IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 April 2025, accepted 13 May 2025, date of publication 21 May 2025, date of current version 30 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3572092

RESEARCH ARTICLE

An FPGA Prototype for Parkinson's Disease Detection Using Machine Learning on Voice Signal

MUJEEV KHAN[1], ABDUL MOIZ[1], GANI NAWAZ KHAN[1], MOHD WAJID[1], (Senior Member, IEEE), MOHAMMED USMAN[2], (Senior Member, IEEE), AND JABIR ALI[3]

[1]Department of Electronics Engineering, ZHCET, Aligarh Muslim University, Aligarh 202002, India
[2]Department of Electronics and Communication Engineering, Bennett University, Greater Noida 201310, India
[3]School of Computer Science Engineering and Technology, Bennett University, Greater Noida 201310, India

Corresponding author: Gani Nawaz Khan (gn2389@myamu.ac.in)

ABSTRACT Parkinson's disease (PD) is a chronic neurological disorder caused by a reduction in dopamine levels in the brain. Early diagnosis is crucial for effective treatment. This paper proposes an efficient machine learning model for PD detection using voice-based features, which offer a non-invasive, cost-effective, and accessible alternative to complex imaging methods. To enhance classification performance and reduce computational complexity, we evaluate three feature selection algorithms — Chi-squared (χ^2), Minimum Redundancy Maximum Relevance (mRMR), and Analysis of Variance (ANOVA) — and adopt an incremental feature selection approach, where each feature set increment is assessed across five classifiers: K-Nearest Neighbors (KNN), Decision Tree (DT), Artificial Neural Network (ANN), Logistic Regression (LR), and Support Vector Machine (SVM). To address dataset imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is applied. Among the evaluated approaches, the ANOVA algorithm yields the most optimal feature set, with the top 10 features enabling the Quadratic SVM to achieve a classification accuracy of 98.86%. The Quadratic SVM, optimized for minimal power consumption, is implemented on a Nexys A7 FPGA. This setup achieves a dynamic power consumption of just 23 mW and delivers a performance acceleration of 155× compared to equivalent computations on a 6th-generation Intel i5 processor. By combining a high-accuracy machine learning model with an energy-efficient FPGA implementation, our approach offers a powerful and portable solution for real-time PD detection.

INDEX TERMS ANOVA, chi-squared, efficient architecture, feature selection, FPGA, hardware design, low power, machine learning, mRMR, Parkinson's disease, quadratic kernel SVM, SMOTE.

I. INTRODUCTION

Parkinson's disease (PD) is a progressive central nervous system disorder that affects the human motor system, causing a variety of motor and non-motor symptoms [1], [2]. The condition arises due to the degeneration of neurons responsible for producing dopamine in the substantia nigra area of the brain [2], [3]. Dopamine is primarily produced in the substantia nigra, a region of the brain crucial for various functions, including movement control, cognitive function, and emotional regulation [4], [5]. Normally, dopamine aids in synchronizing the coordination of many nerve and muscle cells engaged in movement, working alongside other neurotransmitters in a delicate balance. The lack of dopamine disrupts this balance, causing symptoms like tremors in the hands, arms, legs, and jaw; rigidity in the limbs; slowed movements; and challenges with balance and coordination [5], [6], [7]. PD is known as one of the most common movement disorders in people over 60 years old [5]. By 2019, global projections indicated a population exceeding 8.5 million individuals affected by PD, marking an 81% increase since 2000, along with 329,000 fatalities, signifying a more than 100% rise since 2000 [8]. PD has no cure, although treatments such as medications, surgery, and rehabilitation can help alleviate symptoms [8], [9].

The PD diagnosis primarily relies on the common symptoms mentioned above [5], [10]. No X-rays or blood tests can diagnose the condition. Doctors can receive diagnostic

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Quan.

assistance from noninvasive diagnostic imaging procedures such as positron emission tomography (PET). The majority of people with PD depend on medicine to reduce their symptoms [5]. Voice disorder analysis is a non-invasive method for early detection of PD [11], [12], as 90% of patients exhibit dysphonia or voice abnormality [13], [14], distinguishing them from healthy individuals. Diagnosing PD through vocal abnormalities is a promising and useful approach [12], [13], [15]. Early diagnosis is crucial for timely disease detection and treatment [13]. There are no reliable biomarkers for early detection of PD [16], hence, it is vital to utilize artificial intelligence (AI) methods to assist the healthcare industry in accurately and timely diagnosing PD [17]. As a result, creating a hardware-assisted diagnostic system is essential for analyzing voice data and differentiating between individuals with PD from those who are healthy [12], [13]. Despite significant efforts by researchers to achieve satisfactory results in diagnosing PD, effectively implementing the developed algorithms on hardware while maintaining high accuracy remains a challenge. These drawbacks have limited the adoption of PD detection systems in clinical settings. This is the first paper, to the best of the author's knowledge, that proposes a novel machine-learning technique to optimize early PD detection using voice data to differentiate between normal patients and patients affected by PD and realizing the predictive model on field-programmable gate arrays (FPGA), aiding neurologists in making informed diagnostic decisions. Recent advancements in semiconductor technology have made implementing PD detection algorithms in hardware feasible. Many researchers are increasingly considering field-programmable gate arrays (FPGAs) as the target platform for implementing these types of algorithms due to their reconfigurability. Additionally, this approach can be extended to application-specific integrated circuits (ASICs) for higher energy efficiency and smaller chip area. Prototyping and testing on FPGAs offer a practical and efficient way to develop and refine these designs before transitioning to ASICs for specific, fixed applications.

Our research focuses on developing a hardware-friendly and robust PD classification model with high responsiveness and minimal false detection. We prototype the developed PD classification model on FPGAs, demonstrating real-time and energy-efficient computation. This paper makes the following key contributions:

1) Application of an incremental feature selection approach based on feature ranking by various feature selection algorithms, followed by the training and evaluation of five classification models using the best-selected features.
2) Energy-efficient implementation of the classification model on FPGA for real-time application for PD detection.

The remaining portions of this paper are arranged as follows: Section II provides related work. Section III outlines the proposed approach for PD detection. Model Training and Assessment in Section IV. Section V explains the hardware prototyping on FPGA. Section VI has the conclusion.

II. RELATED WORK

To differentiate between people with PD and those who are healthy, our research uses data analysis and machine learning techniques to accurately analyze the voice data feature set and develop a detection prototype that is realized on FPGA. Existing literature has focused on using MRI scans, gait analysis, and genetic data to predict Parkinson's disease, but there has been limited research [18], [19], [20] into utilizing voice data for early detection. A convolutional neural network (CNN) was proposed by [21] to analyze the EEG signals of 15 PD patients and 16 healthy people. A method based on spatial patterns for PD diagnosis was created by [22] for both patients receiving medication and those who are not. With characteristics taken from the beta and alpha ranges, the classifier produced the best results, with 95% accuracy. Reference [23] explored various machine learning algorithms on voice data for classifying Parkinson's disease. Among the tested algorithms, the K-Nearest Neighbors (K-NN) classifier performed the best. The K-NN classifier achieved an impressive accuracy of 97.22% and an F1-score of 97.30%. Using an SVM model, [24] looked at genetic data to predict when PD might manifest in senior patients. They trained their SVM model to attain an 88.9% accuracy rate. The output of LDA was used in clustering algorithms since it performed better than PCA. DBSCAN produced results with a 64% accuracy, 78.13% sensitivity, and 38.89% specificity. To detect PD, Moumita et al. [25] developed three techniques utilizing decision-forest and Systematic Forest (SysFor) algorithms with Forest Pattern Analysis (ForestPA) characteristics. Their method achieves high accuracy with only a few Decision trees (DTs). The accuracy of the DT using ForestPA characteristics was 94.12%. Panda et al. [26] proposed the BIMSSA model, combining Boruta, IMRMR, and Salp Swarm Optimization for effective feature selection from microarray gene expression data. Using an ensemble of ML classifiers, the model achieved up to 97.1% accuracy across multiple cancer datasets, including ALL-AML, Lymphoma, MLL, and SRBCT.

An algorithm was presented by Little et al. [13] to diagnose speech difficulties in patients with PD. Their algorithm assessed dysphonia and examined speech patterns to distinguish between people with PD and healthy individuals based on two unique characteristics: *fractal scaling* and *recurrence*. The accuracy achieved by the approach was 91.4%. Four strategies for selecting and classifying salient characteristics were presented by [27]. Their system's accuracy using 10-fold cross-validation (CV) was 68.94%, while leave-one-subject-out cross-validation yielded 57.5%. An SVM was used by [28] to diagnose mixed features with an accuracy of 82.5%. Reference [29] extracted phonemic pronunciation features using various methods and identified Parkinson's disease using several classifiers. Reference [30] proposed an

explainable recursive feature elimination model, achieving 96.46% accuracy by selecting the most relevant speech features. Reference [31] emphasized speech analysis as a non-invasive diagnostic tool, achieving an accuracy of 94.3%. Reference [32] developed a PD detection model using voice recordings and a group-wise scaling method, achieving 82% accuracy on unseen individuals. Few ML techniques for PD detection have achieved 100% accuracy rates [33], [34], but their techniques are excessively complex. Despite the advancements in developing simulation-based classification models, several gaps remain. First, there is a need for models that balance computational complexity and accuracy, ensuring they are both efficient and highly precise. Second, the existing research lacks a hardware prototype, which is crucial for real-time detection. Our work focuses on creating a model with low computational complexity and high accuracy, alongside developing a hardware prototype on voice based features to facilitate real-time detection.

III. PROPOSED APPROACH FOR PARKINSON'S DISEASE DETECTION

The proposed methodology comprises two primary phases. Phase 1 focuses on developing a software algorithm for detecting PD from a voice dataset. To address class imbalance between normal and disease classes, the Synthetic Minority Over-sampling Technique (SMOTE) is applied to generate synthetic samples for the underrepresented healthy class. Feature selection is conducted using three algorithms — Chi-squared (χ^2), Minimum Redundancy Maximum Relevance (mRMR), and Analysis of Variance (ANOVA) — to identify the most relevant features from a set of 22 extracted features. An incremental feature selection approach is employed, where features are added one by one, and each expanded feature set is evaluated based on the classification performance of five different models: K-Nearest Neighbors (KNN), Decision Tree (DT), Artificial Neural Network (ANN), Logistic Regression (LR), and Support Vector Machine (SVM). The classifier achieving the highest accuracy with the smallest feature set is selected for the subsequent hardware implementation phase. In Phase 2, the Support Vector Machine (SVM) with a quadratic kernel is chosen for prototyping on the FPGA, as it achieved the highest accuracy. This architecture is optimized using a bit suppression technique, enhancing its energy efficiency and making it more suitable for battery-powered and portable systems. The proposed design ensures a balanced trade-off between classification accuracy, hardware complexity, and power consumption. Figure 1 presents a visual overview of the essential steps undertaken in this proposed work, including data preprocessing, feature selection, model training and testing, model evaluation, and FPGA prototyping.

