Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(3): 6668–6698.
DOI: 10.3934/math.2024325
Received: 30 November 2023
Revised: 24 January 2024
Accepted: 25 January 2024
Published: 18 February 2024

Research article

A real-time air-writing model to recognize Bengali characters

Mohammed Abdul Kader[1,2], Muhammad Ahsan Ullah[2], Md Saiful Islam[3], Fermín Ferriol Sánchez[4,5,6], Md Abdus Samad[7,*] and Imran Ashraf[7,*]

[1] Department of Electrical and Electronic Engineering, International Islamic University Chittagong, Kumira-4318, Bangladesh

[2] Department of Electrical and Electronic Engineering, Chittagong University of Engineering & Technology, Chittagong-4349, Bangladesh

[3] Department of Electronics and Telecommunication Engineering, Chittagong University of Engineering & Technology, Chittagong-4349, Bangladesh

[4] Universidad Europea del Atlántico. Isabel Torres 21, 39011 Santander, Spain

[5] Universidad Internacional Iberoamericana Campeche 24560, México

[6] Universidad Internacional Iberoamericana, Arecibo, PR 00613, USA

[7] Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, Republic of Korea

* **Correspondence:** Email: masamad@yu.ac.kr; imranashraf@ynu.ac.kr.

Abstract: Air-writing is a widely used technique for writing arbitrary characters or numbers in the air. In this study, a data collection technique was developed to collect hand motion data for Bengali air-writing, and a motion sensor-based data set was prepared. The feature set as then utilized to determine the most effective machine learning (ML) model among the existing well-known supervised machine learning models to classify Bengali characters from air-written data. Our results showed that medium Gaussian SVM had the highest accuracy (96.5%) in the classification of Bengali character from air writing data. In addition, the proposed system achieved over 81% accuracy in real-time classification. The comparison with other studies showed that the existing supervised ML models predicted the created data set more accurately than many other models that have been suggested for other languages.

Keywords: air-writing; Bengali character; human-computer interaction; hand gestures; machine learning
Mathematics Subject Classification: 68T01, 68T20

1. Introduction

Nowadays, people are used to interacting with the digital world through touch screens, which provide both input and output. People think that as technology improves in the future, it will be possible to make digital connections without using physical products such as laptops and phones. According to [1], it is anticipated that this technology will serve as a further enhancer of our cognitive abilities and provide a seamless connection between individuals and the digital realm. There is an expectation that future developments in virtual and augmented reality will involve the substitution of existing display components with specialized spectacles via which visual output will be directly projected onto the user's eyes. However, a comprehensive and effective method that can successfully integrate input schemes for future generations of technology has yet to be developed. Two well-researched methodologies encompass voice and gesture recognition; however, both exhibit significant constraints. In locations characterized by high levels of noise, the utilization of voice as a means of interacting with gadgets is deemed inefficient. Furthermore, the use of speech in public spaces is seen as unacceptable owing to issues regarding privacy.

There is a finite number of predefined gestures, so gesture recognition may not cover all possible interaction scenarios. In order to address these constraints, the notion of air-writing has been proposed. Air-writing allows users to write linguistic characters in the air without the need to memorize unique movements, making it a natural and user-friendly data feeding procedure for next-generation devices. Additionally, air-writing could be a smart text entry approach for small touchscreen devices, such as smartwatches, where tap-on-screen text input is prone to errors, and voice input is subject to privacy concerns and contamination by ambient noise [2]. Researchers from all over the world have proposed extensive research on air writing; the majority of studies have focused on English, and there have been only a few studies on air writing for Bengali. Bengali is the fifth language spoken worldwide by native speakers and the seventh most spoken language in terms of total speakers, with approximately 267 million people speaking it, including 230 million native speakers [3]. To incorporate this large Bengali-speaking population into modern technology, it is essential to increase the use of Bengali in technology. Despite being the world's seventh-most populous language, Bengali is not among the top 40 languages used on the internet. To achieve progress in business, education, and information technology using the internet, it is necessary to promote one's mother tongue online. However, this cannot be accomplished without recognizing the importance of developing Bengali writing tools that consider future challenges. Since air-writing is a potential text entry approach for future technology, many researchers are working to develop air-writing tools for various languages. Therefore, we must develop air writing tools for Bengali to establish its presence in technology in the future. In this study, an air-writing model is developed for all Bengali characters and the recognition rate of Bengali characters is 96.5% (97.2% for Bengali numerals). The rate of recognition is unaffected by changes in the environment. The following are some of the significant contributions made by this research:

- A portable system is being developed to record hand movements while writing Bengali characters in the air. In addition, an application is being developed that uses Bluetooth to get data from the portable device. The application includes a number of visualization options, including the ability to view received data as bar graphs, box plots, frequency domain plots, and raw and preprocessed data.

- No data set of Bengali symbols based on motion sensors was found for air-writing. In this study, air-writing data set is prepared using 3-dimensional motion sensing technique. All symbols of this language (Vowels, Consonants, and Digits) are written in the air, and the accelerations of the hand in three-dimensional space are recorded using the developed data acquisition system. The motion data for the fifty instances of each character are included in the data set. The data set contains 3050 instances of 40 consonants, 11 vowels, and 10 digits in total.

- The different statistical, time domain, and frequency domain features of the Bengali air-writing data set are examined, and the most useful feature set is determined. The feature set is then used to find the most effective machine learning model for classifying Bengali characters from the data collected by air-writing.

- A real-time application of air-writing of Bengali characters is developed.

2. Related work

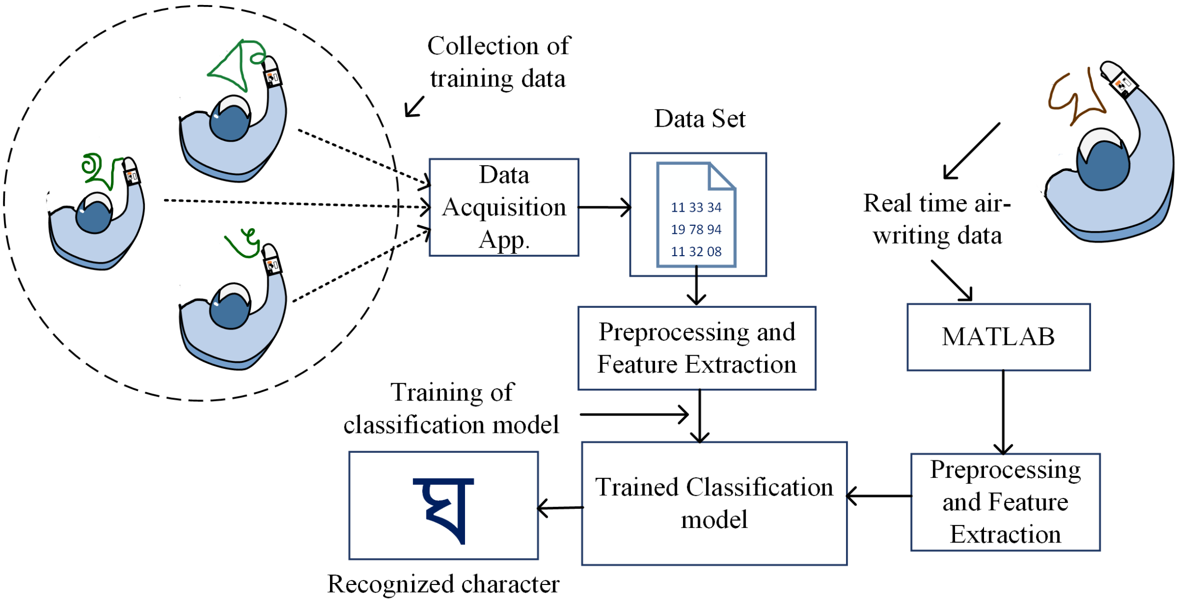
Numerous researchers have already conducted a substantial amount of study on air-writing. The research projects that are being suggested fall into various categories: Computer vision-based writing recognition; radar sensor-based air-writing; WiFi signal-based air-writing; and air-writing based on motion sensor. A set of pictures is captured when a person uses their finger or a fixed-colored item to mimic in writing a symbol in the air for computer vision-based air-writing recognition. After that, computer vision algorithms are used in these pictures to find, separate, and eventually identify the gesture. Some of the research works on air-writing based on computer vision are listed in the reference section [4–6]. Md. Shahinur et al. proposed a trajectory-based air-writing system that enables a user to write a linguistic character or word in open space by waving a finger in front of a camera [4]. In [5], Oyndrila De et al. developed a system that can classify air-written digits from a real-time video stream. In [6], the classification of English digits and letters from hand gestures based on cosine similarity and fast nearest neighbor (NN) techniques is proposed. Chengzhang Qu proposed and implemented a user-friendly human-computer interaction system based on Kinect handwriting [7]. A slope variation detection-based air-writing recognition system for Persian numbers is proposed in [8]. Pradeep Kumar et al. [9] proposed real-time recognition of sign language gestures and air-writing using the leap motion method. They also utilized a hidden Markov model (HMM) and bidirectional long short-term memory neural networks (BLSTM-NNS) to perform 3D text recognition, where they obtained 86.88% and 81.25% accuracies in word recognition for the two methods, respectively [10]. In another study, Xiwen Qu et al. [11] used CNNs to recognize air-written Chinese characters and found that their method achieved high recognition accuracy. Ji Gan and colleagues introduced an innovative system for recognizing 3D in-air handwritten Chinese text (IAHCTR). They employed a new architecture called the temporal convolutional recurrent network (TCRN), specifically designed for online handwritten Chinese text recognition (HCTR) [12]. The architecture design produces superior results when compared to the latest methods in online handwritten Chinese text recognition (HCTR). The review of these research papers reveals that under fixed illumination conditions, the majority of computer vision-based air writing models are highly accurate. Unfortunately, inconsistent lighting and the background have an adverse influence on the accuracy of the models. In recent years, the millimeter wave Radar sensor has also become a viable gesture-based air-writing solution because of its low power use, noncontact type sensing, and independence from the variation of light intensity. In [13] the authors proposed a millimeter wave radar-based air-writing application that includes of a signal processing technique and system design for gesture-based air-writing. They tested the system for five different gestures, 10 numerical symbols, and 9 alphabetic symbols in a two-dimensional space. In [14] a novel 60 GHz millimeter wave radar-based air-writing device has been developed that allows users to write arbitrary characters or numbers in the air while being encircled by a network of radars. Using trilateration and an alpha-beta tracking algorithm, the system is able to locate and follow the user's hand marker. Using a millimeter-wave frequency-modulated continuous wave radar (FMCW) operating at 60 GHz the local hand trajectory was sensed, and a dataset of 3750 character instances was recorded in [14]. The accuracy of recognizing the drawn character was then demonstrated using a 1D temporal convolutional neural network (TCN). Without the use of a handheld device, Faheem Khan et. al. created an impulse radio ultra-wideband (IR-UWB) radar-based system that can detect alphanumeric characters in midair where four IR-UWB radar sensors were arranged in a rectangle layout as the hardware. In character classification, the method was found to perform better than the state-of-the-art [15]. The precision of a radar sensor-based air writing system is good; however, the system is not portable because it needs an environment with several radar sensors. Another approach to air-writing is motion sensor-based air-writing. This method allows users to write in the air by tracking hand movements using motion sensors, such as accelerometers and gyroscopes. The accuracy of this method does not depend on any surrounding parameters. The individual holds a motion sensor either in their hand or on their body and performs a gesture in the air to create a linguistic symbol. Subsequent examination of the sensor readings is conducted to determine the character that the user has drawn in the air. A variety of techniques have been suggested to identify the linguistic nature of a text by examining the unprocessed information from motion detectors. In [1], a novel algorithm named 2-DifViz is presented that converts hand movements in the air (captured by a myo-armband worn by a user) into text. Gesture-based robot control using an accelerometer sensor is proposed in [16] . A contour-based gesture model that converts human gestures into contours in 3D space and then recognizes these contours as characters is presented in [17]. An accelerometer and gyroscope-based air-writing character recognition system using CHMM is proposed in [18]. In [19], Jeen-Shing Wang et al. present an accelerometer-based digital pen for handwritten digit and gesture trajectory recognition.The majority of the research publications cited here are concerned with English character identification by air-writing. Only one study is found where Prasun Roy et al. [20] proposed a video camera-dependent air-writing framework for English, Bengali, and Devanagari numerals. For Bengali numerals, the recognition rate was found 95.4% under constant illumination conditions. Due to color-based segmentation, the performance of this system fluctuates substantially depending on lighting conditions. The recognition of the Bengali alphabet is not taken into account.

