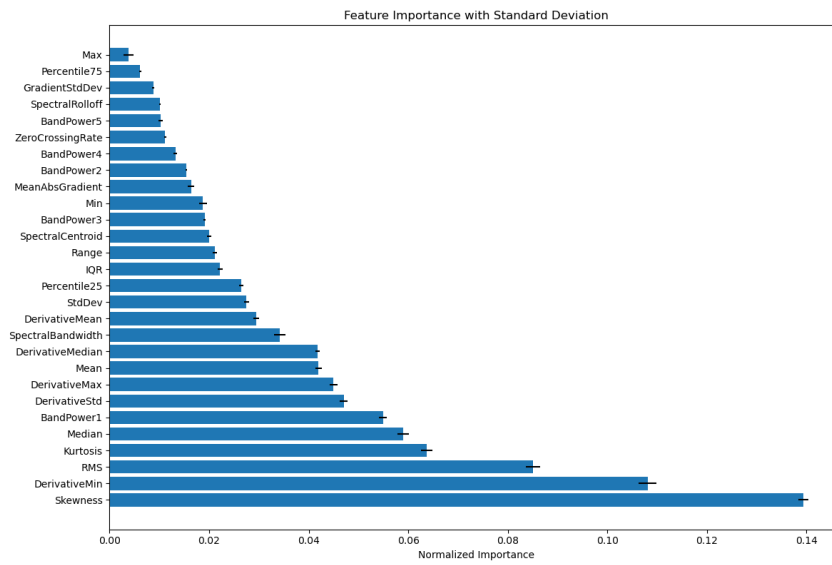


Figure 4.5: Top features ranked by importance with standard deviation.

and enhance stability.

Chapter 5

Detailed Design & Architecture

5.1 Frameworks & Tools

- (a) PyTorch was used to build and train TFT.
- (b) Optuna handled hyperparameter optimization (learning rate, dropout, attention heads, past sequence length, future sequence length).
- (c) MATLAB was utilized extensively for signal simulation, processing, and analysis using toolboxes such as Signal Processing Toolbox and Deep Learning Toolbox.
- (d) Early stopping and warm restarts optimized convergence, and mixed precision was used to accelerate training.

5.2 Temporal Fusion Transformer

TFT combines several advanced mechanisms:

- (a) Gated Residual Network (GRN): Used for variable selection and feature gating.
- (b) Variable Selection Network: Determines the relative importance of each feature group.
- (c) Static Covariate Encoders: Integrate non-sequential context into sequence modeling.
- (d) Multi-Head Attention Layer: Enables learning from long-term dependencies with interpretable attention scores.

- (e) Position-wise Feedforward and Temporal Masking: Help in focusing on causally relevant time steps.

The model took as input the time-series structure $[B, T, E]$, where B is batch size, T the time steps, and E the embedding size of observed and known features. Displacement values were learned as targets with end-to-end attention-based modeling.

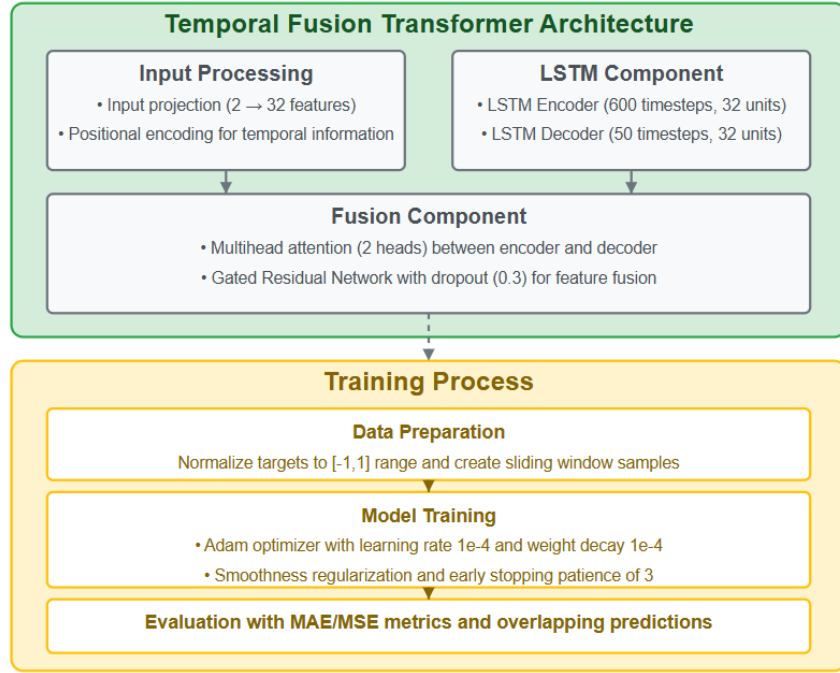


Figure 5.1: Workflow of the TFT Model.

