ENGINEERING RESEARCH JOURNAL (ERJ)
Vol. 1, No. 51 Jan 2022, pp.95-104
Journal Homepage: erj.bu.edu.eg

Fixed-Text vs. Free-Text Keystroke Dynamics for User Authentication

Eng. Shimaa S. Zeid, Prof. Dr. Raafat A. ElKamar, Dr. Shimaa I. Hassan

Abstract : There are many physical biometrics such as iris patterns and fingerprints. There are also interactive gestures like how a person types on a keyboard, moves a mouse, holds a phone, or even taps a touch screen. Keystroke dynamics or typing dynamics is an automatic method that confirms the identity of an individual based on the manner and the way of the user typing on a keyboard. There are two types of keystroke systems, Fixed-text system, and free-text system and each of them has it is own importance. In this research paper, we are investigating the possibility of classifying individuals using features extracted from their keystroke dynamics with two different datasets: (1) fixed-text dataset with different difficulty levels and (2) free-text dataset with no restrictions what a user types on the keyboard. Investigation was done using several classification techniques: RandomForest (RF), Support Vector Machines (SVM), BayesNet (BN), and K-Nearest Neighbors (KNN). The highest accuracy achieved with the fixed-text dataset was 98.8% using RF for classification while the highest achieved accuracy with the free-text dataset was 87.58 % using RF classifier**.**

KEYWORDS: : Keystroke dynamics, User authentication, Continuous authentication, Feature Matching Methods, Machine Learning, fixed-text and free-text.

Introduction

There is no doubt that we need a secure access control method in all security applications, traditional methods such as PINs, passwords, and tokens are not enough because they might easily get stolen or lost. On the other hand, biometric systems

are based on the measurement of distinctive physiological and behavioral characteristics. Finger-scan, facial-scan, iris-scan, hand-scan, and retina-scan are considered physiological biometrics, based on direct measurements of a part of the human body. Voice-scan and signature-scan are considered behavioral biometrics; they are based on measurements and data derived from an action and therefore indirectly measure characteristics of the human body [1].

It's also well known that there is a lot of drawbacks in username/password schemes where passwords might be forgotten, shared, or attacked hence the system will be not safe. As an alternate, the way a user types a pattern could be unique because of neuron-physiological factors that are responsible for making written signatures unique. Consequently, and from this perspective, keystroke dynamics is a better way to authenticate persons based on their typing style [2].

Keystroke dynamics implies that we do not care about what is being typed, but rather how it is being typed. Keystroke biometrics have another interesting property which is that keystroke dynamics data can be collected without user's knowledge or even cooperation. Another interesting point in favor or keystroke dynamics is that passwords can be guessed by a lot of ways like social engineering, dictionary attack, spyware, and brute force attacks. That all made hand keystroke dynamics biometric systems to become the alternate of username/password schemes. Keystroke dynamics is a behavior biometric constitution. It is responsible of the system protection, and responsible of giving high level of usability to the system [1].

There are two main types of keystroke systems: (1) fixed-text systems and (2) free-text systems. Fixed-text systems are applied at the log-in time to make sure of the user's identity and only at the beginning of a user's session. In such systems, users are forced to retype their password a specific number of times, usually fixed, to determine the user's typing behavior for that specific password. On the other hand, free-text systems or dynamic systems do not have such restrictions about the text a user type. Users have all the freedom, in free-text systems - to write any text of any length with and without any constraints [3].

Most of the early research though focused on keystrokes generated by typing fixed words, that is fixed-text systems. It was not before 1995 until Shepherd et al. were the first to be concerned in continuous authentication. In 1997, the overall performance of free-text systems was disappointing for giving only 23% correct classification rate while fixed-text produced roughly 90% classification success rate which clearly indicates how more complex using free-text system is [4].

There are basically two main stages for user's authentication using keystroke dynamic: (1) The Enrollment stage and log-in stage. In the first stage, enrollment, we collect data related to the user such as user's username and user's password in addition to recording the behavior of user's typing. At this stage, the system collects the times of keystrokes and timing features are extracted to build up a template for the typing behavior of each user. That created template is considered as a profile for a user and is stored along with another user information in a directory or a database. The second stage, log-in stage, is any other time the user would like to login through the system that has created in the first stage. This system gathers times of user's keystroke and subsequently extracts the relevant features. A matching between collected features and the corresponding ones stored in the datasets is then performed and according to the results of this process of matching, the result either gets the access to the system or get denied [5].