3. Materials and methods

3.1. Overview of the system

A full illustration of the system is presented in Figure 1. The system has a data acquisition unit. With this unit in hand, each Bengali character is written in the air fifty times. The data acquisition device labeled and recorded the velocity of motion in the x, y, and z directions for all Bengali characters written in the air to create a dataset. Some preprocessing is performed on the dataset and unique features are extracted from these labeled data. These features are used to train the classification model. After training the model, it is now ready for the real-time classification of Bengali air-written characters. In order to employ the categorization model in a real-time scenario, it is necessary for the user to physically trace a Bengali character in the air. The computer receives the data generated by air-writing.

After preprocessing, some features are extracted and given to the classification model. The classification model predicts the air-written character and provides the character label as an output.



Collection of training data
Data Set
Data Acquisition App.
11 33 34
19 78 94
11 32 08
Real time air-writing data
MATLAB
Preprocessing and Feature Extraction
Training of classification model
Trained Classification model
Preprocessing and Feature Extraction
ঘ
Recognized character

Figure 1. Block diagram of the proposed system.

3.2. Development of the data acquisition system and data set

In this article, a data acquisition circuit is developed to record the motion of the hand in a three-dimensional space during air-writing. A data set is created in the air-writing system for data collection by motion sensor. The data acquisition system consists of a portable device that also includes the transmitter unit, a receiver unit, and a computer application. The portable device reads the motion data during air writing and transmits them to the receiver unit via Bluetooth. The receiver is linked to the computer through a USB connection. This unit transmits the data to the computer application instantly when it receives any air-writing data from the portable device. A brief representation of the data acquisition system is depicted in Figure 2.

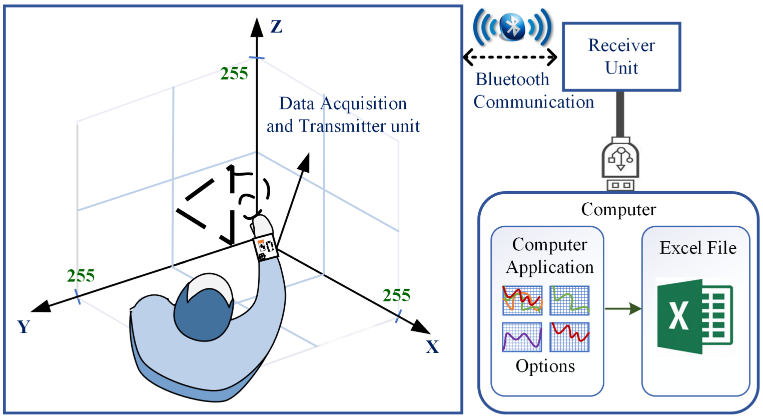


Figure 2. Block diagram of data acquisition system.

3.2.1. Data collection and transmitter unit

This setup includes a microcontroller, an acceleration sensor, a Bluetooth unit, a buzzer, and a push switch for data collection and transmission. As acceleration sensor, a device named ADXL345 was used that can sense 3-axis movement sensor to collect the data. This sensor module is capable of detecting hand motion. The motion recognizing part adjusts the outputs of the x, y, and z channels proportionally to the change in speed along each axis in three dimensions in the voltage from.

The ADXL345 has an adjustable sampling rate between 10 Hz and 3200 Hz, and its sensitivity ranges from +/-2G to +/-16G. The data is read from ADXL345 by the microcontroller through the I2C interface. There is a fixed sampling rate of 128 Hz. In its default state, the microcontroller primarily performs the task of monitoring the status of the push switch. As soon as the push switch is pressed, the microcontroller produces a noise by momentarily applying a logic one signal to the pin that is linked to the buzzer. After reading ADXL345 for two seconds, the microcontroller plays another beep sound to let the user know that the gesture has been read. It reads 256 samples from the sensor at this time. Subsequently, the data is transmitted to the receiver device through a Bluetooth connection. Figure 3 shows the complete circuit diagram of the data collection and transmitter unit.

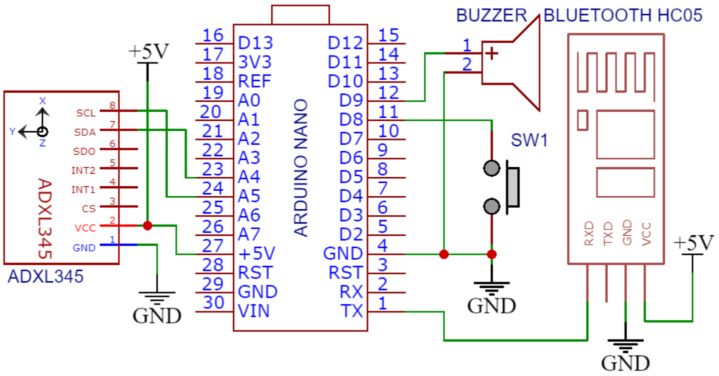


Figure 3. Circuit diagram of data acquisition and transmitter unit.

The ADXL345 is connected to the I2C bus (pins A4 and A5) of Arduino Nano where A4 and A5 act as the Serial Data (SDA) pin and the Serial Clock (SCL) pin. The Bluetooth module (HC05) follows the UART protocol to share data with the microcontroller. Thus, the RX pin of HC05 is connected to the TX pin of Arduino Nano. The Bluetooth module of this circuit acts as a master. The push button and buzzer are connected to the digital input pin D8 and digital output pin D9 of Arduino Nano, respectively. Figure 4 shows the prototype of the data collection and transmitter circuit that was developed.

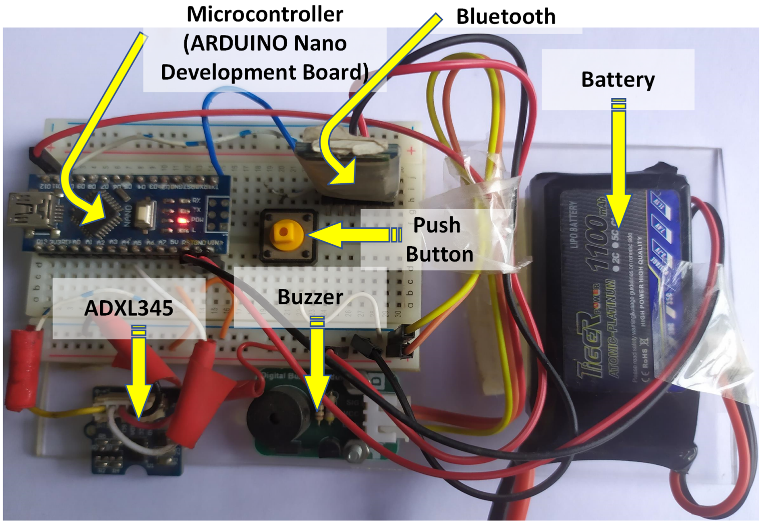


Figure 4. Prototype of data acquisition and transmitter unit.

A sample (instance) of data obtained from the accelerometer sensor has three 10-bit signed numbers. These numbers represent an acceleration in the x, y, and z-direction, respectively. Therefore, the size of each sample is 6 bytes. Before transmitting each sample, the 10-bit signed number is compressed as an 8-bit unsigned number. Thus, a 6-byte of data is compressed as 3-byte data. Then, these three-byte data are converted into a single string and transmitted to the receiver. For an air-written character, this circuit reads 256 instances from the accelerometer sensor and transmits them one by one to the receiver circuit.

3.2.2. Receiver unit

The receiver unit's main purpose is to obtain information from the data collecting unit and help it to be sent to the computer. The device comprises two microcontrollers, namely the Arduino Uno and Arduino Nano development board, together with a Bluetooth module known as the HC05. The diagram illustrating the configuration of the receiver circuit is presented in Figure 5.

The Bluetooth module of this circuit acts as a slave and is connected to the Bluetooth (BT) module of the data collection unit. The TX pin of HC05 is connected to the RX pin of Arduino Nano. The slave BT module receives the data transmitted by the master BT module and sends it to Arduino Nano. The data received is a string consisting of three characters. Then, the received string is converted into three 8-bit unsigned numbers by Arduino Nano. For an air-written character, the microcontroller of the Arduino Nano development board receives 256 instances of accelerometer sensor and stores all the numbers in its RAM. Another circuit microcontroller (Arduino Uno) is interfaced with the Arduino Nano through the I2C bus (A4 and A5) and a digital I/O pin (D7) as shown in Figure 5. The Arduino Uno acts as the master and the Arduino Nano acts like a slave in line I2C (bus). When Arduino Nano receives all the 256 instances of an air-written character, it writes a logic HIGH state on digital pin D7. This pin is connected to the PD7 of Arduino Uno. The microcontroller of the Arduino Uno development board is instructed from the computer. The receiving unit's prototype is depicted in Figure 6.

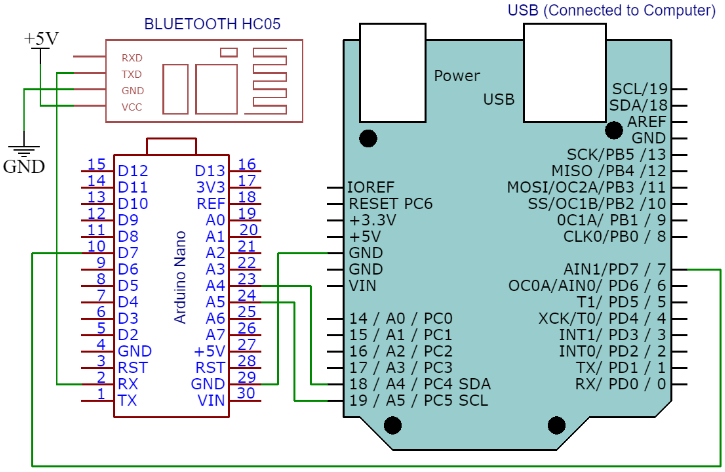


Figure 5. Circuit diagram of the receiver unit.

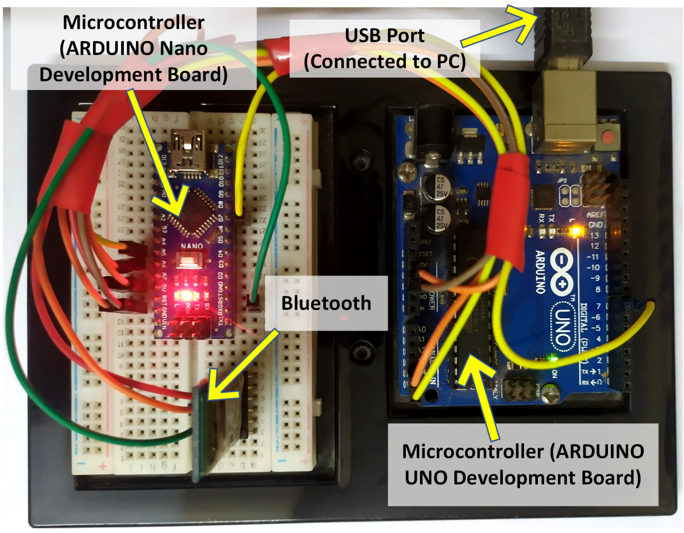

Microcontroller (ARDUINO Nano Development Board)
USB Port (Connected to PC)
Bluetooth
Microcontroller (ARDUINO UNO Development Board)

Figure 6. Prototype of the receiver unit.

3.2.3. Data collection application

The primary goal of the data collection unit is to gather, examine, and preserve Bengali character airwriting data in order to create a data set. Thus, a data collection application is created in order to verify the quality of the data and store them in an organized manner. The developed application's user interface is displayed in Figure 7.

