Optimization of Sign Language Numeral Recognition from Glove Sensor Data: A Comparative Study of Gradient Boosted Trees, Deep Neural Networks, and Feature Selection

Benedict Michael Pepper and Gilbetch Ronaldo Triswanto
Informatics Engineering Study Program, Ma Chung University, Malang, Indonesia
312310007@student.machung.ac.id; 312310018@student.machung.ac.id
Supervisor: Prof. Dr.Eng. Romy Budhi Widodo

Abstract—Sign language recognition systems for the deaf and mute community face challenges in differentiating hand gestures with high morphological similarity, differing only in subtle finger flexion variations and spatial orientation. This paper advances the k-Nearest Neighbors (kNN) baseline by Widodo (2022), which achieved 98.90% classification accuracy on 20 SIBI numeral gestures using a 16-feature glove sensor dataset, but exhibited weaknesses on ambiguous classes (Class 16: F1=0.97, Class 17: F1=0.95). We investigate whether Extreme Gradient Boosting (XGBoost) or Multi-Layer Perceptron (MLP) trained with PyTorch can surpass baseline accuracy, and whether the original reduction from 16 to 11 features is empirically justified. Exploratory data analysis via ExtraTrees reveals that previously excluded magnetometer features carry higher predictive information gain than the retained accelerometer features. After Bayesian hyperparameter optimization via Optuna (100 trials), XGBoost trained on the full 16-feature set achieves 99.83% test accuracy (a 0.93 pp improvement over baseline), resolving Class 16/17 ambiguity with F1-scores of 0.9972 and 0.9978. The optimized MLP achieves 99.08%. Results confirm that decision tree ensemble methods empirically outperform both distance-based and neural network approaches on dense tabular sensor data.

Index Terms—*Sign language recognition, XGBoost, Multi-Layer Perceptron, glove sensor, feature selection, Bayesian optimization, SIBI.*

I. Introduction

Communication barriers between the deaf and mute community and the general public represent a significant challenge in realizing an inclusive society. In Indonesia, the prevalence of speech and hearing impairment is reported at 0.11%–0.15% of the population, with a combined estimate of 36,959 individuals based on the 2012 national disability census. This community primarily communicates through the Sistem Isyarat Bahasa Indonesia (SIBI), a standardized sign language system adopted in 1994 and used throughout Type-B Special Schools nationwide.

Automated Sign Language Recognition (SLR) systems provide a technological approach to bridging this communication gap. By translating hand gestures into text or speech in real-time, SLR systems can facilitate spontaneous interaction without requiring the general public to learn sign language. Feasibility of deployment, particularly in resource-constrained or mobile settings, critically depends on the accuracy, inference speed, and computational efficiency of the underlying machine learning model.

A. Hardware Approach: Sensor Glove vs. Computer Vision

Sign language recognition broadly falls into two hardware paradigms: vision-based systems leveraging cameras or depth sensors, and wearable sensor-based systems using instrumented gloves. Camera-based systems are fundamentally constrained by lighting, fixed viewpoints, and occlusion. Sensor glove systems overcome these limitations by capturing direct physical measurements from finger joints and hand orientation via attached transducers. For the specific task of SIBI numeral classification, where the gesture set is fully distinguishable through hand configuration alone, a sensor glove is suitable choice.

B. Baseline Research and Identified Limitations

The foundational research directly developed in this study is the monograph by Widodo (2022) [1], which applied a kNN classifier to classify all 20 SIBI numeric signs using a glove equipped with five Flexpoint bend sensors (10 bend sensor channels x1–x10) plus a KMX62 MEMS IMU providing a 3-axis accelerometer and 3-axis magnetometer (x11–x16), for 16 total features. The baseline kNN was trained on a manually selected 11-feature subset, excluding x1, x7, x14, x15, x16, an empirical rather than analytical decision. The classifier achieved 98.90% overall validation accuracy but exhibited systematic under-confidence on Class 16 (F1=0.97) and Class 17 (F1=0.95).

C. Research Questions and Contributions

This study addresses: (1) Can XGBoost and MLP outperform the kNN baseline on the same glove sensor dataset? (2) Is the original reduction from 16 to 11 features empirically justified? Contributions include: rigorous EDA using ExtraTrees to objectively assess all 16 sensor channels; implementation of XGBoost and PyTorch-based MLP; a controlled ablation study comparing 11-feature and 16-feature configurations; Bayesian hyperparameter optimization via Optuna; and in-depth per-class analysis resolving Class 16/17 ambiguity.