A. DESCRIPTION OF DATA SET

This paper employs a voice data based feature set for PD detection, curated by M. Little of the University of Oxford, UK, and accessible in the UCI Machine Learning Repository. Voice recordings were made with a head-worn AKG C420 microphone, 8 cm from the subject's lips, in a noise-controlled booth. A Bruel & Kjaer 2238 Type 1 sound-level meter, placed 30 cm from the speaker, calibrated the microphone. Recordings were captured using Kay Elemetrics' CSL 4300B hardware [35], [36]. The signals were sampled at 44.1 kHz with 16-bit resolution. The dataset comprises 195 sustained vowel phonations collected from 31 participants of both genders. Out of these participants, 23 are diagnosed with PD, while the remaining 8 are healthy individuals. Six recordings, ranging in duration from 1 to 36 seconds, were averaged from each participant. Specifically, the dataset includes 147 phonations from healthy individuals and 48 phonations from participants with PD. The participants' ages ranged from 46 to 85 years with a mean and standard deviation of 65.8 and 9.8 years, respectively. The period between the diagnosis and recording of the vowel phonations varied from 0 to 28 years. Following the recording, traditional features were extracted from the signals using Praat [37], matching those in the Kay Pentax *MDVP* for easier comparison. These features analyze short segments of the voice signal, detecting vocal fold vibrations via short-time autocorrelation and peak detection, and calculating the fundamental frequency *(Fo)* or pitch period. Jitter measures pitch variations, with Jitter (%) indicating relative changes in vibration duration, and Jitter (abs) measuring absolute temporal differences. Shimmer assesses amplitude variations between vocal cycles, estimating the signal-to-noise ratio and deriving noise-to-harmonics *(NHR)* and harmonics-to-noise *(HNR)* ratios. Nontraditional features include the correlation dimension, calculated by embedding the signal with a time delay to recreate the speech signal's nonlinear dynamics. Repetition period density entropy *(RPDE)* quantifies periodicity in this phase space. Detrended Fluctuation Analysis *(DFA)* measures self-similarity in noise within the speech signal, indicating turbulent airflow and potential dysphonia. Pitch Period Entropy (PPE) quantifies pitch irregularities by constructing a probability distribution and calculating entropy. See Table 1 for the details of the derived characteristics from the voice signal. This dataset has been extensively utilized, prepared, and assessed by numerous medical professionals [13].

B. DATA AUGMENTATION

Data pre-processing is the process of identifying and transforming (or eliminating) irregular or corrupt recordings from a dataset [38]. This dataset is free from duplicates, outliers, and other forms of corruption, ensuring its reliability for analysis. To address the class imbalance, we applied SMOTE, a data augmentation technique that oversamples the minority class for a more balanced distribution [39]. Additionally, we observed variations in the dynamic ranges of the voice

features, which could potentially hinder the model's learning and performance. To address this, we normalized all

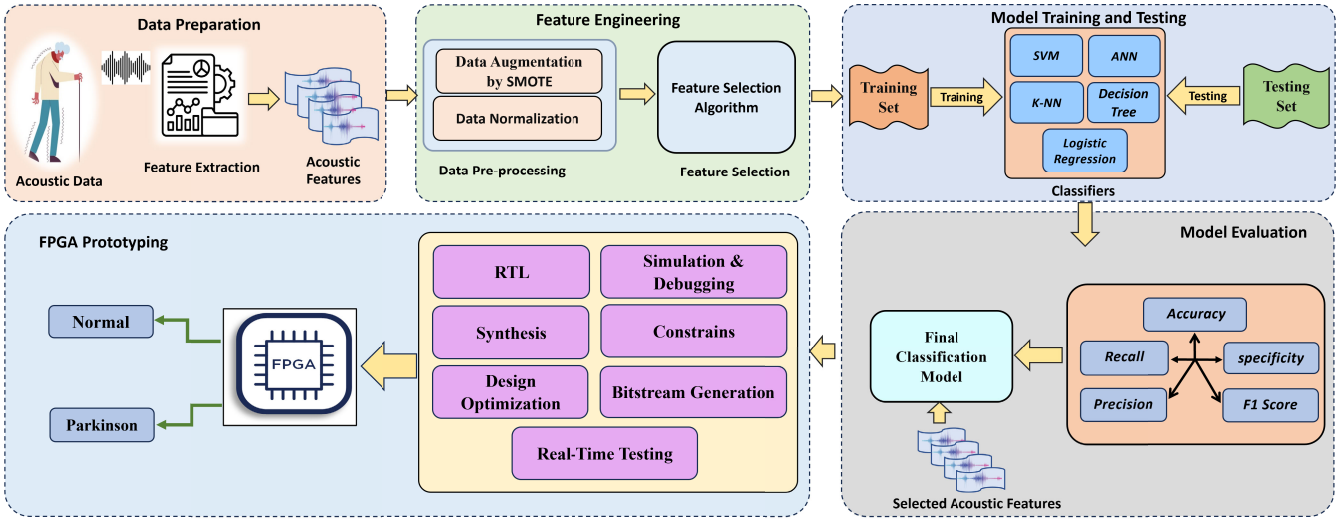

FIGURE 1. The proposed machine learning approach for Parkinson's disease detection, outlining key steps such as feature engineering, model training and testing, model evaluation, and FPGA implementation.

TABLE 1. The PD dataset includes voice measure interpretation.

No.	Feature	Description
1	MDVP :- Jitter (Abs)	Absolute_jitter_in_microseconds
2	DFA	Detrended_Fluctuation_Analysis
3	MDVP :- Shimmer (dB)	local_shimmer_in_decibels
4	MDVP :- PPQ	Five-point_Period_Perturbation_Quotient
5	MDVP :- Jitter (%)	Jitter_as_a_percentage
6	MDVP :- F_o (Hz)	The_average_frequency_ of_fundamental_Frequency
7	Jitter :- DDP	Average_absolute_difference_of_differences_between_cycles , divided_by_the_average period
8	MDVP :- Shimmer	Local_shimmer
9	NHR	Noise_to_Harmonics_Ratio
10	MDVP :- RAP	Relative_Amplitude_Perturbation
11	MDVP :- APQ	11-point_Amplitude_Perturbation_Quotient
12	RDPE	Recurrence_Period_Density_Entropy
13	Shimmer :- DDA	Average_absolute_difference_between_consecutive_differences_between _the_amplitudes_of_consecutive_periods
14	Shimmer :- APQ3	Three_point_Amplitude_Perturbation_Quotient
15	HNR	Harmonics_to_Noise_Ratio
16	Shimmer :- APQ5	Five_point_Amplitude_Perturbation_Quotient
17	MDVP :- F_{low} (Hz)	The_lowest_vocal_fundamental_frequency
18	spread_1 & spread_2	Nonlinear_measures_ of_fundamental_Variation_in_Frequency
19	D_2	Correlation_dimension
20	PPE	Pitch_period_entropy
21	MVDP :- F_{high} (Hz)	The_highest_vocal_fundamental_frequency
22	Status	Subject's_health_status: 1 - Parkinson's, 0 - healthy

features using the min-max normalization technique, which transforms each feature's maximum value to 1, and minimum value to 0, and scales all other values to decimal fractions between 0 and 1 to ensure they were on the same scale. This preprocessing step aids the model in training more effectively and ensures fair treatment of all features.

1) SMOTE

The PD dataset comprises 195 recordings divided into two classes, with an uneven distribution. The healthy class (Class 0) comprises 48 recordings, accounting for 24.62% of the dataset, while the disease class (Class 1) comprises 147 recordings, making up 75.38% of the dataset. This class imbalance can introduce a bias towards the majority class during model training, as the model tends to prioritize learning patterns from the more abundant class. To address this issue, we employed SMOTE. In this technique, synthetic instances are generated for every instance in the minority class by comparing the feature values of the current instance and its nearest neighbor as shown in Eqs. (1) and (2). The difference between these feature values is then added to the current instance's feature value after being multiplied by a number between 0 and 1 which is chosen randomly, as shown in Eqs. (3) and (4). A random point along the line segment connecting the two sets of feature values is chosen [39].

This method efficiently increases the decision boundary of the minority class, leading to better generalization. A simple explanation of how SMOTE augments the data is as follows: Let ($\mathbf{x_1}$,$\mathbf{y_1}$) be the original sample for which we want to generate synthetic instances, and ($\mathbf{x_2}$,$\mathbf{y_2}$) be one of its k-nearest neighbors. The difference between the feature value of the original sample and its nearest neighbor can be represented as:

$$\Delta \mathbf{x} = \mathbf{x_2} - \mathbf{x_1} \tag{1}$$

and

$$\Delta \mathbf{y} = \mathbf{y_2} - \mathbf{y_1} \tag{2}$$

A random number between 0 and 1, denoted as $\mathbf{rand}(0, 1)$, is then generated. The new sample is generated as:

$$\mathbf{x}' = \mathbf{x_1} + \mathbf{rand}(0, 1) \times \Delta \mathbf{x} \tag{3}$$

and

$$\mathbf{y}' = \mathbf{y_1} + \mathbf{rand}(0, 1) \times \Delta \mathbf{y} \tag{4}$$

Our implementation currently utilizes the five nearest neighbors. This choice balances locality and diversity, ensuring that the synthetic samples are sufficiently representative while avoiding noise or overlap with the majority class. Table 2 illustrates the samples of both classes before and after applying SMOTE, demonstrating a balanced distribution where the majority (Parkinson) and minority (healthy) classes are exactly equal.

TABLE 2. The table illustrates the data samples before and after applying the SMOTE technique to achieve dataset balance.