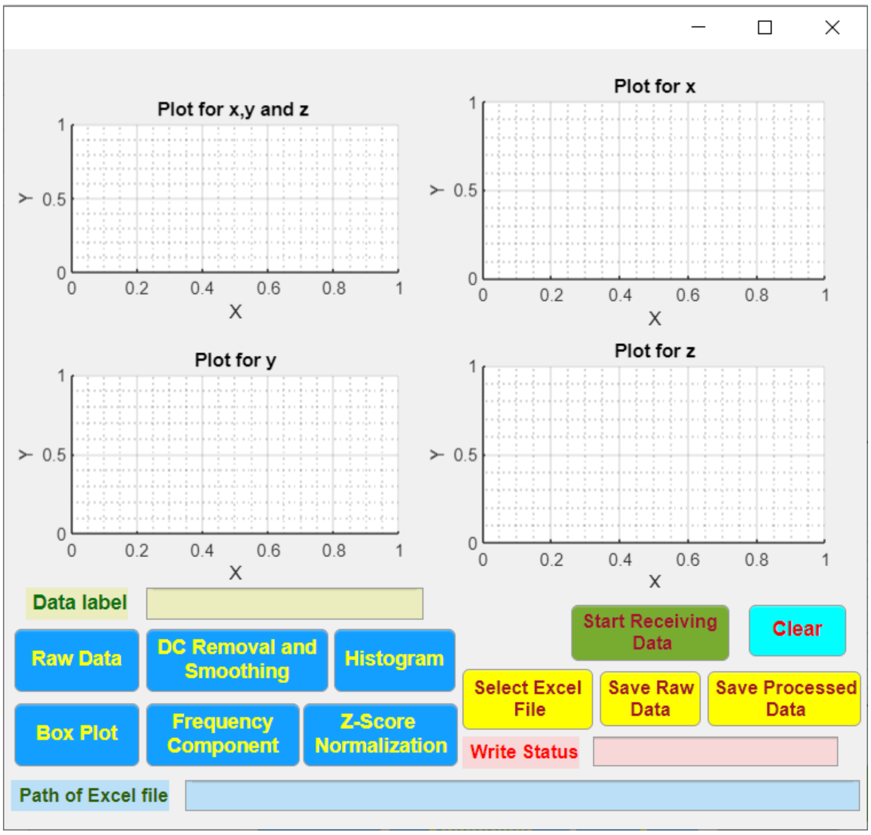
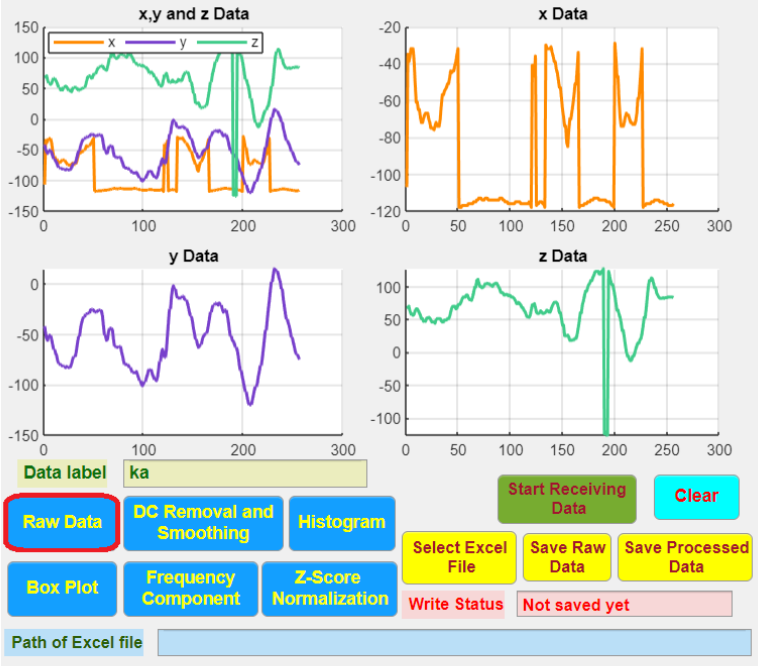


Figure 7. Interface of data collection application.

This application controls the microcontroller-2 (Arduino Uno) of the circuit shown in Figure 5. It reads the digital input pin (PD7) of Arduino Uno. When this pin becomes HIGH, the application reads all the 256 samples of an air writing character that are transmitted from the data acquisition circuit. Then shows the data in the plot area of the application. This application displays four plots to demonstrate the speed of hand movement in the x, y, and z directions. The terms x, y, and z data in the application refer to acceleration data in the x, y, and z axes. There is a button labeled "Start Receiving Data". To initiate the data gathering procedure, users can press this button. The individual can inscribe a character in the air while grasping the data acquisition unit in their hand. The application then receives the data. The first plot area displays the received acceleration data collectively, and the remaining three plot sections display the data separately. Figure 8 illustrates the acceleration in the x, y and z directions while writing the Bengali language symbol 'ka' in the air. The application offers the ability to show both unprocessed and processed data, as well as some notable characteristics, such as histograms, frequency components, and box plots. To understand the quality of the data before integrating them into the data set, these visualizations and data features will be useful. Data can be labeled and saved in an Excel sheet after analysis by the user in order to create the data set. The user can select the location of the Excel spreadsheet where the data will be stored. The option of saving unprocessed or preprocessed data is also available to the user.



Data label: ka

Write Status: Not saved yet

Path of Excel file:

Figure 8. Graphical representation of the acceleration data of Bengali character 'ka'.

3.2.4. Data set

The data obtained from air-writing yields a 3-channel, one-dimensional signal representing characters. The three channels carry information about the acceleration (with respect to time) of the hand in the x,y, and z axes while writing a alphabet in the air. The data for one channel is a vector of 256 samples. The size of each sample is 8 bits (1 byte). The sampling rate is 128 samples per second, and the duration is 2 s. Thus, we can represent the data of air writing of a character (sometimes referred to as an instance) in the data set as follows:

$$\left.\begin{aligned} x_n &= \{x_0, x_1, \cdots, x_{255}\} \\ y_n &= \{y_0, y_1, \cdots, y_{255}\} \\ z_n &= \{z_0, z_1, \cdots, z_{255}\} \\ l &= \{label\} \end{aligned}\right\}. \tag{3.1}$$

In the data set, each instance has four variables: x, y, z, and l. The first three variables (x, y, and z) carry the information on the acceleration of an air-written character, and the last variable (l) is the label of that character. A total of fifty Bengali letters (11 vowels and 39 consonants), as well as ten Bengali digits, must be collected in order to create the Bengali air-writing data set. For each letter and number, data from fifty air-writing instances were gathered from five different people. For all characters and digits combined, 3000 instances have been recorded. To write a character in the air, there should be a pattern for that character. The pattern of the characters should be similar to the visual linguistic characters so that people can easily remember the patterns. Figure 9 shows the pattern of the Bengali character ngo (ঙ) in air writing.

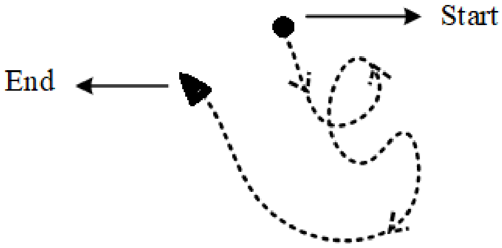


Figure 9. Stroke of Bengali alphabet ngo in air writing.

The area required for air writing in the space is not fixed. It may vary from person to person as well as for different characters. However, the data set is prepared in this research by writing the character in the space covering the area, which is approximately 22 × 20 inches. The air-writing strokes of all Bengali characters are defined before preparing the air-writing data set. The air-writing strokes of Bengali numbers and vowels, and consonants are shown in Figures 10 and 11, respectively.

3.3. *Classification*

The recognition or classification of a Bengali character from air-writing data is discussed in this section. Preprocessing, feature extraction, training, and real-time classification are the steps in the classification process and are explained here in that sequence.

3.3.1. Preprocessing

Data preprocessing is an essential step in improving the accuracy and efficiency of a machine learning model by cleaning and preparing data for use.

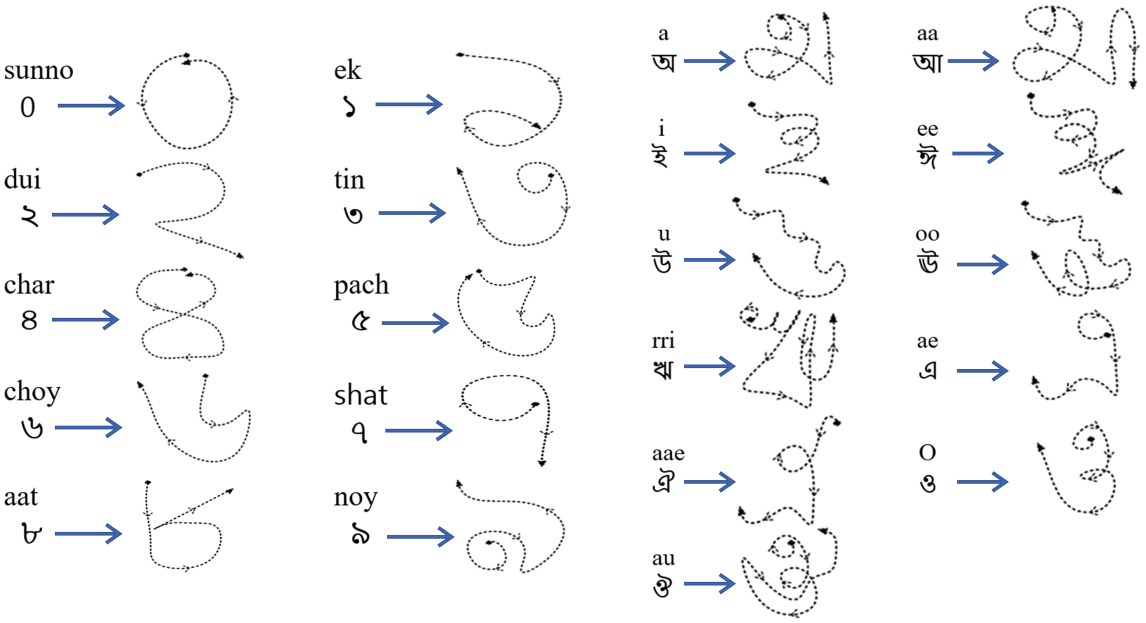


Figure 10. Strokes of Bengali numbers and vowels in air-writing.

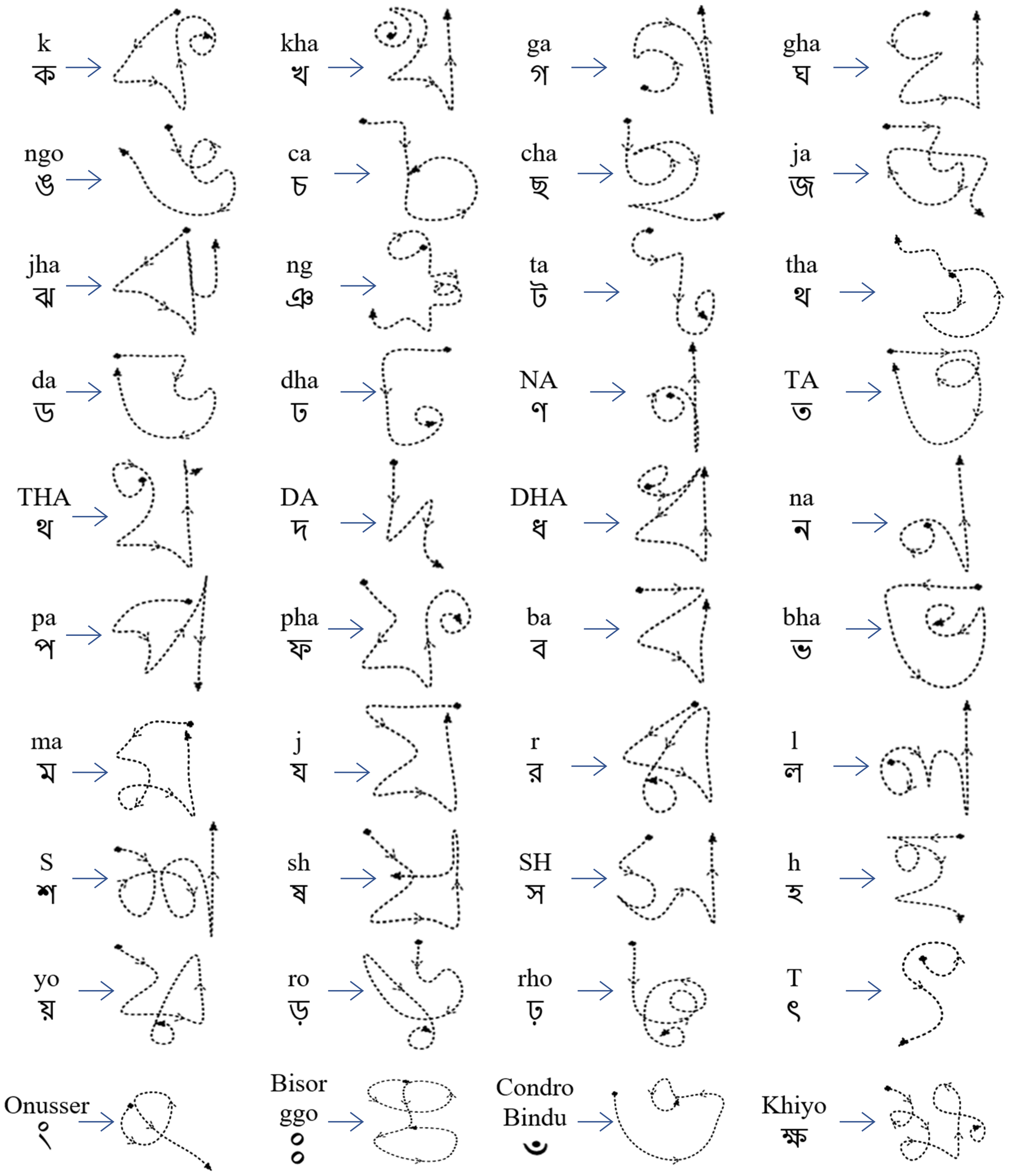


Figure 11. Strokes of Bengali consonants in air-writing.

