electronics

MDPI

Article

Age and Gender as Cyber Attribution Features in Keystroke Dynamic-Based User Classification Processes

Ioannis Tsimperidis [1,*], Cagatay Yucel [2] and Vasilios Katos [2]

[1] Department of Electrical and Computer Engineering, School of Engineering, Democritus University of Thrace, 67100 Xanthi, Greece
[2] Department of Computing and Informatics, Bournemouth University, Poole BH12 5BB, UK; cyucel@bournemouth.ac.uk (C.Y.); vkatos@bournemouth.ac.uk (V.K.)
* Correspondence: itsimper@ee.duth.gr

Abstract: Keystroke dynamics are used to authenticate users, to reveal some of their inherent or acquired characteristics and to assess their mental and physical states. The most common features utilized are the time intervals that the keys remain pressed and the time intervals that are required to use two consecutive keys. This paper examines which of these features are the most important and how utilization of these features can lead to better classification results. To achieve this, an existing dataset consisting of 387 logfiles is used, five classifiers are exploited and users are classified by gender and age. The results, while demonstrating the application of these two characteristics jointly on classifiers with high accuracy, answer the question of which keystroke dynamics features are more appropriate for classification with common classifiers.

Keywords: keystroke dynamics; data mining; user classification; feature selection; feature comparison

Citation: Tsimperidis, I.; Yucel, C.; Katos, V. Age and Gender as Cyber Attribution Features in Keystroke Dynamic-Based User Classification Processes. *Electronics* **2021**, *10*, 835. https://doi.org/10.3390/electronics10070835

Academic Editor: Jemal H. Abawajy

Received: 20 February 2021
Accepted: 26 March 2021
Published: 31 March 2021

1. Introduction

Recognizing certain characteristics of a user is important for a number of processes, such as improving the performance of authentication systems. Various techniques are proposed in the literature [1] for how this can be achieved and are mainly divided into two categories, physiological biometrics and behavioral biometrics. The first are associated with the shape or measurements of the human body, while the second are associated with the behavior of an individual. The former include the measurement and analysis of fingerprints, iris, palm geometry, etc., while the latter includes the measurement and analysis of handwriting, gait, voice, etc.

Keystroke dynamics is a behavioral biometric which exploits data derived from the way users use the keyboard [2], physical or virtual. Its main advantage over other biometric methods is that it does not require any specialized equipment, while its main disadvantage, like all behavioral biometrics, is that a user can change the way he/she types. Research into the keystroke dynamics, which began in the 1970s, has resulted in the implementation of systems that perform user authentication, recognize some inherit and/or acquired user characteristics and recognize mental and/or physical states of the users.

The term "the way a user uses the keyboard" means, among other things, the typing speed, the duration the keys remain pressed, the duration needed for using a series of specific keys, the number and frequency of pauses during typing, the number of typing errors and the way they are corrected, the frequency of use of specific keys, the time of day the typing is performed and the applications in which the typing is performed.

The keystroke dynamics features that have been used in research so far and those that could be used total up to number in the order of millions, but each one of them encloses a small amount of information. This creates a pleasant headache for the researcher, who is asked to choose from a huge number of features, so that his system has the highest possible accuracy in the shortest possible operating time.

This paper attempts to answer two questions. First, which of the most commonly used types of keystroke dynamics features work best. To the best of our knowledge, this analysis has not been done for this field. Second, if the simultaneous and combined characteristics of users are presenting a more appropriate technique than utilization of these characteristics individually. Specifically, in the problem of simultaneously finding of the gender and age of an unknown user, with the help of five machine learning models, the most widely used keystroke dynamics features are used and the system performance is compared for different feature sets.

The rest of the paper is structured as follows: First, a review of the literature that frames the topic of this work is made. The methodology followed is then explained and analyzed. In the Section 4, the system performance results for each feature set and for each of the five machine learning models are presented. Finally, the work is concluded and possible future-work directions of this research are presented.

2. Theoretical Background

Although the user authentication with passwords is an easy to implement security mechanism it has many vulnerabilities such as shoulder surfing, keyloggers, phishing and brute force [3]. For this reason, many alternatives to user authentication have been proposed, including the use of keystroke dynamics. This technology counts almost five decades of research [4] and has now matured, showing authentication systems with very low equal error rate (EER).

Recognizing the aforementioned problems, Lin et al. [5] proposed an authentication system that takes advantage of keystroke dynamics features, including the duration when a key is pressed (keystroke duration (KD)) and the time in between the release of one key and the pressure of the next (up-down diagram latency (UDDL)), in order to detect an unlawful user, even when he/she knew the genuine password for an account. After the data are obtained and the necessary features are extracted, a convolutional neural network is used, performing with the accuracy of 99% for the detection of the legit users.

In another study, Venugopal and Viji [6] used keystroke durations and the four different ways in which diagram latencies can be calculated for their authentication system. This is defined as the time in between pressing one key and pressing the next (down-down diagram latency, DDDL) and, respectively, down-up diagram latency (DUDL), up-up diagram latency (UUDL) and UDDL. The researchers collected data from the passwords entered by the volunteers and developed their system in MATLAB. They managed to achieve an EER of 0.5%.

For a similar purpose, Young et al. [7] in their work attempt to verify the identity of users participating in online courses. To achieve their goal, they collected data from 78 volunteers during typing by copying two texts and answering two questions. They used keystroke durations and up-down diagram latencies as features and with the help of them they extracted the "keyprint signature" and "keyprint profile" for each volunteer. After conducting the experiments, they showed that there is no absolute consistency in the way a user types and, therefore, in order for the authentication to be more successful, a lot of data is required.