Phase	Training Data set 70%		Testing Data set 30%	
Classes	HC	PD	HC	PD
Before OverSampling	34	103	14	44
After OverSampling	103	103	44	44

C. FEATURE SELECTION

After data augmentation, the next critical step is selecting relevant features from the extracted feature set. Some features may be redundant or irrelevant leading to an expanded feature space size. These undesired features can increase computational complexity in the learning algorithms without improving their classification performance. In this study, we employed three feature selection algorithms to rank features based on their importance scores: *a)* Chi-squared test [40], *b)* mRMR test [41], and *c)* ANOVA test [42]. Subsequently, we utilized an incremental feature selection method on the three algorithms to identify the minimum number of features that will yield the highest performance metric for the trained classifier. The following section explains the feature selection algorithms.

1) CHI-SQUARED TEST

The Chi-squared test evaluates the significance of each feature value by computing the Chi-squared statistic concerning the class. It determines the usefulness of a feature by assessing its chi-squared statistic, which examines the initial hypothesis H_0 regarding the independence of the two features [40]. This hypothesis is examined using the chi-squared formula as depicted in Eq. (5).

$$\chi^2 = \sum \frac{1}{E} \times (O - E)^2 \tag{5}$$

where O represents the observed frequency for a group, while E represents the expected frequency for the group under the assumption of independence. Larger values of χ^2 indicate a greater discrepancy between observed and expected frequencies, suggesting a stronger association between the feature and the class.

2) MRMR TEST

The primary goal of the mRMR (minimum Redundancy Maximum Relevance) technique is to minimize the influence of low-scoring features and select the most relevant ones. By computing the mutual information between each feature in the dataset, this method evaluates the similarity between two features based on the calculated I(X, Y) [41], [43], [44].

$$\mathrm{I}(X, Y) = \sum_{y \in Y} \sum_{x \in X} (x, y) \log \left(\frac{p(x, y)}{p_X(x) p_Y(y)} \right) \tag{6}$$

In this equation, $p(x, y)$ represents the joint probability of X and Y, while $p_X(x)$ and $p_Y(y)$ are the marginal probabilities of X and Y, respectively. For feature selection, let F_i represent individual features within the selected feature subset S. The mutual information $I(F_i, F_j)$ quantifies the relationship between features F_i and F_j, while $I(F_i, H)$ represents the mutual information between a feature F_i and the class label H (the target variable). Eqs. (7) and (8) must be met to ensure that the best features are selected. Eq. (7) aims to minimize

redundancy among features and is defined as:

$$\text{Minimize } W, \quad W = \frac{1}{|S|^2} \sum_{\substack{F_i, F_j \in S \\ i \neq j}} I(F_i, F_j) \tag{7}$$

Here, W represents the redundancy within the feature subset S and the summation computes the mutual information between all pairs of features in the subset. Eq. (8) aims to maximize the relevance of the features with respect to the class label, is expressed as:

$$\text{Maximize } V, \quad V = \frac{1}{|S|} \sum_{F_i \in S} I(F_i, H) \tag{8}$$

In this equation, V represents the total relevance of the features in S with respect to the class label H, and the summation computes the mutual information between each feature and the target label. The feature selection process involves iterating through different subsets of features, evaluating their relevance and redundancy according to Eqs. (7) and (8), and selecting the subset that optimally balances relevance and redundancy.

3) ANOVA TEST

The ANOVA test assesses the mean values of a dataset across multiple groups (classes) and identifies significant differences between them [42]. The F-statistic, serving as the test statistic for ANOVA, is computed through the following steps:

(i) The group variance is computed as follows:

$$BSS = \sum_{k=1}^{K} n_k (\bar{X}_k - \bar{X})^2 \tag{9}$$

where *BSS* stands for "Between sum of squares", n_k is the number of observations in group k, $\bar{X}_k$ is the mean of group k, and $\bar{X}$ is the overall mean across all groups. Next, the "Between mean squares" *BMS* is calculated as:

$$BMS = \frac{BSS}{df_{\text{between}}} \tag{10}$$

where $df_{\text{between}} = K - 1$ represents the degrees of freedom between the groups, and K is the number of groups.

(ii) The variation within the groups is calculated as follows:

$$WSS = \sum_{k=1}^{K} (n_k - 1)\sigma_k^2 \tag{11}$$

where *WSS* stands for "Within sum of squares", and σ_k^2 is the variance (or squared standard deviation) of the group k, while n_k is the number of observations in group k.

The "Within mean squares" *WMS* is calculated as:

$$WMS = \frac{WSS}{df_{\text{within}}} \tag{12}$$

where $df_{\text{within}} = N - K$ is the degrees of freedom within the groups, with N being the total number of observations in the dataset.

(iii) Finally, the F-test statistic is calculated as:

$$F = \frac{BMS}{WMS} \tag{13}$$

where *BMS* is the between mean squares and *WMS* is the within mean squares.

A high F-score suggests that the variance between group means surpasses the variance within groups, indicating a probable significant difference between at least two of the group means.

IV. MODEL TRAINING AND ASSESSMENT

The experiment was conducted on a computer with the Windows operating system, equipped with a 6th-generation Intel Core i5 CPU and 8 GB of RAM. A simulation software package was employed for model development.

A. SPLITTING DATASET

After balancing the feature set, it was divided into two subsets: Set 1 contained 70% of the feature set (103 records for both classes) for training the model, while Set 2 comprised the remaining 30% (44 records for both classes) for testing it [45]. Table 2 illustrates the distribution of dataset samples across the training and testing phases.

B. PERFORMANCE METRICS

Five statistical measures—*Accuracy, Precision, Recall, Specificity*, and *F1 score*—were employed to assess the performance of the classification algorithms on the Parkinson's disease (PD) dataset. These metrics are widely recognized for evaluating the effectiveness of classification models [44], [46].

Extensive experimentation guided the selection of optimal hyperparameters for each classifier. The Artificial Neural Network (ANN) was implemented with a single hidden layer and the ReLU activation function, while the output layer used a sigmoid activation. The K-Nearest Neighbors (KNN) classifier was configured with $K = 5$ and the Euclidean distance metric. Logistic Regression utilized the sigmoid activation function and LBFGS solver with L2 regularization. The Decision Tree classifier followed the CART algorithm and employed Gini's diversity index as the splitting criterion. Support Vector Machines (SVM) were applied using both polynomial (quadratic) and radial basis function (RBF) kernels to capture non-linear patterns. The specific learning parameters used for each classifier are summarized in Table 3.

C. PERFORMANCE EVALUATION

The final feature matrix of the voice dataset consists of a training set comprising a normal feature matrix (NFM) and a Parkinson's feature matrix (PFM), each with dimensions 103×22. Similarly, the test set comprises an NFM and a PFM, each with dimensions 44×22, used for training

TABLE 3. Learning parameters for the classifiers used.

Classifier	Learning Parameters
KNN	Number of neighbors: $K = 5$; Distance metric: Euclidean; Search algorithm: Brute-force; Leaf size: 30; Minkowski power: $p = 2$
ANN	Architecture: Single hidden layer; Activation: ReLU; Output activation: Sigmoid; Optimizer: Adam; Loss: Binary cross-entropy;
LR	Activation: Sigmoid; Solver: LBFGS; Regularization: L2; Regularization strength: $C = 1.0$; Max iterations: 100;
DT	Algorithm: CART; Splitting criterion: Gini index; Other hyperparameters: Default
SVM (Poly)	Kernel: Polynomial (degree = 2); Solver: SMO; Regularization: $C = 1.0$; Kernel scale: Auto ($\gamma \approx 0.1$)
SVM (RBF)	Kernel: RBF; Regularization: $C = 1.0$; Kernel scale: Auto ($\gamma \approx 0.1$)

and testing the model, respectively. Feature selection was conducted on the training set using three feature selection algorithms:χ^2 test, mRMR test, and ANOVA test. These algorithms were employed to rank the features based on their importance scores. Subsequently, the incremental feature selection method was applied to determine the minimum number of features that could yield the highest performance metric on the trained classifiers [47]. To initiate the feature selection process, we first selected the top 2 ranked features from each of the three feature ranking algorithms for which the classification error and accuracy are lower than 0.30 and upper than 70%, respectively. These selected features were then used as input for the classifiers, and the resulting classifier performance was recorded. Following this initial selection, the number of features was systematically increased, based on their ranks, in increments of 2. To avoid over-fitting and optimizing the parameters of different classifiers, a 10-fold cross-validation method was utilized [57], where the training data (206 samples) was divided into 10 folds. Each classifier was trained and validated 10 times, with each fold being used exactly once as the validation set, while the remaining 9 folds served as the training set. It was observed that across the three feature selection algorithms, the classifier's performance peaked after selecting 10-12 features. The impact of feature selection from the three algorithms on the accuracy of the classifiers is shown in Table 6 and is discussed in the following sub-section.

(i) Chi-squared test: The Quadratic SVM achieved the highest performance while utilizing minimal features, achieving an accuracy of 97.7% on the test set with the top 14 features ranked by the Chi-squared test.

(ii) mRMR test: ANN achieved the highest performance with a minimal set of features, reaching a 96.6% accuracy on the test set using the top 10 ranked features identified by the mRMR test.

(iii) ANOVA Test: The best performance with minimal features was achieved by the polynomial SVM of order 2 (Quadratic SVM), which achieved a 98.86% accuracy on the test set using the highest 10 ranked features from the ANOVA test.

We observed two significant findings through incremental feature selection across the three feature selection algorithms. Firstly, the ANOVA feature selection method exhibited superior overall performance across all classifiers, achieving optimal results with a minimal selection of features compared to the other algorithms. Secondly, our analysis revealed that the Quadratic SVM delivered the best performance on the test set, utilizing the minimum number of features.Specifically, the selected features included *MDVP:- APQ*, *HNR, D_2, DFA*, MDVP:- F_{low} *(Hz)*, *RPDE, spread_2*, MVDP:- *Shimmer (dB)*, MDVP:- *PPQ, Spread_1*. The quadratic SVM yielded an accuracy of 98.86%, with a classification error rate of 6.7% as shown in Fig. 2.

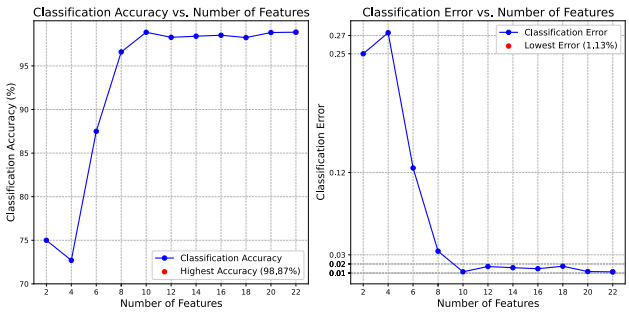

FIGURE 2. Classification accuracy and error of the Quadratic SVM as a function of the number of features.