During air writing, the sensor's disruptions, as well as the users' unconscious trembles, introduce high-frequency noise in the acceleration of hand movement, which usually does not carry any information. A moving average filter is suitable to reduce such high-frequency noise, i.e., smooth the data. However, the smoothing effect in the moving average filter is proportional to the number of data points involved with the filter, and the involvement of larger data points requires higher computation costs and memory storage. There is another filter for smoothing the data, known as a leaky integrator. The same smoothing operation on data can be performed by a leaky integrator with fewer computational costs compared to a moving average filter. The smoothing operation of acceleration data along the x-axis, y-axis, and z-axis is performed by Eqs (3.2)–(3.4) [21], respectively.

$$\mathbf{x}' = \lambda x_{n-1} + (1 - \lambda)x_n, \tag{3.2}$$

$$\mathbf{y}' = \lambda y_{n-1} + (1 - \lambda)y_n, \tag{3.3}$$

$$\mathbf{z}' = \lambda z_{n-1} + (1 - \lambda)z_n. \tag{3.4}$$

Here, λ is a constant that ranges from 0 to 1. The closer the value of λ is to 1, the greater the smoothing.

At this stage, x', y', and z' vectors may have values in different ranges. The features obtained from these data will also have values on dissimilar scales. Therefore, machine learning models have to compare scores on different scales, which creates an uneven contribution of features to prediction. The standardization or normalization process is a solution to this problem. Standardization can be accomplished in a variety of ways. The z-score normalization procedure is employed in this case. This is the process of converting every value in a dataset to a mean of 0 and a standard deviation of 1. Equations (3.5)–(3.7) are used to normalize the data x', y', and z', respectively. Here, μ represents the mean and σ represents the standard deviation of the corresponding data expressed by the vector x' or, y' or, z'.

$$\mathbf{x} = \frac{\mathbf{x}' - \mu_{x'}}{\sigma_{x'}}, \tag{3.5}$$

$$\mathbf{y} = \frac{\mathbf{y}' - \mu_{y'}}{\sigma_{y'}}, \tag{3.6}$$

$$\mathbf{z} = \frac{\mathbf{z}' - \mu_{z'}}{\sigma_{z'}}. \tag{3.7}$$

3.3.2. Feature extraction

Feature extraction is the process of finding meaningful insights by reducing the dimensionality while preserving the information in the original or raw data. Raw data is messy, and machine learning (ML) models cannot work efficiently with messy raw data. The feature extraction process reduces the number of input variables, which lowers the computing cost of the machine learning model and also improves the model's performance. The process of transforming data into features is called feature engineering. The selection of features is a highly iterative process. This may be as simple as selecting the right pieces of data or extracting features, which will require simple or complex statistical calculations and transformations of the data. However, the calculated parameters should have the following properties to be a feature [21]:

- Have variance
- Are not random

- Are unique

In [19], an accelerometer sensor-based handwritten digit and gesture recognition algorithm is developed, where statistical calculations and transformations of sensor data are used to find features in the data set. In this study, primarily the following parameters are selected in the feature set: However, subject to having the above-mentioned properties, the appropriate parameters will be used to train the machine learning models.

(a) **Root mean square (rms) value**: The square root of the average of the squares of all the values in a series is its root mean square, or rms. For example, the rms value of the data sequence x is calculated by Eq (3.8) [19].

$$rms_x = \sqrt{\frac{1}{N}\sum x^2}. \tag{3.8}$$

(b) **Correlation among axes**: The calculation of correlation between data on two axes involves calculating the ratio of covariance to the product of the data's standard deviations. As an illustration, the correlation ($corr_{xy}$) of data x on the x-axis and y on y-axis is obtained by Eq (3.9) [19], where ($cov(x, y)$) is computed by Eq (3.10). Here, N implies the number of sample, σ_x and σ_y are the standard deviations of vector x and y, and $\bar{x}$ and $\bar{y}$ are the average of x and y.

$$corr_{xy} = \frac{cov(x, y)}{\sigma_x \sigma_y}, \tag{3.9}$$

$$cov(x, y) = \frac{\sum (x - \bar{x})(y - \bar{y})}{N}. \tag{3.10}$$

(c) **Entropy**: Entropy is a metric for measuring disorder or uncertainty. It tells us how non-homogeneous a dataset is. The entropy of a discrete sequence is the measure of dissimilarity or diversity of the samples in that sequence. The entropy H_x of the sequence **x** is obtained by Eq (3.11) [22]. Here, c indicates the number of interval $(R1, R2, \cdots, Rc)$ in **x**; P_i is the probability of a sample belonging to the interval R_c.

$$H_x = -\sum_{i=1}^{c} P_i \log_2(P_i). \tag{3.11}$$

In this study, the value of the sequence **x**, **y**, or **z** is an 8-bit signed number and ranges from -128 to +127. This range is divided into 32 intervals. The probability of a sample belonging to an interval is obtained from the histogram of the sequence x. In a similar manner, the entropies H_y and H_z are calculated.

(d) **Zero-crossing rate**: The rate at which the sign of the signal changes during a frame is known as the zero-crossing rate, or ZCR. In other words, it can be expressed as the signal's frequency of shifting from positive to negative and back again, divided by the length of the frame. The ZCR of a signal or sequence can be viewed as a measure of its noisiness. The ZCR of the sequence **x** is calculated using Eq (3.12) [23].

$$Z_x = \frac{1}{2W_L}\sum_{n=1}^{W_L} |sgn(x_n) - sgn(x_{n-1})|, \tag{3.12}$$

Where $sgn(.)$ is the sign function shown in Eq (3.13),

$$sgn(x_n) = \begin{cases} 1 & x_n \geq 0, \\ -1 & x_n < 0. \end{cases} \tag{3.13}$$

(e) **Interquartile range (IQR)**: The interquartile range (IQR) is a measurement of the distribution of data in the middle half. It is the range for the middle half of the data samples. The IQR gives an indication of how samples in a data sequence are distributed. Larger numbers imply that the data is spread out wider in the center. Smaller values, on the other hand, indicate that the middle values are more closely clustered. If we want to find the IQR of the data sequence **x**, we have to sort the samples of **x** into four ranges (quarters), which are called quartiles. The quartiles are labeled from low to high, such as Q1, Q2, Q3, and Q4, which are shown in Figure 12.

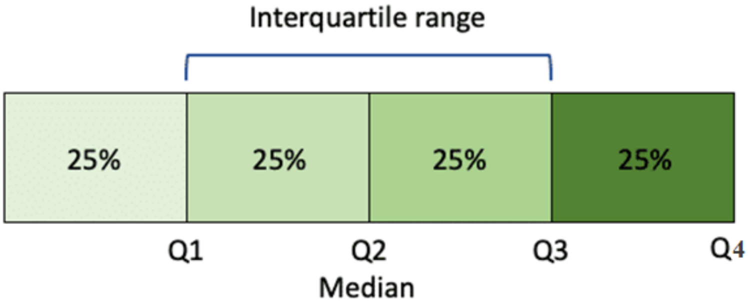


Figure 12. Calculation of IQR.

Now, the IQR of the sequence x will be obtained by subtracting the Q1 value from the Q3 value by Eq (3.14).

$$IQR_x = Q_3 - Q_1. \tag{3.14}$$

(f) **Energy**: The energy of a sequence is determined by taking the square root of the Fast Fourier Transform (FFT) coefficients. The energy of a data sequence **x** is calculated using Eq (3.15) [19], where, N is the number of samples in the sequence, and X_k is the k^{th} FFT co-efficient.

$$E_x = \frac{1}{N}\sum_{k=1}^{N} |X_K|^2. \tag{3.15}$$

Similarly, we can compute the energy E_y of the sequence **y** and E_z of the sequence **z**.

(g) **Mean frequency or spectral centroid**: The frequency at which the energy of a spectrum is centered is indicated by its spectral centroid. If the FFT coefficients or spectral magnitude of the sequence **x** is $S(k)$ and the corresponding frequencies are $f(k)$, then the spectral centroid of the sequence could be found by Eq (3.16) [24].

$$f_c = \frac{\sum_k S(k)f(k)}{\sum_k S(k)}. \tag{3.16}$$

(h) **Spectral roll-off**: The frequency below which a certain percentage of total spectral energy, such as 85%, lies is known as spectral roll-off. The spectral roll-off point of a discrete sequence is calculated by Eq (3.17) as described in [25].

$$\sum_{k=b_1}^{i} S(k) = \gamma \sum_{k=b_1}^{b_2} S(k). \tag{3.17}$$

In this context, S_k represents the spectral value corresponding to the frequency bin k. Variables b_1 and b_2 denote the bin indices that define the band boundaries to calculate the spectral spread. Furthermore, γ represents the proportion of total energy contained within the bin range of b_1 to b_2.

(i) **Spectral bandwidth**: The difference in frequency between the sites where the spectrum falls at least 3 dB below the maximum spectrum level is known as the spectral bandwidth. The spectral bandwidth of order-p of a data sequence is calculated by Eq (3.18) [24].

$$BW_p = (\sum_k S(k)(f_k - f_c)^p)^{\frac{1}{p}}. \tag{3.18}$$

Here, S_k is the spectral magnitude at frequency bin k, f_k is the frequency at bin k, and f_c is the spectral centroid.

(j) **Principal component coefficients**: To determine the principal components of data, principal component analysis, or PCA, is used. Data normalization, covariance matrix computation, and eigenvalue and eigenvector identification are all part of PCA. Finally, the eigenvector of the covariance matrix is used to identify principal components. In this study, standardization is done at the data preprocessing stage and we have found three data sequences x, y, and z. The covariance and covariance matrix of these sequences are found in Eqs (3.19) and (3.20).

$$cov(x, y) = \frac{\sum (x - \bar{x})(y - \bar{y})}{N}, \tag{3.19}$$

$$S = \begin{bmatrix} cov(x, x) & cov(x, y) & cov(x, z) \\ cov(y, x) & cov(y, y) & cov(y, z) \\ cov(z, x) & cov(z, y) & cov(z, z) \end{bmatrix}. \tag{3.20}$$

The eigenvalue of covariance matrix λ is calculated by Eq (3.21). The eigenvector of the first principal component U_1 is obtained by Eq (3.22) taking the maximum eigenvalue found from Eq (3.18).

$$det(S - \lambda I) = 0, \tag{3.21}$$

$$(S - \lambda_{max} I) = 0. \tag{3.22}$$

The computation of the normalized eigenvector and first principal components is performed using the Eqs (3.23) and (3.24) [26].

$$e_1 = \begin{bmatrix} \frac{U_1}{\sqrt{U_1^2+U_2^2+U_3^2}} \\ \frac{U_2}{\sqrt{U_1^2+U_2^2+U_3^2}} \\ \frac{U_3}{\sqrt{U_1^2+U_2^2+U_3^2}} \end{bmatrix}, \tag{3.23}$$

$$P_{1i} = e_1^T \begin{bmatrix} x_i - \bar{x} \\ y_i - \bar{y} \\ z_i - \bar{z} \end{bmatrix}. \tag{3.24}$$

3.3.3. Classification models

Machine learning is a method of data analysis that automates the development of analytical models. It is a subset of artificial intelligence based on the idea that computers can learn from data, identify patterns, and make decisions with little to no human involvement. This research has extensive application of machine learning. The acceleration of hand movements for the air-writing of different Bengali characters will be collected. Based on this data the system has to identify a character from a new air-writing data. The concept of supervised machine learning is very similar to this problem. Supervised learning trains models to produce the desired output using a training set. The model can develop over time because this training dataset contains both accurate and inaccurate outcomes. The loss function is used to evaluate how accurate the method is, and it is changed until the error is sufficiently minimized. Regression and classification are the two categories of supervised learning techniques. The major difference between classification and regression algorithms is that the former predicts continuous values while the latter predicts or classifies discrete values [27]. In our case, the output of the machine learning model is discrete. The model has to select one label from a set of 61 labels (as there are 61 characters in the data set). Therefore, the development of the air-writing model is a classification problem under supervised machine learning. Different supervised machine-learning algorithms are discussed below.

