*applied sciences*

MDPI

*Article*

# Authentication by Keystroke Dynamics: The Influence of Typing Language

**Najwa Altwaijry**

Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; ntwaijry@ksu.edu.sa

**Abstract:** Keystroke dynamics is a biometric method that uses a subject's typing patterns for authentication or identification. In this paper we investigate typing language as a factor influencing an individual's keystroke dynamics. Specifically, we discern whether keystroke dynamics is contingent on the spatial arrangement of letters on the keyboard, or alternatively, whether it is influenced by the linguistic characteristics inherent to the language being used. For this purpose, we construct a new dataset called the Bilingual Keystroke Dynamics Dataset in two languages: English and Arabic. The results show that the authentication system is not contingent on the spatial arrangement of the letters, and is primarily influenced by the language being used, and a system that is used by bilingual users must take into account that each user should have two profiles created, one for each language. An average equal error rate of 0.486% was achieved when enrolling in English and testing on Arabic, and 0.475% when enrolling in Arabic and testing on English.

**Keywords:** keystroke dynamics; bilingual keystroke dynamics; authentication; cybersecurity; biometrics

## 1. Introduction

Identity verification is of paramount importance, especially considering the highly connected nature of today's world that allows remote access to data, information, and user accounts. One approach for user verification is to employ keystroke dynamics for authentication or identification. Authentication aims to show that the user is the actual legitimate user of the system, while identification aims to identify the user of the system. Keystroke dynamics is a behavioural biometric that can be used for both user authentication and identification. The timing information of a user while typing on a keyboard is measured, and then a suitable algorithm is chosen for user verification [1]. Thus, keyboard dynamics is a non-intrusive method that does not require any specialised hardware, making it practical and cost effective. It is an attractive option for security, especially considering that although a password may be known by an attacker, emulating the typing pattern of the user is difficult [2]. Conversely, as a behavioural biometric and not a physical biometric, e.g., a fingerprint, keyboard dynamics can be affected by a wide range of factors, such as the emotional state of the user, illness or lack of sleep, and so on [3]. Practical applications of keystroke dynamics abound. Keystroke dynamics can be used for the authentication of users on a system, as well as for detecting intrusions, either in an offline or online setting. User usage modelling is another area that may benefit from keystroke dynamics in order to develop systems that present services or advertisements appropriate to the user. Illegal activities may also be detected by keystroke dynamics, for example, when typing from anonymous accounts is detected on the web, it can be used to flag suspicious activity. Another area is the use of keystroke dynamics in forensics and surveillance, as regardless of the number of machines used, a criminal still retains their typing pattern. A recent survey [4] presented a comprehensive review and analysis of keystroke dynamics systems.

Researchers can collect many different features from the typing patterns of users. The most widely used is keydown–keydown latency, which is the time taken for a user to press

**Citation:** Altwaijry, N. Authentication by Keystroke Dynamics: The Influence of Typing Language. *Appl. Sci.* **2023**, *13*, 11478. https://doi.org/10.3390/app132011478

Academic Editor: Andrea Prati

Received: 17 September 2023
Revised: 8 October 2023
Accepted: 18 October 2023
Published: 19 October 2023

two consecutive keys [2,5]. It may be divided into two timing measures: keydown–keyup of the first key, known as the hold-time or dwell-time, and keyup–keydown of the first and second keys, known as the flight-time; see Figure 1. Timings involving more than two consecutive keys can also be measured, although they are less common in the literature, as are other measures such as collecting data on key pressure, typing speed, total duration and typing errors [5]. Using these features, a system may either be an anomaly detection system or a classification system. An anomaly detection system is one that is trained on the examples generated by the user, and when a new sample needs to be tested, it is compared to the model of the user saved in the system. The second approach is to use classification, where a system is trained on two or more users, and a new sample is classified into a user category during testing.

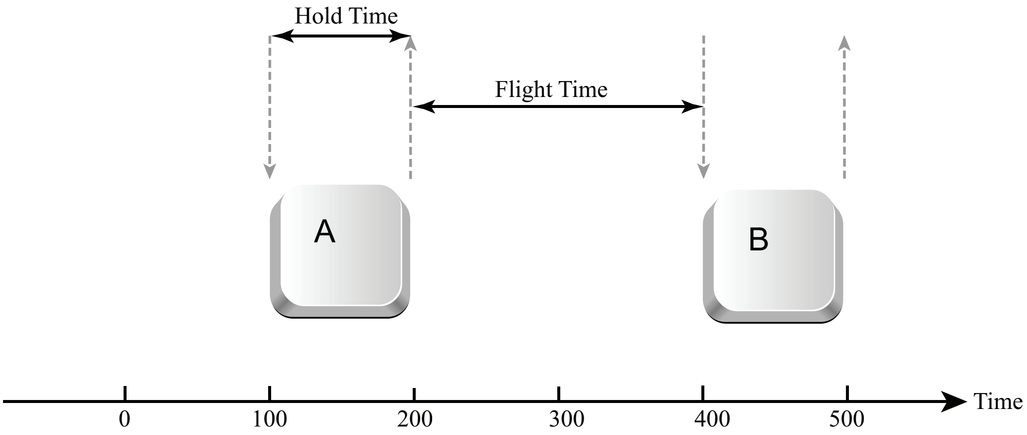


**Figure 1.** Hold and flight times.

Based on the literature, one can differentiate between static-text and dynamic-text keystroke dynamics [5]. In the former, the user enters a predefined string of text, and the system verifies the user at the initial interaction with the system, such as when a user enters a password or a pin code. The latter attempts to verify a user that has already gained access to the system during the period the user is interacting with the system. In other words, it is continuously re-authenticating the user.

Although approximately 43% of people worldwide are bilingual [6], there is a lack of bilingual keystroke dynamics research. Wahab et al. [7] studied the effects of using different keyboards with different language settings. They developed two free-text datasets, one using different keyboard types, and the other for bilingual English and Chinese speaking users. The bilingual dataset was collected using the user's personal keyboard, and was evaluated using two free-text algorithms, the ITAD metric [8] and D-Vectors model [9]. They reported that the use of different languages significantly affected the keystroke dynamics system's performance, with an average EER of 0.210% when enrolling in English and testing on Chinese, and 0.253% when enrolling in Chinese and testing on English.

For a bilingual user using a system employing dynamic-text keyboard dynamics, and in situations with high security requirements, the authentication system must be able to re-authenticate the user in the event the user switches their typing language. A system must not flag the user as an imposter simply for changing the typing language. For this reason, we study the impact of language as a factor that may affect keystroke dynamics. Specifically, in this paper, we are interested in answering the question of whether the keystroke dynamics of a user are dependent on the spatial distances between letters on the keyboard, such that the user's typing pattern is the same if the same keyboard is used, regardless of typing language, or whether the keystroke patterns for the same user are language dependent, and a system cannot verify a user enrolled in a language if the user switches to another. In order to answer our query, we study the effect of the typing language on bilingual users' keystroke dynamics. In particular, we collect a static-text dataset for English and Arabic speakers composed of a short password and a phrase, in both languages, and evaluate our dataset using four classification algorithms. As much as possible, we control the factors that influence keystroke dynamics. For example, we use one mode of data entry: a keyboard that is used by all participants, rather than the user's own keyboard as in Wahab et al. [7]. A user will be more familiar when using their own keyboard rather than a keyboard provided at work or school, and in large corporations with high security requirements, it is expected that the user may access the system from various locations. In addition, the aforementioned work was concerned with free-text entry, while in this work, we study the effects on static-text entry. This paper expands the frontier on bilingual keystroke dynamics by presenting the results of bilingual users that use static-text entry as well as using a single keyboard, and is the first work of its kind to deal with bilingualism with the Arabic language.