Moreover, Sun et al. [8] focus on and present results for novel keystroke dynamics features such as, the keystroke durations of Space, Backspace, arrow keys and the down-down diagram latencies of Shift-"I", Shift-"N", period–Space, comma–Space. They used data from the typing recording of 34 users and after studying the behavior of each of the feature that had isolated, as well as sets of them, with the help of an SVM in user identification, they came up with a system that uses only 13 features and presents an EER of 2.94% and an AUC of 0.994.

In addition to user authentication, which has piqued interest of most researchers, keystroke dynamics also offer solutions to user classification in relation to some of their inherent or acquired characteristics, such as gender, age, handedness, educational level, etc. There are several reasons for this kind of classification and this is because text is the

dominant medium through which the billions of Internet users today communicate [9]. So, for a complete unknown user, who types to communicate with other users, to post announcements and to perform Internet searches, it is possible to identify some of his/her characteristics. In return, results are transferred to, firstly, the optimization of the targeted advertisements so that the user is not to be overwhelmed by messages that do not interest him/her, secondly, the facilitation of the use of Internet services, for example by automatically filling in fields on forms and by making suggestions for participation in discussion groups and visiting websites related to user characteristics and, thirdly, the collection of information on the identity of malicious users, which will help a forensic investigation.

In their work Buker et al. [10] recognize the importance of keystroke dynamics with citing the projected increase in investment in live chat applications, as well as the increase of their usage. Resulting in the consequence that collecting the information of how people type plays a vital social and economic role. In an attempt to find out the gender of a user, they collected data from 60 users, who were communicating in pairs within a live chat application, on a specific topic of discussion. They used only 15 features, including the frequency of use of Backspace, the frequency of use of the question mark and the frequency of use of non-alphabetic characters. They implemented random forest for classification performing with an accuracy of over 95%. Among their findings, an interesting result is that male users have a greater tendency to correct typos and misspellings.

In another study, Bandeira et al. [11] researched on the gender of an individual from data derived by handwriting and by recording of typing. In their endeavor involved 100 volunteers and regarding keystroke dynamics, the data are collected from four tasks completed by the participants that included copying text and typing at will. These two ways of data acquisition in keystroke dynamics are called fixed-text and free-text, respectively. The researchers took advantage of 29 features and used four classifiers. The best results came from SVM with 64% accuracy.

Research on the data from different sources, namely keystroke dynamics, touchscreen dynamics and handwritten signature data are conducted in the work of Da Costa-Abreu and Goncalves [12]. The researchers used data from the recordings of 76 volunteers when typing a specific text. From the data, they extracted from the up-down diagram latencies of 14 diagrams and the keystroke durations of the characters that make them up. Finally, they were able to determine the gender of the user, with the help of an SVM, with an accuracy of 83.5%.

In the field of user age search, Tsimperidis et al. [13] used a dataset consisting of 387 logfiles and extracted 700 features from it, which were keystroke durations and down-down diagram latencies. Utilizing five classifiers, experiments are conducted with different sets of features. The experiments resulted in creation of a system that can distinguish with an accuracy of about 90% the age group of an unknown user, among four options.

In a different field, Tzafilkou and Protogeros [14] tried to relate mouse movements and keystroke dynamics to states that software developers may be found, namely self-efficacy, risk-perception, willingness to learn, perceived usefulness and perceived easy to use. They developed a software for recording the mouse movements and keyboard usage, which also includes a questionnaire to assess the user's status. They collected data from 30 participants and the keystroke dynamics features they extracted were typing speed, keystroke durations, UDDL and DDDL. Their experiments show a correlation between the typing mode and some of the states that are defined for a software developer.

In another paper, Ulinskas et al. [15] used an existing keystroke dynamics dataset that came from recordings from 53 people typing the same password, in order to recognize user fatigue. From the data, they extracted as features the keystroke durations and all the different forms of diagram latencies. They used six classifiers and found that the best results came from the up-up diagram latencies, which made it possible to recognize fatigue at 91%.

A more concise picture of the literature is presented in Table 1, where in the field "Scope" with UA is marked the user authentication, with UV the user verification, with GC the gender classification, with AC the age classification and with SA the state assessment.

Table 1. Summary of related works. (UA—User Authentication; UV—User Verification; GC—Gender Classification; AC—Age Classification; SA—State Assessment; CNN—Convolutional Neural Network; SVM—Support Vector Machine; RF—Random Forest; RBFN—Radial Basis Function Network; k-NN—k Nearest Neighbor; KD—Keystroke Durations; DDDL—down-down diagram latencies; DUDL—down-up; UDDL—up-down; UUDL—up-up; EER—Equal Error Rate).

Work	Scope	Tool	Features	Results
[5]	UA	CNN	KD, UDDL	Accuracy of 99%
[6]	UA	MATLAB	KD, DDDL, DUDL, UDDL, UUDL	EER of 0.5%
[7]	UV	T-test	KD, UDDL	EER of 5%
[8]	UA	SVM	KD, DDDL	EER of 2.94%
[10]	GC	RF	Keys usage frequency	Accuracy of 95%
[11]	GC	SVM	KD, DDDL, DUDL, UDDL, UUDL	Accuracy of 64%
[12]	GC	SVM	KD, UDDL	Accuracy of 83.5%
[13]	AC	RBFN	KD, DDDL	Accuracy of 89.7%
[14]	SA	Pearson correlation	Typing speed, KD, UDDL, DDDL	Correlation of −0.37
[15]	SA	k-NN	KD, DDDL, DUDL, UDDL, UUDL	Accuracy of 91%

Two conclusions drawn from the literature review of keystroke dynamics: (i) it has been observed in the literature that most studies use as key features the keystroke durations and one or more forms of diagram latencies; (ii) there is a diversity and inconsistency in the terminology for these features. Thus, the keystroke duration is found as dwell time, or hold time, or press hold, or key press time. Down-down diagram latency is found as flight time, or press-to-press, or press latency. Up-down diagram latency occurs as interval, or flight time, or release-to-press, or latency. Down-up diagram latency is known as latency, or press-to-release. In addition, down-down diagram latency is met as flight time, or up-to-up, or release-to-release, or release latency.