(a) **Decision tree**: Decision tree (DT) is a statistical approach for classifying data. A decision tree is a structure resembling a flowchart in which each leaf node (terminal node) contains a class name, each internal node represents an attribute test, and each branch shows the test's result. Using decision trees, instances are sorted from the root of the tree to a leaf node, which offers the classification. This method effectively poses a question, and depending on the response (Yes/No), divides the tree into subtrees. Decision trees are prone to errors in classification problems with numerous classes and few training occurrences [28].

(b) **Discriminant analysis**: A collection of prediction equations for classifying people according to independent factors are found using discriminant analysis. Establishing a predictive equation for categorizing new persons or interpreting the predictive equation to ascertain any potential correlations between the variables are two possible objectives of a discriminant analysis. Discriminant analysis is divided into various groups, each with distinct assumptions and properties. The two major categories are quadratic discriminant analysis (QDA) and linear discriminant analysis (LDA). Different trade-offs are made between simplicity and versatility by LDA and QDA. LDA is simpler and works with the assumption of a single covariance matrix; however, QDA is more flexible and allows for different covariance matrices for each class. The selection of one or the other depends on the characteristics of the data and the reasonable assumptions for a given classification task [29].

(c) **Naïve bayes:** This classification technique is predicated on the Bayes Theorem and the idea of predictor independence. Simply put, a Naive Bayes classifier makes the assumption that each characteristic in a class exists independently of each others. This suggests that each predictor has the same effect on the outcome and that the existence of one feature in the probability of an occurrence does not affect the presence of another [29].

(d) **Support vector machine**: A prominent supervised learning method for regression and classification is support vector machine, or SVM. It is most frequently used, nevertheless, in machine learning tasks involving classification. The goal of the support vector machine technique is to identify a decision boundary in an N-dimensional space (where N is the number of features) that categorizes the data points clearly. The optimal decision boundary is known as a hyperplane [28].

(e) **k-nearest neighbor**: Based on how closely related and distant data points are to other readily available data, the KNN algorithm can discern between them. This method assumes that neighboring data points with comparable properties may be discovered. In order to do this, it makes an effort to measure the separation between data points, generally using Euclidean distance, and then assigns a category based on the most prevalent category or average [28].

3.4. Feature selection and training of classification models

For the training of supervised learning models, the parameters discussed above are primarily chosen as features. The parameters are mean, standard deviation, rms value, correlation among axes, entropy, principal component coefficients, zero-crossing rate, interquartile range, energy, mean frequency, spectral roll-off, and spectral bandwidth. However, in an air-writing instance, there are three discrete sequences i.e., acceleration in the x-direction, y-direction, and z-direction (the sequences are represented as x, y, and z in the previous subsection). As a result, we received three features for each parameter. Therefore, 36 features are attained from twelve parameters. All features including the labels are listed in Table 1).

Table 1. List of parameters that are primarily included in the feature set.

Name of parameters	No. of features	Feature label (Assumed)
Mean	3	DC_x, DC_y, DC_z
Standard deviation	3	SD_x, SD_y, SD_z
rms value	3	RMS_x, RMS_y, RMS_z
Correlation among axes	3	$Corr_{xy}, Corr_{yz}, Corr_{zx}$
Entropy	3	$EntropyX, EntropyY, EntropyZ$
Principal component coefficients	3	PCA_x, PCA_y, PCA_z
Zero crossing rate	3	ZCR_x, ZCR_y, ZCR_z
Interquartile Range	3	IQR_x, IQR_y, IQR_z
Energy	3	E_x, E_y, E_z
Mean frequency	3	MF_x, MF_y, MF_z
Spectral roll-off	3	SR_x, SR_y, SR_z
Spectral bandwidth	3	SB_x, SB_y, SB_z
Total features	**36**	

However, having too many features will make the model inefficient. Irrelevant features increase the complexity of the model and make the model computationally inefficient without improving the accuracy of the model. Therefore, to optimize the model accuracy and computational cost we have to avoid irrelevant features. The technique that removes excessive and irrelevant features is known as dimensionality reduction [30].

In this study, two steps are taken to find the useful features from the feature set. First, before including a parameter in the feature list, the variance of the parameter is observed from the box plot of that parameter to ensure how well the feature can distinguish between various characters. The box plot of the parameter $Corr_{yz}$ for different Bengali characters is shown in Figure 13.

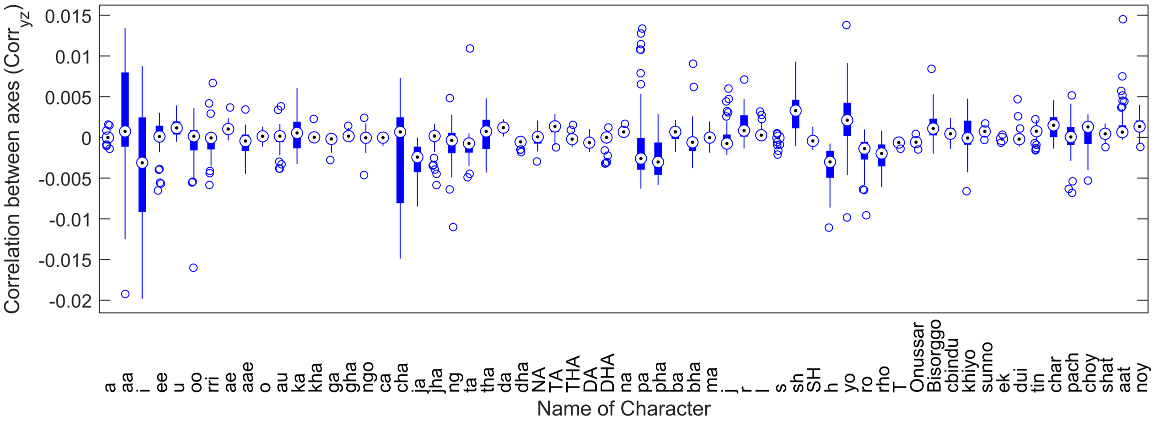


Figure 13. The box plot of the parameter $Corr_{yz}$ for different Bengali characters.

According to observations, there is no discernible variation in this parameter among Bengali characters. In order to differentiate or classify Bengali characters from its air-writing data, this parameter will not be helpful. This parameter will thus not be included in the feature list. The box plot of another parameter, *EntropyZ*, for different Bengali characters is shown in Figure 14. The illustration makes it clear that there is a significant amount of variation in this parameter for various characters.

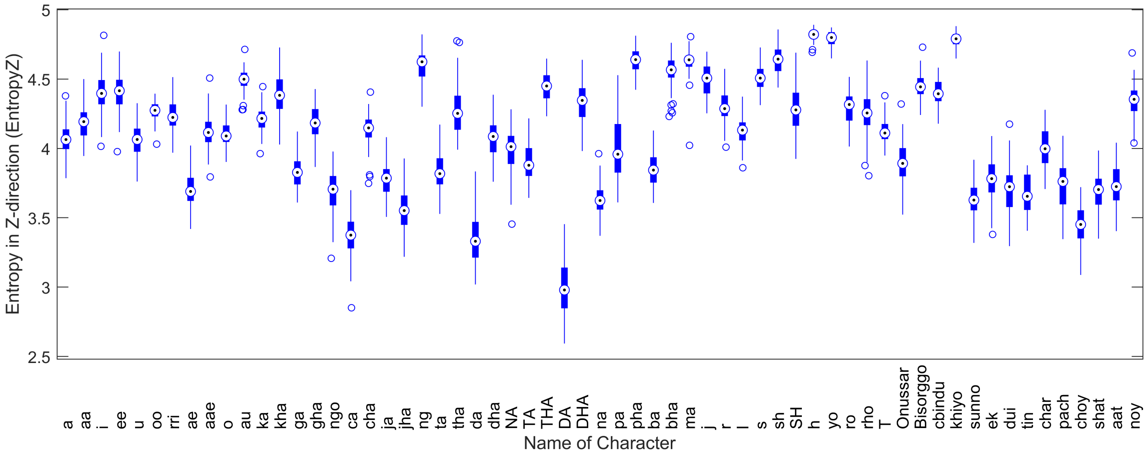


Figure 14. The box plot of the parameter *EntropyZ* for different Bengali characters.

Therefore, machine learning models can easily distinguish among different characters using this parameter. Therefore, this parameter is considered as a feature. As with the individual box plots for each parameter, the boxplot of the parameter's mean value for each Bengali character, which is displayed in Figure 15, provides a measure of the variability for all other parameters. The size of the interquartile range (IQR) demonstrates the parameter's high degree of variability and suggests its use as a feature. The small size of the IQR, on the other hand, indicates little variability. Since the parameters $CorrYZ(Corr_{yz})$ and SBx have little variance, they are not regarded as features.

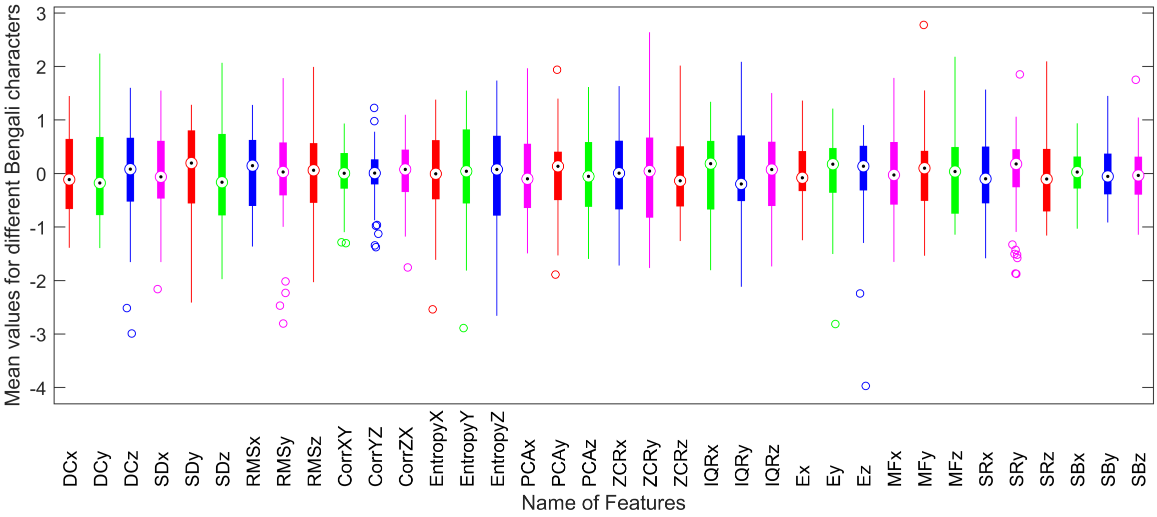


Figure 15. Box plot of the mean values of the parameters for all Bengali characters.

