

Arabic character recognition in air-writing based on motion data from 3-axis accelerometer

Mohammed Abdul Kader

*Department of Electrical and Electronic Engineering
International Islamic University Chittagong (IIUC), Bangladesh*

Article info

Keywords

Accelerometer sensor
Air-Writing
Arabic character
Human-Machine
Interaction
Machine learning

Abstract

Air-writing is a cutting-edge, non-touch human-machine interaction technique that enables users to input text into digital devices by moving their hands in free space, instead of using conventional input devices like keyboards and touch screens. This approach appears to be one of the most effective ways to enter text into digital systems in the future. English air-writing has received significant scholarly attention, but no studies on Arabic air-writing were found. In this research, a system is developed to recognize Arabic characters in air-writing based on motion data from a 3-axis accelerometer. A data acquisition system is constructed to record hand movements during air-writing. Each Arabic letter is written 25 times in the air using this data acquisition system, and a motion-sensor-based Arabic air-writing dataset is prepared. Using this dataset, several supervised machine learning models have been trained, and their accuracy has been determined. It is observed that the Fine KNN and Quadratic SVM models have demonstrated the highest accuracy (98.5%) in identifying Arabic characters from air-writing among the various available supervised machine learning models.

Corresponding author

Mohammed Abdul Kader

Department of Electrical & Electronic Engineering, International Islamic University Chittagong, Chattogram, Bangladesh - 4318. Email: kader05cuet@gmail.com

1. Introduction

Nowadays, people input text or commands through touch screens or keyboards on digital devices. However, with the advent of Virtual Reality and Augmented Reality, it is predicted that in the near future, individuals will be able to access digital content through a simple pair of eyeglasses (Yin, Xie, Gu, Lu & Lu, 2019). This innovation eliminates the need to carry items like mobile phones and laptops. Nevertheless, a challenge arises concerning text input in the absence of traditional input devices like keyboards or touch

screens. Voice commands and voice typing are effective ways to address this issue. However, in this instance, a new issue appears. Voice typing is not at all a good alternative if you wish to input text while maintaining the privacy in public areas, moreover in noisy environment a voice typing is still very inefficient. In all scenarios, using air-writing as a text input method on future devices can be a smart choice. Writing linguistic symbols or words in a free space with hand or finger movements is familiar as "air-writing." People will use this method to write in the air, and the writing will be displayed on the screen i.e. the text will be received by the digital systems as input. Numerous studies on air-writing have already been conducted. However, the majority of them discuss air-writing of English alphabets or words. Arabic air-writing has received very little scholarly attention.

Arabic is one of the Semitic languages and nearly 380 million people speak Arabic as their mother language (Kanan, et al., 2019). Moreover, Arabic is the language of the Holy Quran, due to which people of the Muslim religion around the world use this language. Since air-writing is a potential way for inputting text to the digital devices in the coming days, therefore, in the near future, to increase the use of Arabic language in the digital world, we must develop air-writing model for this language. The main goal of this study is to create an air-writing data set based on motion sensor for Arabic characters, and then use that data set to examine the accuracy of various supervised machine learning models in recognizing Arabic characters in air-writing. There has been a lot of research on air-writing in several languages, where different sensing methods have been applied to identify hand gestures. According to the information obtained from reading the relevant paper, the research on air-writing can be divided into four groups based on the sensing method for detecting hand movements during air-writing such as computer vision based, RADAR sensor based, WiFi signal based, and motion sensor-based air-writing. Computer vision techniques involve using cameras and image processing algorithms to interpret and analyze visual data. It has high accuracy in detecting and tracking hand movements. However, the performance of this technique can be impacted by changes in lighting, and it is also limited in portability because it requires cameras or depth sensors (Mukherjee, Ahmed, Dogra, Kar, & Roy, 2019). WiFi sensor-based techniques utilize variations in WiFi signal strength caused by a user's hand movements to detect air-writing patterns. It is non-invasive and can work in various lighting conditions. However, this method has less accuracy compared to computer vision or other methods, and performance is affected by the presence of obstacles or interference in the WiFi signal (Yin, Xie, Gu, Lu, & Lu, 2019). RADAR (Radio Detection and Ranging)

sensor-based air-writing techniques employ radio waves to detect hand movements and translate them into text or commands. This method is suitable to work in various lighting conditions, including darkness, however, it requires specialized RADAR hardware, making it less accessible (Arsalan, & Santra, 2019). Motion sensor-based air-writing techniques use accelerometers, gyroscopes, or inertial measurement units (IMUs) to detect and track hand movements. This method is able to recognize gestures with high accuracy in any environment. It can be implemented using small and portable devices without using a specialized hardware setup like a camera or RADAR sensor (Amm, Georgi, & Schultz, 2014). The motion sensor-based air-writing method appeared to be the most useful, according to the discussion in the preview. On English air-writing, a lot of earlier work has been done using these techniques. There is also some research on air-writing in other languages. The recognition of English letters in air-writing has been proposed in numerous earlier works (Baig, Fahad Khan, & Beg, 2013; Mall, Rani, Khatri, 2021; Vaidya, Pravanth, & Viji, 2022). Some proposed models to identify English digits in air-writing was made in several researches (Alam, Kwon, Alam, Abbass, Imtiaz, & Kim, 2020; Wang, Su, & Lin, 2015; Roy, Ghosh, & Pal, 2018). Additionally, in some research papers (Joseph, Talpade, Suvarna, & Mendonca, 2018; Chen, Su, & Chien, 2019; Ghosh, Mitra, 2022; Anjaneyulu, Jampaiah, Karthik, & Vijetha, 2017), methods to distinguish both English letters and digits in air-writing were suggested. Recognition of air-writing based on computer vision of selected Chinese letters (Zhang, Ye, Jin, Feng, & Xu, 2013), Japanese letters (Hayakawa, Goncharenko, & Gu, 2022), Persian digits (Mohammadi, & Maleki, 2020), and Bengali digits (Roy, Ghosh, & Pal, 2018) are proposed in the mentioned research papers. It has been found that earlier studies have predominantly identified English alphabet and number in air-writing. Air-writing of Arabic alphabet has not been the subject of any research.