II. Related Work

A. Evolution of Sign Language Recognition

Cooper, Holt, and Bowden (2011) [6] surveyed the evolution from frame-based classification to continuous sign sequence recognition, noting that isolated gesture classification, as in this study, can be treated as a standard multi-class classification problem once a reliable feature representation is established. Ambar et al. (2023) [8] demonstrated a wearable glove combining flexible sensors with an inertial measurement unit for real-time sign language translation, achieving successful recognition of 13 sign gestures and demonstrating that bend sensor and IMU data provide complementary discriminative information in glove-based systems.

B. Gradient Boosted Trees and Deep Learning for Tabular Data

XGBoost, introduced by Chen and Guestrin (2016) [2], extends the gradient boosted decision tree (GBDT) framework with second-order Taylor expansion objectives, L1/L2 regularization, and a histogram-based greedy algorithm. It performs implicit feature selection through splitting criteria, unlike kNN which computes distances uniformly. For deep learning, Grinsztajn et al. [11] have shown that on tabular benchmarks with fewer than 10,000 features, gradient boosted trees consistently outperform deep learning models when data is neither sequential nor spatial.

C. Bayesian Hyperparameter Optimization with Optuna

Optuna, developed by Akiba et al. (2019) [4], implements the Tree-Parzen Estimator (TPE) sampler, a form of Bayesian optimization that models the distribution of hyperparameter configurations yielding good versus poor objective values, allocating more evaluation budget to promising regions of hyperparameter space compared to grid or random search.

III. Dataset and Preprocessing

A. Hardware Architecture and Data Acquisition

The dataset was collected using a wearable glove with two sensor types: (1) Five Flexpoint bend sensors placed along the

dorsal surface, one per finger, producing two readings per finger at the MCP and PIP joints, yielding 10 bend sensor channels (x1–x10). (2) A KMX62 MEMS IMU providing a 3-axis accelerometer (x11–x13) measuring gravitational acceleration decomposition, and a 3-axis magnetometer (x14–x16) measuring Earth's ambient magnetic field decomposition along local sensor axes.

TABLE I. SENSOR FEATURE MAPPING

Feature	Name	Anatomy	Type	In Baseline
x1	Pinky MCP	Pinky - MCP Joint	Bend Sensor	No
x2	Pinky PIP	Pinky - PIP Joint	Bend Sensor	Yes
x3	Ring MCP	Ring - MCP Joint	Bend Sensor	Yes
x4	Index PIP	Index - PIP Joint	Bend Sensor	Yes
x5	Middle MCP	Middle - MCP Joint	Bend Sensor	Yes
x6	Middle PIP	Middle - PIP Joint	Bend Sensor	Yes
x7	Index MCP	Index - MCP Joint	Bend Sensor	No
x8	Thumb MCP	Thumb - MCP Joint	Bend Sensor	Yes
x9	Ring PIP	Ring - PIP Joint	Bend Sensor	Yes
x10	Thumb PIP	Thumb - PIP Joint	Bend Sensor	Yes
x11	Accel X	IMU - X Axis	Accelerometer	Yes
x12	Accel Y	IMU - Y Axis	Accelerometer	Yes
x13	Accel Z	IMU - Z Axis	Accelerometer	Yes
x14	Magneto X	IMU - X Axis	Magnetometer	No
x15	Magneto Y	IMU - Y Axis	Magnetometer	No
x16	Magneto Z	IMU - Z Axis	Magnetometer	No

B. Dataset Characteristics and Integrity Verification

The raw dataset consists of 45,147 samples distributed across 20 classes (SIBI numerals 1–20), stored as a CSV file with 17 columns. A comprehensive integrity audit confirmed: no null/missing values, no duplicate rows, and a near-uniform class distribution with samples per class ranging from 1,802 to 2,867 (mean = 2,257). Fig. 1 shows the class distribution.

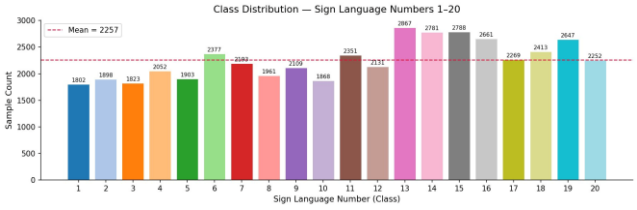

Fig. 1. Class Distribution — Sign Language Numbers 1–20 (Mean = 2,257 samples/class).

C. Exploratory Data Analysis (EDA)

A Pearson correlation heatmap across all 16 features was computed to identify linear redundancy. Most feature pairs exhibit low mutual correlation, confirming largely independent informational content. A notable exception is pair (x1, x4), Pinky MCP and Index PIP, with Pearson r = 0.88. Even highly correlated features can carry complementary discriminative information in non-linear models. Fig. 2 shows the full correlation heatmap.