Furthermore, this work provides an answer to the question posed above: whether the spatial arrangement of letters on a keyboard is the primary factor that influences the keystroke dynamics of people and results in unique typing patterns, or whether a person would type differently depending on the language. If the former is true, then a classification system trained on one language and tested on another language for the same user should have similar results. If, however, the typing patterns are affected by other factors, such as the language itself or familiarity with the typing sequences of the most used letters or *n*-grams in languages—making a user more proficient at typing those sequences—then a person would not be able to replicate their typing patterns when typing in another language.

The contributions of this study are as follows: (1) a static-text bilingual dataset in English and Arabic in two parts (password and phrase) from 85 total users. This dataset is publicly available to researchers, and is the first of its kind in the field of bilingual keystroke dynamics for English/Arabic text, and the first of its kind for Arabic static-text. (2) Whether the typing language affects keystroke dynamics. This is an important consideration for security systems, especially in cases when users speak two or more languages. (3) The expected deterioration in EER when using an unfamiliar keyboard. This is discussed at length and compared with previous work in Section 4.2.3. Finally, (4) some interesting areas for future work.

The rest of this paper is organised as follows: Section 2 presents the literature review. Section 3 introduces our dataset, the data collection process, data preprocessing, and the classifiers used in this study. Section 4 presents the detailed results achieved by the four classifiers on the dataset, as well as discussing these results, and finally, Section 5 concludes this paper.

## 2. Literature Review

Most research on keystroke dynamics is focused on fixed- or static-text entry on computer keyboards. Recent trends in the literature, however, are more concerned with modern technology, including mobile devices [10]. Mobile devices introduce a number of new and different challenges compared with traditional keyboards [11], such as limited computational resources, functions such as predictive text, and keyboard sizes that affect typing patterns: typing with one or two hands, or just a finger, by the same user [10]. Existing research shows that statistical classification methods [12] as well as machine learning methods [13] can be used for the verification of users; however, the results fail to meet secure verification requirements [10]. The most used algorithms are support-vector-based classifiers and tree-based classifiers, including random forest, AdaBoost, and decision trees, all of which produce results with impressive accuracy [4]. Machine learning methods have been shown to outperform traditional methods when it comes to the accuracy of predictions [14].

Many factors influence keystroke dynamics. We can divide these factors into two categories: internal and external. Internal factors are those that are personal to the user, for example, emotional state, language, or amount of sleep. External factors are those that involve the hardware being used, such as keyboard type or computer type. We detail a number of studies that have dealt with external factors next.

A study by Oyebola [15] collected a dataset using a desktop computer and handheld devices, and found that different keyboards produced different results using the log-logistic distribution. These results corroborate other studies that found differences between personal computer keyboards and laptop keyboards [7]. In addition, the varying size of screens on handheld devices leads to changing system performance in various measures, such as accuracy and acceleration [16]. The system clock resolution also affects the detector's performance, with higher clock speeds achieving better performance [17]. The use of additional features obtained from specialised sensors available on smart phones has shown improved performance for keystroke dynamics user recognition, for example, pressure sensors [18], accelerometers [19], and gyroscopes [20].

As mentioned previously, keystroke dynamics are affected by factors internal to the user. Tsimperidis and Arampatzis [21] performed gender, age, and handedness classifications in social networks using keystroke dynamics. Another study by Tsimperidis et al. [22] attempted to profile the user's mother tongue based on their typing patterns for the Albanian, Bulgarian, English, Greek, and Turkish languages. The use of keystroke dynamics for emotion recognition was studied by Sağbaş et al. [23] to detect stress in users via data collected from accelerometers and gyroscopes using smartphones. Indeed, keystroke dynamics has been used to identify many emotional states, both positive emotions such as happiness, confidence and excitement, as well as negative emotions such as anger, sadness and fear, and a recent review by Yang and Qin [24] surveyed papers presenting many more emotional states. Cascan et al. [25] showed that keystroke dynamics can be used to classify user experience. A study by Lamiche et al. [26] combined gait characteristics from smartphones' accelerometers to improve the accuracy of keystroke dynamics systems.

The previous studies were all concerned with factors that affect keystroke dynamics. However, keystroke dynamics was found to be helpful in other fields such as medical, human–computer interaction (HCI), as well as the most well-known field, security. In the medical field, a number of studies have used keystroke dynamics for disease detection, such as the diagnosis of fine motor decline [27] and Parkinson's disease [28]. Keystroke dynamics has also been used to detect fatigue [29]. In the field of HCI, a study by Roy et al. [30] identified the age group of users, in order to prevent children from accessing specific content for their own protection, and also identified gender in order to target consumers with appropriate content. Keystroke dynamics was used in online examinations [31] and to accurately predict the educational level of users [32].

Finally, in the field of security, a recent paper [33] used metaheuristics to augment a neural network to improve keystroke dynamics. Touch gestures were used by Stylios et al. [34] to improve authentication, while Wang et al. [35] used mouse clicks. A key concern of security is preserving privacy, and while keystroke dynamics help with identification and authentication, the nature of the data collected raises privacy concerns [36], which can be ameliorated via privacy preserving methods, such as used by Acar et al. in [37]: fuzzy hashing and fully homomorphic encryption.

From the above papers, summarised in Table 1, it is clear that the literature on keystroke dynamics is rich and varied. However, there is room for more research in the area of keystroke dynamics in bi- or multi-language settings. This research is intended to address language as a factor that affects keystroke dynamics in a static-text setting, with implications for dynamic-text environments.

**Table 1.** Summary of papers presented in the literature review.

| Paper | Field | Input Type | Application |
|---|---|---|---|
| Campisis [12] | security | smartphone | verification |
| Kambourakis [13] | security | smartphone | authenticating |
| Oyebola [15] | dataset collection | keyboard, smartphone, tablet | any |
| Wahab [7] | security | keyboards | verification |
| Alsuhibany [16] | security | tablet | authentication |
| Killourhy [17] | clock testing | keyboard | any |
| Shekhawat [18] | security | keyboard with sensor | authentication |
| Ning [19] | health | smartphone | brain health |
| Senerath [20] | security | smartphone | authentication |
| Tsimperidis [21] | HCI | keyboard | gender, age, handedness |
| Tsimperidis [22] | HCI | keyboard | language |
| Saugbacs [23] | emotion | smartphone | stress |
| Cascone [25] | HCI | smartphone | age, gender, user experience |
| Lamiche [26] | security | smartphone | authentication |
| Roy [28] | health | keyboard | Parkinson's disease |
| Acien [29] | health | keyboard | mental fatigue |
| Roy [30] | HCI | smartphone | age, gender |
| Chen [31] | security | keyboard | authentication |
| Tsimperidis [32] | HCI | keyboard | education level |
| ElKenawy [33] | security | smartphone | authenticating |
| Stylios [34] | security | smartphone | authenticating |
| Wang [35] | security | keyboard, mouse | authenticating |
| Hatin [36] | security | smartphone | authentication with privacy |
| Acar [37] | security | keyboard | authentication with privacy |

## 3. Methodology

In this section, we describe the dataset design and data gathering method we used. We also describe data preprocessing and the models used for classification.

### 3.1. Bilingual Keystroke Dynamics Dataset (BKSD)

Datasets are integral to learning systems. Not only do they allow researchers to compare various classifiers, they also aid in answering specific questions put forth by researchers.