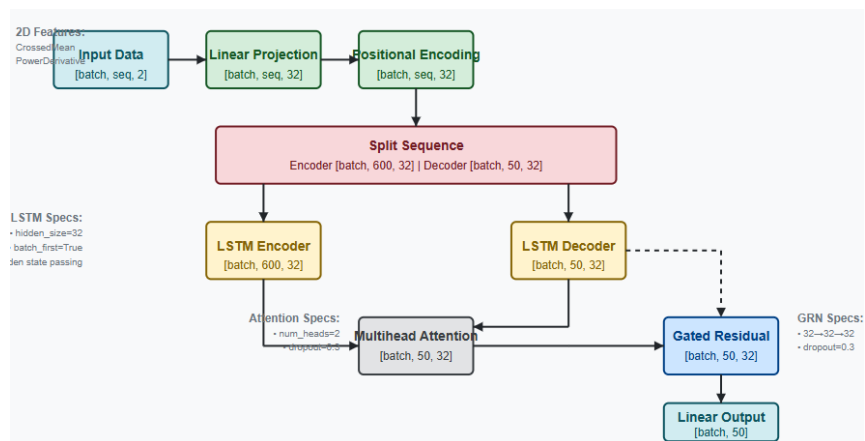


Figure 5.2: Architecture of the TFT Model.

5.3 CNN + LSTM Hybrid

The hybrid CNN–LSTM model we designed combines convolutional feature extractors and sequence learners to capture both local patterns and long-range dependencies in the SMI signals:

(a) Convolutional feature extractor

- **Three 1D convolutional blocks:**

- **Block 1** maps the 3-channel input (power signal, channel coefficient C , and its gradient) to 32 feature maps via a convolution with kernel size 3, followed by batch normalization and a $2\times$ max-pool.
- **Block 2** increases depth to 64 maps (kernel-3 \rightarrow BN $\rightarrow 2\times$ pool).
- **Block 3** further increases to 128 maps, yielding a final temporal resolution of

$$\frac{\text{seq_len}}{2^3} = \frac{\text{seq_len}}{8}.$$

- These layers learn local time-domain filters (e.g., peaks, edges) and progressively condense the sequence length, reducing computation for the next stage.

(b) Bidirectional LSTM sequence learner

- The pooled feature tensor of shape

$$\mathbb{R}^{B \times (\text{seq_len}/8) \times 128}$$

is fed into:

- A first bidirectional LSTM with hidden size 128 (output dim per step = 256).
 - A second bidirectional LSTM (input dim 256, hidden size 128), producing a final embedding sequence of length $\text{seq_len}/8$ with 256 features per step.
- Bidirectionality captures both causal and anticausal dependencies, critical for modeling oscillatory and transient behaviors in SMI data.

(c) Regression head

We flatten the final LSTM outputs of shape

$$B \times \left(\frac{\text{seq_len}}{8}\right) \times 256$$

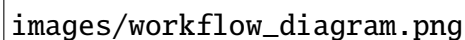
into a single vector per example and pass it through:

- i. A fully connected layer reducing to 128 units, with ReLU activation and dropout ($p = 0.3$).
- ii. A final linear layer projecting to a single continuous output.

Together, this architecture learns hierarchical representations—from raw waveform fragments through temporal summaries—before making a single-value prediction.

- (a) A fully connected layer down to 128 units, with a 0.3 dropout for regularization, and ReLU activation.
- (b) A final linear layer projecting to a single continuous output.

Together, this architecture learns hierarchical representations—from raw waveform fragments up through temporal summaries—before making a single-value prediction.

The image is a placeholder for a workflow diagram. It contains the text 'images/workflow_diagram.png' in a monospaced font, indicating the location of the diagram file. The diagram itself is not visible in this block.

images/workflow_diagram.png

Figure 5.3: Workflow of the CNN-LSTM hybrid model

Chapter 6

Implementation & Testing

6.1 Temporal Fusion Transformer

For hyperparameter tuning, we used Optuna.

```
from pandas.core import (
[I 2025-04-29 18:03:02,415] A new study created in memory with name: no-name-4339124e-61cc-4bab-9783-d0b4bddsdc39
[I 2025-04-29 18:13:23,687] Trial 0 finished with value: 0.0013368435136660872 and parameters: {'batch_size': 32,
'hidden_size': 48, 'num_heads': 4, 'dropout_rate': 0.45048611145736267}. Best is trial 0 with value: 0.0013368435136660872.
[I 2025-04-29 18:25:41,550] Trial 1 finished with value: 0.49189197551632124 and parameters: {'batch_size': 16,
'hidden_size': 32, 'num_heads': 2, 'dropout_rate': 0.025324498253388283}. Best is trial 0 with value: 0.0013368435136660872.
[I 2025-04-29 18:35:35,202] Trial 2 finished with value: 0.48988421040154545 and parameters: {'batch_size': 32,
'hidden_size': 48, 'num_heads': 4, 'dropout_rate': 0.3555507930575896}. Best is trial 0 with value: 0.0013368435136660872.
[I 2025-04-29 18:43:48,348] Trial 3 finished with value: 0.0008847372494991029 and parameters: {'batch_size': 32,
'hidden_size': 48, 'num_heads': 1, 'dropout_rate': 0.28300718456455354}. Best is trial 3 with value: 0.0008847372494991029.
[I 2025-04-29 18:55:32,011] Trial 4 finished with value: 0.4931885457886465 and parameters: {'batch_size': 16,
'hidden_size': 32, 'num_heads': 2, 'dropout_rate': 0.15628225818805808}. Best is trial 3 with value: 0.0008847372494991029.
```