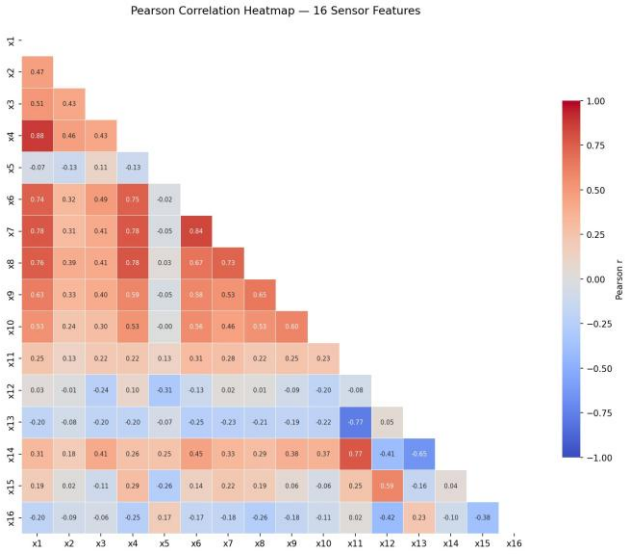

Fig. 2. Pearson Correlation Heatmap — 16 Sensor Features.

To objectively evaluate each feature's predictive contribution, an ExtraTreesClassifier with 200 estimators was trained on the full 16-feature dataset and mean impurity decrease (Gini importance) was extracted. As shown in Fig. 3, magnetometer axes (x14, x15, x16) ranked among the highest-importance features, while the accelerometer axes retained in the 11-feature subset occupied the bottom ranks, which directly contradicts the baseline feature selection strategy.

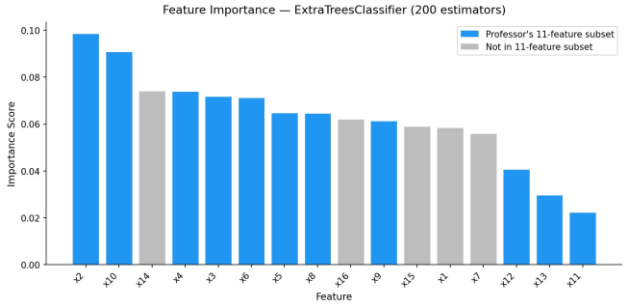

Fig. 3. Feature Importance via ExtraTreesClassifier (200 estimators) — All 16 Features.

Box plots of all 16 features, stratified by class, were generated to visually inspect sensor reading separation across 20 gestures. Fig. 4 shows all 16 feature distributions per class. Features such as x5 (Middle Finger MCP) and x2 (Pinky PIP) exhibit clear multimodal distributions with distinct class centroids. Magnetometer features display more overlapping per-class distributions, but their orthogonal spatial encoding provides additive discriminative value.

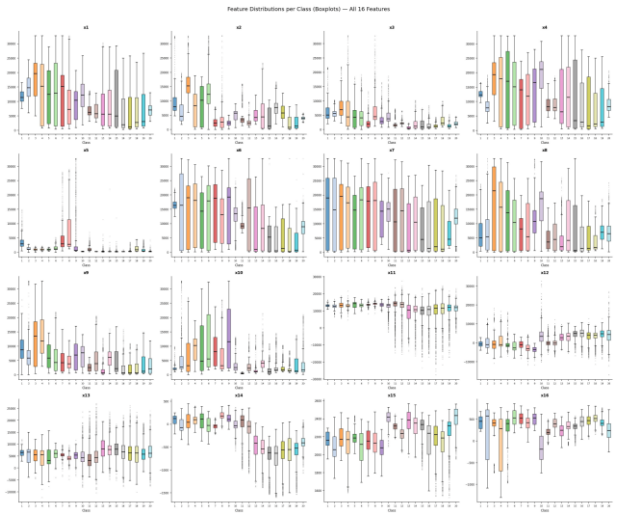

Fig. 4. Feature Distributions per Class (Box Plots) — All 16 Features (x1–x16).

D. Preprocessing Pipeline

The 45,147-sample dataset was partitioned using stratified random splitting (random_state=42). Table II summarizes partition sizes. StandardScaler was applied exclusively to MLP inputs (fitted on training only, then applied to validation/test), preventing data leakage. XGBoost required no scaling due to its monotone transformation invariance. Class labels were converted from 1-indexed (1–20) to 0-indexed (0–19) for PyTorch and XGBoost compatibility.