In this paper, the motion data (acceleration towards x, y, and z axes) of hand during air writing of Arabic characters are used to test different supervised machine learning models in the classification of Arabic characters in air-writing. The noteworthy contributions of this research are as follows- A data acquisition system is developed to record the data of hand movement during air-writing of Arabic characters.

A motion sensor-based air-writing data set of Arabic characters is developed.

This data set has been used to train a number of supervised machine learning models, and the performance of those models has been evaluated.

The remaining parts of the article are arranged as follows. The overview

of the system is illustrated in Section 2. The methodology, which covers gradual development of the Arabic character recognition system in air-writing, is explained in Section 3. Sections 4 is dedicated to discuss the experimental results. Conclusion and further work are discussed in Section 6.

2. Overview of the system

The main challenge of this research is to prepare a data set. A data acquisition system is developed that can record hand movements in real-time in order to prepare the data set. For the machine learning model to be able to classify the various Arabic characters in the data set, the acceleration data for each character must be distinct. It is ensured by preparing a unique, simple-to-move, and memorable strokes for each letter when air-writing. Figure 1 represents the complete overview of the Arabic character recognition system in air-writing. The first objective of this system is to construct a data acquisition system to collect the data of hand movement during air writing of Arabic characters. The user has to carry this unit in hand. The data acquisition system consists of a 3-axis accelerometer sensor. The output (acceleration towards x, y, and z axes) of the accelerometer sensor changes with the change of orientation and acceleration of the sensor while air-writing. The data of the accelerometer sensor which carries the information of hand movement during air-writing is then transmitted to the computer via Bluetooth. An application using MATLAB software is developed which receives the air-writing data. The application has the option to save data with a label to an excel file.

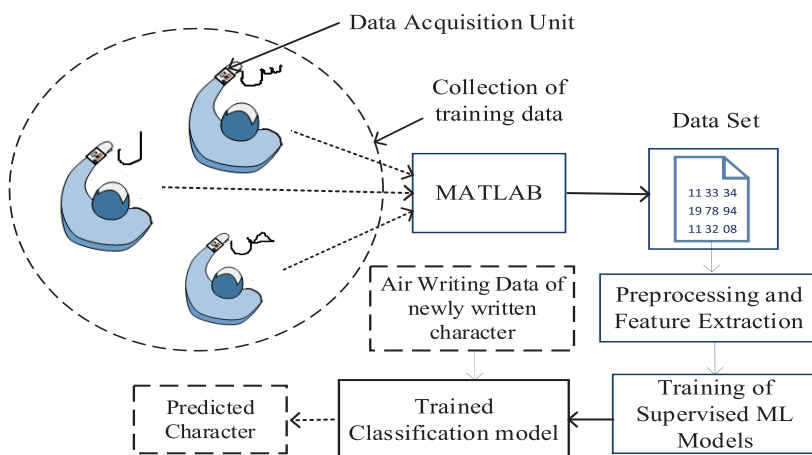


Figure 1
Complete Block Diagram of the System.

The second objective is to use this data acquisition system to prepare a data set of air-writing of Arabic Characters. While holding the data collecting unit in hand, each of the Arabic alphabet is repeatedly written in the air and a data set is created by saving the air-writing data together with a label for each instance of air-writing in an excel file. This data set is then utilized to train supervised machine learning models like KNN, SVM, Decision Tree, etc. after preprocessing, and the accuracy of these models in identifying Arabic characters is evaluated.

3. Methodology

This section outlines the gradual development of the Arabic character recognition system in air-writing. The development of the data acquisition system and data set, data preprocessing, feature extraction, and training of the classification model are addressed in this section.

3.1 Development of data acquisition system and data set

The data acquisition system's goals are to collect real-time air-writing data for preparing an air-writing data set of Arabic alphabets based on motion sensor and to receive data for the real-time recognition of Arabic air writing. The data acquisition system consists of a data collection unit, receiving unit, and an application developed in MATLAB software. Figure 2 displays an overall functional diagram of the data acquisition unit, where dotted blocks indicating the various units. The main part of data collection unit is a 3-axis accelerometer (ADXL345). To share data with the microcontroller (Microcontroller-1), it follows to the Inter-Integrated Circuit (I2C) communication protocol. The data collection unit also has a PUSH button and buzzer. Before writing, the user must click the push button. After that the buzzer will give an alarm to indicate that the microcontroller has begun reading the accelerometer sensor. The user will get two seconds to finish the air-writing. The microcontroller then stops reading the accelerometer sensor, which will be indicated by the buzzer beeping once again. The air-writing data is subsequently transferred over Bluetooth from the data collection unit to the receiver unit.

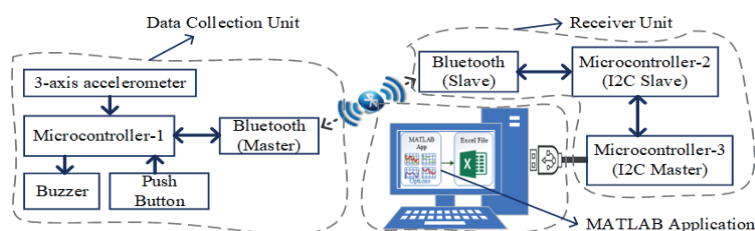


Figure 2
Overall functional diagram of data acquisition system.