3. Methodology

In the present study, the dataset created for [16] is used. This work describes the keylogger used, the process of recording volunteers during their daily computer use, the size and format of logfiles, the number of participants and logfiles and the process of extracting keystroke durations and down-down diagram latencies used as features to create four systems, each of which recognizes the gender, age group, handedness and educational level of the users with high accuracy.

In contrast, in the present work, a system is designed for the simultaneous identification of the gender and age group of the user. In the available dataset, which consists of 387 logfiles, users are divided into two genders and four age groups. Therefore, eight classes are created. The number of logfiles with respect to these age and gender groups are presented in Table 2.

Table 2. Number of logfiles per class.

Class	Number of Logfiles	Percentage in Dataset
Female 18–25	23	5.9
Female 26–35	86	22.2
Female 36–45	55	14.2
Female 46+	20	5.2
Male 18–25	73	18.9
Male 26–35	43	11.1
Male 36–45	62	16.0
Male 46+	25	6.5

As shown in Table 2, the dataset is not balanced, but each class is adequately represented.

The extraction of keystroke durations and down-down diagram latencies led to a feature set with over one hundred thousand features. In order to select the appropriate features that include the most information in the separation of users according to their gender and age, a procedure was followed, which is described in detail in [17], in which the information gain (*IG*) for each feature is calculated. In a brief description, the *IG* of a feature f is calculated from the reducing of the entropy that causes to a system x, as follows:

$$IG(x, f) = H(x) - H(x|f) \quad (1)$$

The entropy $H(x)$ of system x is calculated as:

$$H(x) = -\sum_{i=1}^{m} P(x_i) ln P(x_i) \quad (2)$$

with m being the number of classes and $P(x_i)$ being the probability of class x_i.

The term $H(x|f)$ is calculated by splitting the dataset into groups according to the value of the particular feature f. Then, the entropy of each group is calculated and $H(x|f)$ is given by:

$$H(x|f) = \frac{1}{N} \sum_{j=1}^{k} n_j H(x_j) \quad (3)$$

with N being the number of instances of the initial dataset, k being the number of groups that the initial dataset was split, n_j being the number of instances of the j-th group and $H(x_j)$ being the entropy of the j-th group, which can be calculated from Equation (2).

The results showed that there are 811 features with non-zero *IG* and the first 30 are presented in Table 3, where features with one number, corresponding to the virtual key code (VKC) of the key used, are the keystroke durations and features with two numbers, corresponding to the VKCs of the diagram used, are the down-down diagram latencies.

Table 3. Keystroke dynamics features with the highest information gain (*IG*) in gender–age classification.

#	Feat.	Keys	*IG*	#	Feat.	Keys	*IG*	#	Feat.	Keys	*IG*
1	69	E	0.2045	11	65–32	A-(space)	0.1113	21	66	B	0.0975
2	87	W	0.1546	12	89	Y	0.1099	22	39	(right arrow)	0.0885
3	84	T	0.1460	13	32	(space)	0.1095	23	65–82	A-R	0.0871
4	72	H	0.1379	14	70	F	0.1063	24	73–83	I-S	0.0856
5	73	I	0.1256	15	86	V	0.1057	25	75–65	K-A	0.0770
6	79	O	0.1232	16	73–78	I-N	0.1050	26	71–73	G-I	0.0766
7	82	R	0.1184	17	71	G	0.1040	27	77–79	M-O	0.0750
8	68	D	0.1150	18	84–79	T-O	0.0994	28	37	(left arrow)	0.0741
9	83	S	0.1143	19	76	L	0.0994	29	79–78	O-N	0.0731
10	65	A	0.1138	20	87–32	W-(space)	0.0993	30	77–186	M-;:	0.0714

An important observation made in Table 3 is that keystroke durations seem to play a more important role in classifying users according to their gender and age. Table 4 shows the number and percentage of KD and DDDL per 100 features.

Table 4. Number and percentage of KD and DDDL in features with the highest *IG*.

# of Features	# of KD	% of KD	# of DDDL	% of DDDL
100	32	32.0	68	68.0
200	36	18.0	164	82.0
300	39	13.0	261	87.0
400	40	10.0	360	90.0
500	40	8.0	460	92.0
600	41	6.8	559	93.2
700	41	5.9	659	94.1
800	42	5.3	758	94.7

Considering that there are around 100 KD and 100,000 DDDL, the striking thing is that 0.1% of the available features represent 32% in the top 100 with the highest *IG*. One explanation that can be given for this phenomenon is that in the data recorded from the volunteers the KD of a key is found many more times than the DDDL of a diagram. However, in the present study the value of the feature is taken as the average value and therefore the number of appearances of the feature does not play a direct role. Therefore, if there is no hidden correlation between the number of appearances of each feature in the raw data and the enclosed information, an explanation should be sought for how important they are in keystroke dynamics studies, or at least in user classification according to their gender and age.