TABLE II. DATASET PARTITION SUMMARY

Partition	Samples	Percentage	Purpose
Training	27,087	60%	Model fitting
Validation	9,029	20%	Hyperparameter selection & early stopping
Test	9,030	20%	Final evaluation only

IV. Methodology

A. Experimental Design

Two independent parallel experiments were conducted. Experiment A: both XGBoost and MLP trained on the 11-feature subset (x2, x3, x4, x5, x6, x8, x9, x10, x11, x12, x13) replicating the original selection, establishing an architecture-controlled comparison against the kNN baseline. Experiment B: both models retrained using the full 16-feature set, adding the five excluded channels (x1, x7, x14, x15, x16). Within each experiment, a baseline phase (default hyperparameters) and a tuning phase (Bayesian optimization via Optuna) were executed.

B. MLP Architecture (PyTorch)

A fully-connected MLP with pyramidal compression was implemented using PyTorch v2.0+ [3]. The baseline architecture consists of an input layer (D neurons, D ∈ {11, 16}), three hidden layers of 256, 128, and 64 neurons respectively (each with BatchNorm, ReLU, and Dropout p=0.3), and a 20-neuron output layer. Training used the Adam optimizer (lr = 1×10^{-3}), CrossEntropyLoss, mini-batch size 64, and early stopping with patience 15. The baseline run converged at epoch 79 (of max 100) with validation accuracy 98.50%. Post-Optuna, the optimal 16-feature configuration used hidden layer sequence [128, 64, 256] with dropout p=0.3.

C. XGBoost Architecture

XGBoost builds a sequential ensemble of decision trees. At each boosting round, a new tree is fitted to the negative gradient (pseudo-residuals) of the current ensemble's loss function, concentrating capacity on misclassified samples. Table III presents the baseline hyperparameter configuration.

TABLE III. XGBOOST BASELINE HYPERPARAMETER CONFIGURATION

Parameter	Value	Rationale
n_estimators	300	Number of sequential trees in ensemble
learning_rate	0.1	Shrinks each tree's contribution for better generalization
max_depth	6	Maximum depth per tree, controls complexity
subsample	0.8	Stochastic row subsampling per tree to reduce variance
colsample_bytree	0.8	Stochastic feature subsampling per tree
objective	multi:softprob	Probabilistic multi-class output
eval_metric	merror	Multi-class classification error rate
tree_method	hist	Histogram-based approximate greedy algorithm

D. Bayesian Hyperparameter Optimization via Optuna

Optuna's TPE sampler executed 100 trials per architecture-feature-set combination, systematically identifying optimal hyperparameter configurations. Table IV presents the full search space. Figs. 5 and 6 show the Optuna optimization history for XGBoost and MLP, respectively, illustrating convergence behavior across trials.

TABLE IV. OPTUNA HYPERPARAMETER SEARCH SPACE

Model	Hyperparameter	Range	Scale
XGBoost	n_estimators	[50, 200]	Integer
XGBoost	learning_rate	[0.05, 0.20]	Log-uniform
XGBoost	max_depth	[3, 6]	Integer
XGBoost	subsample	[0.6, 1.0]	Uniform
XGBoost	colsample_bytree	[0.6, 1.0]	Uniform
MLP	learning_rate	[1e-4, 5e-3]	Log-uniform
MLP	dropout_rate	[0.1, 0.4]	Uniform
MLP	hidden_layer_config	{[256,128,64], [128,64,32], [128,64,256], [256,64,128]}	Categorical
MLP	batch_size	{32, 64, 128}	Categorical

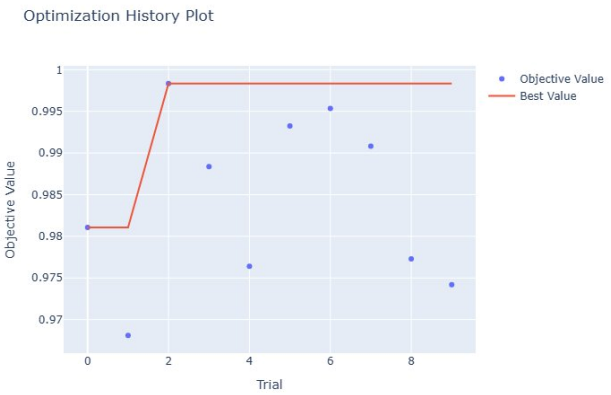

Fig. 5. Optuna Optimization History — XGBoost.