The Bilingual Keystroke Dynamics Dataset (BKSD) is a publicly available dataset (BKSD can be found at https://github.com/ntwaijry/BKSD (accessed on 17 October 2023)) collected during the period January 2022 to April 2022 between 8 a.m. and 2 p.m. As much as possible, we controlled various factors that affect keystroke dynamics, such as age, gender, dominant hand, and time of day. Participants were screened to select bilingual English- and Arabic-proficient typists from King Saud University, Riyadh, Saudi Arabia, in the Diriyah Campus. All typists were right-handed female students aged from 19 to 24 years old.

The total number of participants enrolled was 102, which we divided into two groups of 51 each, one for typing a password, and one for typing a phrase. Each participant entered the password or phrase a total of 400 times in each language, and a keylogger running in the background captured the timings of keystrokes: the system time for each keydown–keyup sequence for each key pressed. Each password or phrase attempt is recorded on one line for each user. The number of participants and the number of features collected in BKSD are in line with other datasets in the literature.

#### 3.1.1. Data Collection

**Password:** In order to compare a password entered in Arabic and in English, we selected a meaningful word with the same character sequence in both languages; see Figure 2. The password is a mix of letters, numbers and a special character, and is at least 8 characters long, fulfilling the general requirements for a strong password. We do not use a capital letter as those do not exist in Arabic. In English, the password is “thug-173”, and in Arabic, it is “فاعل - ١٧٣”. The letter positions for the English and Arabic passwords are the same on the keyboard used, and follow the same sequence. Both words are meaningful in both languages. This is an intentional choice, as it allows us to ascertain whether the most important factor is the letter positions on the keyboard or the language being used.

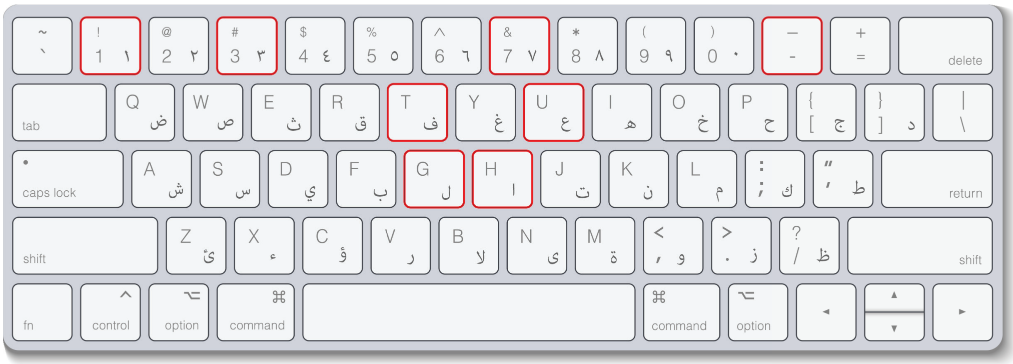
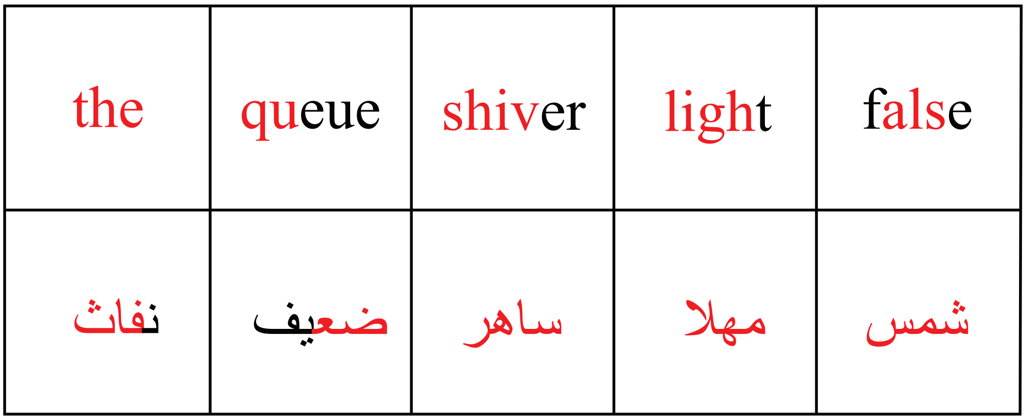


**Figure 2.** Password chosen in the dataset BKSD, with characters highlighted.

**Phrase:** Finding meaningful words in both languages with corresponding keystroke patterns is difficult, and so we do not follow the procedure outlined above for an entire phrase. Instead, we constructed two sentences, one in English and one in Arabic, where each sentence was composed of meaningful words separated by spaces, and some characters within each word overlap. The English sentence was “the queue shiver light false”, and the Arabic sentence was “نفاث ضعيف ساهر مهلا شمس”. The overlapping characters are highlighted in red in Figure 3.



| | | | | |
|---|---|---|---|---|
| the | queue | shiver | light | false |
| نفاث | ضعيف | ساهر | مهلا | شمس |

**Figure 3.** Overlapping characters, shown in red, in the phrase chosen in BKSD.

The most common bigrams in the English language present in the words selected in our dataset are listed in Table 2, with the corresponding most frequent bigrams in Arabic. The rank column indicates the rank of the bigram as measured by its frequency. As we only collect the overlapping characters, as highlighted in Figure 3, we denote the available bigrams in our dataset in Table 2 using an asterisk. Note that the dataset bigrams in English are more commonly used than the dataset bigrams in Arabic.

**Table 2.** Most frequent bigrams in Arabic and English. The text available in the BKSD dataset is denoted by an asterisk.

| Rank | English Bigram | Rank | Arabic Bigram |
|---|---|---|---|
| 1 | th * | 3 | لا * |
| 2 | he * | 14 | عل * |
| 4 | er | 53 | سا * |
| 19 | al * | 55 | مس * |
| 25 | se | 88 | فا * |
| 31 | ve | 116 | عي |
| 35 | hi * | 169 | مه * |

### 3.1.2. Data Cleaning and Preparation

The data collected contained raw system timings for each keydown–keyup sequence, and thus is unsuitable for use as is. First, we convert the data from raw system time into the actual time required: for each row, we subtract the first keydown time from all columns in the row. Next, we extract the hold time, the keydown–keydown time, and the keyup–keydown time (flight time) for each row. We also calculate the total time required for the entire phrase or password. The user ID is finally used as the last column for each user as the class label.

The dataset is composed of four files, two for the password and two for the phrase, with one file in Arabic and one file in English for each. We removed any rows where the user made a mistake during typing, and so, the final number of rows for each user is 400 or less. The numbers of users who completed the data entry process were 50 and 35 for the password and phrase data, respectively.

## *3.2. Classification Models*

In this section, we present the four classification models implemented to test our dataset, as well as the preprocessing performed on the raw data provided in the BKSD dataset.

### 3.2.1. Data Preprocessing

We performed a number of preprocessing steps on the BKSD dataset. First, we used deep feature synthesis (DFS) [38], which provides automated feature engineering. The input data were transformed using two primitives from those provided by DFS: the *add* and *multiply* primitives. The resulting data were then normalised using Equation (1), such that the mean became zero and the variance became 1.

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

where $x$ is the input, $\mu$ is the mean, $\sigma$ is the standard deviation, and $x'$ is the resulting normalised output.