The ability to simultaneously find of the gender and age of an unknown user, using the features highlighted by the feature selection process, as well as all available features, all available KDs and all available DDDLs, is tested in this research using five known machine learning models, namely, support vector machine (SVM), simple logistic (SL), naïve Bayes (NB), Bayesian network classifier (BNC), and radial basis function network (RBFN). The selection of these classifiers was made with two criteria. Firstly, to make a direct comparison with the results presented in the work [16] and secondly because these models showed the best performance over many others tested, such as multilayer perceptron (MLP), random forest (RF), naïve Bayes tree (NBtree), etc., which showed low accuracy and/or very high training time.

Experiments were performed for each of the classifiers in order to find the best performance, using as feature sets the 100 features with the highest *IG*, the 200 features with the highest *IG*, etc., up to 800 features with the highest *IG*, which is the hundred closest to the number of features with non-zero *IG*. In addition, for the best performing classifiers, all available features, all available KDs and all available DDDLs were used as additional feature sets, so that a direct comparison can be made and conclusions can be drawn as to which type of features is most suitable for solving the problem of user classification.

The comparison of the performance between the classifiers and the different feature sets was done with the criterion of accuracy (Acc.) and the required training time of the system (time to build model (TBM)). In addition, the F-score (F1), which is the harmonic means between precision and recall [18] and is a safe measurement even for unbalanced datasets and the area under the ROC curve (AUC), which is the area below the receiver operating characteristic curve [19] were used as comparison measures.

Each experiment was performed with the 10-fold cross-validation method [20] in order to obtain a more unbiased picture of its statistical measures, since it is performed ten times with a different training set and testing set each time and their average value is calculated. This avoids the possibility of calculating an outlier as accuracy, F-score and AUC.

In our case, where the dataset consists of 387 logfiles, each fold consists of 38 or 39 files. Thus, the 348 or 349 files are used to train the model and the rest as a testing set. This is repeated in a round robin manner. In addition, another approach is the leave-one-out mode, where in our case the dataset would be divided into 387 folds, so that only one logfile is examined at a time. This approach is more appropriate for the better estimation of model performance, but it costs in computational time; this is planned to be used in future research.

4. Results and Discussion

For each of the five machine learning models and for each of the eight feature sets of 100 to 800 features, a number of experiments was performed in order to find those classifier parameters that lead to the best performance.

The best results in simultaneously finding the gender and age of an unknown user, for each different feature set, using SVM, along with classifier parameters, are presented in Table 5.

Table 5. The best performance of SVM over different feature sets. Where "Acc." denotes the accuracy, "TBM" denotes the time to build model (training time), "F1" denotes the F-score, and "AUC" denotes the area under the ROC curve.

# of Features	Performance Values				Classifier Parameters	
	Acc.	TBM	F1	AUC	C	Kernel
100	60.0%	0.38	0.600	0.844	9.0	Polykernel
200	65.4%	0.27	0.653	0.861	4.0	Polykernel
300	63.3%	0.30	0.629	0.870	1.5	Polykernel
400	64.9%	0.32	0.645	0.871	1.5	Polykernel
500	64.1%	0.41	0.640	0.870	2.8	Polykernel
600	64.1%	0.47	0.630	0.868	1.0	Polykernel
700	66.7%	0.49	0.653	0.874	1.0	Polykernel
800	65.6%	0.50	0.657	0.875	2.2	Polykernel

SVM shows the best performance using the polynomial kernel in each different feature set, while the highest accuracy is achieved in a feature set with less than 800 features, which is the largest tested.

Table 6 shows the results and classifier parameters for SL, for each different feature set.

Table 6. The best performance of SL over different feature sets.

# of Features	Performance Values				Classifier Parameters	
	Acc.	TBM	F1	AUC	Last Iteration	Weight Trimming
100	57.9%	0.90	0.580	0.835	65	85%
200	61.8%	3.02	0.615	0.853	70	95%
300	61.0%	6.73	0.608	0.853	40	100%
400	61.2%	3.60	0.611	0.852	20	95%
500	60.7%	28.44	0.605	0.863	120	100%
600	63.1%	7.92	0.630	0.853	55	95%
700	63.3%	8.89	0.631	0.864	50	95%
800	64.1%	15.37	0.641	0.863	155	95%

The best SL performance is achieved in the feature set with the largest number of features. Regarding NB, the results of the experiments showed better performance for each feature set, as presented in Table 7.

Table 7. The best performance of NB over different feature sets.

# of Features	Performance Values			
	Acc.	TBM	F1	AUC
100	47.3%	<0.01	0.464	0.801
200	51.2%	<0.01	0.503	0.816
300	55.0%	<0.01	0.540	0.832
400	54.3%	0.01	0.529	0.831
500	56.1%	0.07	0.546	0.839
600	57.9%	0.12	0.564	0.847
700	57.9%	0.13	0.564	0.843
800	57.9%	0.12	0.564	0.844

The accuracy and time complexity of the NB increase as more features are used, while the best performance is the one of 600 features since has the higher AUC.

Similarly, the best results and BNC settings that lead to them are shown in Table 8.

Table 8. The best performance of BNC over different feature sets.