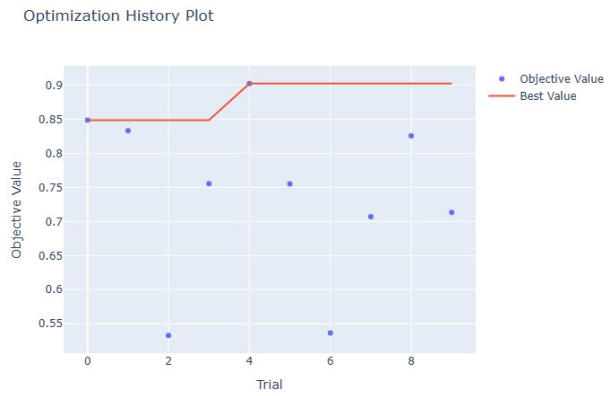

Fig. 6. Optuna Optimization History — MLP.

E. Evaluation Protocol

All reported final performance metrics derive exclusively from the held-out test set (9,030 samples), not accessed during any model development phase. Computed metrics include: global accuracy, per-class Precision/Recall/F1-Score for all 20 classes, Macro-Averaged F1 (unweighted average), and Normalized Confusion Matrix (rows normalized by true class support).

V. Results

A. Baseline Performance (11-Feature Subset, Experiment A)

The XGBoost baseline, configured with default parameters and trained in approximately 11.5 seconds on CPU, before any hyperparameter optimization.

The MLP baseline achieved 98.50% validation accuracy, below the kNN reference, as MLPs on tabular data are known to underperform without tuning. The model trained for 94 epochs (early stopping, best at epoch 79) in 257 seconds on CPU. Table V summarizes baseline results.

TABLE V. BASELINE RESULTS: 11-FEATURE SUBSET (EXPERIMENT A)

Model	Val. Accuracy	F1 Class 16	F1 Class 17
kNN (Baseline Reference)	98.90%	0.9700	0.9500
XGBoost Baseline	99.53%	0.9898	0.9878
MLP Baseline	98.50%	0.9619	0.9569

XGBoost baseline feature importance on 11 features (Fig. 7) confirmed EDA findings: x5 (Middle Finger MCP) emerged as most important (importance 0.213), followed by x2 (Pinky PIP, 0.126). Notably, x11 (Accelerometer X) and x13 (Accelerometer Z) ranked last (0.038 and 0.036), confirming that channels retained in the baseline at the expense of magnetometer channels provided the least discriminative information.

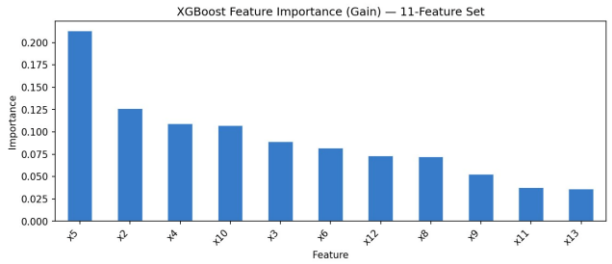

Fig. 7. XGBoost Feature Importance (Gini Gain) — 11-Feature Set.

B. MLP Training Dynamics

The MLP baseline training curves (Fig. 8) show ideal convergence: rapid learning in the first 20 epochs followed by gradual refinement. No catastrophic overfitting was observed; the training/validation loss gap remained small throughout, indicating effective dropout regularization. Fig. 9 shows the final tuned MLP (16-feature) training curves, which converged faster and reached higher final validation accuracy than the baseline.

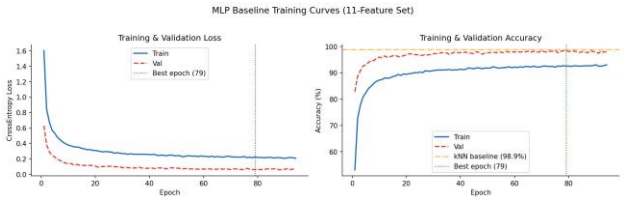

Fig. 8. MLP Baseline Training Curves (11-Feature Set) — Loss and Accuracy per Epoch.

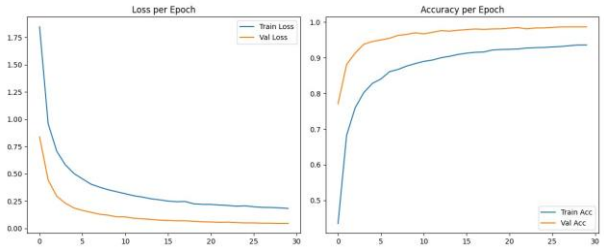

Fig. 9. MLP Final Tuned Training Curves (16-Feature Set) — Loss and Accuracy per Epoch.