Second, there may have some features which are highly correlated with each other. They will then have a similar influence on the target variable, and including both in the model will be unnecessary. Therefore, we can eliminate one of them without affecting the model's performance. Such redundant features from the feature set can be identified by analyzing the heatmap of the covariance of different features [29]. Figure 16 shows a heatmap illustrating the covariance between different features of Bengali consonants. It is clear from the heatmap that the features DC_x, SD_x, and IQR_x are highly correlated. We can consider only one feature instead of three. Likewise, the correlation is found for some other features that are shown in Figure 17. From this figure, it is observed that six groups of features are highly correlated having 18 features. Thus, we can take only six features out of eighteen without a considerable effect on the accuracy of the models. Selected features are marked in green color and eliminated in red. The features having low computational cost are selected from the correlated feature groups.

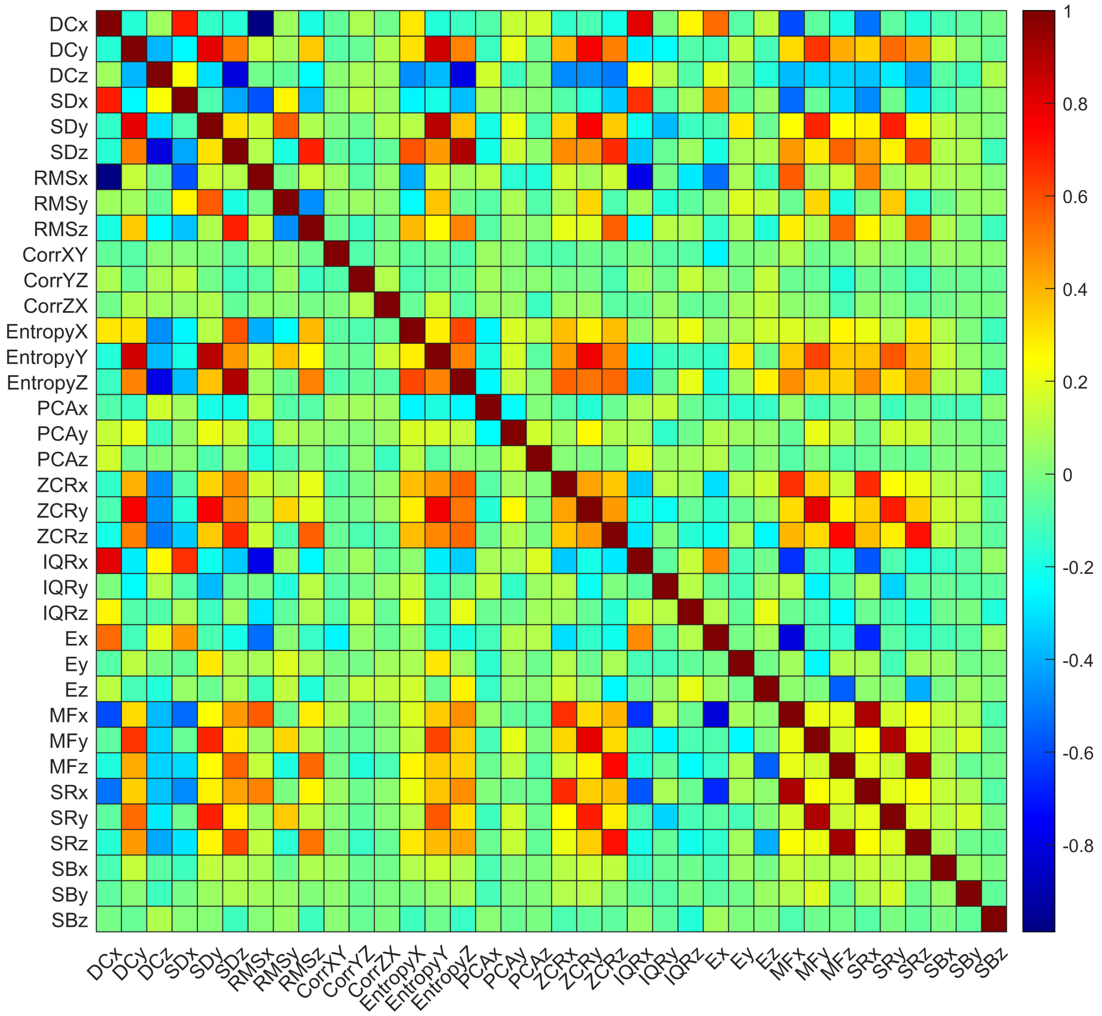


Figure 16. Heatmap illustrating the covariance between different features.

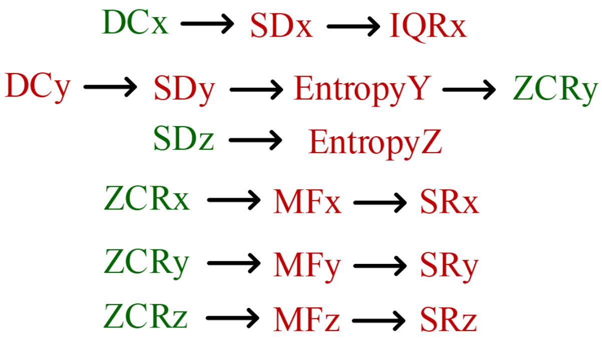


Figure 17. Correlated features.

The redundant features are: $SD_x, SD_y, Corr_{yz}$ IQR_x, DC_y, SB_x $EntropyY$, $EntropyZ, MF_x, MF_y, MF_z, SR_x$, SR_y, SR_z. Finally removing the excessive and irrelevant features, the useful feature set is listed in Table 2.

Table 2. List of useful features.

Name of parameters	**No. of features**	**Selected features**
Mean	2	DC_x, DC_z
Standard deviation	1	SD_z
rms value	3	RMS_x, RMS_y, RMS_z
Correlation among axes	2	$Corr_{xy}, Corr_{zx}$
Entropy	1	$EntropyX$
Principal component coefficients	3	PCA_x, PCA_y, PCA_z
Zero crossing rate	3	ZCR_x, ZCR_y, ZCR_z
Interquartile range	2	IQR_y, IQR_z
Energy	3	E_x, E_y, E_z
Spectral bandwidth	2	SB_y, SB_z
Total features	**22**	

A program is developed to calculate the features from the data set. The features are calculated for all the instances of the data set of Bengali characters. Then, the features, with their labels are inserted into an application for training the classification models. Following are the steps to train models:

- Inserting features with label
- Specifying a validation scheme
- Selecting useful features from the feature set
- Selecting models to train
- Observe results

Validation techniques in machine learning are used to calculate the ML model's error rate, that is close to the actual error rate as possible, i.e., it protects the models against overfitting. There are three validation options in the application used for training: Cross-Validation, Holdout validation, and No-Validation. The holdout validation technique is preferable for large data sets. The data set prepared in this study is comparatively small as there are only fifty instances for a character. Therefore, the Cross-Validation technique is selected to train the models. By partitioning the data set into folds and evaluating the accuracy of each fold, this validation technique protects against overfitting. In this case, the data are partitioned into five folds. After specifying the validation technique, the features listed in Table 2 are selected from the set of features to train the ML models.

The supervised learning models included in the application are Quadratic Discriminant, Cosine KNN, Coarse Gaussian SVM, Gaussian Naïve Bayes, Cubic SVM, Linear SVM, Weighted KNN, Coarse KNN, Linear Discriminant, Medium Gaussian SVM, Medium KNN, Fine Gaussian SVM, Coarse Tree, Fine KNN, Cubic KNN, Fine Tree, Quadratic SVM, and Medium Tree. Each of these models is trained individually using the specified characteristics from the Bengali air-writing dataset.

4. Results

One of our objectives is to create a comprehensive dataset of Bengali characters using a motion sensor-based air-writing technique. A data set is developed that contains 3050 instances of 61 Bengali characters. The data included in the data set are raw data, that is, no modification or processing is performed on data received from the sensor. The graphical representation of raw air-writing data of the symbol ‘ka (ক)’ is shown in Figure 18(a) and the preprocessed form of this symbols of air-writing is shown in Figure 18(b). This initial step of processing involves smoothing and standardizing the data using the z-score technique. It is observed from Figure 18(b) that after preprocessing the high-frequency noise is removed, and all the samples in the data sequences are organized on the same scale due to the normalization process.

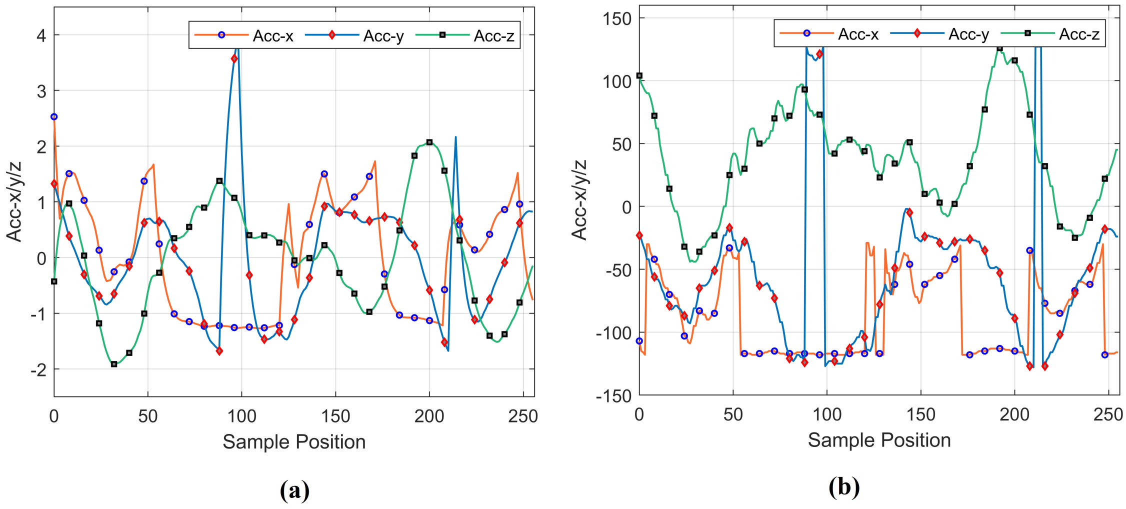


Figure 18. Graphical representation of (a) preprocessed data, and (b) raw sensor data of Bengali character ka (ক).

After preprocessing, Figures 19 and 20 show the graphical representation of air-writing for all Bengali characters. Despite the fact that fifty occurrences of one character are taken, only one instance of each character is displayed in the figures.

After preprocessing, the features listed in Table 2 are calculated. These features are used to train the classification model mentioned in the preceding section. Then, the accuracy of different models is evaluated. In Table 3, the accuracy of various classification models is listed. The medium Gaussian SVM shows height accuracy (96.5%) in recognition of the air-writing of Bengali characters. In addition, the Linear Discriminant, Gaussian Naïve Bayes, Kernel Naïve Bayes, Linear SVM, Quadratic SVM, Cubic SVM, Fine KNN, Medium KNN, Cosine KNN, Weighted KNN, and Cubic KNN all have greater than 90 percent accuracy.