# of Features	Performance Values				Classifier Parameters	
	Acc.	TBM	F1	AUC	Initial Count	Max Number of Parents
100	50.4%	0.19	0.503	0.822	0.250	3
200	55.0%	1.96	0.547	0.854	0.400	5
300	56.1%	0.05	0.560	0.848	0.020	1
400	56.1%	3.96	0.553	0.851	0.180	3
500	56.9%	0.05	0.568	0.853	0.035	1
600	58.9%	0.05	0.587	0.862	0.024	1
700	59.4%	0.08	0.592	0.865	0.020	1
800	60.0%	0.61	0.598	0.865	0.021	1

BNC performance, as expected, improves as the number of features used increases, while time complexity depends on the number of features and the max number of parents.

Finally, Table 9 presents the best performance and the corresponding RBFN settings for each different feature set.

Table 9. The best performance of RBFN over different feature sets.

# of Features	Performance Values				Classifier Parameters	
	Acc.	TBM	F1	AUC	# of Clusters	Min Std. Deviation
100	74.2%	0.87	0.744	0.895	190	1.20
200	78.3%	1.33	0.785	0.916	140	1.10
300	77.0%	1.92	0.773	0.917	190	1.15
400	77.8%	2.46	0.781	0.913	200	1.15
500	77.0%	2.83	0.773	0.908	160	1.15
600	77.0%	3.25	0.772	0.910	200	1.20
700	77.0%	3.83	0.772	0.909	100	1.20
800	77.0%	4.34	0.772	0.909	140	1.20

As in SVM, so in RBFN, the best performance is presented in a feature set with less than 800 features, while the time complexity increases steadily with the number of features. To explain why the accuracy of the system slightly decreases as the number of features in the feature set increases, meticulous experiments over the subsets of the feature set which lie beyond the scope of this work need to be conducted. This task is left as a future research goal.

At this point two comparisons can be made. One regards the performance of the classifiers used. Figure 1 shows the optimal performance of each classifier.

As shown in Figure 1, the RBFN is superior to all other classifiers in accuracy exceeding 78%, in the F-score and in the area under the ROC curve, with SVM second in each case having an optimal accuracy approaching 67%, while NB has the lowest values in all three sizes. Regarding the time required to train the model, the NB runs faster than other classifiers, followed by the SVM, while the SL presents the greatest time complexity.

The confusion matrix of the best run, which is that of the RBFN model in the feature set of 200 features, in order to give a picture of the distribution of predictions, is presented in Table 10.

One observation made in Table 10 is that a large proportion of erroneous predictions, 27 out of 84, are in adjacent age groups. This, in part, was to be expected, as the classification of users according to the age group they belong to is not clear enough. For example, a 26-year-old user probably has more in common with a 25-year-old user than with a 35-year-old user. However, while the user is in the same group with the latter, they will be in a different age group than the former.

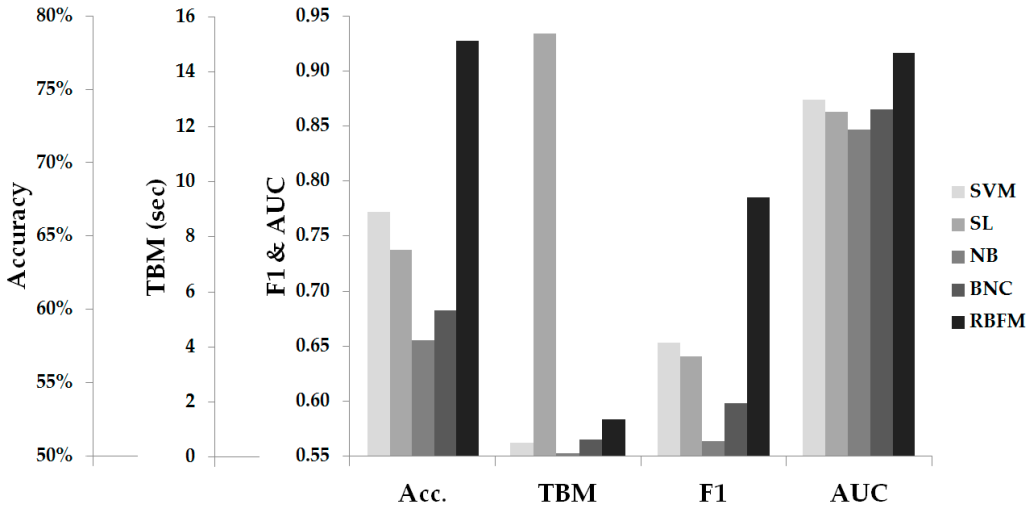

Figure 1. The best performances of the five classifiers. (SVM—support vector machine; SL—simple logistic; NB—Naïve Bayes; BNC—Bayesian Network Classifier; RBFN—Radial Basis Function Network).

Table 10. Confusion matrix of the experiment with the highest accuracy.

Predicted as	Male 18–25	Male 26–35	Male 36–45	Male 46+	Female 18–25	Female 26–35	Female 36–45	Female 46+
M18–25	60	2	1	3	4	3	0	0
M26–35	1	37	0	0	2	2	1	0
M36–45	0	3	49	2	1	4	3	0
M46+	1	0	0	16	0	5	3	0
F18–25	0	0	1	0	14	4	2	2
F26–35	1	2	3	0	4	72	4	0
F36–45	0	2	4	2	2	6	38	1
F46+	0	0	2	0	0	1	0	17

The second comparison is between the performance of the system presented in the present work in simultaneously searching for a user's gender and age group and the performance of the systems for separately searching for these two characteristics in the study [16]. This comparison is made in Table 11.