C. Full Model Comparison After Tuning

Following Optuna optimization, a comprehensive comparison across all model-feature-set combinations was evaluated on the held-out test set, as summarized in Table VI. XGBoost on the full 16-feature set achieves the best performance at 99.83% test accuracy, surpassing the baseline by 0.93 pp.

TABLE VI. COMPLETE RESULTS: ALL MODELS AND FEATURE SETS (TEST SET)

Model	Feature Set	Test Acc.	F1 Cls.16	F1 Cls.17
kNN (Baseline Ref.)	11 features	98.90%	0.9700	0.9500
XGBoost (Tuned)	11 features	99.60%	0.9898	0.9878
XGBoost (Tuned)	16 features	99.83%	0.9972	0.9978
MLP Baseline	11 features	97.90%	0.9619	0.9569
MLP (Tuned)	16 features	99.08%	0.9662	0.9604

D. Confusion Matrix Analysis

Figs. 10–13 present normalized confusion matrices for all model-feature-set combinations, providing visual evidence of where each classifier's errors concentrate.

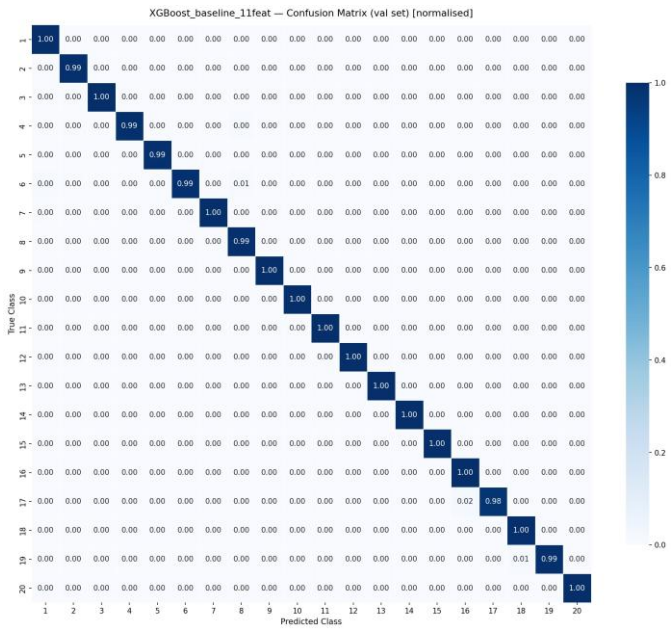

Fig. 10. Confusion Matrix — XGBoost Baseline, 11-Feature Set (Validation Set, Normalized).

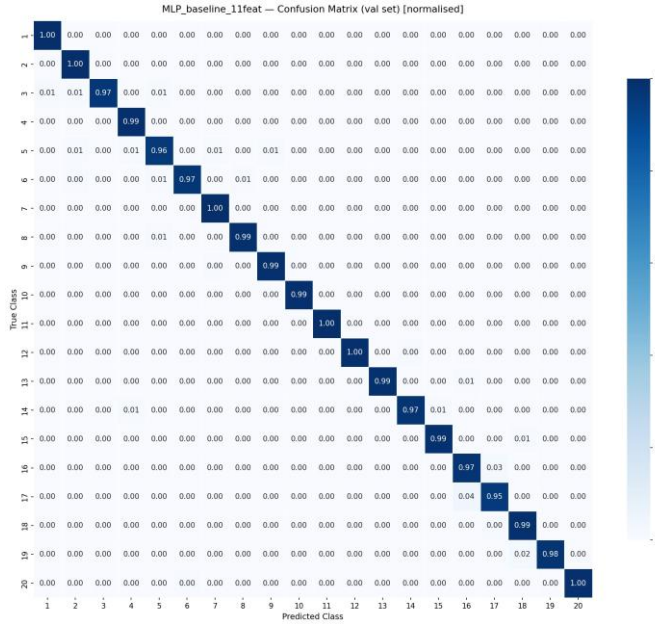

Fig. 11. Confusion Matrix — MLP Baseline, 11-Feature Set (Validation Set, Normalized).