The confusion matrix for the quadratic SVM model is given in Fig. 3. See Table 5 for the performance metric of different classifiers using the 10 best-selected features

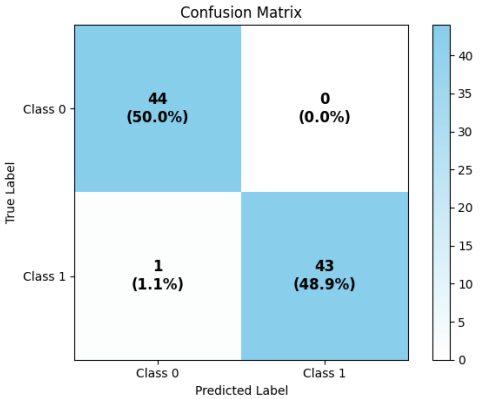

FIGURE 3. Illustration of the Confusion Matrix, displaying the classification results, including true positives, true negatives, false positives, and false negatives, for evaluating the performance of a classifier.

TABLE 4. Table illustrating the effectiveness of our approach relative to prior research, emphasizing advancements and distinctions.

Study	Year	Data Source	No. of Subjects (PD/HC)	Machine Learning Method	Results
[18]	2023	Machine learning repository at UCI	31 (23/8)	LR, RF, SVM, KNN 75% training + 25% testing	RF Acc. = 91.835% Sens. = 0.95
[20]	2022	Machine learning repository at UCI	31 (23/8)	Naïve Bayes, ANN, KNN 80% training + 20% testing	ANN Acc. = 96.7% Sens. = 92.42% Spec. = 91.25%
[48]	2021	Machine learning repository at UCI	252 (188/64)	SVM, LR, KNN, AdaBoost	SVM Acc. = 94.1%
[49]	2020	Machine learning repository at UCI	31 (23/8)	CART, SVM, ANN	SVM Acc. = 93.84%
[50]	2019	Machine learning repository at UCI	31 (23/8)	Random Forest, Boosted C5.0, SVM, RPART, C4.5, PART, Bagging SVM: CART 70% training + 30% testing	Acc. = 97.57% Sens. = 0.9756 Spec. = 0.9987 NPV = 0.9995
[19]	2019	Machine learning repository at UCI	31 (23/8)	EFMM-OneR with 10 Cross Validation	Acc. = 94.21%
[51]	2018	Machine learning repository at UCI	31 (23/8)	NNge with Adaboost with 10-fold Cross Validation	Acc. = 96.30%
[52]	2018	Machine learning repository at UCI	31 (23/8)	Linear Regression, KNN, Naive Bayes, SVM, Decision Tree, Random Forest, DNN with 10-Cross Validation	KNN Acc.=95.513%
[53]	2018	Machine learning repository at UCI	1st Data Set: 31 (23/8) 2nd Data Set: 68 (48/20)	SVM, FKNN, KELM with 10-Cross Validation	FKNN Acc. = 97.89%
[54]	2017	Machine learning repository at UCI	31 (23/8)	Boosted Decision Tree, Average Perceptron, BPM, Decision Forest, Decision Jungle, Locally Deep SVM, Logistic Regression, NN, SVM with 10-Fold Cross Validation	Boosted Decision Tree: Acc.=91.21% Prec.=0.935 F-Score=0.9423 AUC=0.9662
[55]	2016	Machine learning repository at UCI	31 (23/8)	DBN of 2 RBMs	FKNN Acc. = 97.89%
[56]	2015	Machine learning repository at UCI	31 (23/8)	SVM-linear, Random Tree, FBANN with 10-fold Cross Validation 50% Training + 50% Testing	FBANN Sens.=98.60% Spec.=93.62% Acc.=97.37% FPR=6.38% Prec.=0.979 MSE=0.027
[30]	2025	Machine learning repository at UCI (extended speech dataset)	252 (188/64)	KMeansSMOTE + RFE + Logistic Regression + SHAP 80% training + 20% testing	XRFILR Acc. = 96.46% Prec. = 96.46% AUC = 0.99
Proposed Research Work	**2025**	**Machine learning repository at UCI**	**31 (23/8)**	**SVM (Quadratic, Gaussian), ANN, KNN, Logistic Regression, Decision Tree with 10-Fold Cross Validation 70% Training + 30% Testing**	**SVM (Quadratic): Acc.=98.86% Prec.=100% Recall=97.77% Spec.=100% F1-Score=98.87**

using the ANOVA test. High precision (the ratio of correctly identified Parkinson's cases among all cases identified as Parkinson's) and specificity (the ratio of correctly identified.

Non-Parkinson's cases among all cases identified as non-Parkinson's) play a critical role in disease detection problems as they minimize false positives and false negatives, ensuring accurate diagnoses and minimizing unnecessary treatments or missed opportunities for intervention. Another important factor in the detection of PD is the computational speed of the classifier. To ensure robustness in our evaluation, we conducted a comparative study of the average detection rate across different classifiers and feature selection methods. The goal was to determine if any classifier could outperform the Quadratic SVM in terms of computational speed while maintaining comparable accuracy, as a simple classifier with high computational speed having larger feature space is often more desirable than a complex classifier with low computational speed having smaller feature space, provided their accuracies are comparable, see Table 7. In our simulations, the Quadratic SVM, using the top 10 features selected via the ANOVA test, achieved an average computational speed of 10, 221 μsec and a classification accuracy of 98.86%.

We prioritized classifiers and feature sets that demonstrated accuracy levels near or comparable to the Quadratic SVM. We aimed to determine if any classifier could outperform the Quadratic SVM in terms of computational speed without significantly compromising accuracy. Some of the classifiers gave better computational speed but were compromised in accuracy without much difference in classifier complexity or overall implementation overhead.

D. STATISTICAL SIGNIFICANCE TESTING

To assess the statistical robustness of the observed performance differences between the proposed Quadratic SVM model and alternative classification algorithms, we conducted a series of pairwise hypothesis tests using the two-matched-samples Student's t-test. This choice of test is appropriate given that each model was evaluated under identical experimental conditions across the same dataset, satisfying the assumption of dependent samples.

TABLE 5. Performance comparison of various classifiers, detailing accuracy, precision, recall, F1 score, and specificity. For the hardware SVM, values are presented with full precision.

Classifier	Accuracy	Precision	Recall	F1 Score	Specificity
KNN	94.32%	100.00%	89.79%	94.62%	100.00%
Decision Tree	90.90%	100.00%	84.61%	91.66%	100.00%
ANN	95.45%	97.72%	95.65%	96.67%	97.61%
Logistic Regression	95.42%	100.00%	91.66%	95.64%	100.00%
SVM (Fine Gaussian)	97.73%	95.45%	100.00%	97.67%	95.65%
SVM (Quadratic) (Software)	98.86%	100.00%	97.77%	98.87%	100.00%
SVM (Quadratic) (Hardware)	98.865188%	99.986231%	97.773438%	98.867188%	99.896712%

TABLE 6. Table depicting classification accuracy versus the number of features selected using Chi-Squared, mRMR, and ANOVA feature selection tests respectively for the classifiers, showing the impact of feature selection on classification performance.

Classifier	(Chi-squared, mRMR, ANOVA Test)										
	No. of Features Based on Rank										
	2	4	6	8	10	12	14	16	18	20	22
K-NN	62.5, 69.3, 77.31	60.2, 80.7, 87.51	58, 75, 87.51	62.5, 84.1, 89.88	56.8, 92, 94.32	90.9, 90.9, 92.11	89.8, 92, 93.23	90.45, 94.1, 95.11	92.35, 94.1, 95.11	90.67, 96.18, 96.22	94.81, 97.25, 96.81
Decision Tree	50, 68.2, 77.3	55.5, 70.25, 79.5	58.15, 74.15, 90.7	60.25, 77.24, 89.8	64.19, 79.56, 90.9	88.6, 80.55, 89.41	94.3, 82.44, 86.2	90.9, 84.87, 90.89	94.3, 86.65, 91.43	94.45, 90.67, 91.88	95.15, 95.00, 92.00
ANN	86.4, 73.9, 67	84.1, 84.1, 80.7,	80, 83, 88.6	92, 84.1, 94.3	87.5, 96.6, 95.45	94.7, 93.2, 96.1	95.5, 94.3, 90.25	95.54, 95.8, 94.24	96.31, 95.61, 95.31	96.45, 96.62, 96.82	97.85, 97.45, 97.89
Logistic Regression	64.8, 72.7, 70.7	52.3, 80.7, 81.8	76.1, 80.7, 81.8	73.9, 81.8, 93.2	73.9, 90.9, 95.42	90.9, 97.7, 95.42	94.3, 97, 94.74	94.49, 97.25, 94.9	95.07, 97.45, 96.07	96.5, 98.25, 96.51	97.2, 98.5, 97.29
SVM (fine Gaussian)	59.3, 59.3, 69.3	63.4, 63.4, 73.4	75.12, 65.12, 85.12	88.45, 68.45, 88.45	90.78, 70.78, 97.73	92.21, 75.1, 92.1	93.25, 78.35, 93.35	94.45, 84.5, 94.5	95.65, 86.67, 95.67	96.17, 90.12, 96.12	97.59, 95.56, 97.56
SVM (Quadratic)	76.1, 72.7, 75	81.8, 85.2, 72.7	89.8, 86.4, 87.5	93.2, 81.8, 96.6	93.2, 94.9, 98.86	96.6, 96.6, 98.28	97.7, 97.7, 98.41	97.8, 97.53, 98.52	98.15, 98.25, 98.25	98.46, 98.78, 98.83	98.79, 98.9, 98.87

[1] In each cell of table, the first, 2nd, and 3rd elements represent accuracy from features selected using Chi-squared, mRMR, and ANOVA tests, respectively.

TABLE 7. Table depicting computational time for different classifiers.