We worked on two online datasets, The MOBIKEY Keystroke Dynamics Password Dataset as a fixed-text dataset [6], and The Politehnica University Timisoara keystroke dataset as a free-text dataset [7]. It is a very new dataset which has no published work on it up till now. We made a lot of data preprocessing on it as it was almost a row data. We applied four classifiers for both of these two datasets such as *RandomForest (RF), Support Vector Machines (SVM), BayesNet (BN), and K-Nearest Neighbors (KNN)*

The remaining of this paper is structured as follows: In the following section we introduce related work on keystroke authentication, in section three we will discuss methodology of our work by showing tools used, extracted features and applied methods, in section four we will present results and make a discussion about these results, in the fifth section we made a conclusion for the results and whole work in this paper.

Related Work

Meng et al. [8] questioned if we could use keystroke dynamics as a biometric by building a training interface and make users train themselves in simulating another person's password typing rhythm. In this study they used two groups, each one contained 8-character length passwords and they used an easy and a complex one. They found that passwords that were easier to type were also easier to simulate.

Complexity measurements that are related to the typing of a password were listed by Monda et al. [9] and that led later to several performance measurements. They claimed that easier passwords are better choice for keystroke dynamics biometrics which happened to be against what Meng said [8].

It was reported by Hala H. Zayed et al. [10] that their system was measured using four distance measures: Manhattan, Euclidean, Manhattan with standard deviation, and Mahalanbois. They took the standard deviation into account which increased the performance of the matching process. Manhattan with standard deviation had the most accurate results because it concentrated the standard deviation of the training samples. Results included an EER of 4.9 with majority voting (MV) considered for selecting specific features and an EER of 6.6 when all features were considered.

Robert Cockell and Basel Halak [11] found two statistics-based ways for data analysis. The first was based on the simple averages computation while the second was based on a probability estimation, with both ways depending on the characteristics extracted from each button press. As they expected, the results were the same for the same user for each of the two techniques. Users who pressed the buttons in a fixed way were less distinct from other users doing the same thing compared to others who did not. This is a relatively predictable result, the second factor is that the values for correct data varied per user, without affecting how well that user was recognized. This shows that no fixed threshold should be used for verification with their algorithms.

H. Elmiligi et al. [12] focused on classifying users' behavior when a computer device is to be accessed and authentication is required. Their work used keystroke dynamics by capturing the behavioral biometric of a user and subsequently applying concepts of machine learning to classify users. It was claimed then that the best classification performance was from the SVM RBF class. They have also found that touch coordinates, size, and pressure are the most relevant when it comes to user’s authentication.

Iapaet al. [7] noticed that while more than 9000 researchers did their work on free-text keystroke field, one common problem among all of them was the lack of availability of public datasets of free-text. Hence, they provided a dataset that included free-text keystroke data for 80 users. They have also done some analysis to the collected data that should help researchers to know where to start from when it comes to things such as feature selection. They have also implemented an authentication algorithm using the collected dataset and reported an EER of 13.89% and 6.55% using Manhattan distance and the proposed distance respectively for distance measurement.

Jianwei Li, Han-Chih Chang and Mark Stamp [13] worked on verifying user identity based on keystroke dynamics problem. They made a novel feature engineering method which creates an image similar to transition matrices.A convolution neural network (CNN) with cutout achieves the best results for this image-like feature.

Augustin-Catalin Iapa; Vladimir-Ioan Cretu [14] aimed to analyze the possibilities of increasing the efficiency of an authentication algorithm based on keystroke dynamics by reducing the value of the Equal Error Rate (EER). They modified the Manhattan distance calculation formula to generates better performances, EER was improved by 38.53%, so the EER value became 3.27%, compared to 5.32% obtained with the classic Manhattan formula.

Methodology

The flow diagram of our methodology is shown in figure 1.

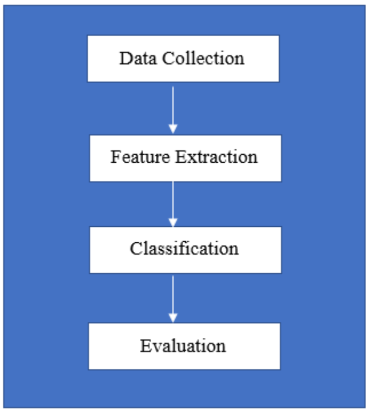


Fig 1: flow diagram of the methodology used in this paper [15]

1-Datasets

Two datasets were used in our work. One for fixed-text and another for free-text.