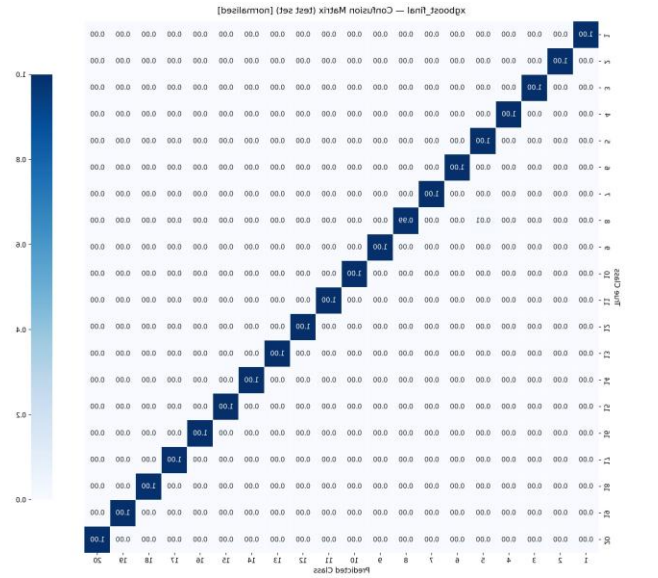

Fig. 12. Confusion Matrix — XGBoost Final Tuned, 16-Feature Set (Test Set, Normalized).

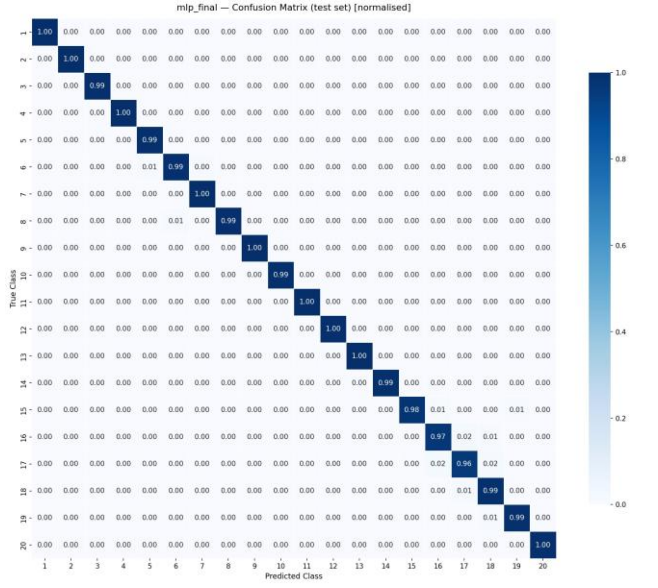

Fig. 13. Confusion Matrix — MLP Final Tuned, 16-Feature Set (Test Set, Normalized).

The final XGBoost confusion matrix (Fig. 12) displays a near-perfect diagonal with negligible off-diagonal probability mass. The final MLP confusion matrix (Fig. 13) reveals residual confusion among Classes 15-17, where approximately 0.03-0.04 probability mass appears in off-diagonal cells between adjacent class pairs, which explains its lower F1-scores for these classes compared to XGBoost.

VI. Discussion

A. Resolution of Class 16/17 Ambiguity

In SIBI, numeral 16 involves the index finger pointing upward with the thumb extended sideways, while numeral 17 involves a similar configuration with a slight variation in thumb overlap and minor forward index tilt. These micro-gestures produce sensor readings occupying geometrically adjacent regions in 16-dimensional feature space, causing Euclidean-distance-based classifiers (kNN) to struggle at establishing hard boundaries.

XGBoost's decision tree mechanism resolves this ambiguity by learning axis-aligned splitting conditions that can isolate specific sensor channels and value thresholds. A representative rule might be: 'If Magneto_Z > threshold$_1$ AND Index_PIP < threshold$_2$, predict Class 16.' The result is quantifiable: from F1=0.97/0.95 in kNN to F1=0.9972/0.9978 in the 16-feature XGBoost, representing near-complete resolution of the Class 16/17 ambiguity.

B. Critical Role of Magnetometer Data

The baseline's decision to exclude magnetometer channels was based on practitioner intuition; accelerometers were considered more physically interpretable while magnetometers were viewed as potentially noisy. Our EDA (Fig. 3) and experimental results contradict this assumption.

The physical distinction between these sensor types is as follows: accelerometers measure gravitational field decomposition along local hand axes, making them sensitive to static tilt but insensitive to yaw rotation. Magnetometers measure Earth's magnetic field decomposition along local axes, providing absolute compass heading and encoding the hand's rotational orientation in the horizontal plane. For SIBI numeric signs differentiated primarily by horizontal wrist rotation, and among the available sensor channels, the magnetometer is the only one encoding horizontal wrist rotation as a direct measurement.

Quantitatively: the improvement from 11 to 16 features on XGBoost yields +0.23 pp accuracy (99.60% → 99.83%), F1 Class 16 +0.0074, F1 Class 17 +0.0100. For MLP, the improvement is larger (+1.18 pp: 97.90% → 99.08%), reflecting that XGBoost can partially compensate for missing features through its ensemble mechanism, while MLP depends more directly on input completeness.