The column "Both" in the results [16] in Table 11 is calculated from the product of the corresponding values in the columns "Gender" and "Age". Indeed, gender and age of a person are two events statistically independent of each other and thus, to calculate the probability of a user belonging to a specific gender and age group, it is sufficient to multiply the individual probabilities [21]. The conclusion is that all machine learning models, except RBFN, have a higher accuracy of 3 to 6% in simultaneously finding the gender and age of an unknown user, compared to the finding of these characteristics separately. Although the phenomenon is not universal, it seems that the choice of such features that do not focus only on the segregation of users according to their gender, or only according to their age, leads to systems with better performance in the simultaneous finding of user characteristics.

Table 11. Accuracy comparison between research results.

Model	[16]			This Research
	Gender	Age	Both	Both
SVM	86.1%	74.2%	63.9%	66.7%
SL	84.2%	71.8%	60.5%	64.1%
NB	77.0%	66.9%	51.5%	57.9%
BNC	77.5%	69.8%	54.1%	60.0%
RBFN	92.0%	89.2%	82.1%	78.3%

In addition, in the work [16], 514 features were used for the gender classification and 690 for the age classification. Some of these features are common to both classification problems, but many others are found in only one of them, with the result that in order to achieve the performance presented in Table 11, a total of 947 keystroke dynamics features are needed. This is a number of features 18% larger than the largest feature set used in this study and it is almost five times more than the 200 features that give the best result in RBFN. The consequence of this is that a shorter processing time is required to extract the appropriate features. This phenomenon is observed in the simultaneous search for additional user characteristics, such as handedness, educational level, etc. Therefore, one of the novel contributions of this work, to the best of our knowledge, the simultaneous search of more than one user characteristics using keystroke dynamics. Conducting experiments for this effort shows that, the simultaneously searching for multiple characteristics of an unknown user, using a single feature set for all characteristics, leads to similar or better results, but while spending less time extracting features.

4.1. Comparing Keystroke Durations and Down-Down Digram Latencies

The findings of Table 4 lead to the investigation of the importance of the two most frequently used types of features in keystroke dynamics, namely KD and DDDL, and to their direct comparison. Specifically, the two best performing classifiers, SVM and RBFN, were used to test which of the two types of features best separates users by gender and age.

In the available dataset there are actions recorded from 108 keys and from about 11,500 diagrams. Additional experiments were performed to find the classifier parameters that lead to the best performance for the dataset with all the available features, all KD and all DDDL. The results are presented in Table 12.

Table 12. Comparison of effectiveness of features by classifier performances.

Features	SVM				RBFN			
	Acc.	**TBM**	**F1**	**AUC**	**Acc.**	**TBM**	**F1**	**AUC**
All KD	49.9%	0.56	0.500	0.809	66.7%	4.40	0.666	0.857
All DDDL	60.5%	5.98	0.601	0.839	68.2%	90.53	0.681	0.858
All KD & DDDL	64.9%	5.55	0.646	0.864	79.3%	79.78	0.792	0.914

From Table 12, it is clear that including all available DDDLs leads to higher performing systems than those using all available KDs. However, using such a large feature set, such as one that contains all DDDLs, results in very high time complexity. Thus, using all available DDDLs requires tens of times more training time than using all available KDs. In the case of RBFN, an increase in accuracy of 1.5% requires 20 times more execution time. The results are not so diverse in SVM, where increasing the accuracy by 10% takes 10 times more time. So, on the one hand, using only all KDs leads to systems that run much faster than those that use only all DDDLs, but on the other hand, they have lower accuracy and in some cases much lower.

The claim that far fewer KDs are used and that the comparison should be made on the same number of features is refuted by the fact that there are many more available DDDLs that can contribute with information that contain. Therefore, it is not possible to draw a safe conclusion, with the existing data and results, as to which of the two types of features is considered the most important. However, what can be safely extracted is that the combined use of both types of features leads to even better results. In fact, in the case of RBFN the system has an accuracy of 79.3%, which is even higher than those shown in Figure 1. In addition, further proof of this claim, as shown in Tables 5, 9 and 12, is that experimenting with the same number of features, systems that use both KDs and DDDLs perform much better than those that use only KDs.

The fact that the combination of KDs and DDDLs leads to systems with better accuracy sets the basis for a research that will involve even more types of keystroke dynamics features.

4.2. Improving the Accuracy

The main requirement of a system that attempts to recognize certain characteristics from a completely unknown user is the accuracy it presents and in our research is the most important criterion for its evaluation. For this reason, some of the algorithms which use the results of various classifiers apply numerous techniques to improve the accuracy and usually are called meta-algorithms, have been utilized in this research.

As such, AdaBoost [22] was initially used with "weak" classifier the RBFN, which showed the best results, as shown in Figure 1. The experiments were conducted in the feature set of 200 features, in which the highest accuracy of 78.3% is presented and in the feature set of 800 features, which is the largest of the feature sets formed with the features that have non-zero *IG*. The best performance of the model, for every 10 iterations of the AdaBoost algorithm, is presented in Table 13, which does not include experiments with the number of iterations where the training time was prohibitively long.

Table 13. Improving accuracy with AdaBoost and RBFN.

Features	Iterations	Acc.	TBM	F1	AUC
200	10	77.5%	152.30	0.778	0.943
200	20	80.1%	261.87	0.805	0.957
200	30	80.6%	617.82	0.808	0.958
200	40	81.1%	995.85	0.814	0.959
200	50	81.9%	1205.98	0.821	0.960
800	10	79.3%	677.89	0.793	0.946
800	20	79.6%	1161.81	0.796	0.955

As shown in Table 13, improving the accuracy of the RBFN using AdaBoost is possible, but the cost is very high in terms of the required computational time. Indicatively, for an improvement of 3.6% it takes about 900 times more time.