The receiver unit has two microcontrollers. Data is transferred from one microcontroller (Microcontroller-2), which is connected to the data acquisition unit via Bluetooth, to another microcontroller (Microcontroller-3), which is linked to the computer via USB. Finally, the data is received by an application developed by MATLAB software. Figure 3 shows the implementation of the data collection unit and receiving unit, and Figure 4 displays a screenshot of the MATLAB application. This application has an option to visualize the data and also to save the data in an excel file. Finally, this data collection system is utilized to develop the data set of the Arabic alphabets written in the air. Each Arabic letter was written ten times in the air by five different persons who were holding the data collection unit in hand. The data was consistently saved in an Excel file i.e. data has been taken 25 times for each Arabic character. The strokes of the Arabic alphabet during air-writing to prepare the data set are shown in Table I.

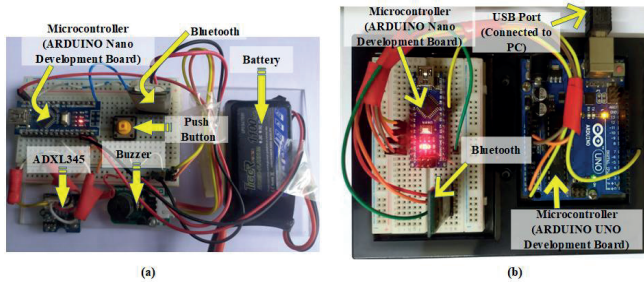


Figure 3
Implementation of (a) data collection unit and (b) receiver unit.

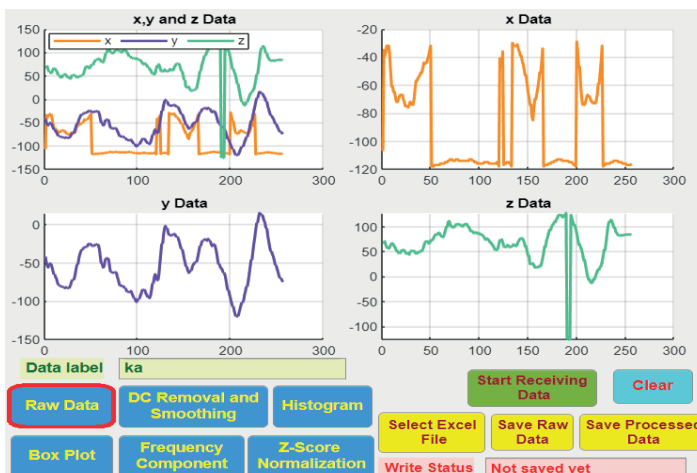


Figure 4
MATLAB application to receive and visualize air-writing data.

Table I
Proposed strokes of Arabic character in air-writing.

Character	Stroke	Character	Stroke	Character	Stroke	Character	Stroke
ا alif		ب baa		ت taa		ث thaa	
ج jim		ح haa		خ Khaa		د daal	
ذ dhaal		ر raa		ز zay		س siin	
ش shiin		ص saad		ض daad		ط TAA	
ظ zaa		ع ayn		غ Ghayn		ف faa	
ق qaaf		ك kaaf		ل laam		م miim	
ن nun		ه HAA		و waaw		ي yaa	
ء hamza		لا lam alif					

N.B. dot (●) indicates the start and triangle (▶) indicated the end of a stroke.
 Since there are 30 letters (including ء and لا) in Arabic, therefore in total 750 instances were taken in the data set. The length of each air-writing instance is 2 seconds where the sampling rate of reading the acceleration data is 128 Hz and each sample is an 8-bit signed number (value ranges from -128 to 127). Thus, we can represent the data of air writing of a character (sometimes referred as an instance) in the data set as shown in Eq. (1). Here, x, y, and z are three vectors of length 256 each and carry the information of the hand motion towards x, y, and z-axes respectively during air-writing.

$$\begin{aligned}
 x(n) &= \{x_0, x_1, x_2, \dots, x_{255}\} \\
 y(n) &= \{y_0, y_1, y_2, \dots, y_{255}\} \\
 z(n) &= \{z_0, z_1, z_2, \dots, z_{255}\} \\
 l &= \{label\}
 \end{aligned}
 \tag{1}$$

3.2 Data preprocessing

Data preprocessing is a crucial stage to prepare it for the extraction of valuable insights for a machine learning model. Sometimes, real-world data

has noise, missing values, or the data is in an improper format that prevents it from being easily incorporated into machine learning models. In this application, during air writing, the sensor's disturbances and the users' unintentional trembling invariably contaminate the measured signals with high-frequency noise. Smoothing operation on data can remove this high-frequency noise. A leaky integrator is a smoothing filter which has a low computational cost compared to other smoothing filters (Kader, M. A., Hasan, M. J., Emon, M. A. I., Karim, A., Mahmud, S., & Tahsin, T., 2022). Therefore, a leaky integrator is used in this application. The smoothing operation is performed by using Eq. (2) to Eq. (4).

$$x(n) = \lambda x(n-1) + (1-\lambda)x(n) \quad (2)$$

$$y(n) = \lambda y(n-1) + (1-\lambda)y(n) \quad (3)$$

$$z(n) = \lambda z(n-1) + (1-\lambda)z(n) \quad (4)$$

Here, n indicates the sample position, and λ is a constant which ranges from 0 to 1. The closer the value of λ to 1, the greater the smoothing (Vetterli, M., 2014).