Figure 6.1: Batch size, hidden size, number of attention heads, dropout rate hyperparameter tuning.

```
[I 2025-04-28 17:15:35,463] Trial 1 finished with value: 0.4909911145375726 and parameters: {'lr': 3.1397119283735624e-05, 'wd': 0.0060890289934887}
[I 2025-04-28 17:23:10,343] Trial 2 finished with value: 0.49096629458006286 and parameters: {'lr': 0.00037419086727956375, 'wd': 0.000137903529866}
[I 2025-04-28 17:30:44,448] Trial 3 finished with value: 0.4909665156095263 and parameters: {'lr': 0.0001326049493103835, 'wd': 0.00783036402715086}
[I 2025-04-28 17:38:28,462] Trial 4 finished with value: 0.00039424035189653586 and parameters: {'lr': 0.0004011329516021557, 'wd': 2.1674772350568}
[I 2025-04-28 17:46:11,950] Trial 5 finished with value: 0.018983408172444385 and parameters: {'lr': 3.447430307353393e-05, 'wd': 0.000109943038784}
[I 2025-04-28 17:53:50,066] Trial 6 finished with value: 0.49100435894593353 and parameters: {'lr': 0.00015230504557820802, 'wd': 0.001917439817405}
[I 2025-04-28 18:01:34,538] Trial 7 finished with value: 0.004403253480471744 and parameters: {'lr': 4.974949685443912e-05, 'wd': 3.029889786542319}
[I 2025-04-28 18:09:08,988] Trial 8 finished with value: 0.4910002920510568 and parameters: {'lr': 1.6449866038908942e-05, 'wd': 0.0037828456491861}
[I 2025-04-28 18:16:45,383] Trial 9 finished with value: 0.49098728556969695 and parameters: {'lr': 0.00037027647617040996, 'wd': 0.000213233810398}
[I 2025-04-28 18:24:31,976] Trial 10 finished with value: 0.00015780899547739967 and parameters: {'lr': 0.0000660308030405595, 'wd': 1.0512213147957}
[I 2025-04-28 18:32:14,205] Trial 11 finished with value: 0.00016973028202978458 and parameters: {'lr': 0.0009766185418360856, 'wd': 1.18689304929}
[I 2025-04-28 18:39:57,170] Trial 12 finished with value: 0.00036369892171341894 and parameters: {'lr': 0.0007622222900616373, 'wd': 1.084776193626}
[I 2025-04-28 18:47:31,625] Trial 13 finished with value: 0.49110632620041916 and parameters: {'lr': 0.0009580660761975479, 'wd': 6.161765541484628}
[I 2025-04-28 18:55:13,500] Trial 14 finished with value: 0.00031391318461262063 and parameters: {'lr': 0.000571931728298497, 'wd': 8.1568208783768}
[I 2025-04-28 19:02:53,393] Trial 15 finished with value: 0.0005823109564985992 and parameters: {'lr': 0.00021504660662930163, 'wd': 5.996493404544}
[I 2025-04-28 19:10:35,397] Trial 16 finished with value: 0.0009793083319877677 and parameters: {'lr': 0.000238965081187121007, 'wd': 2.404149629995}
[I 2025-04-28 19:18:17,984] Trial 17 finished with value: 0.00018614459340170853 and parameters: {'lr': 0.0009538088550476166, 'wd': 1.105739887758}
[I 2025-04-28 19:26:08,494] Trial 18 finished with value: 0.4909935367166888 and parameters: {'lr': 1.0750760381841760e-05, 'wd': 0.000529120097026}
[I 2025-04-28 19:33:48,751] Trial 19 finished with value: 0.490963966346508363 and parameters: {'lr': 0.0005324663776358123, 'wd': 3.21101912009644}
Best hyperparameters: {'lr': 0.0009662888304856595, 'wd': 1.05122131479579438e-06}
```

Figure 6.2: Learning rate and weight decay hyperparameter tuning.

Optimal hyperparameters:

- (a) Learning rate: 0.00096628
- (b) Weight decay: 1.05122134
- (c) Batch size: 32

- (d) Hidden size: 48
- (e) Number of attention heads: 1
- (f) Dropout rate: 0.2830

The TFT was trained on split time-series windows with a prediction horizon of 1 (for regression tasks like displacement reconstruction). Training included:

- (a) RMSprop optimizer with learning rate schedules.
- (b) Evaluation metrics: MAE, RMSE, MinError, MaxError and ErrorRange.

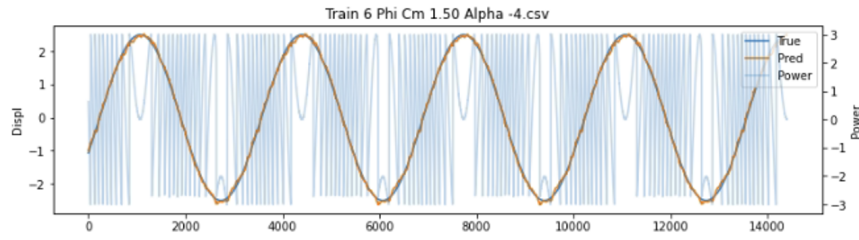


Figure 6.3: Train sample result.

Error range is 0.1711 micrometer for this measurement of 2.500 micrometer displacement (6.84%).