Another algorithm used is Rotation Forest [23] with base classifier C4.5 decision tree. The experiments were conducted in the feature set of 800 features and the best performance of the model, for every 10 iterations of the Rotation Forest algorithm, up to 100 iterations, is presented in Table 14.

Among the classifiers that are tested for the requirements of the presented study is the C4.5 decision tree, which presents 36.7% as higher accuracy with training time of 1.75 s, which was the reason for not being included in the models with the best performance. However, Table 14 shows that using the rotation forest algorithm, the accuracy of C4.5 is significantly improved reaching 80%, although the processing time has to be multiplied due to the iterative approach.

Table 14. Improving accuracy with Rotation Forest and C4.5 decision tree.

Iterations	Acc.	TBM	F1	AUC
10	65.4%	11.19	0.651	0.895
20	69.5%	25.55	0.691	0.927
30	72.9%	42.69	0.721	0.924
40	75.7%	47.49	0.751	0.940
50	76.0%	56.95	0.750	0.946
60	75.5%	78.66	0.749	0.940
70	77.0%	109.74	0.761	0.947
80	77.5%	105.44	0.771	0.949
90	78.8%	128.56	0.780	0.952
100	76.2%	114.35	0.757	0.945

Although many more experiments have to be conducted, using more boosting algorithms, more classifiers and more fine-tuning, which goes beyond the scope of this research,

it is concluded that the accuracy of the system can be improved, however, with the cost of the much longer training time.

4.3. Method Limitations

As can be seen from the results of the present study, as well as others in the field of keystroke dynamics, it is possible to find some user characteristics with quite high percentages in accuracy. However, there are some issues to be discussed regarding limitations and objections to the use of the proposed method.

The user typing pattern is quite hidden and it is not clear what separates the males from the females, the users of one age group from those of another, etc. However, it is not known whether a user can modify the way they type to hide their characteristics. If this is possible, then new techniques should be sought to overcome this problem.

In addition, the collection of keystroke dynamics data is a limitation of the method. This is because, while the data exploited by keystroke dynamics do not reveal personal or sensitive user data, the way they are obtained leaves questions as to whether a malicious user can exploit them.

5. Conclusions and Future Work

Keystroke dynamics are a subject of research mainly in the field of user authentication, but also in the detection of the physical and mental condition of users, as well as in their classification according to some inherent or acquired characteristics. The latter can find a variety of applications, such as obtaining valuable information about the person who committed a cybercrime in a digital forensics investigation, facilitating Internet users in exploiting useful services on a case-by-case basis, improving targeted advertising and warning of unsuspecting users about the danger of becoming victims of deception. In addition, keystroke dynamics can be added as a side authentication and verification mechanisms for the cases where a continuous authentication is needed. The contributions from this research can be applied to user sessions dynamically to provide security measures for an unattended computer terminal that is left unlocked to verify the user activity to raise alerts or flags.

In almost all keystroke dynamics studies the features used were keystroke durations and the four forms of diagram latencies. Due to this fact, in this work, it is investigated which type of features includes the most information. For this purpose, this research contributes to the existing keystroke dynamics literature by introducing a system that simultaneously recognizes the gender and age of unknown users and by exploiting the features of an existing keystroke dynamics dataset with five known classifiers. The results showed that (i) the identification of the class, among eight, to which a user belongs according to his gender and age group can be achieved with a percentage approaching 80% and (ii) it is not possible to draw a safe conclusion as to which of the two types of keystroke dynamics features examined contains the most information and is most appropriate for user segregation. However, one conclusion reached was that the combined use of both types gives significantly better results.

As a conclusion, the extension of the research will be directed towards the extraction of other types of features and the implementation of systems that will use a combination of them. Such features can come from trigrams, tetragrams, etc., typing pauses, typing corrections, etc. Other possible extensions are to approach the problem with Dempster-Shafer theory [24] to extend the existing dataset by recording volunteers which have different native languages in order to examine the possibility of recognizing this characteristic and to test the robustness of our methodology using different keystroke dynamics datasets.

Author Contributions: Conceptualization I.T. and V.K.; methodology, I.T.; software, I.T.; validation, C.Y. and I.T.; writing—original draft preparation, I.T.; writing—review and editing, C.Y.; supervision, V.K.; project administration, C.Y.; funding acquisition, V.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from the EU's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 778229 (Ideal-Cities) and under the grant agreement No. 830943 (ECHO).

Data Availability Statement: The logfiles contain sensitive and/or personal data of the volunteers who participated in the typing recording and are therefore not available.

Acknowledgments: A preliminary version of this research was presented in the 12th International Network Conference (INC 2020): "User Attribution Through Keystroke Dynamics-Based Author Age Estimation", by Tsimperidis, I.; Rostami, S.; Wilson, K.; Katos, V.