The fact that the samples of x , y , and z are not in the same range for the air writing of various Arabic characters is another significant issue. It may confuse the machine learning models that employ distance as a metric in decision-making since variables do not contribute evenly to a distance calculation. The standardization or normalization process is a solution to this problem. The concept of putting different variables on the same scale is known as standardization and this idea allows us to compare scores from several ranges of variables. Standardization can be accomplished in a variety of ways. The z-score normalization procedure is employed in this work. This is the process of converting every value in a dataset with an average of 0 and a standard deviation of 1. Eq. (5) to Eq. (7) are used to normalize the data of hand movement along the x , y , and z -axis respectively, where n indicates sample position and N is the total number of samples.

$$x(n) = \frac{x(n) - \frac{\sum x(n)}{N}}{\sqrt{\frac{\sum [x(n) - \frac{\sum x(n)}{N}]^2}{N}}} \quad (5)$$

$$y(n) = \frac{y(n) - \frac{\sum y(n)}{N}}{\sqrt{\frac{\sum [y(n) - \frac{\sum y(n)}{N}]^2}{N}}} \quad (6)$$

$$z(n) = \frac{z(n) - \frac{\sum z(n)}{N}}{\sqrt{\frac{\sum [z(n) - \frac{\sum z(n)}{N}]^2}{N}}} \tag{7}$$

3.3 Feature extraction

A feature is a detail in the data that can be used to make predictions. The process of choosing features is extremely iterative. This could be as simple as choosing the appropriate pieces of data, or it could involve extracting features from the data and performing simple/complex statistical computations on them. However, a parameter must have variance, be random, and be unique in order to be a feature (Michael, 2021). An accelerometer sensor-based method for handwritten digit and gesture identification is developed by Wang, & Chuang (2011), where they employed statistical analysis and sensor data transformations to identify features from the data set. In this study, primarily the parameters mean, standard deviation, rms value, correlation among axis, entropy and spectral energy are selected in the feature set. However, some parameters have been employed as features, which is addressed in the result section, subject to having the properties of a parameter in order to be a feature. The equations used to calculate the features are discussed below. Recall that we had three discrete sequences (x, y, and z) at one data instant, where, the sequences represent acceleration of hand in x-direction, y-direction, and z-direction respectively during air writing.

The mean of sequence x denoted by \bar{x} can be calculated by Eq. (8), where N implies the number of values in x. Similarly, we can calculate the mean of sequence y and z.

$$\bar{x} = \frac{1}{N} \sum x \tag{8}$$

The standard deviation (STD) is the square root of the variance. Eq. (9) for instance, compute the STD of the vector x. STD convey the information about the scattering of the values in a data sequence. We can calculate the standard deviation of the sequences y and z in the same way.

$$\sigma_x = \sqrt{\frac{1}{N} \sum (x - \bar{x})^2} \tag{9}$$

Root mean square (rms) of a sequence is the square root of the average of the square of all the values in the sequence. For example, the rms value of the data vector x is calculated by Eq. (10).

$$rms_x = \sqrt{\frac{1}{N} \sum x^2} \quad (10)$$

The rms values of the sequences y and z can be determined using the same equation.

The correlation between the vectors of two axis is obtained taking the fraction of the covariance to the product of standard deviations of the vector. For example, the correlation ($corr_{xy}$) of data x on x-axis and y on y-axis is found by Eq. (11), where $cov(x, y)$ is computed by Eq. (12). Here, N implies the size of data, σ_x and σ_y which are the standard deviations of vector x and y, and \bar{x} and \bar{y} are the average of x and y. Similarly, we can calculate the correlation of acceleration between y and z axis ($corr_{yz}$), and z and x axis ($corr_{zx}$).

$$corr_{xy} = \frac{cov(x, y)}{\sigma_x \sigma_y} \quad (11)$$

$$cov(x, y) = \frac{\sum(x - \bar{x})(y - \bar{y})}{N} \quad (12)$$

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 the Eq. (13). Here, c indicates the number of interval (R1, R2, . . . , Rc) in x; P_i is the probability of a sample belonging to the interval Rc. In similar manner, the entropy H_y and H_z are calculated.

$$H_x = - \sum_{i=1}^c P_i \log_2(P_i) \quad (13)$$

The sum of the square of the Fast Fourier Transform coefficients gives the spectral energy of a sequence. The spectral energy of a discrete sequence x is obtained by Eq. (14), where, N is the size of the sequence, and X_k is the k^{th} FFT coefficient. Similarly, we can compute the energy E_y of the sequence y and E_z of the sequence z.

$$E_x = \frac{1}{N} \sum_{k=1}^N |X_k|^2 \quad (14)$$

The list of selected parameters for feature selection are shown in Table II.

Table II

Description of the model parameters.

Name of the parameter	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	$EntropyX, EntropyY, EntropyZ$
Energy	3	E_x, E_y, E_z
Total Features	18	

3.4 training of classification model

The data set developed for Arabic air-writing consists of labeled data. Such labeled data are handled with the supervised learning model in machine learning. There are numerous varieties of supervised learning models, including Naive Bayes Classifiers, K-nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Trees, etc. In this study, the classification accuracy of various models has been evaluated using a variety of features from the features set that are derived from the data set to determine the best suitable supervised learning model for the classification of Arabic air-written characters. Training of different models and accuracy calculation for selecting various features are performed using an application (App) in MATLAB name ‘Classification Learner’. The Classification Learner app has feature that allows users to train models to classify data. This program allows users to do experiment with supervised machine learning using a variety of classifiers. An interface of ‘Classification Learner App’ is shown in Figure 5. The app has options to explore data, select features, define validation schemes, train models, and evaluate results. It can also perform automatic training to find the optimal classification model type, such as support vector machines, decision trees, discriminant analysis, logistic regression, closest neighbors, naive Bayes, kernel approximation, ensemble classification, and neural network classification. The validated model's output is displayed in the app. The verified model results are reflected in diagnostic measurements like model accuracy and visualizations like a scatter plot or a confusion matrix chart. In Classification Learner Application (CLA), a model is trained using two validation schemes. K-fold cross-validation is the default method by which the app protects against overfitting. Another option is holdout validation. Furthermore, there is the option to train a model on all accessible data without validation.