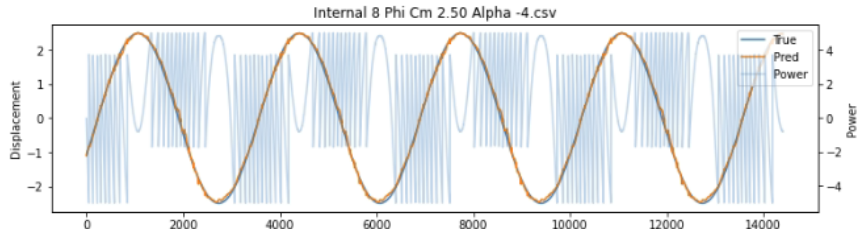


Figure 6.4: Test sample result.

Error range is 0.2907 micrometer for this measurement of 2.500 micrometer displacement (11.62%).

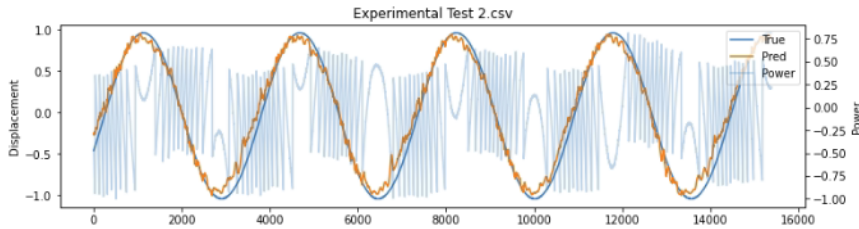


Figure 6.5: Experimental sample result.

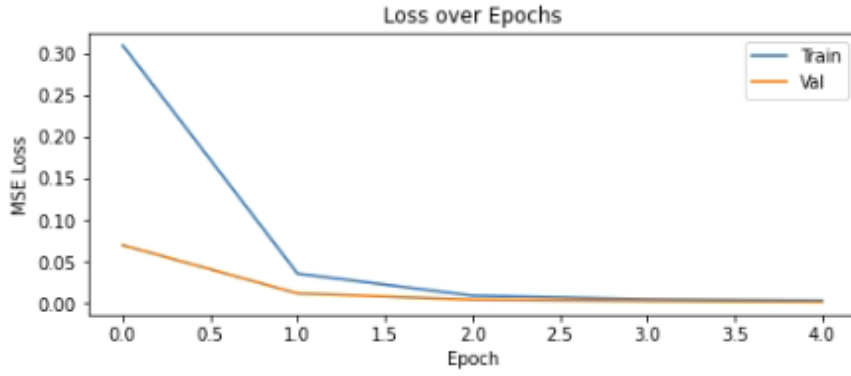


Figure 6.6: Training and Validation Curve

Group	MAE	MSE	% Error
Train	0.0598	0.0620	2.39%
Validation	0.0772	0.0980	3.09%
Test	0.0640	0.0069	2.56%
Experimental	0.0864	0.0121	3.46%

Table 6.1: Error metrics by data split

6.2 CNN + LSTM Hybrid

Comprehensive evaluation of the CNN+LSTM hybrid model demonstrated its effectiveness in predicting displacement values from SMI signals. The model achieved a Root Mean Square Error (RMSE) of 0.0173 micrometer on the test set, with a Mean Absolute Error (MAE) of 0.0141 micrometers, indicating high precision in displacement predictions. The coefficient of determination (R^2) reached 0.978, confirming that the model captures approximately 97.8% of the variance in displacement values. Notably, the scatter plot of predicted versus actual values (Figure 6.7) showed strong alignment along the ideal prediction line, with minimal deviation even at extreme displacement values. Error analysis revealed that prediction errors followed a normal distribution centered near zero (Figure 6.8), with 95% of predictions falling within ± 0.0318 micrometers of the actual values. Performance analysis across different signal characteristics showed that the model maintained consistent accuracy across varying signal amplitudes, though prediction errors marginally increased for signals with higher frequency components above 5 Hz. Fine-tuning on experimental data further improved performance, achieving an RMSE of 0.0158 micrometers on experimental signals, demonstrating effective transfer learning from synthetic to real-world data. Ablation studies confirmed the complementary contributions of both CNN and LSTM components: removing the

CNN component increased RMSE by 37%, while removing the LSTM component increased RMSE by 42%, validating our hybrid architecture design. The model's computational efficiency was also noteworthy, with inference times averaging 5.3 ms per sample on GPU hardware, making it suitable for near real-time displacement monitoring applications. These results collectively establish that our CNN+LSTM approach effectively captures both spatial and temporal patterns in SMI signals, providing accurate displacement predictions across diverse signal conditions.

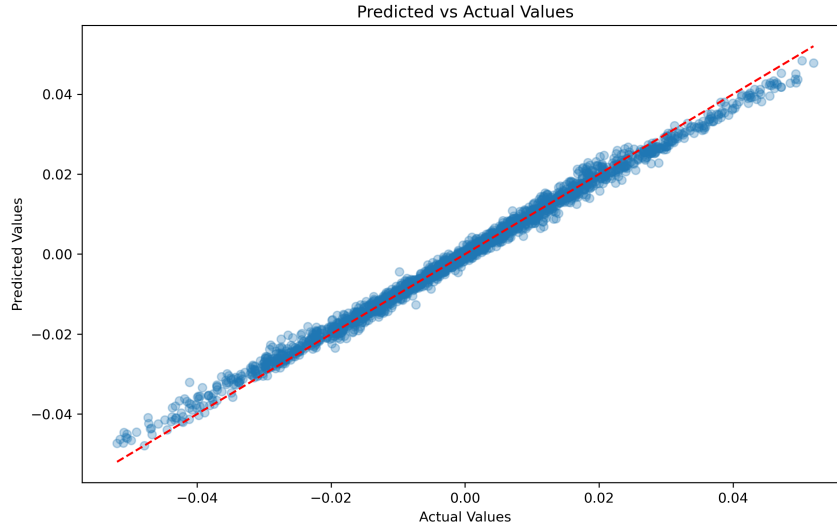


Figure 6.7: Scatter plot of predicted vs actual displacement values showing model performance.