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

References

1. Sabhanayagam, T.; Venkatesan, V.P.; Senthamaraikannan, K. A Comprehensive Survey on Various Biometric Systems. *Int. J. Appl. Eng. Res.* **2018**, *13*, 2276–2297.
2. He, L.; Li, Z.; Shen, C. Performance Evaluation of Anomaly-Detection Algorithm for Keystroke-Typing based Insider Detection. *Tsinghua Sci. Technol.* **2018**, *23*, 513–525. [CrossRef]
3. Subangan, S.; Senthooran, V. Secure Authentication Mechanism for Resistance to Password Attacks. In Proceedings of the 19th International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 2–5 September 2019; pp. 1–7. [CrossRef]
4. Banerjee, S.P.; Woodard, D.L. Biometric Authentication and Identification using Keystroke Dynamics: A Survey. *J. Pattern Recognit. Res.* **2012**, *7*, 116–139. [CrossRef]
5. Lin, C.H.; Liu, J.C.; Lee, K.Y. On Neural Networks for Biometric Authentication Based on Keystroke Dynamics. *Sens. Mater.* **2018**, *30*, 385–396.
6. Venugopal, P.C.; Viji, K.S.A. Applying Empirical Thresholding Algorithm for a keystroke Dynamics Based Authentication System. *Int. J. Inf. Commun. Technol.* **2019**, *18*, 383–413.
7. Young, J.R.; Davies, R.S.; Jenkins, J.L.; Pfleger, I. Keystroke Dynamics: Establishing Keyprints to Verify Users in Online Courses. *Comput. Sch.* **2019**, *36*, 48–68. [CrossRef]
8. Sun, Y.; Ceker, H.; Upadhyaya, S. Anatomy of Secondary Features in Keystroke Dynamics—Achieving More with Less. In Proceedings of the 2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA), New Delhi, India, 22–24 February 2017; pp. 1–6. [CrossRef]
9. Brindha, S.; Sukumaran, S. Pattern Taxonomy Deploying Model for Text Document Classification. *Int. J. Comput. Sci. Inf. Secur.* **2017**, *15*, 212–218.
10. Buker, A.A.N.; Roffo, G.; Vinciarelli, A. Type Like a Man! Inferring Gender from Keystroke Dynamics in Live-Chats. *IEEE Intell. Syst.* **2019**, *34*, 53–59. [CrossRef]
11. Bandeira, D.R.C.; Canuto, A.M.P.; Costa-Abreu, M.D.; Fairhurst, M.; Li, C.; Nascimento, D.S.C. Investigating the Impact of Combining Handwritten Signature and Keyboard Keystroke Dynamics for Gender Prediction. In Proceedings of the 8th Brazilian Conference on Intelligent Systems (BRACIS), Salvador, Brazil, 15–18 October 2019; pp. 126–131. [CrossRef]
12. Da Costa-Abreu, M.; Goncalves, J.C. An Evaluation of a Three-Modal Hand-Based Database to Forensic-Based Gender Recognition. In Proceedings of the 19th Brazilian Symposium on Information and Computer System Security (SBSeg 2019), São Paulo, Brazil, 2–5 September 2019; pp. 1–6.
13. Tsimperidis, I.; Rostami, S.; Wilson, K.; Katos, V. User Attribution through Keystroke Dynamics-Based Author Age Estimation. In *Selected Papers from the 12th International Networking Conference*; Lecture Notes in Networks and Systems; Ghita, B., Shiaeles, S., Eds.; Springer: Cham, Switzerland, 2020; Volume 180, pp. 47–61. [CrossRef]
14. Tzafilkou, K.; Protogeros, N. Mouse Behavioral Patterns and Keystroke Dynamics in End-User Development: What Can They Tell Us About Users' Behavioral Attributes? *Comput. Hum. Behav.* **2018**, *83*, 288–305. [CrossRef]
15. Ulinskas, M.; Wozniak, M.; Damasevicius, R. Analysis of Keystroke Dynamics for Fatigue Recognition. In *Computational Science and Its Applications*; Lecture Notes in Computer Science; Gervasi, O., Ed.; Springer: Cham, Switzerland, 2017; Volume 10408, pp. 235–247. [CrossRef]
16. Tsimperidis, I.; Arampatzis, A. The Keyboard Knows About You: Revealing User Characteristics via Keystroke Dynamics. *Int. J. Technoethics* **2020**, *11*, 34–51. [CrossRef]
17. Kalpana, P.; Mani, K. A New Hybrid Framework for Filter based Feature Selection using Information Gain and Symmetric Uncertainty. *Int. J. Eng. Trans. B Appl.* **2017**, *30*, 659–667. [CrossRef]
18. Chauhan, J.; Rajasegaran, J.; Seneviratne, S.; Misra, A.; Seneviratne, A.; Lee, Y. Performance Characterization of Deep Learning Models for Breathing-based Authentication on Resource-Constrained Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 1–26. [CrossRef]
19. Feng, D.; Cortese, G.; Baumgartner, R. A Comparison of Confidence/Credible Interval Methods for the Area Under the ROC Curve for Continuous Diagnostic Tests with Small Sample Size. *Stat. Methods Med. Res.* **2017**, *26*, 2603–2621. [CrossRef] [PubMed]

20. Raju, K.S.; Murty, M.R.; Rao, M.V.; Satapathy, S.C. Support Vector Machine with K-fold Cross Validation Model for Software Fault Prediction. *Int. J. Pure Appl. Math.* **2018**, *118*, 321–334.
21. Mundici, D. De Finetti Coherence and the Product Law for Independent Events. *Synthese* **2019**, *196*, 265–271. [CrossRef]
22. Wang, F.; Li, Z.; He, F.; Wang, R.; Yu, W.; Nie, F. Feature Learning Viewpoint of Adaboost and a New Algorithm. *IEEE Access* **2019**, *7*, 149890–149899. [CrossRef]
23. Chen, T. An Improved Rotation Forest Algorithm Based on Heterogeneous Classifiers Ensemble for Classifying Gene Expression Profile. *Adv. Model. Anal. B* **2017**, *60*, 1–24. [CrossRef]
24. Xiao, F. Generalization of Dempster–Shafer Theory: A Complex Mass Function. *Appl. Intell.* **2020**, *50*, 3266–3275. [CrossRef]