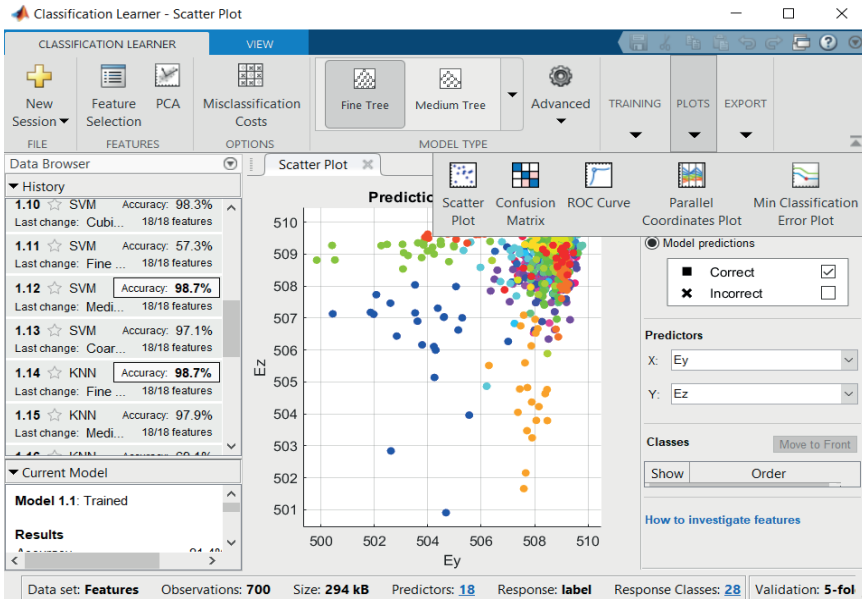


Figure 5
Interface of classification learner App

This study employs k-fold cross-validation with k set to 5. Data is divided by the CLA into 5 nearly equal subsets (or folds) at random. Four subsets (20 instances) are utilized to train the model, and one subset (5 instances) is used to validate the model. There are 5 iterations of this method, with each subset being applied for validation exactly once. This allows the use of every instance of a character as training and testing data. The accuracy of the model is expressed as the mean accuracy over the five folds.

4. Results and discussions

Creating an air-writing data set of Arabic characters based on motion sensor is one of the key objectives of this research. The data set has been prepared, and section 3.1 discusses the details of how it is developed. The data in the data set are raw data; no modifications or processing were made to the data before it was added to the data set. However, before extracting features, some preprocessing is carried out, such as smoothing and normalization. Figure 6(a) and 6(b) depict the graphical depiction of the raw and preprocessed air-writing data for the Arabic character

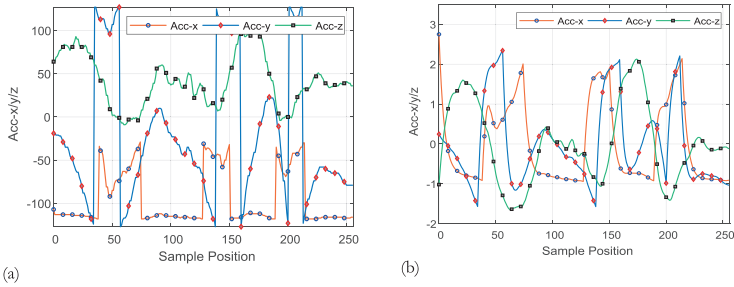


Figure 6
 Graphical representation of the air-writing data of Arabic character baa (ب) Illustration in (a) Raw sensor data, in (b) Preprocessed data (After smoothing and z-score normalization).

“ba (ب)”. As shown by the Figure 6(b), the high-frequency noise and DC components are eliminated after preprocessing, and the normalization procedure helped to place all of the samples in the data sequences on the same scale. The graphical representation of air-writing of all Arabic characters after preprocessing are shown in Figure 7.

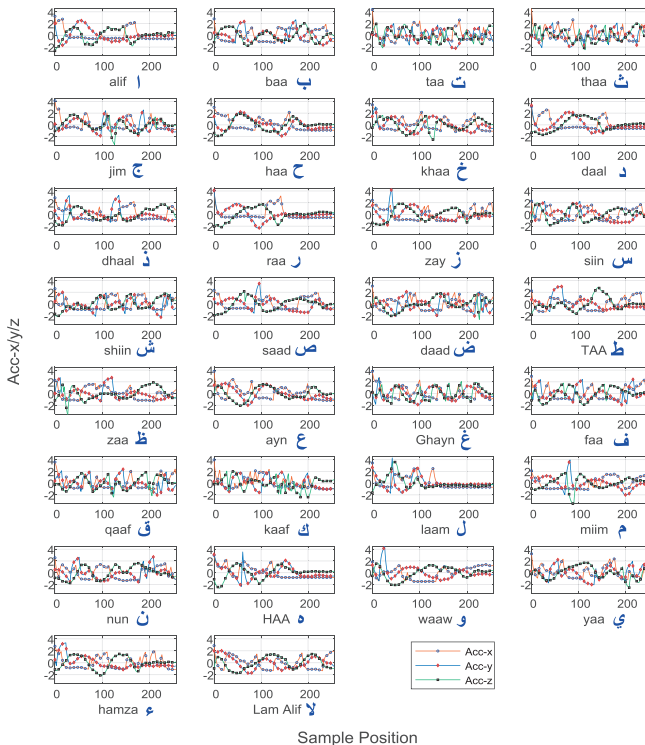


Figure 7
 Graphical representation of air-writing data of all Arabic alphabets

Despite the fact that one character has 25 occurrences obtained from five distinct individuals, only one instance of each character is displayed in the illustration.

For the training of supervised learning models, six parameters or variables are primarily chosen as features. However, in an air-writing instance there are three discrete sequences i.e., acceleration towards x-axis, y-axis and z-axis (the sequences are represented as x, y, and z in previous sections). As a result, there are three features for each parameter. Therefore, 18 features are attained from six parameters which are already listed in Table II. The features that have high variance in distinguishing various Arabic characters from air writing data are included in the Table II. Now, among the selected features there may have some features which are highly correlated to 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 feature set can be identified by analyzing the heatmap of the covariance of different features. Eq. (12) is used to determine the covariance between two features. Figure 8 shows a heatmap illustrating the covariance between different features. The heatmap clearly shows that the mean of acceleration in x-direction ($\bar{D}x$) is closely associated with its standard deviation (SDx) and spectral energy ($RMSx$). As a result, it will not be necessary to include all three features in the model because their effects on the target variable will be identical. Only $\bar{D}x$ is therefore thought of as a feature that requires less computation compared to SDx and $RMSx$. We can also ignore the parameters standard deviation of acceleration in the y-direction (SDy) and entropy of acceleration in all directions (EntropyX, EntropyY, and EntropyZ) for the same reason. It is shown in Figure 9.