Classifier	Chi-squared Test	mRMR Test	ANOVA Test
	Time in μsec (Feature Size)	Time in μsec (Feature Size)	Time in μsec (Feature Size)
KNN	-	26450 (22)	25650 (22)
DT	-	-	-
ANN	22603 (22)	25655 (22)	27650 (22)
LR	9603 (22)	10732 (22), 10224 (20), 9374.1 (18), 9275 (16)	10550 (22)
SVM (Gaussian)	11750 (22)	-	18250 (22)
SVM (Quadratic)	12219 (22), 11162 (20), 11167 (18),10656 (16)	9567 (22), 9600 (20), 9732 (18), 9947 (16)	12821 (22), 12673 (20), 11750 (18),11650 (16), 11550 (14), 10750 (12), 10221 (10)

Let $\mathbf{X} = \{x_i\}_{i=1}^n$ and $\mathbf{Y} = \{y_i\}_{i=1}^n$ denote the performance accuracy of Quadratic SVM and a baseline classifier, respectively, across n cross-validation folds. The null hypothesis is formally defined as:

$$H_0 : \mu_D = \mathbb{E}[x_i - y_i] = 0, \tag{14}$$

where μ_D is the expected mean difference in performance. The alternative hypothesis is:

$$H_A : \mu_D \neq 0. \tag{15}$$

The test statistic is given by:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}, \tag{16}$$

where $\bar{d}$ is the sample mean of the differences $d_i = x_i - y_i$, and s_d is the standard deviation of the differences. A significance level of $\alpha = 0.05$ was adopted, and all results were evaluated using a 95% confidence interval.

As reported in Table 8, all computed p-values fall significantly below the critical threshold of $p < 0.05$. Consequently, we reject H_0 in all cases and accept H_A, concluding that the observed improvements in classification performance by Quadratic SVM over the other models are statistically significant. In particular, Quadratic SVM demonstrated superior performance compared to KNN, DT, ANN, LR, and the SVM-RBF. These outcomes reinforce the hypothesis that the quadratic kernel more effectively models the nonlinear structures intrinsic to the dataset's feature space.

TABLE 8. Paired t-test results for SVM-Quad ($p < 0.05$ indicates statistical significance).

No.	Compared Algorithms	p-Value
1.	SVM-Quad vs. KNN	1.03×10^{-6}
2.	SVM-Quad vs. DT	6.41×10^{-8}
3.	SVM-Quad vs. ANN	9.62×10^{-7}
4.	SVM-Quad vs. LR	7.78×10^{-7}
5.	SVM-Quad vs. SVM-RBF	1.10×10^{-6}

These statistically significant results reinforce the empirical findings presented earlier and highlight the efficacy of the proposed quadratic SVM model in comparison with widely-used conventional classifiers.

With these analyses, we conclude that the Quadratic SVM is the best classifier that gave a better performance in all domains when compared with the existing works of PD detection using voice-based features (see Table 4), and is thus chosen for hardware prototyping. While we acknowledge that differences in experimental protocols across studies—such as data splits, preprocessing techniques, and feature selection—may limit direct one-to-one comparison, Table 4 serves to provide a general contextual benchmark to highlight the relative advancement and robustness of our proposed method.

V. HARDWARE PROTOTYPING

In this section, we describe the implementation of quadratic SVM classifier on the Nexys A7 FPGA board for the detection of Parkinson's disease using voice signal features. The Nexys A7 board was selected for its simplicity and portability, demonstrating the feasibility of deploying the model on lightweight, resource-constrained platforms. This implementation highlights the potential for integrating machine-learning algorithms into portable systems for medical diagnostics. The Quadratic SVM was chosen for its effectiveness in binary classification tasks, as demonstrated by its high accuracy and computational efficiency in our analyses.

Equation 17 represents the mathematical model of a quadratic SVM classifier, which was implemented on FPGA.

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i (\mathbf{X} \cdot \mathbf{Y_i} + c)^2 + b \tag{17}$$

In Eq. (17), $f(x)$ represents the decision function of the SVM classifier. The symbols in this equation have the following meanings: $\mathbf{X}$ is the input vector to be classified, $\mathbf{Y_i}$ is the ith support vector, c is the trade-off parameter, α_i is the Lagrange multiplier associated with the ith support vector, y_i is the class label (either $+1$ or -1), b is the bias term, and N is the total number of support vectors. For classification, if $f(x) > 0$, the point is classified as class 1. Conversely, if $f(x) < 0$, the point is classified as class 0. Figure 4 shows the quadratic SVM decision boundary separating Class A (PD patients) and Class B (healthy individuals). To model the SVM classifier at the Register Transfer Level (RTL), Verilog HDL has been utilized within AMD's Vivado synthesis tool. The RTL development method was chosen over HLS (High-Level Synthesis) due to our need for fine-grained control over the hardware implementation. By directly writing Verilog code, we could implement detailed optimizations such as bit suppression techniques and specific arithmetic operations, ensuring minimal power consumption and efficient resource utilization. An SVM classifier can be built using a combination of arithmetic circuits and logic gates. The SVM model is based on a hyperplane that separates data into different classes, with support vectors, input features, and a bias term. The RTL description in Verilog HDL involves defining these circuits and ensuring proper data flow. Using AMD's Vivado synthesis tool, the HDL code is converted into a physical FPGA design. This implementation serves as a foundational step toward real-time, edge-based diagnostic tools for neurological disorders.

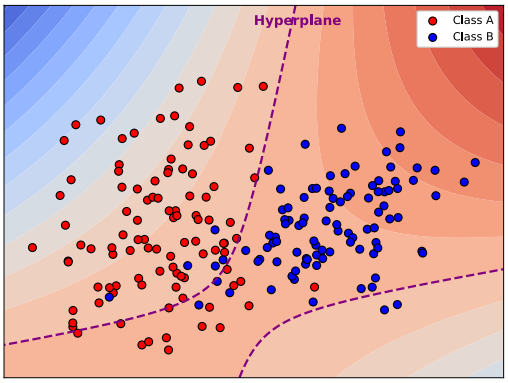

FIGURE 4. Decision boundary of a quadratic SVM classifier separating two classes: Class A (Patients with Parkinson's Disease) and Class B (Healthy Individuals). The dashed purple line represents the learned quadratic hyperplane.

A. HARDWARE ARCHITECTURE OF SVM

The hardware architecture of the selected SVM classifier is optimized for processing voice signal features and detecting early signs of PD. To ensure efficient prototyping on hardware, we limit the number of input feature space to 10 as obtained in the previous section which resulted in 47 support vectors for training the model. This approach minimizes memory usage and other resource requirements.

For this classification problem, we opt for the quadratic kernel function to implement the SVM classifier on hardware. For testing, we used data from 88 patients: the first 44 were Healthy controls, while the remaining 44 were diagnosed with PD. The data was stored in the FPGA board's on-chip memory to support a proof-of-concept implementation and simplify testing. In the following section, multiple key components and their hardware architecture are discussed to provide a comprehensive understanding of how the classifier is implemented. This includes the hardware-based Multiply-Accumulate (MAC) operations, the kernel function, and the overall structure of the SVM classifier.

1) THE MULTIPLY-ACCUMULATE (MAC)

The Multiply Accumulate (MAC) operation is a fundamental computational block in hardware architectures for digital signal processing (DSP), neural networks, digital filters, and machine learning algorithms. It performs multiplication and accumulation as defined by Eq. (18).

$$MAC = Accumulator + (operand_1 \times operand_2) \quad (18)$$

The MAC module is implemented using a synchronized circuit comprising a register, a multiplier, and an adder as shown in Fig. 5. The register stores intermediate results, the multiplier computes the product of input values, and the adder accumulates these products with the stored results. This efficient architecture is integral to the SVM classifier's hardware implementation, ensuring precise and reliable performance.

2) KERNEL FUNCTION

The kernel function is a crucial module in SVMs, enabling the classifier to separate data in higher dimensions. This separation is key to addressing complex, non-linear problems that traditional linear classifiers might struggle with. Thus to address this problem and proper classification we opt for the second-order quadratic kernel. The hardware architecture for implementing this function is depicted in Fig. 5, which comprises several arithmetic blocks. To facilitate easier processing, all the support vectors extracted during training are stored in Block RAM (BRAM) [58]. The extracted features are also saved in BRAM, providing quick local access for further computations. The entire system is synchronized by the FPGA's onboard 100 MHz clock, ensuring consistent timing throughout the architecture. The first step in this system involves using a MAC module to compute the dot product between the support vectors and the transposed feature vectors. Following this, a constant value c, that is 1 in this case, is added to the dot product. This addition serves as an offset, providing flexibility in the kernel function, and is easily accomplished using an adder. Next, the output from the adder is squared, which is a part of polynomial kernels. This squaring operation allows the classifier to capture non-linear relationships between data points, a crucial capability for accurate classification. The architecture achieves this by utilizing a multiplier that receives its input from the previous stage the output from the adder representing the dot product with the added constant. The multipliers M_1 and M_2 accept 16-bit inputs and generate a 32-bit output, reflecting the increased data width after the squaring operation to ensure no quantization error. This entire sequence of operations helps create the kernel function that the SVM uses to determine the decision boundary for classifying data as described in Eq. (17). By enabling the classifier to map input data into a more complex feature space, the kernel function improves its ability to accurately predict outcomes, even when the data structure contains intricate patterns or non-linear relationships. This flexibility helps SVM identify subtle class differences, resulting in more precise decision boundaries.

3) SVM CLASSIFIER

Once all modules and combinational circuits were integrated, the design was capable of performing the kernel function, completing the computation process for the kernel. This integration ensured that the system could manage data flow, execute required calculations, and perform the entire sequence necessary for accurate classification. A simplified block diagram, shown in Fig. 6, depicts the SVM architecture, including the key components and their interconnections. The implementation of the SVM classifier involved two MAC units. The first MAC unit calculates the inner product between the input feature vectors and the support vectors, a critical step in the SVM's classification process. The second MAC unit handles the final partial sum calculation, integrating the results from the earlier stages. The output of the kernel function through the multiplier M_2 is fed into the multiplier M_3 to calculate the product with the Lagrange multipliers and data labels, obtained from offline training, which are critical to the SVM classifier. Lagrange multipliers indicate the significance of each data point in defining the SVM's decision boundary, while data labels specify the class to which a given data point belongs. These parameters are stored in BRAM to keep the system is lightweight and efficient. After obtaining the product of these multipliers with the data labels, a bias value is added with the help of adder A_3, providing additional flexibility to the classifier's output. followed by a comparator that determines whether the final output is positive or negative. This step helps classify whether the given output indicates a patient with Parkinson's disease or a healthy individual.