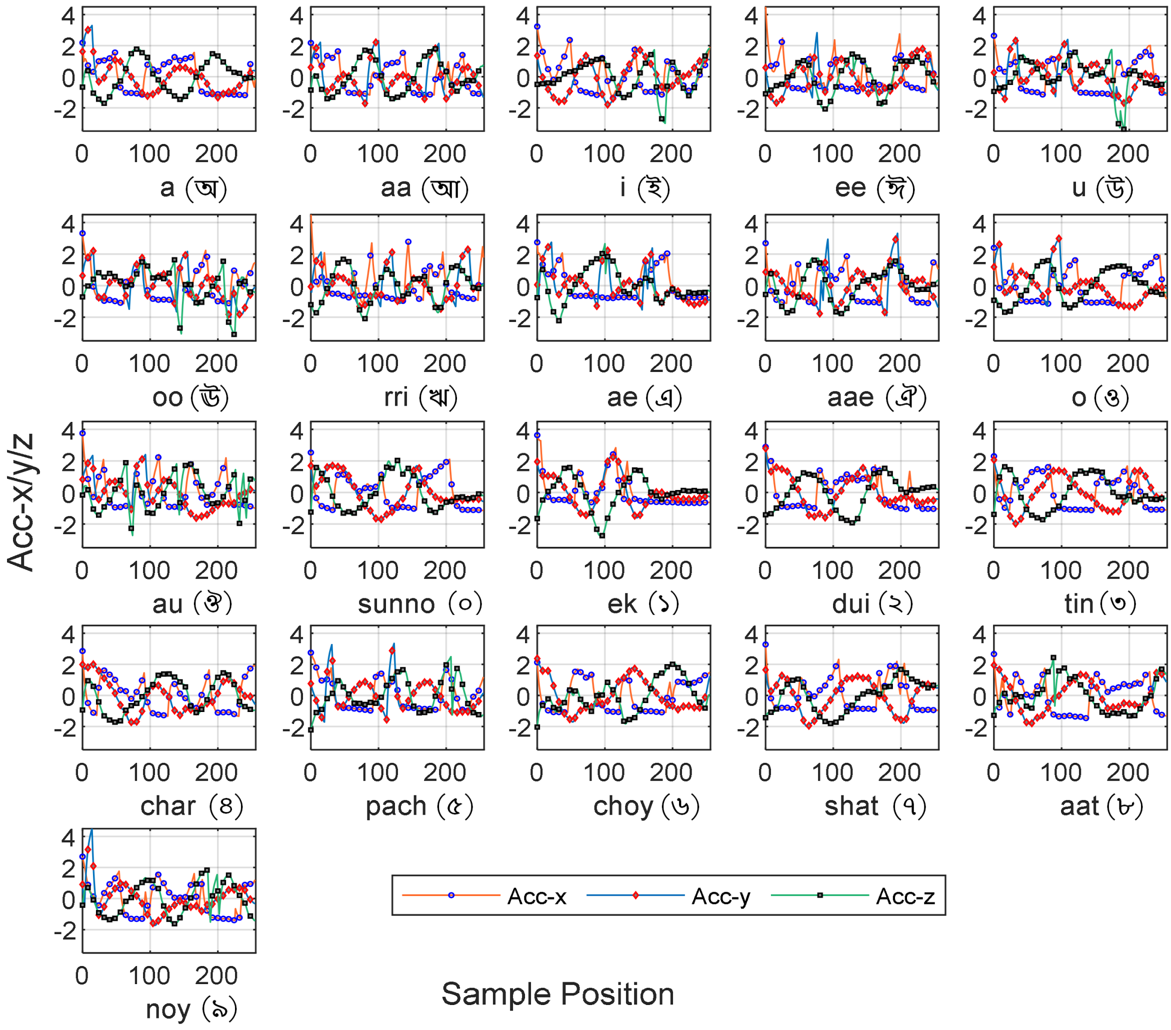


Figure 19. Graphical representation of air-writing data of Bengali vowels and digits.

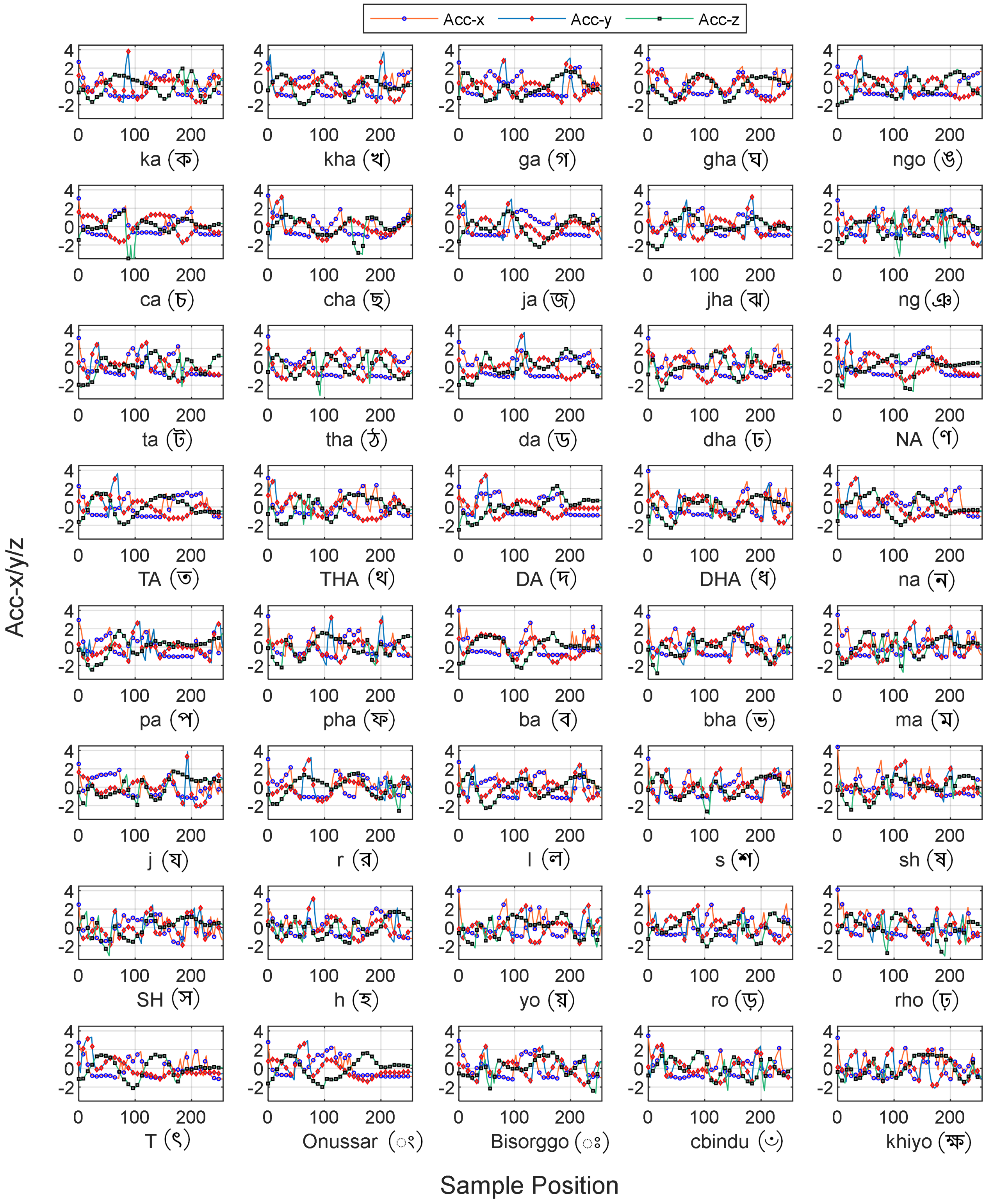


Figure 20. Graphical representation of air-writing data of Bengali consonants.

Table 3. Accuracy of Bengali character recognition by different classification models.

Classification model	Accuracy	Classification model	Accuracy
Fine Tree	60.3%	Medium Tree	22.0%
Coarse Tree	6.1%	Linear Discriminant	95.4%
Gaussian Naïve Bayes	93.3%	Kernel Naïve Bayes	92.9%
Linear SVM	95.3%	Quadratic SVM	95.8%
Cubic SVM	94.9%	Fine Gaussian SVM	35.4%
Medium Gaussian SVM	**96.5%**	Coarse Gaussian SVM	91.7%
Fine KNN	94.1%	Medium KNN	91.7%
Coarse KNN	80.5%	Cosine KNN	90.8%
Weighted KNN	93.4%	Cubic KNN	90.6%

A bar chart in Figure 21 illustrates the accuracy of individual characters in air-writing using the classification model "Medium Gaussian SVM", which provides the highest average accuracy among the mentioned models in Bengali character recognition from their air-writing. In the bar chart, the horizontal axis denotes different Bengali characters, and the vertical axis indicates the accuracy of the corresponding characters in air-writing. All character has an accuracy greater than 90% except for u (উ), jha (ঝ), and ro (ড়). The average accuracy for all characters is 96.5%. The average accuracy for Bengali consonants, Bengali vowels, and Bengali numbers is 96.6%, 95.81%, and 97.2%, respectively.

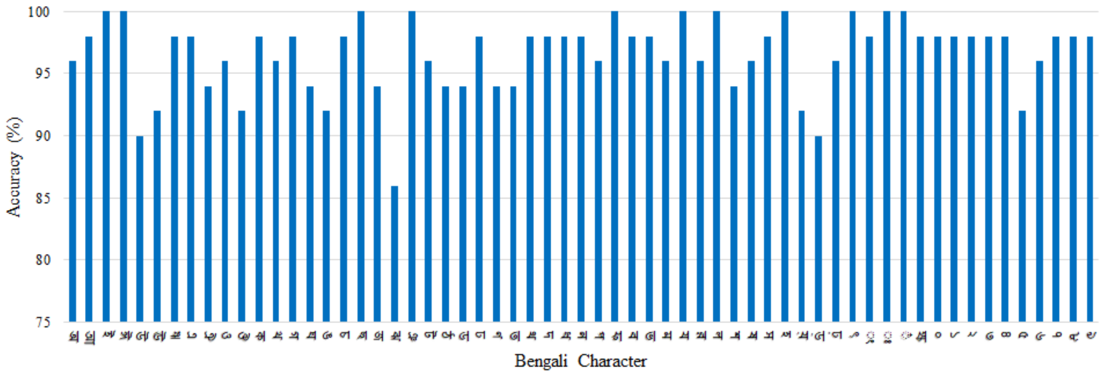
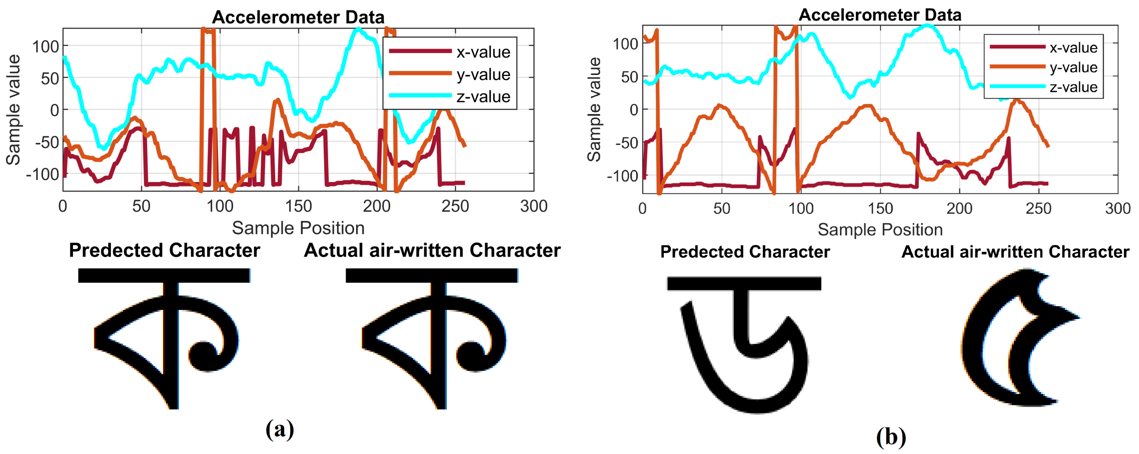


Figure 21. Accuracy of individual Bengali characters in air-writing.

The performance of the model (Medium Gaussian SVM) that exhibits the highest accuracy during the training session is evaluated in real-time after the training session. For this task, an application is developed. The application has taken the acceleration data from the data acquisition system in real-time when Bengali characters were written in the air. Following that, the application immediately calculated the features from the collected data, provided them to the trained classification model, and displayed the model's decision. Two cases of real-time air-writing classification are shown in Figure 22. In the instance of Figure 22(a), ka (ক) is written in the air. The top of the image displays the hand movement data that was received for writing the letter ka (ক). The bottom left portion of the figure displays the predicted character, while the bottom right portion displays the actual character. In this case, the character was accurately classified by the trained classifier. The trained classifier incorrectly predicted the character written in the air in case 2 (shown in Figure 22(b) to be da (ড) when it was actually the Bengali digit pach (৫).



(a)

(b)

Figure 22. An illustration of a Bengali character being **(a)** successfully classified **(b)** misclassified in real-time air-writing.

In this way, the classification model was evaluated 100 times more for categorizing ten randomly selected Bengali characters from the air-writing data. The test outcome is represented in Table 4. The randomly chosen characters were ka (ক), kha (খ), ek (১), i (ই), cha (ছ), tin (৩), rri (ঋ), ngo (ঙ), pach (৫), and jha (ঝ). Each character was written ten times in the air. The character ka (ক) was observed to be correctly identified six times out of ten attempts, while being incorrectly classified three times as pha (ফ) and once as r (র).

Table 4. Real-time performance of the trained classification model in recognition of Bengali character from air-writing.