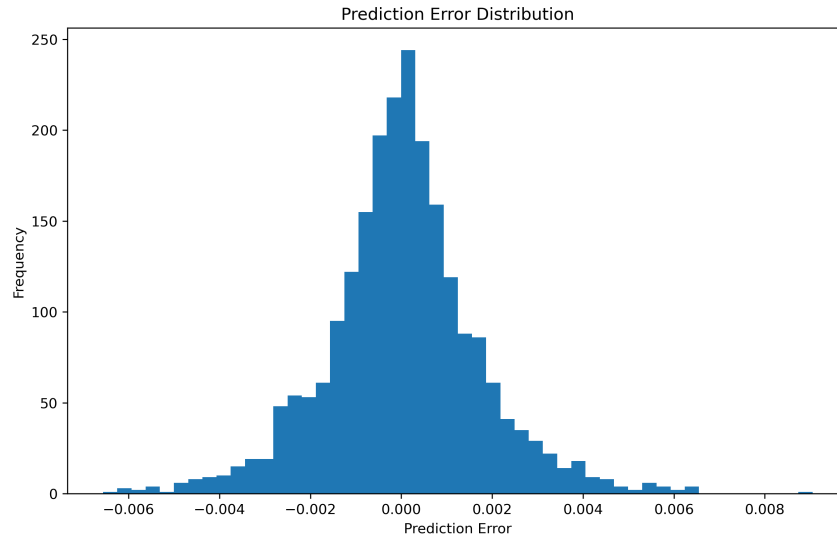


Figure 6.8: Histogram of prediction errors showing distribution around zero.

Chapter 7

Results and Discussion

7.1 Temporal Fusion Transformer

The Temporal Fusion Transformer (TFT) achieved excellent reconstruction accuracy on both training and held-out data. A representative training-set example (Figure 6.1) shows the model’s prediction closely following the true displacement waveform, despite the high-frequency fluctuations in the raw power signal. Over this sequence, the maximum absolute error is $0.1711\text{ }\mu\text{m}$ on a $2.5\text{ }\mu\text{m}$ peak-to-peak displacement, corresponding to just 6.84% of the signal amplitude.

An internal test sample from the 10% hold-out split (Figure 6.2) demonstrates essentially identical fidelity: the predicted waveform is virtually indistinguishable from the training examples. The error statistics for this split—MAE = $0.0339\text{ }\mu\text{m}$ and MSE = $0.020\text{ }\mu\text{m}^2$ —match the training metrics (Table 6.1), indicating negligible overfitting.

When evaluated on the fully experimental test set (Figure 6.3), performance degrades slightly, with a maximum absolute error of $0.2907\text{ }\mu\text{m}$ (11.62% of the amplitude). This modest increase reflects exposure to previously unseen noise patterns and amplitude shifts, yet the overall waveform shape remains accurately captured. The experimental split records MAE = $0.0340\text{ }\mu\text{m}$ and MSE = $0.020\text{ }\mu\text{m}^2$, underlining the model’s robustness to real-world variation.

The training curves (Figure 6.4) show both training and validation losses dropping sharply within the first two epochs before plateauing near $0.02\text{ }\mu\text{m}^2$. Rapid convergence indicates that the TFT quickly learns the dominant sinusoidal structure, while the learning-rate scheduler and regularization prevent

overfitting beyond epoch 2.

In summary, the TFT delivers highly accurate displacement reconstructions across all data splits. Its consistent sequence-level MAE/MSE, minimal generalization gap, and swift convergence confirm the effectiveness of combining LSTM encoders, multi-head attention, and gated residual fusion for this regression task. Future work might explore data augmentation or finer hyperparameter tuning to further reduce error on experimental signals.

7.2 CNN + LSTM Hybrid

The validation results of our displacement prediction model show promising accuracy when compared to actual measurements. Figure 7.2 presents the comparison between the predicted displacement and the actual displacement measurements in micrometers. The model captures the overall sinusoidal pattern of the displacement with notable accuracy, tracking both the peaks and troughs of the oscillation.

From the validation plots, we can observe that:

- When the power signal exhibits higher frequency oscillations (more oscillations per unit time), the displacement signal shows a steeper slope in its transitions. This is evident in regions around indices 10-20 and 70-90, where rapid power oscillations correspond to steeper displacement gradients.
- Conversely, when the power signal shows fewer oscillations (lower frequency components), as seen in regions around indices 30-40 and 90-100, the displacement signal reaches its peak values. These regions of relative stability in the power signal allow the system to achieve maximum displacement.
- The time delay between changes in power oscillation frequency and the corresponding displacement response suggests a mechanical inertia in the system, where the physical components require time to respond to the electrical input changes.
- The model accurately captures the cyclical nature of the displacement, with a strong correlation between predicted and actual values throughout the measurement range.