4) OPTIMIZATION TECHNIQUE

In this design, a 16-bit fixed-point number representation ($Q_{10.6}$ format) is employed to create a compact and reliable SVM classifier while maintaining high prediction and calculation accuracy. This format allocates 10 bits for the integer part and 6 bits for the fractional part, offering a resolution of 0.015625. The fixed-point representation ensures precise computations during key operations, including kernel function computation, multiplication by Lagrange multipliers,

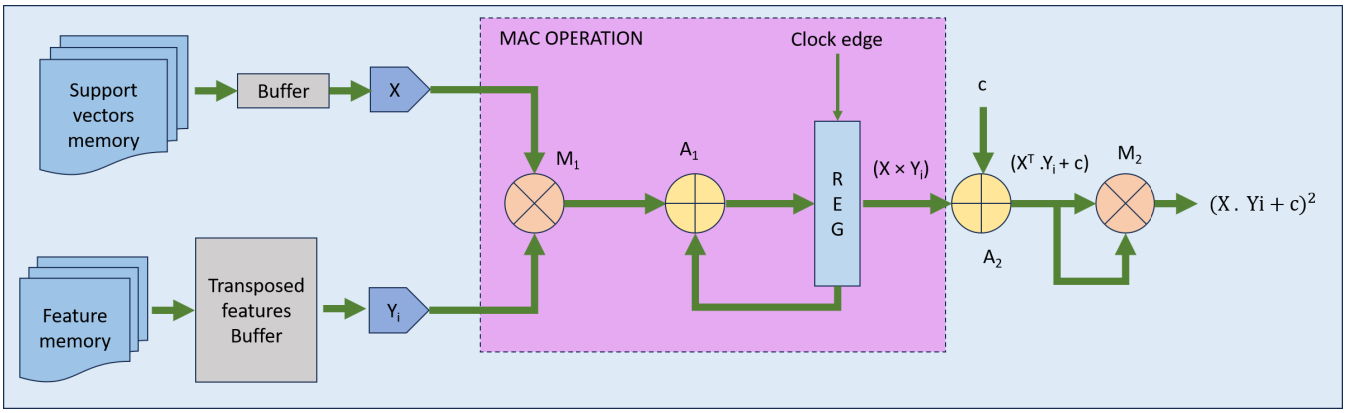

FIGURE 5. Data path for the second-order polynomial kernel function, illustrating how it uses a single MAC operation for computation, BRAM memory for storing intermediate results, and transposed input features for efficient processing.

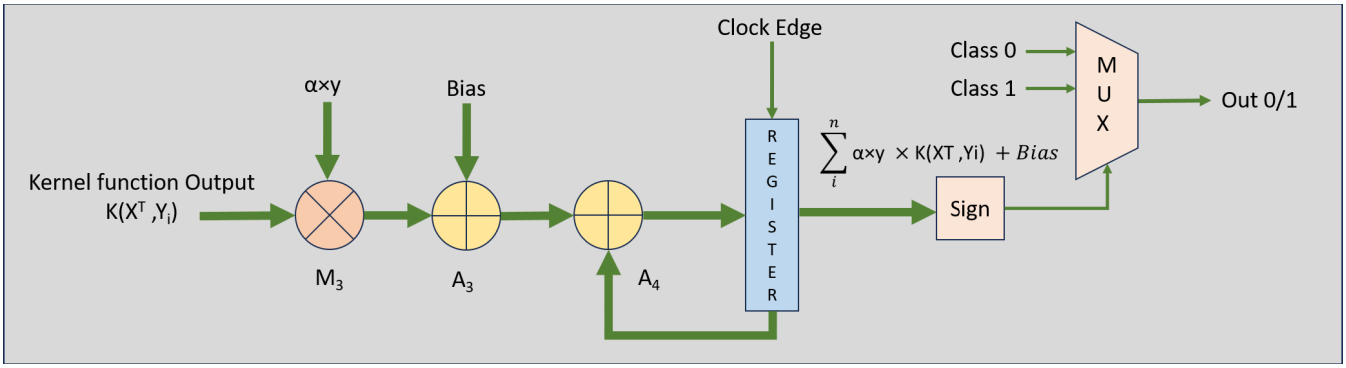

FIGURE 6. Data path illustrating the complete integration of the SVM classifier, including the kernel function computation, addition of Lagrange multipliers, and bias adjustment for precise prediction.

and data label adjustments, all of which can increase bit width.

To mitigate the challenges of increased bit width—such as higher power consumption, greater resource utilization, and delays—bit suppression or truncation techniques are applied at each stage. These techniques effectively control bit width growth while ensuring that any calculation errors remain negligible. This approach improves FPGA resource utilization, enhances computational efficiency, and reduces power consumption. Consequently, the SVM classifier achieves real-time processing capabilities with optimized performance, making it suitable for portable, battery-operated devices.

B. HARDWARE PERFORMANCE

Given the resource constraints of the Nexys A7 FPGA, optimization of resource utilization was a primary consideration in our implementation. The decision to implement the Quadratic SVM using only the top 10 features—selected through ANOVA-based incremental feature selection—enabled a reduction in input dimensionality without sacrificing performance. As detailed in Table 5, this subset yielded the highest classification accuracy in software (98.86%) and was therefore selected for hardware deployment. To validate the trade-off, we performed fixed-point quantization and mapped the model to the FPGA. The results confirmed that the FPGA implementation achieved the same classification accuracy of 98.86% as the floating-point software model. Any differences introduced by quantization were negligible, appearing only beyond the third decimal place. These fixed-point accuracy results have been explicitly reported in Table 5. Through careful design and synthesis, the classifier was mapped efficiently to Configurable Logic Blocks (CLBs), DSP slices, BRAM, and I/O resources while ensuring real-time responsiveness. Bit suppression further reduced unnecessary switching activity, contributing to substantial power and resource savings. Specifically, the use of this technique led to a reduction in LUT utilization by 74.16% (from 2229 to 576) and Flip-Flop usage by 65.65%, as summarized in Table 9.

A comprehensive power analysis, conducted during both idle and active states, confirmed the classifier's energy efficiency. The dynamic power consumption was measured at just 23 mW, underscoring the system's suitability for real-time, battery-powered, and portable health monitoring applications. The simulation results presented in Fig. 7 validate the classifier's functionality, while Fig. 8 showcases the physical FPGA prototype. Taken together, these results

TABLE 9. Comparative analysis of resource utilization before and after optimization, highlighting changes in FPGA resources such as LUTs, Flip-Flops, and BRAM, demonstrating the impact of optimization techniques.

Resource	Before Bit Optimization			After Bit Optimization			Saving %
	Utilization	Available	Utilization %	Utilization	Available	Utilization %	
LUT	2229	63400	3.52	576	63400	0.91	74.16
FF	361	126800	0.28	124	126800	0.10	65.65
BRAM	5	135	3.70	5	135	3.70	0
DSP	1	240	0.42	1	240	0.42	0
IO	12	210	5.71	12	210	5.71	0

TABLE 10. Performance comparison of the proposed Parkinson's disease diagnosis implementation with some recently proposed implementations.

Characteristics	This work	[59]	[60]	[61]	[62]
Technology	Artix-7	Zynq-7Z020	Zynq-7Z020	Cyclone V	ZCU102
Operating frequency (MHz)	100	100	667	50	-
Data width (bits)	16	32	32	-	16
Latency (μs)	65.8	667.39	-	57000	-
LUTRAM	576	682	11,589	50,120	110,435
Power (W)	0.114	-	2.297	0.089	-
Accuracy (%)	98.86	98.3	-	-	99.54

TABLE 11. Comparative analysis of power consumption before and after optimization.

	Dynamic Power	Total on Chip Power
Before Bit Optimization	0.054 (W)	0.145 (W)
After Bit Optimization	0.023 (W)	0.114 (W)
Saving Power (%)	57.407	21.379

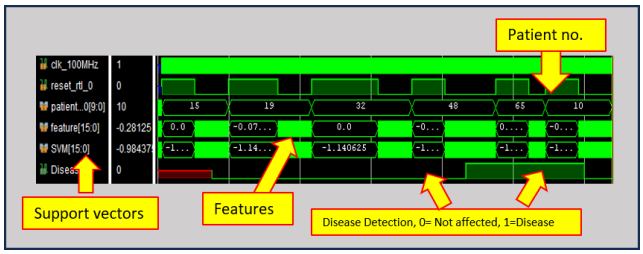

FIGURE 7. Simulation results of the SVM classifier on AMD's Vivado, illustrating the classifier's performance and functionality in a hardware-based environment.

highlight a balanced and optimized hardware implementation that preserves high classification accuracy while minimizing resource usage and energy consumption—an essential requirement for real-world deployment in constrained environments.

Table 12 shows the significant acceleration of the proposed system—approximately **155**× faster than the software-based solution—when comparing the core classification latency alone (excluding data transfer time). This measurement reflects the time taken from input availability on the FPGA to the final classification output. Such acceleration demonstrates the system's suitability for real-time embedded applications, where input data is expected to be locally acquired and processed on-chip. The same algorithm was implemented in the software based solution as well as FPGA. In terms of power and energy consumption as shown in Table 11 there was a notable decrease in dynamic power usage after bit optimization, dropping by 57.41% from 0.054 W to 0.023 W. This significant reduction can be attributed to the bit suppression technique, which minimized the activity within the FPGA, leading to lower energy requirements during operation. However, the static power remained relatively unchanged, with only a slight decrease of 1.09% because this power is related to leakage issues in the FPGA board. Overall, the total on-chip power consumption decreased by 21.38%, a substantial saving that underscores the effectiveness of the optimization technique.This reduction in power consumption is crucial for FPGA-based systems, particularly in medical devices where efficiency and battery life are important. These results indicate that bit suppression is an effective technique for reducing FPGA resource utilization, allowing more complex designs to fit within the constraints of the hardware. This reduction in resource usage also contributes to lower power consumption, an essential factor in hardware-based medical applications. Additionally, Table 10 presents a comparative analysis of the proposed architecture with the state-of-the-art.

The results indicate that the proposed approach excels in achieving high accuracy while efficiently utilizing FPGA resources, reducing latency and lowering power consumption.

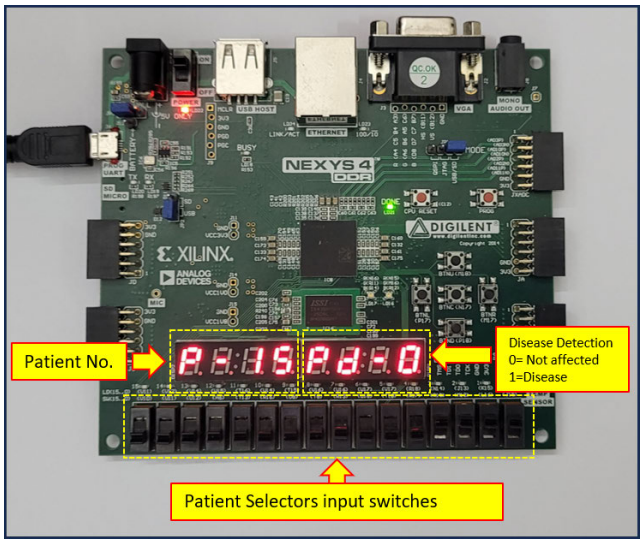

FIGURE 8. Implementation of the SVM classifier on the Nexys 4 Artix-7 FPGA kit, showcasing the hardware setup and integration process for executing the classifier.