Recall that the four files in BKSD contain a list of all participants with their corresponding attempts, with each attempt on one line. This format is unsuitable for testing the dataset. So, the data are split such that each participant has her own attempts only in her file. These data represent the positive case for that user. The negative case (attack case) is simulated by collecting the first few attempts from the other participants' files. The final file is, thus, constructed for each user containing her positive attempts and negative attempts by other participants. Each user has two files, one for the password and one for the phrase. The password file contains the first five entries from 50 users as the negative class, while the phrase file contains the first 10 entries from 34 users as the negative class. We supplement the minority class (the attack class) using safe-level SMOTE [39].

#### 3.2.2. Classifiers

We implemented four classifiers to evaluate our datasets. As the purpose of this work is to study language as a factor affecting keystroke dynamics, we use four well-known classifiers: AdaBoost, decision tree (DT), random forest (RF), and support vector machine (SVM). These classifiers were carefully selected to present a consolidated view of the dataset. AdaBoost is considered an excellent algorithm that is scalable and can easily deal with missing values while avoiding overfitting. On the other hand, AdaBoost is sensitive to outliers and not interpretable. Decision trees are easy to use and do not require normalisation or scaling of the data. They also provide explainable results, and are insensitive to outliers; however, they are prone to overfitting and usually have worse prediction results compared to other machine learning algorithms. Random forests are also robust to outliers, and have a lower risk of overfitting while maintaining good prediction results. Unfortunately, they are very slow to train and can be sensitive to categorical features. SVMs work well with smaller datasets, such as ours, by defining the boundary between the classes. They are robust to noise and can generalise well, as well as avoiding overfitting. They are, however, expensive to train and their results cannot be interpreted. To choose the best parameters for each classifier, we employed the grid search algorithm to find the best parameters. For each user, we trained two instances of the classifier: one for Arabic, and one for English. The total number of classifiers trained for the password dataset was $4 \times 2 \times 50 = 400$, and for the phrase dataset, a total of $4 \times 2 \times 35 = 280$ classifiers were trained. Each classifier was tested on its corresponding testing data, as well as the test set data from the other language: 800 tests were performed on the password dataset, and 560 tests were performed on the phrase dataset.

For example, assume we are training an SVM for user-1, who entered password data in English and in Arabic. We train and test as follows:

1. SVM-Arabic-user-1: Trained on Arabic, tested on Arabic and English.
2. SVM-English-user-1: Trained on English, tested on Arabic and English.

If the keystroke dynamics of a user are indistinguishable across languages, i.e., a language has no effect on typing patterns, the results from 1 and 2 above should be similar. If, however, a language does affect the keystroke dynamics, then the classifiers should not be able to distinguish the same user typing in a different language. In the next section, we describe the performance measures used to test our dataset, and present the results of the four classifiers selected for the experiments, along with a discussion of those results. In addition, we test the effects of bigram frequencies to classify users.

### *3.3. Ethical Considerations*

Any study that includes human information analysis and the use of artificial intelligence may suffer from unintended bias, because AI itself is biased [40,41]. For example, the use of keystroke dynamics for targeted advertising based on factors such as age, college education, or gender can have the unintended results of unequal opportunities available to those involved when the audience receives "tailored" ads, such as career STEM ads [42]. When AI tools are used, gender bias can occur in recruitment [41], as well as racial discrimination [43]. Similarly, identifying emotions through AI raises issues regarding users' rights to privacy and the risks and harms that follow, which can be further exacerbated because "*legal consent is not required to capture data about emotions that is not personal (i.e., capable of identifying or singling-out a person)*" [44].

To address issues arising from the use of AI, researchers, legislators, and industry experts must be aware of the inherent dangers of using AI on society and individuals, in order to ensure privacy, fairness, and accountability [45]. For our dataset, we employ a doubly anonymous data collection process: at the beginning of the data collection process, informed consent was obtained from the study participants. They were asked to select a unique identifier only they knew, to log onto the system, and to submit their data. These identifiers were not connected to any identifying information of the participants. The identifiers were shuffled and mapped into unique numbers at the end of the experiment, such that neither the researchers nor the participants would be able to identify or map data to any participant; therefore, all data are anonymised.

## 4. Results and Discussion

### *4.1. Experimental Setup*

We investigate the effects of language on keystroke dynamics using the BKSD dataset and four classification algorithms. We evaluate our models and dataset using the Google Colab environment.

To evaluate the performance of keystroke dynamics classifiers, criteria that measure the frequency of errors that occur during imposter detection are used. There are two types of errors:

1. False rejection rate (FRR): Also known as a type I error. In this error, the classifier rejects a genuine user, i.e., classifies a user as an imposter. A small FRR is desirable, as only a small number of genuine users are refused access. The following equation is used to calculate the FRR if the attacker is considered in the positive class (i.e., 0: user, 1: attacker):
$$FRR = \frac{FP}{TN + FP} \tag{2}$$
where $FP$ stands for false positives, and $TN$ stands for true negatives.
2. False acceptance rate (FAR): Also known as a type II error. In this error, the classifier accepts an imposter, i.e., classifies an imposter as an authentic user. FAR rates are also known as miss rates. FAR errors are more critical than FRR errors, and thus, it is more desirable to have this error as low as possible. FAR is calculated as follows if the attacker is considered in the positive class (i.e., 0: user, 1: attacker):
$$FAR = \frac{FN}{TP + FN} \tag{3}$$
where $FN$ stands for false negatives, and $TP$ stands for true positives.

In a biometric system, both error measures outlined above should be minimised to ensure accurate and reliable system performance. In addition, a threshold is selected that bounds error rates in a system to enhance system performance. To determine the threshold value, two measures are employed:

1. Equal error rate (EER): Sometimes called the crossover error rate (CER), the EER is the point where both type i and type II errors are equal; see Figure 4. A smaller EER value implies lower FRR and FAR rates, thus a better performing verification system.
2. Zero-miss false alarm rate (ZM-FAR): This measure ensures that no attacker can gain access to the system by selecting a threshold value such that the miss rate of attackers is zero. ZM-FAR prioritises security by not accepting imposters at the expense of increasing FRR values. Note that the letter “A” in ZM-FAR stands for alarm and not acceptance.

The relationship between FAR and FRR is an inverse relationship. A system administrator considers the trade-off between user experience and the level of security. If the administrator wishes to increase security by minimising FAR, a rise in FRR occurs, which translates into a negative user experience because of frequent rejections by the system of legitimate users. On the other hand, lower FAR values translate into more accurate biometric systems with higher security profiles. The system administrator selects the threshold based on the required security measures, and a classifier that has lower EER and ZM-FAR values surpasses a classifier with higher values.