Fixed-Text Dataset

The MOBIKEY Keystroke Dynamics Password Database was used as a fixed-text dataset [6]. This dataset contains 54 subjects (49 males, 5 females) with an ages range of 19 to 26 years and an average of 20.61 years. Three passwords with different difficulty levels were used by subjects (easy: kicsikutyatarka, logicalstrong: Kktsf2!20, strong (.tie5Roanl). There are 60 samples per each subject and at least 3 sessions per subject.

Free-Text Dataset

The Politehnica University Timisoara keystroke dataset was used as a free-text dataset [7]. This dataset contained keystrokes on the keyboard by 80 users (35 males, 44 females, 1 unknown) with ages that ranged from 16 to 59 years and had an average of 28.19 years. Data was collected in a single session via a web platform using a keyboard of a desktop computer or laptop (64 laptops, 15 desktops, 1 unknown). There were a total number of 410,633 key-events collected with an average of 5132 key-events per user and a total time interval of almost 24 hours to collect the whole dataset from all users. Language used by users in the collection of this dataset was Romanian.

Data is Presenting as in the following form:

Table 1:Data representation of the free dataset

Pressed key code	**Type**	**Timestamp**
16	0	434889
86	0	435006
86	1	435146
16	1	435221
82	0	435308
…	…	…

Where the first column of the datasets represents the pressed key code, the second column represents the type of the event that occurred (0 for press, 1 for release), and the third column is the timestamp at which this key event has taken place.

2-Features:

Figure 2 shows extracted features like hold time, flight time, down down time and up down time.

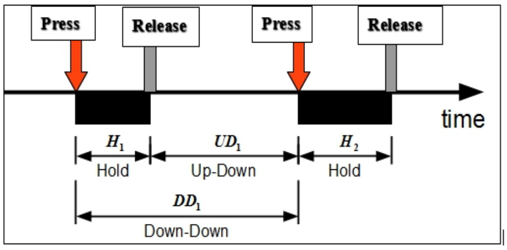


Fig 2: Extracted features[15]

As shown in figure 2, hold Time is the time between press and release of a key, while the flight Time is time between pressing a key and pressing the next one, hold time is the time between one Press and the sequential release. Down-Down time is the time between two sequential Presses, Up-Down time is the time between one release and the next press [16].

In the fixed-text dataset [6]relevant features were used in our system for each key were key hold time (HT), down-down time (DD), up-down time (UD), key press pressure (P).

In the free-text dataset [7] there were only 3 features provided in this dataset for each user: the code of a key, whether it was a press or release event, and the corresponding timestamp at which this key event (press/release) has taken place

3-Methods:

Four classification techniques were used such as BN classifier, SVMs (SVM), KNN (KNN), and RF.

BN is a Bayesian classification network that is based on biased random competition by using Gaussian kernels [17]. It is a neural network architecture that is capable of learning the probability density functions (PDFs) of individual pattern classes using a collection of learning trails as shown in figure3, it is designed for pattern classification based on the Bayesian decision rule [17].

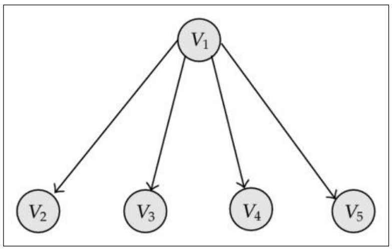


Figure 3: Navi Bayes[15]

SVMs are a combination of supervised learning techniques used for classification, regression and detection as shown in figure 4. One of the very important advantages of SVMs is that they are very effective in high dimensional spaces. It is also effective when the number of dimensions is relatively more than the number of samples.[18]