TABLE 12. Execution time comparison between an FPGA board and MATLAB.

Platform	Time taken
FPGA	65.8 μs
MATLAB	10221 μs
	Total acceleration $\approx \times 155$

VI. CONCLUSION

In this paper, we proposed a comprehensive design flow from algorithm development to hardware implementation for Parkinson's disease (PD) detection using machine learning on an FPGA platform with voice-based features. To address class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied, and feature selection was optimized using the ANOVA algorithm, leading to an efficient feature subset. The Quadratic SVM, chosen for its superior performance, achieved a classification accuracy of 98.86% and an F1 score of 98.87%. The FPGA-based implementation was optimized using a bit suppression technique, significantly reducing resource utilization and power consumption, making it suitable for portable applications. This optimized hardware model delivered a 155× acceleration compared to software-based computation on a standard system while maintaining high accuracy and efficiency. Future work will focus on enhancing diagnostic capabilities through advanced feature extraction techniques, such as deep learning models, and integrating additional data modalities like imaging and sensor data. Expanding the system's scalability to other hardware platforms, such as Application-Specific Integrated Circuits (ASICs), also presents promising opportunities for broader clinical deployment. This work demonstrates an efficient, portable, and high-accuracy solution for real-time PD detection, paving the way for accessible and advanced diagnostic tools for neurological disorders.

ACKNOWLEDGMENT

The authors would like to thank the Ministry of Electronics and Information Technology (MeitY), Government of India, for providing FPGA to the Department of Electronics Engineering, AMU, Aligarh, under the Chips-to-Startup (C2S) Program, which has excellent support in student training.

(Mujeev Khan and Abdul Moiz contributed equally to this work.)

REFERENCES

[1] E. Tolosa, A. Garrido, S. Scholz, and W. Poewe, "Challenges in the diagnosis of Parkinson's disease," *Lancet Neurol.*, vol. 20, no. 5, pp. 385–397, 2021.
[2] S. Yousefvand and F. Hamidi, "Role of lateral hypothalamus area in the central regulation of feeding," *Int. J. Peptide Res. Therapeutics*, vol. 28, no. 3, p. 83, May 2022.
[3] L. Marsili, G. Rizzo, and C. Colosimo, "Diagnostic criteria for Parkinson's disease: From James Parkinson to the concept of prodromal disease," *Frontiers Neurol.*, vol. 9, Mar. 2018, Art. no. 313069.
[4] Z.-L. Wang, L. Yuan, W. Li, and J.-Y. Li, "Ferroptosis in Parkinson's disease: Glia-neuron crosstalk," *Trends Mol. Med.*, vol. 28, no. 4, pp. 258–269, 2022.
[5] AANS. (2024). *Parkinson's Disease*. [Online]. Available: https://www.aans.org/en/Patients/Neurosurgical-Conditions-and-Treatments/Parkinsons-Disease
[6] M. Alfonsetti, V. Castelli, and M. d'Angelo, "Are we what we eat? Impact of diet on the gut–brain axis in Parkinson's disease," *Nutrients*, vol. 14, no. 2, p. 380, Jan. 2022.
[7] L. Wang, Z. Gao, G. Chen, D. Geng, and D. Gao, "Low levels of adenosine and GDNF are potential risk factors for Parkinson's disease with sleep disorders," *Brain Sci.*, vol. 13, no. 2, p. 200, Jan. 2023.
[8] WHO. (2024). *Parkinson Disease*. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/parkinson-disease
[9] R. Pahwa and K. Lyons, *Handbook of Parkinson's Disease*. Boca Raton, FL, USA: CRC Press, 2008.
[10] D. Mata-Marín, J. A. Pineda-Pardo, J. A. Molina, L. Vela, F. Alonso-Frech, and I. Obeso, "Aberrant salient and corticolimbic connectivity in hypersexual Parkinson's disease," *Brain Connectivity*, vol. 11, no. 8, pp. 639–650, Oct. 2021.
[11] S. Sapir, L. O. Ramig, J. L. Spielman, and C. Fox, "Formant centralization ratio: A proposal for a new acoustic measure of dysarthric speech," *J. Speech, Lang., Hearing Res.*, vol. 53, no. 1, pp. 114–125, Feb. 2010.
[12] A. Tsanas, M. A. Little, P. E. McSharry, J. Spielman, and L. O. Ramig, "Novel speech signal processing algorithms for high-accuracy classification of Parkinson's disease," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 5, pp. 1264–1271, May 2012.
[13] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman, and L. O. Ramig, "Suitability of dysphonia measurements for telemonitoring of Parkinson's disease," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 4, pp. 1015–1022, Apr. 2009.
[14] J. A. Logemann, H. B. Fisher, B. Boshes, and E. R. Blonsky, "Frequency and cooccurrence of vocal tract dysfunctions in the speech of a large sample of Parkinson patients," *J. Speech Hearing Disorders*, vol. 43, no. 1, pp. 47–57, Feb. 1978.
[15] A. K. Ho, R. Iansek, C. Marigliani, J. L. Bradshaw, and S. Gates, "Speech impairment in a large sample of patients with Parkinson's disease," *Behavioural Neurol.*, vol. 11, no. 3, pp. 131–137, Jan. 1999.
[16] S. Saravanan, K. Ramkumar, K. Adalarasu, V. Sivanandam, S. R. Kumar, S. Stalin, and R. Amirtharajan, "A systematic review of artificial intelligence (AI) based approaches for the diagnosis of Parkinson's disease," *Arch. Comput. Methods Eng.*, vol. 29, no. 6, pp. 3639–3653, Oct. 2022.
[17] R. Maskeliūnas, R. Damaševičius, A. Kulikajevas, E. Padervinskis, K. Pribuišis, and V. Uloza, "A hybrid U-lossian deep learning network for screening and evaluating Parkinson's disease," *Appl. Sci.*, vol. 12, no. 22, Nov. 2022, Art. no. 11601.
[18] A. Govindu and S. Palwe, "Early detection of parkinson's disease using machine learning," *Proc. Comput. Sci.*, vol. 218, pp. 249–261, Jun. 2023.

[19] O. N. Al Sayaydeha and M. F. Mohammad, "Diagnosis of the Parkinson disease using enhanced fuzzy min-max neural network and OneR attribute evaluation method," in *Proc. Int. Conf. Adv. Sci. Eng. (ICOASE)*, Apr. 2019, pp. 64–69.

[20] A. Rana, A. Dumka, R. Singh, M. Rashid, N. Ahmad, and M. K. Panda, "An efficient machine learning approach for diagnosing Parkinson's disease by utilizing voice features," *Electronics*, vol. 11, no. 22, p. 3782, Nov. 2022.

[21] H. W. Loh, C. P. Ooi, E. Palmer, P. D. Barua, S. Dogan, T. Tuncer, M. Baygin, and U. R. Acharya, "GaborPDNet: Gabor transformation and deep neural network for Parkinson's disease detection using EEG signals," *Electronics*, vol. 10, no. 14, p. 1740, Jul. 2021.

[22] M. Aljalal, S. A. Aldosari, K. AlSharabi, A. M. Abdurraqeeb, and F. A. Alturki, "Parkinson's disease detection from resting-state EEG signals using common spatial pattern, entropy, and machine learning techniques," *Diagnostics*, vol. 12, no. 5, p. 1033, Apr. 2022.

[23] A. Ouhmida, A. Raihani, B. Cherradi, and Y. Lamalem, "Parkinson's disease classification using machine learning algorithms: Performance analysis and comparison," in *Proc. 2nd Int. Conf. Innov. Res. Appl. Sci., Eng. Technol. (IRASET)*, Mar. 2022, pp. 1–6.

[24] S. Moradi, L. Tapak, and S. Afshar, "Identification of novel noninvasive diagnostics (don't short) biomarkers in the Parkinson's diseases and improving the disease classification using support vector machine," *BioMed Res. Int.*, vol. 2022, no. 1, Jan. 2022, Art. no. 5009892. [Online]. Available: https://doi.org/10.1155/2022/5009892

[25] M. Pramanik, R. Pradhan, P. Nandy, A. K. Bhoi, and P. Barsocchi, "Machine learning methods with decision forests for Parkinson's detection," *Appl. Sci.*, vol. 11, no. 2, p. 581, Jan. 2021.

[26] P. Panda, S. K. Bisoy, A. Panigrahi, A. Pati, B. Sahu, Z. Guo, H. Liu, and P. Jain, "BIMSSA: Enhancing cancer prediction with salp swarm optimization and ensemble machine learning approaches," *Frontiers Genet.*, vol. 15, Jan. 2025, Art. no. 1491602.

[27] İ. Cantürk and F. Karabiber, "A machine learning system for the diagnosis of Parkinson's disease from speech signals and its application to multiple speech signal types," *Arabian J. Sci. Eng.*, vol. 41, no. 12, pp. 5049–5059, Dec. 2016.

[28] Y. Li, C. Zhang, Y. Jia, P. Wang, X. Zhang, and T. Xie, "Simultaneous learning of speech feature and segment for classification of Parkinson disease," in *Proc. IEEE 19th Int. Conf. E-Health Netw., Appl. Services (Healthcom)*, Oct. 2017, pp. 1–6.

[29] J. S. Almeida, P. P. R. Filho, T. Carneiro, W. Wei, R. Damaševičius, R. Maskeliūnas, and V. H. C. de Albuquerque, "Detecting Parkinson's disease with sustained phonation and speech signals using machine learning techniques," *Pattern Recognit. Lett.*, vol. 125, pp. 55–62, Jul. 2019.

[30] K. Velu and N. Jaisankar, "Design of an early prediction model for Parkinson's disease using machine learning," *IEEE Access*, vol. 13, pp. 17457–17472, 2025.

[31] P. A. Ponni, A. Seireena, and R. M. Shiny, "Early detection of Parkinson's disease through vocal features," in *Proc. Int. Conf. Multi-Agent Syst. Collaborative Intell. (ICMSCI)*, Jan. 2025, pp. 1214–1219.