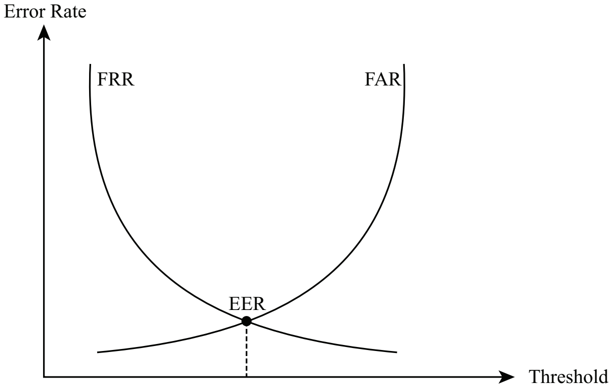


**Figure 4.** The relationship between EER, FAR, and FRR.

In addition, we also provide results for the following metrics, which are frequently reported in classification problems:

1. *Recall (R)* is the percentage of instances correctly classified out of the total number of instances that are actually true. It can be calculated as follows:

$$R = \frac{TP}{TP + FN} \tag{4}$$

2. *Precision (P)* is the percentage of instances correctly classified as true out of the total number of instances classified as true. Precision is calculated as follows:

$$P = \frac{TP}{TP + FP} \tag{5}$$

3. *Accuracy* is the percentage of records classified correctly and is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

### *4.2. Performance Evaluation*

In this section, we present the experimental results achieved when evaluating the dataset. In the first subsection, we present the password results for the four classifiers, and in the next subsection, we present the phrase results for the four classifiers. In general, when training a classifier, the dataset is split into parts to ensure proper training is performed, making the reported results accurate and meaningful. The first part is the training set, which is a part of the dataset that is used to fit the classification model. The model uses these data to learn the various parameters within the model. The validation set is another part of the data that evaluates the learned parameters the model was trained upon using the training set. The last set is the test set, which the model does not encounter at all during training; it is held back and presented to the model after training is completed to produce an unbiased assessment of the model.

The data were split into a training set and a testing set of sizes 75% and 25%, respectively. The classifiers were trained using 5-fold cross-validation on the training set, and results are reported for the testing sets.

#### 4.2.1. BKSD—Password

Tables 3–6 present the average results achieved by each of the four classifiers on the password dataset (average of the 50 users) by training and testing on Arabic, training and testing on English, training on Arabic and testing on English, and training on English and testing on Arabic, respectively.

**Table 3.** Average results achieved by each of the four classifiers on the password dataset when training and testing on Arabic. Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|
| AdaB | 0.043 | 0.406 | 0.951 | 0.963 | 0.957 | 0.037 | 0.049 |
| DT | 0.129 | 0.980 | 0.849 | 0.890 | 0.871 | 0.106 | 0.151 |
| RF | 0.067 | 0.516 | 0.936 | 0.931 | 0.933 | 0.071 | 0.064 |
| SVM | 0.101 | 0.689 | 0.884 | 0.913 | 0.899 | 0.085 | 0.116 |

**Table 4.** Average results achieved by each of the four classifiers on the password dataset when training and testing on English. Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|
| AdaB | 0.041 | 0.450 | 0.95 | 0.969 | 0.959 | 0.031 | 0.050 |
| DT | 0.114 | 0.933 | 0.876 | 0.898 | 0.886 | 0.104 | 0.124 |
| RF | 0.059 | 0.553 | 0.928 | 0.954 | 0.941 | 0.046 | 0.072 |
| SVM | 0.081 | 0.705 | 0.901 | 0.936 | 0.919 | 0.063 | 0.099 |

**Table 5.** Average results achieved by each of the four classifiers on the password dataset when training on Arabic and testing on English. Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|
| AdaB | 0.491 | 0.796 | 0.961 | 0.511 | 0.513 | 0.943 | 0.039 |
| DT | 0.480 | 0.979 | 0.859 | 0.523 | 0.523 | 0.818 | 0.141 |
| RF | 0.476 | 0.818 | 0.929 | 0.523 | 0.527 | 0.880 | 0.071 |
| SVM | 0.477 | 0.969 | 0.780 | 0.530 | 0.526 | 0.733 | 0.220 |

**Table 6.** Average results achieved by each of the four classifiers on the password dataset when training on English and testing on Arabic. Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|
| AdaB | 0.499 | 0.826 | 0.921 | 0.505 | 0.503 | 0.918 | 0.079 |
| DT | 0.493 | 0.999 | 0.826 | 0.508 | 0.507 | 0.813 | 0.174 |
| RF | 0.486 | 0.794 | 0.897 | 0.514 | 0.515 | 0.869 | 0.103 |
| SVM | 0.478 | 0.971 | 0.784 | 0.522 | 0.523 | 0.741 | 0.216 |

Recall that the password is exactly the same keystroke pattern in Arabic and in English. From Tables 3 and 4, when the classifiers are trained and tested on the same language, one can see that both AdaBoost and random forest performed similarly. Both algorithms achieved excellent EER, recall, precision, accuracy, FRR, and FAR values. AdaBoost was able to achieve better ZM-FAR values. Although AdaBoost and random forest outperformed decision tree and SVM, the latter still achieved excellent performance. The algorithms are all able to distinguish imposters well, as seen in the last column's (FAR) values.

When training and testing on different languages, as seen in Tables 5 and 6, all measures deteriorate. The classifier accuracies are close to 50% in these tables, showing that the classifiers' performance is close to random guessing. The accuracy improves marginally in Table 5. In general, all measures are somewhat better when training on Arabic and testing on English than when training on English and testing on Arabic; however, none are good. Using the appropriate threshold for EER; where FRR and FAR are equal, results in a system that would reject close to half the users while giving access to close to half the imposters in Tables 5 and 6. In fact, a look at the FRR column shows that the best classifier would deny access to almost 73% and 74% of legitimate users in Tables 5 and 6, respectively, whilst simultaneously granting access to 22% of imposters. What all these measures are indicating is that the classifiers are unable to distinguish users when they type in a different language, e.g., if a user enters their password in Arabic, then goes on to type in English, the system cannot recognise their new language keystroke dynamics pattern, although the pattern sequence on the keyboard is identical. This means that each user has their own distinct pattern of typing for each language, and that the primary factor affecting keystroke dynamics is language based, or possibly *n*-gram based. Any system that verifies the user in one language should have the user enrol in their other language.

#### 4.2.2. BKSD—Phrase

Tables 7–10 present the average results (of 35 users) achieved by each of the four classifiers on the phrase dataset by training and testing on Arabic, training and testing on English, training on Arabic and testing on English, and training on English and testing on Arabic, respectively.

**Table 7.** Average results achieved by each of the four classifiers on the phrase dataset when training and testing on Arabic. Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|
| AdaB | 0.014 | 0.196 | 0.983 | 0.989 | 0.986 | 0.011 | 0.017 |
| DT | 0.114 | 0.983 | 0.869 | 0.899 | 0.886 | 0.097 | 0.131 |
| RF | 0.051 | 0.454 | 0.948 | 0.950 | 0.949 | 0.051 | 0.052 |
| SVM | 0.106 | 0.676 | 0.886 | 0.901 | 0.894 | 0.097 | 0.114 |

**Table 8.** Average results achieved by each of the four classifiers on the phrase dataset when training and testing on English. Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|
| AdaB | 0.011 | 0.211 | 0.987 | 0.991 | 0.989 | 0.009 | 0.013 |
| DT | 0.096 | 0.948 | 0.897 | 0.912 | 0.904 | 0.090 | 0.103 |
| RF | 0.043 | 0.460 | 0.947 | 0.967 | 0.957 | 0.033 | 0.053 |
| SVM | 0.088 | 0.682 | 0.896 | 0.926 | 0.911 | 0.072 | 0.104 |

**Table 9.** Average results achieved by each of the four classifiers on the phrase dataset when training on Arabic and testing on English. Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|
| AdaB | 0.482 | 0.841 | 0.958 | 0.511 | 0.514 | 0.922 | 0.042 |
| DT | 0.472 | 0.986 | 0.867 | 0.523 | 0.528 | 0.811 | 0.133 |
| RF | 0.463 | 0.864 | 0.934 | 0.531 | 0.538 | 0.860 | 0.066 |
| SVM | 0.458 | 0.967 | 0.785 | 0.540 | 0.543 | 0.700 | 0.215 |

**Table 10.** Average results achieved by each of the four classifiers on the phrase dataset when training on English and testing on Arabic. Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|
| AdaB | 0.490 | 0.537 | 0.985 | 0.499 | 0.501 | 0.964 | 0.015 |
| DT | 0.488 | 0.988 | 0.848 | 0.507 | 0.510 | 0.824 | 0.152 |
| RF | 0.483 | 0.707 | 0.917 | 0.511 | 0.514 | 0.884 | 0.083 |
| SVM | 0.471 | 0.960 | 0.787 | 0.522 | 0.528 | 0.728 | 0.213 |

Tables 7 and 8 show similar results to the password dataset: when a classifier is trained and tested on the same language, it achieves excellent results and is able to distinguish imposters with minimal authentic user rejection. Again, AdaBoost and random forest outperform decision tree and SVM, although their performance is excellent as well.