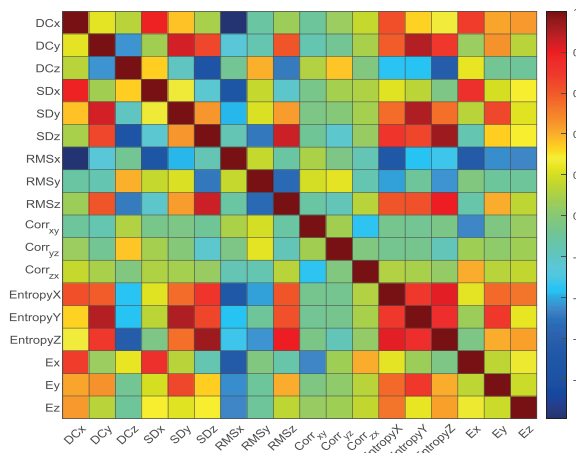


Figure 8
Heatmap illustrating the covariance between different features.

The number of features left after eliminating the redundant feature is 12. Various supervised machine learning models are trained using these features. The accuracy of the models and their parameter settings in identifying Arabic characters from air writing is reported in Table III. For the developed air writing data set of Arabic characters, it is seen that the majority of K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) models exhibit outstanding classification accuracy in recognizing Arabic characters from air-writing data. Both the medium Gaussian SVM and the fine KNN models show maximum accuracy of 98.5%.

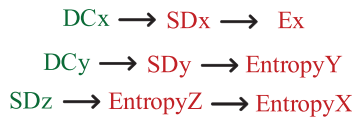


Figure 9
Reduction of highly correlated feature.

Table III
Classification accuracy of different supervised learning models for the recognition of air-writing of Arabic Characters.

SL.	Name of the Model	Accuracy	Parameter Setting
1	Linear Discriminant	97.9%	Covariance structure: Full
2	Quadratic Discriminant	76.3%	Covariance structure: Diagonal
3	Gaussian Naïve Bayes	97.6%	Distribution of numeric Predictors: Gaussian Distribution of categorical Predictors: MVMN Distribution for numeric Predictors: kernel
4	Kernel Naïve Bayes	97.5%	Distribution for categorical Predictors: MVMN Kernel type: Gaussian Support: Unbounded
5	Linear SVM	97.9%	Kernel function: Linear, Kernel scale: Automatic, Box constraint level: 1, Multiclass method: One-vs-One, Normalize data: true
6	Quadratic SVM	98.5%	Kernel function: Quadratic, Kernel scale: Automatic, Box constraint level: 1, Multiclass method: One-vs-One, Normalize data: true
7	Cubic SVM	98.1%	Kernel function: Cubic, Kernel scale: Automatic, Box constraint level: 1, Multiclass method: One-vs-One, Normalize data: true
8	Fine Gaussian SVM	56.7%	Kernel function: Gaussian, Kernel scale: 1.7, Box constraint level: 1, Multiclass method: One-vs-One, Normalize data: true
9	Medium Gaussian SVM	98.4%	Kernel function: Gaussian, Kernel scale: 6.7, Box constraint level: 1, Multiclass method: One-vs-One, Normalize data: true
10	Coarse Gaussian SVM	96.7%	Kernel function: Gaussian, Kernel scale: 27, Box constraint level: 1, Multiclass method: One-vs-One, Standardize data: true
11	Fine KNN	98.5%	No. of neighbors:1, Distance metric: Euclidean, Distance weight: Equal, Standardize data: true
12	Medium KNN	96.1%	No. of neighbors:10, Distance metric: Euclidean, Distance weight: Equal, Standardize data: true
13	Coarse KNN	68.5%	No. of neighbors:10, Distance metric: Euclidean, Distance weight: Equal, Standardize data: true

14	Cosine KNN	95.3%	No. of neighbors:10, Distance metric: Cosine, Distance weight: Equal, Standardize data: true
15	Cubic KNN	94%	No. of neighbors:10, Distance metric: Minkowski, Distance weight: Equal, Standardize data: true
16	Weighted KNN	97.5%	No. of neighbors:10, Distance metric: Euclidean, Distance weight: Squared inverse, Standardize data: true

The confusion matrix for Fine KNN model is shown in Figure 10. It demonstrates that, out of 25 instances, the model correctly classified 23 to 25 instances for each Arabic character. Figure 11 displays the Fine KNN model's accuracy in detecting each individual Arabic character from air-writing. The findings of this research have been compared with some of the most recent studies on air-writing for other languages since there was no research on Arabic air-writing. The comparison is shown in Table IV. The comparison shows that the existing supervised machine learning models predict the developed data set for Arabic air-writing more accurately than many other models that have been proposed for other languages.