C. Tree-Based vs. Neural Network Approaches

This study adds to the growing body of evidence that gradient boosted tree methods empirically outperform feed-forward neural networks on structured tabular sensor data as encountered in wearable assistive devices. The performance hierarchy XGBoost > MLP > kNN is consistent across all feature set configurations. XGBoost's advantage on this dataset stems primarily from its inherent feature selection through splitting criteria and its efficient non-linear decision boundaries via rectangular space partitioning. Additionally, tree ensembles are less sensitive to irrelevant input features than distance-based or gradient-descent methods.

D. Research Limitations

The dataset contains only isolated static gestures; real sign language communication involves continuous sequences with co-articulation effects. All data was collected from a single glove design, so generalization to other designs is unknown. Cross-user variability was not explicitly analyzed. The Optuna budget was limited to 100 trials per configuration. The feature importance

hierarchy (magnetometer > accelerometer) reflects SIBI numeric gesture geometry and may not generalize to other sign language systems.

VII. Conclusion

This paper presented a systematic comparative evaluation of kNN (baseline), XGBoost, and MLP for classifying 20 SIBI sign language numerals from a 16-channel glove sensor dataset. Principal findings are:

(1) XGBoost outperforms kNN in all configurations; even with 11 features, optimized XGBoost achieves 99.60% test accuracy (+0.70 pp over baseline). (2) The full 16-feature set consistently outperforms: XGBoost achieves 99.83% and MLP achieves 99.08% when all sensor channels are included. (3) XGBoost resolves Class 16/17 ambiguity with F1-scores of 0.9972 and 0.9978, compared to 0.97 and 0.95 in baseline. (4) Gradient boosted trees outperform neural networks on this tabular sensor data. (5) The original feature reduction is suboptimal; magnetometer channels carry higher discriminative value than the retained accelerometer channels.

The most direct extension is TinyML deployment, converting the trained XGBoost model to C/C++ for microcontroller execution. A more fundamental limitation is the static-gesture constraint; future work should address continuous sign sequence recognition using LSTM or Transformer architectures. Cross-subject validation protocols are also needed before real-world deployment.

Acknowledgment

The authors thank Prof. Dr.Eng. Romy Budhi Widodo for providing the original glove sensor dataset and his foundational research that motivated this comparative study. This work was conducted as part of the Applied Intelligent System practicum, Informatics Engineering Study Program, Ma Chung University, 2025.

References

[1] R. B. Widodo, Machine Learning Metode k-Nearest Neighbors: Klasifikasi Angka Bahasa Isyarat, Malang: Media Nusa Creative, 2022, ISBN 978-602-462-907-6.

[2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD, San Francisco, CA, USA, Aug. 2016, pp. 785–794.

[3] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in NeurIPS, vol. 32, Dec. 2019, pp. 8024–8035.

[4] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in Proc. 25th ACM SIGKDD, Anchorage, AK, USA, Aug. 2019, pp. 2623–2631.

[5] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis," Intelligent Systems with Applications, vol. 12, p. 200056, Dec. 2021.

[6] H. Cooper, B. Holt, and R. Bowden, "Sign Language Recognition," in Visual Analysis of Humans, T. Moeslund et al., Eds. London: Springer, 2011, pp. 539–562.

[7] M. S. Amin, S. T. H. Rizvi, and M. M. Hossain, "A Comparative Review on Applications of Different Sensors for Sign Language Recognition," J. Imaging, vol. 8, no. 4, p. 98, Apr. 2022.

[8] R. Ambar, S. Salim, M. H. Abd Wahab, M. M. Abdul Jamil, and T. C. Phing, "Development of a Wearable Sensor Glove for Real-Time Sign Language Translation," Annals of Emerging Technologies in Computing (AETiC), vol. 7, no. 5, pp. 25–38, Oct. 2023.

[9] Depdiknas RI, Kamus Sistem Isyarat Bahasa Indonesia (SIBI), Jakarta: Departemen Pendidikan Nasional, 1994.

[10] M. Malika and Berlianti, "Penggunaan Bahasa Isyarat SIBI dan BISINDO untuk Siswa Tuli di Sekolah Luar Biasa Negeri Pembina Medan," Concept: Journal of Social Humanities and Education, vol. 4, no. 3, pp. 78–87, Sep. 2025.

[11] Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). "Why tree-based models still outperform deep learning on tabular data." NeurIPS 2022.