Tables 9 and 10 also show that when classifiers are trained on a different language than the testing language, the classifier performance deteriorates, with accuracies close to random guessing. In addition, the performance of Arabic trained and English tested slightly exceeds that of English trained and Arabic tested. The phrase experiment further corroborates the findings from the password experiment: the user's typing pattern is different when the user switches languages. The high recall and low precision values in Tables 5, 6, 9 and 10 indicate that the classifiers identify most cases as attackers, rejecting many authentic users, and they are correct about half of the time. This corresponds with the high FRR and low FAR values: the classifiers reject authentic users whilst still identifying intruders. We remind our reader that the positions of letters used in both languages are all the same. Keystroke dynamics are not affected by merely the key positions on the keyboard, but by the language the user is typing in.

### 4.2.3. Bigram Keystroke Dynamics

In this section, we present the results of testing a segment of BKSD based on bigram frequencies. Table 2 shows the most frequent bigrams in both languages present in our dataset. The most frequent English bigram is "th" and the corresponding Arabic letters on the keyboard give the bigram "فا", which is the 88th most frequent bigram in Arabic. The third most frequent Arabic bigram is present in our dataset: "لا", and the corresponding English bigram "gh" does not appear in our dataset. We design a simple experiment where we compare the classifier performance on a sub-dataset composed of the most frequent bigram in English, "th", and the third most frequent bigram in Arabic, "لا", as well as the most frequent English bigram, "th", and its corresponding Arabic bigram, "فا", and report the results in this section. We do not select bigrams with the same key positions, in order to test the system's ability to verify users based on frequent bigrams. Tables 11–14 present the average results achieved by each of the four classifiers by training and testing on Arabic, training and testing on English, training on Arabic and testing on English, and training on English and testing on Arabic, respectively.

**Table 11.** Average results achieved by each of the four classifiers on the bigram subset when training and testing on Arabic. Results labelled "First" compare the frequent bigrams "لا" and "لا", and results labelled "Second" compare the bigrams "فا" and "فا". Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Exp | Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|---|
| First | AdaB | 0.052 | 0.469 | 0.937 | 0.955 | 0.947 | 0.042 | 0.063 |
| | DT | 0.107 | 0.941 | 0.872 | 0.906 | 0.893 | 0.087 | 0.128 |
| | RF | 0.062 | 0.272 | 0.933 | 0.94 | 0.938 | 0.058 | 0.067 |
| | SVM | 0.049 | 0.628 | 0.935 | 0.966 | 0.951 | 0.033 | 0.065 |
| Second | AdaB | 0.049 | 0.461 | 0.945 | 0.965 | 0.951 | 0.042 | 0.055 |
| | DT | 0.094 | 0.997 | 0.882 | 0.939 | 0.904 | 0.070 | 0.118 |
| | RF | 0.047 | 0.223 | 0.948 | 0.965 | 0.953 | 0.041 | 0.052 |
| | SVM | 0.043 | 0.548 | 0.950 | 0.970 | 0.957 | 0.036 | 0.050 |

**Table 12.** Average results achieved by each of the four classifiers on the bigram subset when training and testing on English. Results compare the frequent bigrams "th" and "th". Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|
| AdaB | 0.050 | 0.449 | 0.940 | 0.958 | 0.951 | 0.040 | 0.060 |
| DT | 0.079 | 0.909 | 0.905 | 0.933 | 0.921 | 0.063 | 0.095 |
| RF | 0.050 | 0.252 | 0.941 | 0.958 | 0.950 | 0.041 | 0.059 |
| SVM | 0.044 | 0.599 | 0.951 | 0.960 | 0.956 | 0.039 | 0.049 |

**Table 13.** Average results achieved by each of the four classifiers on the bigram subset when training on Arabic and testing on English. Results labelled “freq” compare the frequent bigrams “th” and “لا”, and results labelled “same” compare the same bigrams “th” and “فا”. Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Exp | Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|---|
| Freq | AdaB | 0.516 | 0.966 | 0.893 | 0.488 | 0.480 | 0.926 | 0.107 |
| | DT | 0.505 | 1.000 | 0.823 | 0.503 | 0.492 | 0.833 | 0.177 |
| | RF | 0.501 | 0.911 | 0.906 | 0.504 | 0.496 | 0.908 | 0.094 |
| | SVM | 0.505 | 0.957 | 0.864 | 0.496 | 0.493 | 0.873 | 0.136 |
| Same | AdaB | 0.510 | 0.930 | 0.915 | 0.491 | 0.487 | 0.934 | 0.085 |
| | DT | 0.509 | 0.971 | 0.869 | 0.494 | 0.488 | 0.888 | 0.131 |
| | RF | 0.497 | 0.858 | 0.944 | 0.500 | 0.499 | 0.939 | 0.056 |
| | SVM | 0.500 | 0.917 | 0.917 | 0.500 | 0.497 | 0.917 | 0.083 |

**Table 14.** Average results achieved by each of the four classifiers on the bigram subset when training on English and testing on Arabic. Results labelled “freq” compare the frequent bigrams “th” and “لا”, and results labelled “same” compare the same bigrams “th” and “فا”. Cls: classifier; R: recall; P: precision; Acc: accuracy; AdaB: AdaBoost.

| Exp | Cls | EER | ZM-FAR | R | P | Acc | FRR | FAR |
|---|---|---|---|---|---|---|---|---|
| Freq | AdaB | 0.506 | 0.743 | 0.869 | 0.497 | 0.488 | 0.882 | 0.131 |
| | DT | 0.494 | 0.914 | 0.825 | 0.502 | 0.501 | 0.812 | 0.175 |
| | RF | 0.504 | 0.731 | 0.886 | 0.499 | 0.489 | 0.894 | 0.114 |
| | SVM | 0.488 | 0.946 | 0.863 | 0.510 | 0.506 | 0.840 | 0.137 |
| Same | AdaB | 0.510 | 0.909 | 0.862 | 0.551 | 0.527 | 0.882 | 0.138 |
| | DT | 0.505 | 1.000 | 0.800 | 0.550 | 0.525 | 0.810 | 0.200 |
| | RF | 0.512 | 0.766 | 0.870 | 0.550 | 0.526 | 0.894 | 0.130 |
| | SVM | 0.481 | 0.949 | 0.875 | 0.569 | 0.554 | 0.838 | 0.125 |

As expected, due to the restricted number of features used in training, the classifiers’ accuracy deteriorates slightly when training and testing on the same language due to underfitting. In addition, the classifiers continue to exhibit an inability to differentiate users based on them typing a very frequent bigram in English with a very frequent bigram in Arabic, as well as when comparing the frequent English bigram and its corresponding (positional) Arabic bigram.