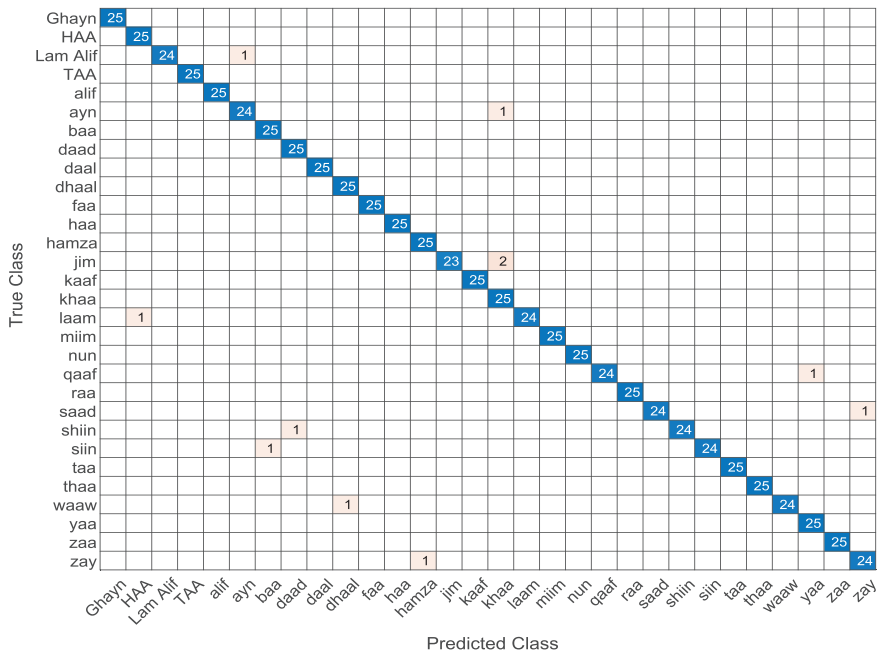
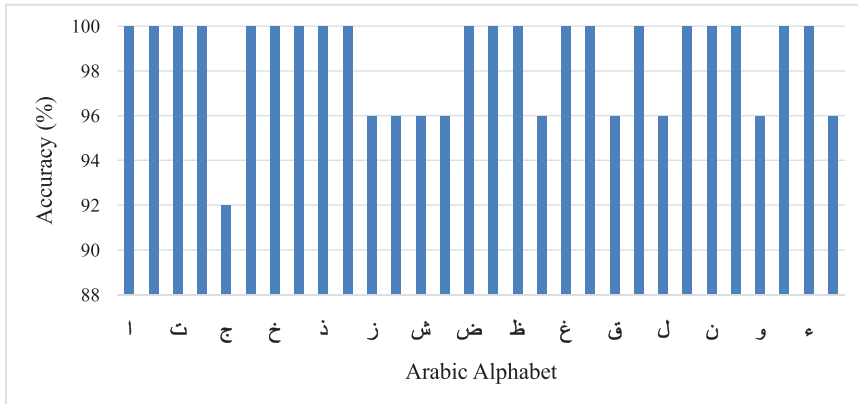


Figure 10
Confusion matrix in recognition of Arabic characters in air-writing by Fine KNN.

**Figure 11**

Accuracy in identifying Arabic characters from air-writing by Fine KNN model.

Table IV

Comparison of the results with recent studies.

Research work	Language	Method	Accuracy
Mukherjee, Ahmed, Dogra, Kar, & Roy, 2019	English	Computer vision based	96.11%
Joseph, Talpade, Suvarna, & Mendonca, 2018	English	Computer vision based	86.9%
Hayakawa, Goncharenko, & Gu, 2022	Japanese	Computer vision based	92.5%
Wang, Su, & Lin., 2015	English Digit	Kinnect Sensor based	96.8%
Xu, Pathak, & Mohapatra, 2015	English letter	Motion sensor based	95%
Yin, Xie, Gu, Lu, & Lu, 2019	English letter	Motion sensor based	94.3%
Luo, Liu, & Shimamoto, 2021	English letter and number	Motion sensor based	88.4%
Arsalan, & Santra, 2019	English letter	Radar sensor based	98.33%
Fu, Xu, Zhu, Liu, & Sun, 2018	English letter	WiFi signal based	88.74%
Proposed	Arabic Alphabet	Motion sensor based	98.5%

5. Conclusion

In this research, a data acquisition system is constructed to capture data of Arabic characters written in the air, and a motion sensor-based air-writing dataset is created using this system. The developed dataset is then used to assess the accuracy of various supervised machine learning models in recognizing Arabic characters in air-writing. It has been observed that existing machine learning models effectively recognize Arabic letters from air-writing data, achieving a maximum accuracy of 98.5%. This technique holds the potential to facilitate inputting Arabic text into digital devices without the need for a keyboard or touch screen, thereby enhancing user-friendly human-machine interaction. However, there are some drawbacks to the suggested air-writing technique. It requires the entire hand to move, making it time-consuming, labor-intensive, and demanding in terms

of space. In the future, the proposed system could be further developed to recognize Arabic air-writing based on finger movements, potentially making the air-writing process quicker, more effortless, and space-efficient.

References

- Alam, M. S., Kwon, K. C., Alam, M. A., Abbass, M. Y., Imtiaz, S. M., & Kim, N. (2020). Trajectory-based air-writing recognition using deep neural network and depth sensor. *Sensors*, *20*(2), 376.
- Amma, C., Georgi, M., & Schultz, T. (2014). Airwriting: A wearable handwriting recognition system. *Pers Ubiquit Comput* *18*, 191–203. <https://doi.org/10.1007/s00779-013-0637-3>.
- Anjaneyulu, P., Jampaiah, Y., Karthik, R., & Vijetha, T. (2017, December 07-08). *Air writing recognition modeling and recognition of characters, words and connecting motions*. Paper presented at the IEEE International Conference on Intelligent Sustainable Systems (ICISS) (pp. 1112-1115), Palladam, India.
- Arsalan, M., & Santra, A. (2019). Character recognition in air-writing based on network of radars for human-machine interface. *IEEE Sensors Journal*, *19*(19), 8855-8864, doi: 10.1109/JSEN.2019.2922395.
- Baig, F., Fahad Khan, M., & Beg, S. (2013). Text writing in the air. *Journal of Information Display*, *14*(4), 137-148, DOI: 10.1080/15980316.2013.860928.
- Chen, Y. H., Su, P. C., & Chien, F. T. (2019, October 15-18). *Air-writing for smart glasses by effective fingertip detection*. Paper presented at the IEEE 8th Global Conference on Consumer Electronics (GCCE) (pp. 381-382), Osaka, Japan.
- Fu, Z., Xu, J., Zhu, Z., Liu, A. X., & Sun, X. (2018). Writing in the air with WiFi signals for virtual reality devices. *IEEE Transactions on Mobile Computing*, *18*(2), 473-484.
- Ghosh, S., Mitra, A. (2022). Air writing-based automated room. In M. L. Kolhe, S. B. Jaju, Diagavane, P.M. (Eds.), *Smart Technologies for Energy, Environment and Sustainable Development*, Vol 1. Springer Proceedings in Energy. Springer, Singapore. https://doi.org/10.1007/978-981-16-6875-3_53.
- Hayakawa, S., Goncharenko, I., & Gu, Y. (2022, March 7-9). *Air writing in Japanese: A CNN-based character recognition system using hand tracking*. Paper presented at the IEEE 4th Global Conference on Life Sciences and Technologies (LifeTech) (pp. 437-438), Osaka, Japan.