Character	No. of attempts										**Accuracy**
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	
ka (ক)	ফ	✓	✓	ফ	ফ	✓	র	✓	✓	✓	60.0%
kha (খ)	✓	✓	✓	✓	✓	✓	✓	✓	গ	✓	90.0%
ek (১)	✓	✓	✓	✓	✓	✓	০	✓	✓	✓	90.0%
i (ই)	✓	✓	ঈ	✓	✓	✓	০	✓	✓	✓	80.0%
cha (ছ)	✓	ই	হ	✓	✓	✓	✓	✓	হ	✓	70.0%
tin (৩)	ত	ত	✓	✓	ও	✓	✓	ত	✓	✓	60.0%
rri (ঋ)	✓	✓	✓	✓	✓	✓	✓	ধ	✓	✓	90.0%
ngo (ঙ)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100.0%
pach (৫)	ড	✓	ড	✓	✓	✓	✓	✓	✓	✓	80.0%
jha (ঝ)	✓	✓	✓	✓	✓	✓	ব	✓	✓	✓	90.0%
	Overall accuracy										81%

Our results have been compared with some of the most current studies on air-writing in other languages, including research on Bengali air-writing. The comparison is shown in Table 5. The recognition of Bengali characters (only numbers) in air-writing has only been the subject of one published study. The study was based on computer vision, and the identification rate for Bengali numbers was determined to be 95.4%; however, our study, which includes not only numerals but also all Bengali characters, reported 96.5% accuracy. The comparison with other studies shows that the existing supervised machine learning models predict the created data set for Bengali air-writing more accurately than many other models that have been suggested for other languages.

Table 5. Comparison of the result with recent studies.

Research work	Language	Method	Accuracy
[31]	English	Computer vision based	96.11%
[32]	English	Computer vision based	86.9%
[33]	Chinese	Computer vision based	98.11%
[34]	Japanize	Computer vision based	92.5%
[35]	English Digit	Kinnect sensor based	96.8%
[36]	English Letter	Motion Sensor based	95.0%
[17]	English Letter	Motion sensor based	94.3%
[37]	English	Motion sensor based	88.4%
[14]	English Letter	Radar sensor based	98.33%
[38]	English Letter	WiFi signal based	88.74%
[39]	Latin	Leap Motion	72.25%
[20]	Bengali digit	Computer vision based	95.4%
Proposed	Bengali Alphabet	Motion sensor based	96.5%

5. Conclusions

The idea of using air writing to input text on devices is very new, and studies are being conducted to improve the technique. Based on the research papers we have collected, the majority of the studies are being performed to recognize air-writing for the English language. This research is focused on the development of the air-writing model for the Bengali language. We introduce a data collection approach for gathering hand motion data for Bengali air-writing. Additionally, a motion sensor-based dataset is created for Bengali characters. The data set contains 50 occurrences of each Bengali character from five distinct people. Then, the useful features of the Bengali air-writing data set are recognized and the most effective machine learning model is identified for recognizing Bengali air-writing among existing well-known supervised machine learning models. The medium Gaussian SVM classifier showed the maximum training accuracy, which is 96.5%. For this model, the accuracy achieved by providing real-time air-writing data is also measured for 10 randomly selected Bengali characters, and the accuracy is 81%. In the suggested method, air-writing requires the entire hand to move, which is quite time-consuming and laborious. In the future, we can create a system that allows users to write Bengali characters by moving only one finger.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by the European University of Atlantic.

Conflict of interest

The authors declare no conflict of interest.

References

1. A. Dash, A. Sahu, R. Shringi, J. Gamboa, M. Z. Afzal, M. I. Malik, et al., Airscript-creating documents in air, In: *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, 2017, 908–913. https://doi.org/10.1109/ICDAR.2017.153
2. X. Lin, Y. Chen, X. Chang, X. Liu, X. Wang, Show: Smart handwriting on watches, In: *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, **1** (2018), 151. https://doi.org/10.1145/3161412
3. The Bengali language and the history of its evolution, LingoStar, 2021. Available from: https://lingo-star.com/bengali-language/?v=4326ce96e26c.
4. M. S. Alam, K. C. Kwon, M. A. Alam, M. Y. Abbass, S. M. Imtiaz, N. Kim, Trajectory-based air-writing recognition using deep neural network and depth sensor, *Sensors*, **20** (2020), 376. https://doi.org/10.3390/s20020376
5. O. De, P. Deb, S. Mukherjee, S. Nandy, T. Chakraborty, S. Saha, Computer vision based framework for digit recognition by hand gesture analysis, In: *2016 IEEE 7th annual information technology, electronics and mobile communication conference (IEMCON)*, 2016. https://doi.org/10.1109/IEMCON.2016.7746361
6. S. Poularakis, I. Katsavounidis, Low-complexity hand gesture recognition system for continuous streams of digits and letters, *IEEE T. Cybernetics*, **46** (2016), 2094–2108. https://doi.org/10.1109/TCYB.2015.2464195
7. C. Qu, D. Zhang, J. Tian, Online kinect handwritten digit recognition based on dynamic time warping and support vector machine, *J. Inform. Comput. Sci.*, **12** (2015), 413–422.
8. S. Mohammadi, R. Maleki, Air-writing recognition system for Persian numbers with a novel classifier, *The Visual Comput.*, **36** (2020), 1001–1015. https://doi.org/10.1007/s00371-019-01717-3
9. P. Kumar, R. Saini, S. K. Behera, D. P. Dogra, P. P. Roy, Real-time recognition of sign language gestures and air-writing using leap motion, In: *2017 fifteenth IAPR international conference on machine vision applications (MVA)*, 2017. https://doi.org/10.23919/MVA.2017.7986825
10. P. Kumar, R. Saini, P. P. Roy, D. P. Dogra, Study of text segmentation and recognition using leap motion sensor. *IEEE Sens. J.*, **17** (2017), 1293–1301. https://doi.org/10.1109/JSEN.2016.2643165

11. X. Qu, W. Wang, K. Lu, J. Zhou, Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network, *Pattern Recogn. Lett.*, **111** (2018), 9–15. https://doi.org/10.1016/j.patrec.2018.04.001
12. J. Gan, W. Wang, K. Lu, In-air handwritten Chinese text recognition with temporal convolutional recurrent network, *Pattern Recogn.*, **97** (2020) 107025. https://doi.org/10.1016/j.patcog.2019.107025
13. P. Wang, J. Lin, F. Wang, J. Xiu, Y. Lin, N. Yan, et al., A gesture air-writing tracking method that uses 24 GHz SIMO radar SoC, *IEEE Access*, **8** (2020), 152728–152741. https://doi.org/10.1109/ACCESS.2020.3017869
14. M. Arsalan, A. Santra, K. Bierzynski, V. Issakov, Air-writing with sparse network of radars using spatio-temporal learning, In: *2020 25th international conference on pattern recognition (ICPR)*, 2021. https://doi.org/10.1109/ICPR48806.2021.9413332
15. F. Khan, S. K. Leem, S. H. Cho, In-air continuous writing using UWB impulse radar sensors, *IEEE Access*, **8** (2020), 99302–99311. https://doi.org/10.1109/ACCESS.2020.2994281
16. M. K. Chakravarthi, R. K. Tiwari, S. Handa, Accelerometer based static gesture recognition and mobile monitoring system using neural networks, *Procedia Comput. Sci.*, **70** (2015), 683–687. https://doi.org/10.1016/j.procs.2015.10.105
17. Y. Yin, L. Xie, T. Gu, Y. Lu, S. Lu, AirContour: Building contour-based model for in-air writing gesture recognition, *ACM T. Sensor. Network*, **15** (2019), 44. https://doi.org/10.1145/3343855
18. S. Xu, Y. Xue, Air-writing characters modelling and recognition on modified CHMM, In: *2016 IEEE international conference on systems, man, and cybernetics (SMC)*, 2016. https://doi.org/10.1109/SMC.2016.7844452
19. J. S. Wang, F. C. Chuang, An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition, *IEEE T. Ind. Electron.*, **59** (2012), 2998–3007. https://doi.org/10.1109/TIE.2011.2167895
20. P. Roy, S. Ghosh, U. Pal, A CNN based framework for unistroke numeral recognition in air-writing, In: *2018 16th international conference on frontiers in handwriting recognition (ICFHR)*, 2018. https://doi.org/10.1109/ICFHR-2018.2018.00077
21. Coursera, Data processing and feature engineering with MATLAB, Available form: https://www.coursera.org/learn/feature-engineering-matlab.
22. Entropy calculation, information gain & decision tree learning, 2020. Available form: https://medium.com/analytics-vidhya/entropy-calculation-information-gain-decision-tree-learning-771325d16f
23. T. Giannakopoulos, A. Pikrakis, *Introduction to audio analysis: A MATLAB® approach*, 1st Eds, Cambridge, Massachusetts, US: Academic Press, 2014.
24. E. Scheirer, M. Slaney, Construction and evaluation of a robust multifeature speech/music discriminator, In: *1997 IEEE international conference on acoustics, speech, and signal processing*, 1997. https://doi.org/10.1109/ICASSP.1997.596192
25. M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications*, Springer Cham, 2015. https://doi.org/10.1007/978-3-319-21945-5
26. M. A. Kader, M. A. Ullah, M. S. Islam, A real-time classification model for Bengali character recognition in air-writing, In: *Computer vision and image analysis for industry 4.0*, 1st Eds, Chapman and Hall/CRC, 2023.

27. Javatpoint, Regression vs. classification in machine learning, Available from https://www.javatpoint.com/regression-vs-classification-in-machine-learning.
28. A. Burkov, *The hundred-page machine learning book*, 1st Eds, Quebec City, QC, Canada: Andriy Burkov, 2019.
29. M. Mohammed, M. B. Khan, E. B. M. Bashier, *Machine learning: Algorithms and applications*, 1st Eds, Boca Raton: CRC Press, 2016. https://doi.org/10.1201/9781315371658
30. B. Dickson, Machine learning: What is dimensionality reduction? 2021. Available from: https://bdtechtalks.com/2021/05/13/machine-learning-dimensionality-reduction/.
31. S. Mukherjee, S. A. Ahmed, D. P. Dogra, S. Kar, P. P. Roy, Fingertip detection and tracking for recognition of air-writing in videos, *Expert Syst. Appl.*, **136** (2019), 217–229. https://doi.org/10.1016/j.eswa.2019.06.034
32. V. Joseph, A. Talpade, N. Suvarna, Z. Mendonca, Visual gesture recognition for text writing in air, In: *2018 second international conference on intelligent computing and control systems (ICICCS)*, 2018. https://doi.org/10.1109/ICCONS.2018.8663176
33. J. Gan, W. Wang, K. Lu, A new perspective: Recognizing online handwritten Chinese characters via 1-dimensional CNN, *Inform. Sci.*, **478** (2019), 375–390. https://doi.org/10.1016/j.ins.2018.11.035
34. S. Hayakawa I. Goncharenko, Y. Gu, Air writing in Japanese: A CNN-based character recognition system using hand tracking, In: *2022 IEEE 4th global conference on life sciences and technologies (LifeTech)*, 2022. https://doi.org/10.1109/LifeTech53646.2022.9754825
35. C. Wang C. Y. Su, C. L. Lin, A novel recognition system for digits writing in the air using coordinated path ordering, In: *HotMobile '15: Proceedings of the 16th international workshop on mobile computing systems and applications*, 2015, 9–14. https://doi.org/10.1109/ICIIBMS.2015.7439500
36. C. Xu, P. H. Pathak, P. Mohapatra, Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch, In: *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, 2015, 9-14. https://doi.org/10.1145/2699343.2699350
37. Y. Luo, J. Liu, S. Shimamoto, Wearable air-writing recognition system employing dynamic time warping, In: *2021 IEEE 18th annual consumer communications & networking conference (CCNC)*, 2021. https://doi.org/10.1109/CCNC49032.2021.9369458
38. Z. Fu, J. Xu, Z. Zhu, A. X. Liu, X. Sun, Writing in the air with WiFi signals for virtual reality devices *IEEE T. Mobile Comput.*, **18** (2019), 473–484. https://doi.org/10.1109/TMC.2018.2831709
39. P. Kumar, R. Saini, P. P. Roy, U. Pal, A lexicon-free approach for 3D handwriting recognition using classifier combination, *Pattern Recogn. Lett.*, **103** (2018), 1–7. https://doi.org/10.1016/j.patrec.2017.12.014