The study by Wahab et al. [7] corroborates our findings, with an average EER of 0.210% when enrolling in English and testing on Chinese, and 0.253% when enrolling in Chinese and testing on English, while on our dataset, an average EER of 0.486% was achieved when enrolling in English and testing on Arabic, and 0.475% when enrolling in Arabic and testing on English. A direct comparison of results between the two studies is not beneficial. The two studies differ in languages used, the dataset itself and the dataset collection approach (static vs. free-text), the familiarity of the user with the keyboard, as well as the algorithms used to assess the dataset. The deterioration of EER in our dataset might be caused by users having to use an unfamiliar keyboard compared with users using their own keyboard in [7]. Allowing users to type on their own keyboards might have inflated the EER results. Conversely, the discrepancy may be attributed to language-specific issues or simply to the differences in the algorithms used. More study is needed to conclusively answer this question.

In summary, the results show that users typing in English and Arabic have different typing patterns for each language, and each person has their own distinct typing rhythms that are language dependent. When a classifier is trained on one language, the user’s own typing pattern or keystroke dynamics in another language is rejected as an attacker. This signifies that other factors affect a person’s keystroke dynamics, factors that are likely language-specific. This holds for the Chinese [7] and Arabic languages, where both languages are significantly different from the English language.

## 5. Conclusions

This study presented the Bilingual Keystroke Dynamics Dataset (BKSD), a dataset designed to test whether keystroke dynamics are affected by language. BKSD is divided into two components: password and phrase, with a total of 50 and 35 users, respectively. Each user has approximately 400 attempts in English, and 400 attempts in Arabic.

The results show that using different languages (Arabic and English) prevents authentication systems from accurately identifying users, despite the letter patterns typed being identical, i.e., the finger movements of the subject were the same. EER values deteriorated to an average of 0.486% when enrolling in English and testing on Arabic, and 0.475% when enrolling in Arabic and testing on English. Previous work reported similar results for the Chinese and English languages. This work suffers from a number of limitations. As keystroke dynamics is used to authenticate users, authentication becomes susceptible to emotions or tiredness, and other human factors, as these factors all affect keystroke dynamics. In addition, using keystroke dynamics in an authentication system places an extra energy load on those systems. Finally, a keystroke dynamics authentication system's performance may differ when hardware is changed.

For future work, a number of interesting directions present themselves. We plan to study the effect of frequent $n$-grams on keystroke dynamics in depth. The location of frequent $n$-grams on the keyboard might also come into play, where a cluster of letters may have a faster typing time. Another direction is the effect of languages and language families on keystroke dynamics, which necessitates the collection of data in the same and different language families. Finally, what effect does the letter layout on keyboards have on keyboard dynamics? In Arabic, for example, there are two major keyboard layouts that require further exploration.

**Funding:** This work has been funded by King Saud University, Riyadh, Saudi Arabia, through Researchers Supporting Project number (RSPD2023R857).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The Bilingual Keystroke Dynamics Dataset (BKSD) is freely available at https://github.com/ntwaijry/BKSD (accessed on 17 October 2023).

**Acknowledgments:** Special thanks to Raghad Alarwan, Yara Alzahrani, Maryam Alajlan, Leenah Alfulaij, and Lubna Aldibasi for overseeing the collection of the raw keystroke data from participants, and to Najd Altwaijry for the fruitful discussions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| BKSD | Bilingual Keystroke Dynamics Dataset |
| AdaB | AdaBoost |
| DT | Decision tree |
| RF | Random forest |
| FRR | False rejection rate |
| FAR | False acceptance rate |
| EER | Equal error rate |
| ZM-FAR | Zero-miss false alarm rate |

## References

1. Bleha, S.; Slivinsky, C.; Hussien, B. Computer-access security systems using keystroke dynamics. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 1217–1222. [CrossRef]
2. Banerjee, S.P.; Woodard, D.L. Biometric authentication and identification using keystroke dynamics: A survey. *J. Pattern Recognit. Res.* **2012**, *7*, 116–139. [CrossRef] [PubMed]
3. Epp, C.; Lippold, M.; Mandryk, R.L. Identifying emotional states using keystroke dynamics. In Proceedings of the Sigchi Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, 7–12 May 2011; pp. 715–724.
4. Roy, S.; Pradhan, J.; Kumar, A.; Adhikary, D.R.D.; Roy, U.; Sinha, D.; Pal, R.K. A systematic literature review on latest keystroke dynamics based models. *IEEE Access* **2022**, *10*, 92192–92236. [CrossRef]
5. Teh, P.S.; Teoh, A.B.J.; Yue, S. A survey of keystroke dynamics biometrics. *Sci. World J.* **2013**, *2013*, 408280. [CrossRef] [PubMed]
6. Multilingual People. Available online: https://ilanguages.org/bilingual.php (accessed on 12 March 2023).
7. Wahab, A.A.; Hou, D.; Banavar, M.; Schuckers, S.; Eaton, K.; Baldwin, J.; Wright, R. Shared multi-keyboard and bilingual datasets to support keystroke dynamics research. In Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy, Baltimore, MD, USA, 25–27 April 2022; pp. 236–241.
8. Ayotte, B.; Banavar, M.; Hou, D.; Schuckers, S. Fast free-text authentication via instance-based keystroke dynamics. *IEEE Trans. Biom. Behav. Identity Sci.* **2020**, *2*, 377–387. [CrossRef]
9. Baldwin, J.; Burnham, R.; Meyer, A.; Dora, R.; Wright, R. Beyond speech: Generalizing d-vectors for biometric verification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 842–849.
10. Zhong, Y.; Deng, Y. A survey on keystroke dynamics biometrics: Approaches, advances, and evaluations. *Recent Adv. User Authentication Using Keystroke Dyn. Biom.* **2015**, *2*, 1–22.
11. Botha, R.A.; Furnell, S.M.; Clarke, N.L. From desktop to mobile: Examining the security experience. *Comput. Secur.* **2009**, *28*, 130–137. [CrossRef]
12. Campisi, P.; Maiorana, E.; Bosco, M.L.; Neri, A. User authentication using keystroke dynamics for cellular phones. *IET Signal Process.* **2009**, *3*, 333–341. [CrossRef]
13. Kambourakis, G.; Damopoulos, D.; Papamartzivanos, D.; Pavlidakis, E. Introducing touchstroke: Keystroke-based authentication system for smartphones. *Secur. Commun. Netw.* **2016**, *9*, 542–554. [CrossRef]
14. Anusas-Amornkul, T. Strengthening password authentication using keystroke dynamics and smartphone sensors. In Proceedings of the 9th International Conference on Information Communication and Management, Prague, Czech Republic, 23–26 August 2019; pp. 70–74.
15. Oyebola, O. Examining the Distribution of Keystroke Dynamics Features on Computer, Tablet and Mobile Phone Platforms. In *Mobile Computing and Sustainable Informatics, Proceedings of the ICMCSI 2023, Lalitpur, Nepal, 11–12 January 2023*; Springer: Singapore, 2023; pp. 613–620.
16. Alsuhibany, S.A.; Almuqbil, A.S. Impact of using different-sized touch keyboards on free-text keystroke dynamics authentication in the Arabic language. *Sci. Rep.* **2022**, *12*, 15866. [CrossRef]
17. Killourhy, K.; Maxion, R. The effect of clock resolution on keystroke dynamics. In Proceedings of the Recent Advances in Intrusion Detection: 11th International Symposium, RAID 2008, Cambridge, MA, USA, 15–17 September 2008; Proceedings 11; Springer: Berlin/Heidelberg, Germany, 2008; pp. 331–350.
18. Shekhawat, K.; Bhatt, D.P. A novel approach for user authentication using keystroke dynamics. *J. Discret. Math. Sci. Cryptogr.* **2022**, *25*, 2015–2027. [CrossRef]
19. Ning, E.; Cladek, A.T.; Ross, M.K.; Kabir, S.; Barve, A.; Kennelly, E.; Hussain, F.; Duffecy, J.; Langenecker, S.L.; Nguyen, T.; et al. Smartphone-derived Virtual Keyboard Dynamics Coupled with Accelerometer Data as a Window into Understanding Brain Health: Smartphone Keyboard and Accelerometer as Window into Brain Health. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, Hamburg, Germany, 23–28 April 2023; pp. 1–15.
20. Senerath, D.; Tharinda, S.; Vishwajith, M.; Rasnayaka, S.; Wickramanayake, S.; Meedeniya, D. BehaveFormer: A Framework with Spatio-Temporal Dual Attention Transformers for IMU enhanced Keystroke Dynamics. In Proceedings of the 7th IEEE International Joint Conference on Biometrics (IJCB 2023), Ljubljana, Slovenia, 25–28 September 2023.
21. Tsimperidis, I.; Arampatzis, A. User Profiling Using Keystroke Dynamics and Rotation Forest. In *Applications of Machine Learning and Deep Learning for Privacy and Cybersecurity*; IGI Global: Hershey, PA, USA, 2022; pp. 1–24.
22. Tsimperidis, I.; Grunova, D.; Roy, S.; Moussiades, L. Keystroke Dynamics as a Language Profiling Tool: Identifying Mother Tongue of Unknown Internet Users. *Telecom* **2023**, *4*, 369–377. [CrossRef]
23. Sağbaş, E.A.; Korukoglu, S.; Balli, S. Stress detection via keyboard typing behaviors by using smartphone sensors and machine learning techniques. *J. Med. Syst.* **2020**, *44*, 68. [CrossRef] [PubMed]
24. Yang, L.; Qin, S.F. A review of emotion recognition methods from keystroke, mouse, and touchscreen dynamics. *IEEE Access* **2021**, *9*, 162197–162213. [CrossRef]
25. Cascone, L.; Nappi, M.; Narducci, F.; Pero, C. Touch keystroke dynamics for demographic classification. *Pattern Recognit. Lett.* **2022**, *158*, 63–70. [CrossRef]
26. Lamiche, I.; Bin, G.; Jing, Y.; Yu, Z.; Hadid, A. A continuous smartphone authentication method based on gait patterns and keystroke dynamics. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 4417–4430. [CrossRef]