[32] N. Momeni, S. Whitling, and A. Jakobsson, "Interpretable Parkinson's disease detection using group-wise scaling," *IEEE Access*, vol. 13, pp. 29147–29161, 2025.

[33] R. H. Abiyev and S. Abizade, "Diagnosing Parkinson's diseases using fuzzy neural system," *Comput. Math. Methods Med.*, vol. 2016, Jun. 2016, Art. no. 1267919.

[34] M. Hariharan, K. Polat, and R. Sindhu, "A new hybrid intelligent system for accurate detection of Parkinson's disease," *Comput. Methods Programs Biomed.*, vol. 113, no. 3, pp. 904–913, Mar. 2014.

[35] M. Little, "Parkinsons," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 4, pp. 1015–1022, Apr. 2009, doi: 10.1109/TBME.2008.2005954.

[36] M. Little, P. Mcsharry, S. Roberts, D. Costello, and I. Moroz, "Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 4, pp. 1010–1022, Apr. 2009.

[37] P. Boersma and V. Van Heuven, "Speak and unspeak with PRAAT," *Glot Int.*, vol. 5, nos. 9–10, pp. 341–347, 2001.

[38] M. Kang and J. Tian, "Machine learning: Data pre-processing," *Prognostics Health Manage. Electron., Fundam., Mach. Learn., Internet Things*, vol. 2018, pp. 111–130, Aug. 2018.

[39] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[40] A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," *Knowl.-Based Syst.*, vol. 36, pp. 226–235, Dec. 2012.

[41] M. Toğaçar, B. Ergen, Z. Cömert, and F. Özyurt, "A deep feature learning model for pneumonia detection applying a combination of mRMR feature selection and machine learning models," *IRBM*, vol. 41, no. 4, pp. 212–222, Aug. 2020.

[42] M. Kumar, N. K. Rath, A. Swain, and S. K. Rath, "Feature selection and classification of microarray data using mapreduce based ANOVA and K-nearest neighbor," *Proc. Comput. Sci.*, vol. 54, pp. 301–310, May 2015.

[43] B. Sahu, A. Panigrahi, A. Pati, M. N. Das, P. Jain, G. Sahoo, and H. Liu, "Novel hybrid feature selection using binary portia spider optimization algorithm and fast mRMR," *Bioengineering*, vol. 12, no. 3, p. 291, Mar. 2025.

[44] A. Houkan, A. K. Sahoo, S. P. Gochhayat, P. K. Sahoo, H. Liu, S. G. Khalid, and P. Jain, "Enhancing security in industrial IoT networks: Machine learning solutions for feature selection and reduction," *IEEE Access*, vol. 12, pp. 160864–160883, 2024.

[45] K. M. Kahloot and P. Ekler, "Algorithmic splitting: A method for dataset preparation," *IEEE Access*, vol. 9, pp. 125229–125237, 2021.

[46] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015.

[47] S.-U. Guan, J. Liu, and Y. Qi, "An incremental approach to contribution-based feature selection," *J. Intell. Syst.*, vol. 13, no. 1, pp. 15–42, Jan. 2004.

[48] A. Pramanik and A. Sarker, "Parkinson's disease detection from voice and speech data using machine learning," in *Proc. Int. Joint Conf. Adv. Comput. Intell. (JCACI)*. Cham, Switzerland: Springer, 2021, pp. 445–456.

[49] Z. K. Senturk, "Early diagnosis of parkinson's disease using machine learning algorithms," *Med. Hypotheses*, vol. 138, Jun. 2020, Art. no. 109603.

[50] S. Aich, H.-C. Kim, K. Younga, K. L. Hui, A. A. Al-Absi, and M. Sain, "A supervised machine learning approach using different feature selection techniques on voice datasets for prediction of Parkinson's disease," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2019, pp. 1116–1121.

[51] E. J. Alqahtani, F. H. Alshamrani, H. F. Syed, and S. O. Olatunji, "Classification of Parkinson's disease using nnge classification algorithm," in *Proc. 21st Saudi Comput. Soc. Nat. Comput. Conf. (NCC)*, Jun. 2018, pp. 1–7.

[52] A. Anand, M. A. Haque, J. S. R. Alex, and N. Venkatesan, "Evaluation of machine learning and deep learning algorithms combined with dimentionality reduction techniques for classification of Parkinson's disease," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2018, pp. 342–347.

[53] Z. Cai, J. Gu, C. Wen, D. Zhao, C. Huang, H. Huang, C. Tong, J. Li, and H. Chen, "An intelligent Parkinson's disease diagnostic system based on a chaotic bacterial foraging optimization enhanced fuzzy KNN approach," *Comput. Math. Methods Med.*, vol. 2018, pp. 1–24, Jun. 2018.

[54] A. Dinesh and J. He, "Using machine learning to diagnose Parkinson's disease from voice recordings," in *Proc. IEEE MIT Undergraduate Res. Technol. Conf. (URTC)*, Nov. 2017, pp. 1–4.

[55] A. H. Al-Fatlawi, M. H. Jabardi, and S. H. Ling, "Efficient diagnosis system for Parkinson's disease using deep belief network," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 1324–1330.

[56] M. S. Islam, I. Parvez, H. Deng, and P. Goswami, "Performance comparison of heterogeneous classifiers for detection of Parkinson's disease using voice disorder (dysphonia)," in *Proc. Int. Conf. Informat., Electron. Vis. (ICIEV)*, May 2014, pp. 1–7.

[57] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial," 2019, *arXiv:1905.12787*.

[58] M. Khan, P. Mahajan, G. N. Khan, D. Chaudhary, J. Benny, M. Wajid, and A. Srivastava, "Design and implementation of FPGA based system for object detection and range estimation used in ADAS applications utilizing FMCW radar," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2024, pp. 1–5.

[59] H. Majidinia, F. Khatib, S. J. S. M. Chabok, H. R. Kobravi, and F. Rezaeitalab, "Diagnosis of Parkinson's disease using convolutional neural network-based audio signal processing on FPGA," *Circuits, Syst., Signal Process.*, vol. 43, no. 7, pp. 4221–4238, Jul. 2024.

[60] G. Conti, M. Quintana, P. Malagón, and D. Jiménez, "An FPGA based tracking implementation for Parkinson's patients," *Sensors*, vol. 20, no. 11, p. 3189, Jun. 2020.
[61] D. De Venuto, V. F. Annese, G. Mezzina, and G. Defazio, "FPGA-based embedded cyber-physical platform to assess gait and postural stability in Parkinson's disease," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 8, no. 7, pp. 1167–1179, Jul. 2018.
[62] J. Mishra and R. K. Sharma, "Optimized FPGA architecture for CNN-driven voice disorder detection," *Circuits, Syst., Signal Process.*, vol. 44, no. 6, pp. 4455–4467, Jun. 2025.

MUJEEV KHAN received the B.Tech. degree in electronics and communication engineering from MDU, Haryana, in 2021. He is currently pursuing the M.Tech. degree in electronic circuits and systems design with Aligarh Muslim University, Aligarh. He is particularly focused on developing efficient real-time embedded systems using FPGA and ASIC technologies. His work bridges the domains of hardware acceleration and intelligent computing, emphasizing the design of optimized machine learning and digital signal processing architectures for edge and biomedical applications. He has contributed to several projects involving AI/ML hardware architecture, embedded systems, and VLSI design. His research interests include AI/ML-based reconfigurable hardware architectures, DSP architecture design, low-power digital systems, and hardware prototyping.

ABDUL MOIZ received the Bachelor of Technology (B.Tech.) degree in electronics engineering from AKTU, Uttar Pradesh, in 2021. He is currently pursuing the Master's of Technology (M.Tech.) degree in electronic circuits and systems design with Aligarh Muslim University, Aligarh. His work aims to develop efficient, real-time intelligent systems for edge computing and embedded applications. His research interests include machine learning and deep learning models, with a particular interest in their integration into reconfigurable hardware platforms, such as FPGAs.

GANI NAWAZ KHAN received the B.E. degree in electronics and communication engineering from Jamia Millia Islamia, India, in 2021. He is currently pursuing the M.Tech. degree in electronic circuits and system design with Aligarh Muslim University, India. His research focuses on FPGA-based signal processing, angle of arrival (AoA) estimation, and hardware acceleration of machine learning models. His research interests include VLSI design, embedded systems, and real-time computing.

MOHD WAJID (Senior Member, IEEE) received the B.Tech. degree in electronics engineering from Aligarh Muslim University (AMU), Aligarh, the M.Tech. degree from IIIT Hyderabad, and the Ph.D. degree in signal processing from Indian Institute of Technology Delhi, India. He is currently an Associate Professor with the Department of Electronics Engineering, AMU. Before joining AMU, he was associated with Jaypee University of Information Technology, Texas Instruments, Xilinx India Technology Services Pvt. Ltd., and Blue Star Ltd. He serves as a Reviewer for reputed journals, including IEEE TRANSACTIONS, Springer, and Elsevier.

MOHAMMED USMAN (Senior Member, IEEE) received the B.E. degree in electronics and communication engineering from the University of Madras, India, in 2002, and the M.S. degree in communications, control, and digital signal processing, and the Ph.D. degree in electronic and electrical engineering from the University of Strathclyde, Glasgow, U.K., in 2003 and 2008, respectively. In 2009, he was a Product Manager with Wimax Communications Ltd., U.K. He is currently a Full Professor and the HoD of the Electronics and Communication Engineering Department, Bennett University, India. He has more than 20 years' experience in academia, conducting research and teaching at bachelor's and graduate levels. His research interests include mathematical modeling, signal processing, and AI techniques, with a focus on wireless communication and biomedical signal processing applications. He received the Academic Excellence Award and Research Excellence Award at King Khalid University, in 2016 and 2020, respectively. He has served as the organizing chair/TPC chair of IEEE international conferences. He serves as a reviewer for IEEE/Springer/Elsevier journals.

JABIR ALI has been an Assistant Professor with the School of Computer Science Engineering and Technology, Bennett University, Greater Noida, since 2024. With more than 13 years of teaching experience at various educational institutions, he has guided more than five postgraduate students and is supervising two Ph.D. scholars. He has published more than 12 papers in SCI/SCIE/Scopus-indexed journals and contributed to more than five conference proceedings and book chapters. His research interests include machine learning, data science, cybersecurity, and brain–computer interfaces.