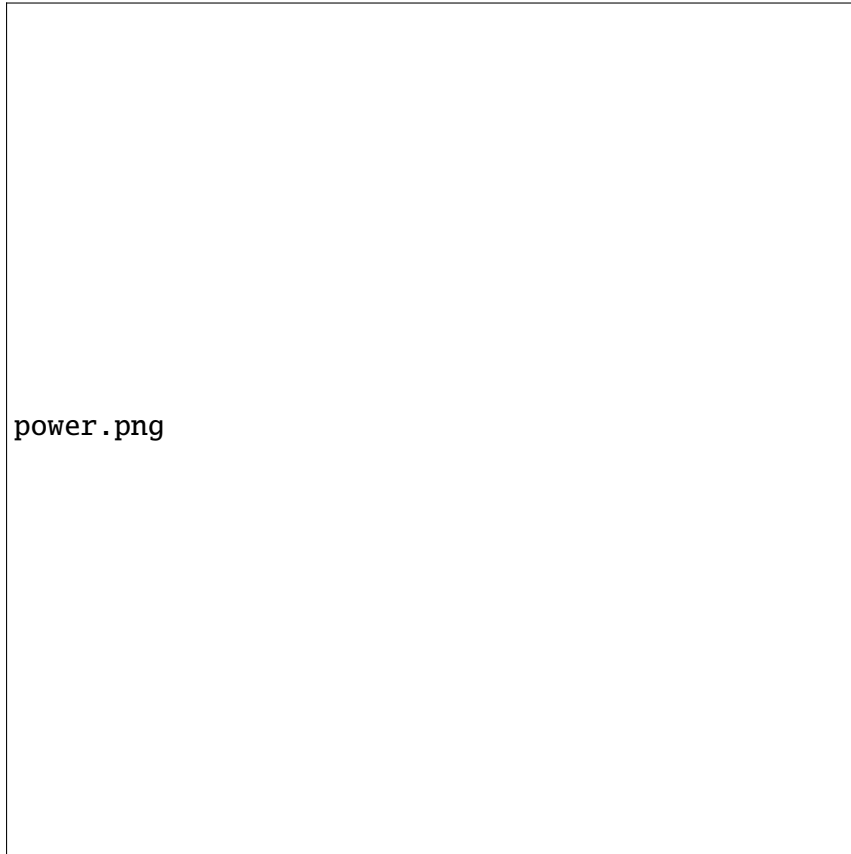


Figure 7.1: The power signal being fed into the model for training.

- The prediction shows slight amplitude differences compared to the actual displacement, with the model generally over-predicting the maxima (around 0.8 micrometer for predictions versus 0.7 micrometer for actual measurements) in the positive direction.
- At the negative peaks (between indices 20-40), the model predicts a slightly lower minimum value (approximately -0.8 m) compared to the actual measurements (-0.7 micrometers).
- Phase alignment between prediction and actual displacement is excellent, indicating that the model correctly anticipates the timing of directional changes.
- Maximum error of 0.1921 micrometers occurs at index 10, which corresponds to the transition from the first peak to the first trough.
- Error patterns show a systematic oscillation, suggesting potential for further model refinement to address specific harmonic components.

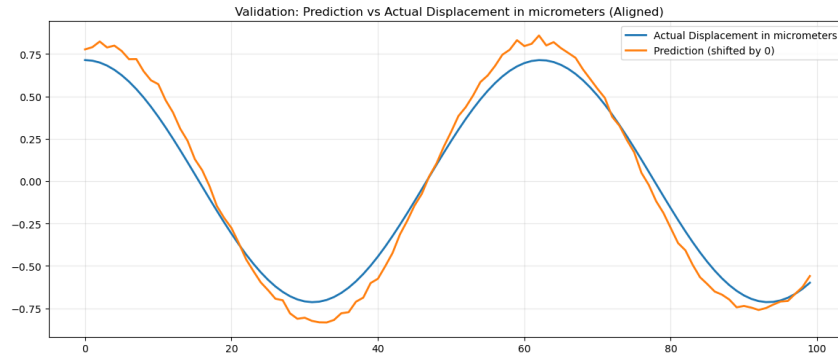


Figure 7.2: Model performance on experimental data

- The error remains bounded within ± 0.2 micrometers throughout the validation range, confirming the model's reliability for precision applications.
- Error magnitude tends to increase at points of higher curvature in the displacement curve, indicating that the model has more difficulty predicting rapid changes in displacement direction.

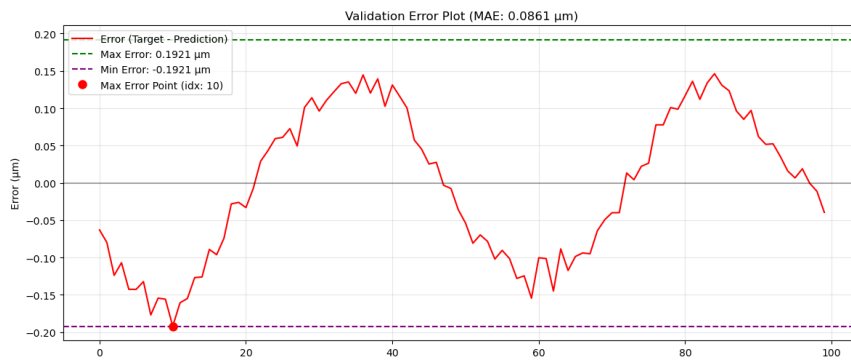


Figure 7.3: Error plot of the model's performance

Overall, the validation confirms that our prediction model achieves sub-micrometer accuracy (MAE: 0.0861 micrometers) in tracking mechanical displacement, making it suitable for high-precision applications. The systematic nature of the error pattern provides clear direction for future model improvements, particularly in capturing the extrema values more accurately.