27. Alfalahi, H.; Khandoker, A.H.; Chowdhury, N.; Iakovakis, D.; Dias, S.B.; Chaudhuri, K.R.; Hadjileontiadis, L.J. Diagnostic accuracy of keystroke dynamics as digital biomarkers for fine motor decline in neuropsychiatric disorders: A systematic review and meta-analysis. *Sci. Rep.* **2022**, *12*, 7690. [CrossRef]
28. Roy, S.; Roy, U.; Sinha, D.; Pal, R.K. Imbalanced ensemble learning in determining Parkinson's disease using Keystroke dynamics. *Expert Syst. Appl.* **2023**, *217*, 119522. [CrossRef]
29. Acien, A.; Morales, A.; Vera-Rodriguez, R.; Fierrez, J.; Mondesire-Crump, I.; Arroyo-Gallego, T. Detection of Mental Fatigue in the General Population: Feasibility Study of Keystroke Dynamics as a Real-world Biomarker. *JMIR Biomed. Eng.* **2022**, *7*, e41003. [CrossRef]
30. Roy, S.; Sinha, D.; Roy, U. Identifying age group and gender based on activities on touchscreen. *Int. J. Biom.* **2022**, *14*, 61–82. [CrossRef]
31. Chen, Z.; Cai, H.; Jiang, L.; Zou, W.; Zhu, W.; Fei, X. Keystroke dynamics based user authentication and its application in online examination. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; pp. 649–654.
32. Tsimperidis, I.; Yoo, P.D.; Taha, K.; Mylonas, A.; Katos, V. R2BN: An Adaptive Model for Keystroke-Dynamics-Based Educational Level Classification. *IEEE Trans. Cybern.* **2020**, *50*, 525–535. [CrossRef]
33. El-Kenawy, E.S.M.; Mirjalili, S.; Abdelhamid, A.A.; Ibrahim, A.; Khodadadi, N.; Eid, M.M. Meta-Heuristic Optimization and Keystroke Dynamics for Authentication of Smartphone Users. *Mathematics* **2022**, *10*, 2912. . [CrossRef]
34. Stylios, I.; Chatzis, S.; Thanou, O.; Kokolakis, S. Continuous Authentication with Feature-Level Fusion of Touch Gestures and Keystroke Dynamics to Solve Security and Usability Issues. *Comput. Secur.* **2023**, *132*, 103363. [CrossRef]
35. Wang, X.; Shi, Y.; Zheng, K.; Zhang, Y.; Hong, W.; Cao, S. User Authentication Method Based on Keystroke Dynamics and Mouse Dynamics with Scene-Irrelated Features in Hybrid Scenes. *Sensors* **2022**, *22*, 6627. [CrossRef] [PubMed]
36. Hatin, J.; Cherrier, E.; Schwartzmann, J.J.; Rosenberger, C. Privacy preserving transparent mobile authentication. In *International Conference on Information Systems Security and Privacy*; SCITEPRESS: Setúbal, Portugal, 2017; Volume 2, pp. 354–361.
37. Acar, A.; Liu, W.; Beyah, R.; Akkaya, K.; Uluagac, A.S. A privacy-preserving multifactor authentication system. *Secur. Priv.* **2019**, *2*, e88. [CrossRef]
38. Kanter, J.M.; Veeramachaneni, K. Deep feature synthesis: Towards automating data science endeavors. In Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, France, 19–21 October 2015; pp. 1–10. [CrossRef]
39. Bunkhumpornpat, C.; Sinapiromsaran, K.; Lursinsap, C. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Advances in Knowledge Discovery and Data Mining, Proceedings of the 13th Pacific-Asia Conference, PAKDD 2009, Bangkok, Thailand, 27–30 April 2009*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 475–482.
40. Huang, M.H.; Rust, R.T. A strategic framework for artificial intelligence in marketing. *J. Acad. Mark. Sci.* **2021**, *49*, 30–50. [CrossRef]
41. Varsha, P. How can we manage biases in artificial intelligence systems—A systematic literature review. *Int. J. Inf. Manag. Data Insights* **2023**, *3*, 100165. .: 10.1016/j.jjimei.2023.100165. [CrossRef]
42. Lambrecht, A.; Tucker, C. Algorithmic bias? An empirical study of apparent gender-based discrimination in the display of STEM career ads. *Manag. Sci.* **2019**, *65*, 2966–2981. [CrossRef]
43. Murphy, L.W. Airbnb's work to fight discrimination and build inclusion. *Rep. Submitt. Airbnb* **2016**, *8*, 2016.
44. McStay, A. *Emotional AI: The Rise of Empathic Media*; Sage: Thousand Oaks, CA, USA, 2018. [CrossRef]
45. Mikalef, P.; Conboy, K.; Lundström, J.E.; Popovič, A. Thinking responsibly about responsible AI and 'the dark side'of AI. *Eur. J. Inf. Syst.* **2022**, 31, *257–268.* [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.