- Joseph, V., Talpade, A., Suvarna, N., & Mendonca, Z. (2018, June 14-15). *Visual gesture recognition for text writing in air*. Paper presented at the Second International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 23-26), Madurai, India.
- Kader, M. A., Hasan, M. J., Emon, M. A. I., Karim, A., Mahmud, S., & Tahsin, T. (2022, February 26-27). *Hand Gesture Based Speed and Direction Control of DC Motor Using Machine Learning Algorithm*. Paper presented at the International Conference on Innovations in Science, Engineering and Technology (ICISSET) (pp. 284-289), Chittagong, Bangladesh.
- Kanan, T., Sadaqa, O., Aldajeh, A., Alshwabka, H., AlZu'bi, S., Elbes, M., ... & Alia, M. A. (2019, April 9-11). *A review of natural language processing and machine learning tools used to analyze arabic social media*. Paper presented at the IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) (pp. 622-628), Amman, Jordan.
- Luo, Y., Liu, J., & Shimamoto, S. (2021, January 9-12). *Wearable air-writing recognition system employing dynamic time warping*. Paper presented at the IEEE 18th Annual Consumer Communications & Networking Conference (CCNC) (pp. 1-6), Las Vegas, NV, USA.
- Mall, J., Rani, K., & Khatri, D. (2021). Air Writing: Tracking and Tracing. In: Singh, S.K., Roy, P., Raman, B., Nagabhushan, P. (eds.) *Computer Vision and Image Processing. CVIP 2020. Communications in Computer and Information Science*, vol 1378. Springer, Singapore. <https://doi.org/10.1007/978-981-16-1103-2>
- Mall, J., Rani, K., & Khatri, D. (2021). Air Writing: Tracking and Tracing. In *Computer Vision and Image Processing: 5th International Conference, CVIP 2020, Prayagraj, India, December 4-6, 2020, Revised Selected Papers, Part III 5* (pp. 12-24). Springer Singapore.
- Mohammadi, S., & Maleki, R. (2020). Air-writing recognition system for Persian numbers with a novel classifier. *The Visual Computer*, 36(5), 1001-1015. <https://doi.org/10.1007/s00371-019-01717-3>
- Michael, R. (2015). Data Processing and Feature Engineering with MATLAB [MOOC]. *Coursera*. Retrieved May 28, 2021, from <https://www.coursera.org/learn/feature-engineering-matlab>.
- Mukherjee, S., Ahmed, S. A., Dogra, D. P., Kar, S., & Roy, P. P. (2019). Fingertip detection and tracking for recognition of air-writing in videos. *Expert Systems with Applications*, 136, 217-229.

- Mukherjee, S., Ahmed, S. A., Dogra, D. P., Kar, S., & Roy, P. P. (2019). Fingertip detection and tracking for recognition of air-writing in videos. *Expert Systems with Applications*, 136, 217-229. <https://doi.org/10.1016/j.eswa.2019.06.034>.
- Roy, P., Ghosh, S., & Pal, U. (2018, August 05-08). *A CNN based framework for unistroke numeral recognition in air-writing*. Paper presented at the 16th international conference on frontiers in handwriting recognition (ICFHR) (pp. 404-409), Niagara Falls, NY, USA.
- Vaidya, V., Pravanth, T., & Viji, D. (2022, March 10-11). *Air writing recognition application for dyslexic people*. Paper presented at the International Mobile and Embedded Technology Conference (MECON), pp. 553-558, Noida, India.
- Vetterli, M. (2014). The leaky integrator. *Coursera*. Retrieved April 2, 2021 from <https://www.coursera.org/lecture/dsp2/2-1-2-b-the-leaky-integrator-FNKpz>
- Wang, J. S., & Chuang, F. C. (2011). An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition. *IEEE Transactions on Industrial Electronics*, 59(7), 2998-3007. DOI: 10.1109/TIE.2011.2167895
- Wang, C., Su, C. Y., & Lin, C. L. (2015, November 28-30). *A novel recognition system for digits writing in the air using coordinated path ordering*. Paper presented at the International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS) (pp. 244-249), Okinawa, Japan.
- Xu, C., Pathak, P. H., & Mohapatra, P. (2015, February). *Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch*. Paper presented at the 16th International Workshop on Mobile Computing Systems and Applications (pp. 9-14), NY, USA.
- Yin, Y., Xie, L., Gu, T., Lu, Y., & Lu, S. (2019). AirContour: Building contour-based model for in-air writing gesture recognition. *ACM Transactions on Sensor Networks (TOSN)*, 15(4), 1-25.
- Zhang, X., Ye, Z., Jin, L., Feng, Z., & Xu, S. (2013). A new writing experience: Finger writing in the air using a kinect sensor. *IEEE MultiMedia*, 20(4), 85-93.