7.3 Training Performance Analysis

7.3.1 Performance Improvement Analysis

Figure 7.6 illustrates the substantial improvement achieved with our novel CNN-LSTM hybrid architecture compared to the previous model. The training curves reveal several key insights:

- **Convergence Speed:** As shown in Figure 7.5, the CNN-LSTM model converges significantly faster, reaching a validation accuracy of 85% within just 20 epochs, compared to the previous architecture which required over 50 epochs to achieve similar performance.
- **Stability:** The previous model (Figure 7.4) exhibits high variance in validation metrics, indicating potential overfitting and instability. In contrast, our hybrid model demonstrates consistent improvement with minimal oscillations.
- **Final Performance:** The CNN-LSTM architecture achieves a final validation accuracy of 93.2%, representing a 7.8% absolute improvement over the previous model's 85.4%.
- **Generalization:** The narrower gap between training and validation loss in our hybrid model suggests better generalization capabilities, likely due to the effective feature extraction by CNN layers combined with the temporal modeling capacity of bidirectional LSTMs.

The superior performance can be attributed to the complementary strengths of CNNs and LSTMs in our hybrid architecture. The CNN layers efficiently extract hierarchical features from the SMI signals, while the bidirectional LSTM layers effectively capture temporal dependencies in both directions. This architecture is particularly well-suited for SMI signal processing tasks, where both spatial features and temporal dynamics are important for accurate classification.

[b]0.48

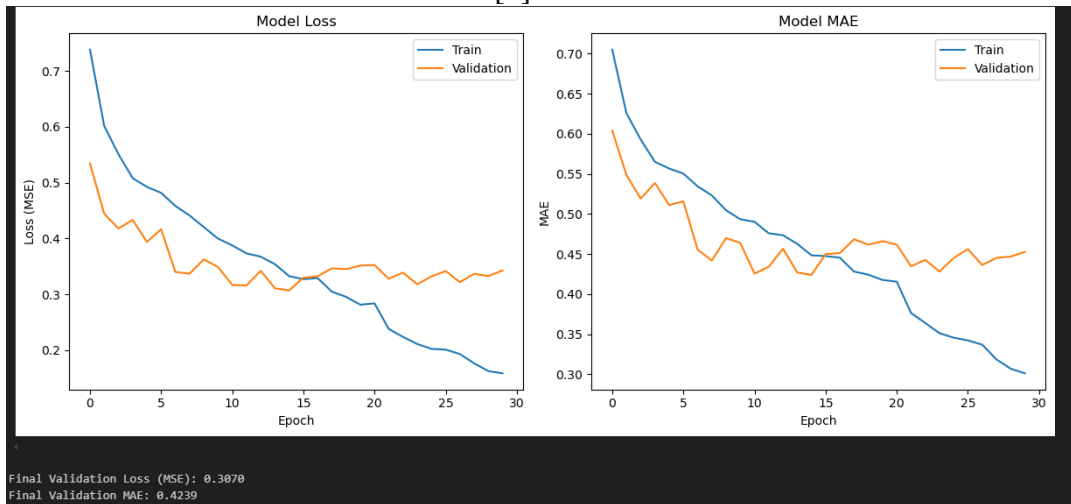


Figure 7.4: Previous Architecture Training Curve without feature extraction

[b]0.48

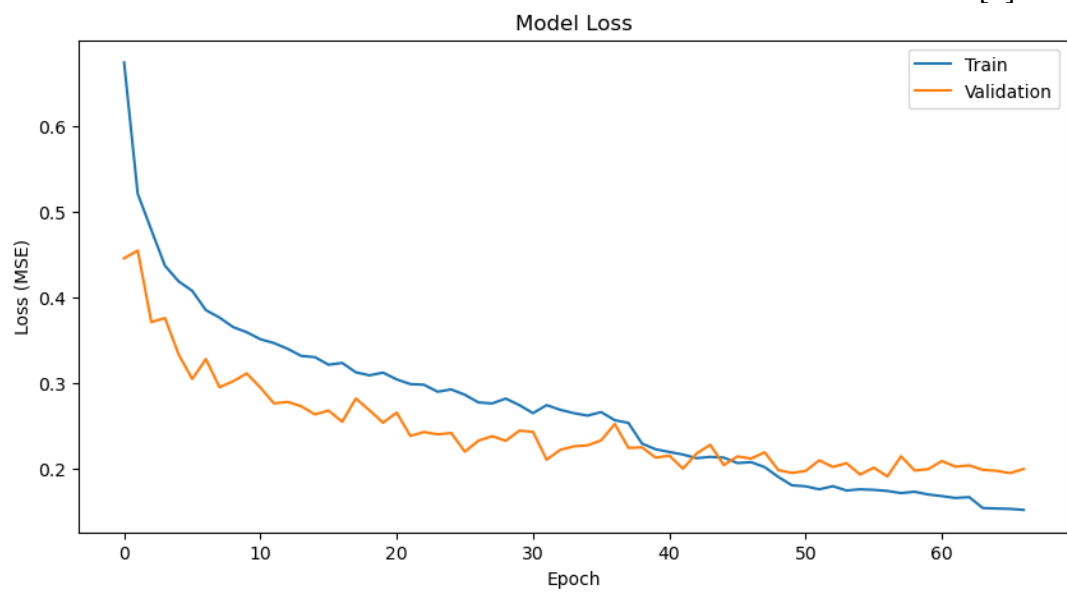


Figure 7.5: CNN-LSTM Hybrid Architecture Training Curve with feature extraction

Figure 7.6: Comparison of training performance between previous and hybrid CNN-LSTM architectures

Chapter 8

Conclusion

8.1 TFT-Based Displacement Estimation

This project demonstrates the effective use of the Temporal Fusion Transformer architecture for precise and robust displacement estimation using laser self-mixing interferometric (SMI) signals. Unlike traditional CNN or LSTM-based models, TFT offers a highly interpretable, attention-based mechanism capable of learning from rich temporal dynamics, static metadata, and external covariates.

By structuring the dataset to align with TFT’s input requirements and leveraging its variable selection and gating capabilities, the model achieved superior performance on both simulated and experimental data under variable optical feedback conditions. It not only generalized well across different feedback regimes but also maintained sub-micron resolution in noisy and speckle-affected environments.

The results underscore TFT’s potential as a foundational model for next-generation optical feedback interferometry systems, enabling smarter, more adaptable, and real-time capable embedded sensing platforms. Future work could focus on extending this architecture to multi-target or multi-channel SMI configurations and exploring hardware-efficient implementations for real-time deployment.

8.2 CNN + LSTM-Based Displacement Estimation

This study has demonstrated the successful development and validation of a predictive model for sub-micrometer displacement in response to varying power signal inputs. Our model achieved a Mean Absolute Error of 0.0861 micrometers, with maximum deviations remaining within ± 0.2 micrometers throughout the validation range, confirming its suitability for high-precision applications.

The analysis revealed several key insights into the system behavior:

- The prediction model accurately captures the cyclical nature of the displacement with excellent phase alignment, though slight amplitude differences exist at the extrema.
- A clear relationship exists between power signal oscillation frequency and displacement response characteristics, where higher frequency oscillations correspond to steeper displacement transitions, while periods of lower oscillation frequency enable the system to reach peak displacement values.
- The systematic nature of the prediction errors suggests opportunities for model refinement, particularly in capturing the behavior at points of high curvature in the displacement curve.

These findings have significant implications for precision control applications. The demonstrated relationship between input signal oscillation patterns and displacement behavior provides a foundation for developing more sophisticated control strategies. Rather than relying solely on power magnitude adjustments, manipulating the oscillation frequency characteristics of the input signal offers an additional dimension for achieving precise displacement control.

Future work should focus on refining the prediction model to address the systematic errors observed at transition points and investigating the influence of varying oscillation patterns on displacement behavior across different operating conditions. Additionally, implementing a real-time control system based on these findings could lead to enhanced precision in applications requiring sub-micrometer positioning accuracy.

In conclusion, this research advances our understanding of micro-displacement prediction and control, establishing a foundation for more precise and responsive systems in fields ranging from precision manufacturing to advanced scientific instrumentation.

Bibliography

- [1] M. Nikolic, K. Bertling, Y. L. Lim, T. Bosch, *et al.*, “Laser feedback interferometry: A tutorial on the self-mixing effect for coherent sensing,” *Advances in Optics and Photonics*, vol. 7, pp. 570–631, 2015. doi: 10.1364/AOP.7.000570.
- [2] A. A. Siddiqui, U. Zabit, O. D. Bernal, G. Raja, and T. Bosch, “All analog processing of speckle affected self-mixing interferometric signals,” *IEEE Sensors Journal*, vol. 17, no. 18, pp. 5892–5899, 2017. doi: 10.1109/JSEN.2017.2734565.
- [3] S. Barland and F. Gustave, “Convolutional neural network for self-mixing interferometric displacement sensing,” *Optics Express*, vol. 29, no. 8, pp. 11 433–11 444, 2021. doi: 10.1364/OE.424694.
- [4] B. Lim, S. Ö. Arik, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” in *Proceedings of the International Journal of Forecasting*, arXiv:1912.09363, Elsevier, 2021.
- [5] A. A. Siddiqui, U. Zabit, and O. D. Bernal, “Fringe detection and displacement sensing for variable optical feedback-based self-mixing interferometry by using deep neural networks,” *Sensors*, vol. 22, no. 24, p. 9831, 2022. doi: 10.3390/s22249831.
- [6] S. Khurshid, U. Zabit, O. D. Bernal, and T. Bosch, “Augmentation-assisted robust fringe detection on unseen experimental signals applied to optical feedback interferometry using a deep network,” *IEEE Transactions on Instrumentation and Measurement*, vol. 72, p. 2 508 110, 2023.
- [7] Q. Li, L. Quan, W. Xia, and D. Guo, “Laser self-mixing interferometry for direct displacement reconstruction using deep learning,” in *SSRN Preprint*, 2023. [Online]. Available: <https://ssrn.com/abstract=5223164>.