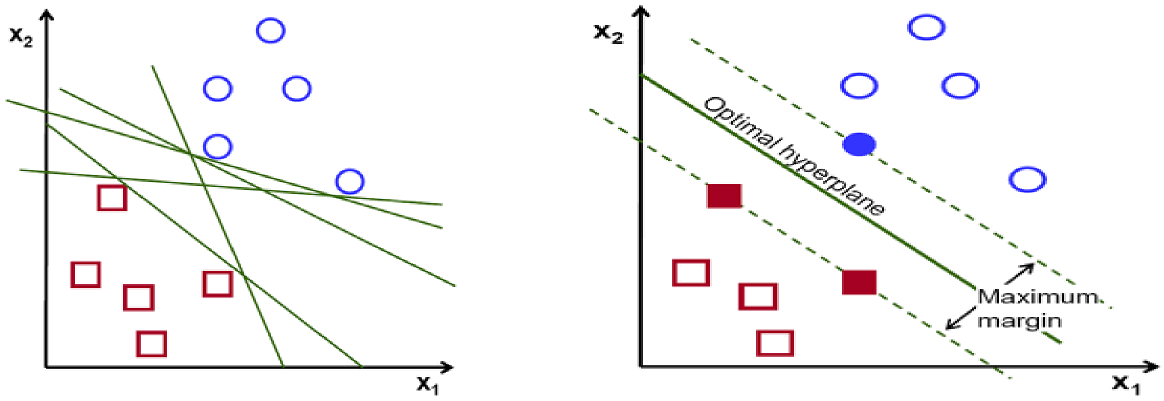


Figure 4: Possible hyper plane for SVM[18]

K-Nearest Neighbour whose idea is shown in figure5 is one of the topmost machine learning algorithms. It is very easy to understand, simple, and adaptable. KNN can be used in a lot of applications like healthcare, finance, and handwriting. It is a lazy learning algorithm which means all training data are also used in the testing phase. One drawback is that the testing phase becomes slower and costlier. This means much time spent and more memory used [19].

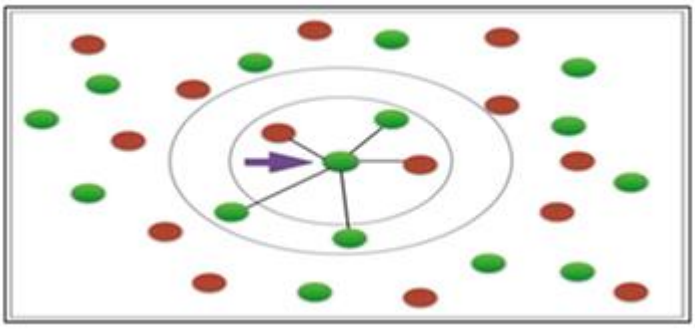


Fig 5: :KNN[17]

RF whose idea is shown in figure6 is classifiers is one of the ensemble-based learning techniques. The advantages of such techniques are being fast, easy, simple, and very successful in a variety of domains. The RF technique comprises the construction of a number of "simple" decision trees in the training phase then makes the majority vote combination rule for them.[18]

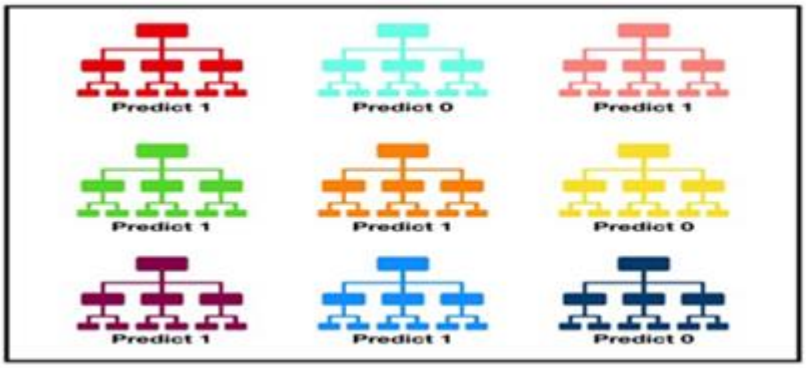


Fig 6: Visualization of a RF Model Making a Prediction[18]

Experimental Results and Discussion

Tools

The main tools employed is Weka which is tool that combines several machine learning algorithms that are used in data mining tasks. Algorithms could be applied directly to a dataset or could be called from another program implemented in Java programming language. Weka is a very powerful tool that provides several methods such as data pre-processing, feature extraction, classification, clustering, regression, association rules, and visualization.

Evaluation metrics

Evaluation parameters that were calculated and used are true prediction rate, false prediction Rate, ROC average, and accuracy.

True Positive (TP): This happens when the model predicts that the claimed person is the real one and this prediction is correct [20].

True Positive Rate = True Positives / (True Positives + False Negatives)

Equation [1] How to calculate The True positive rate term[20]

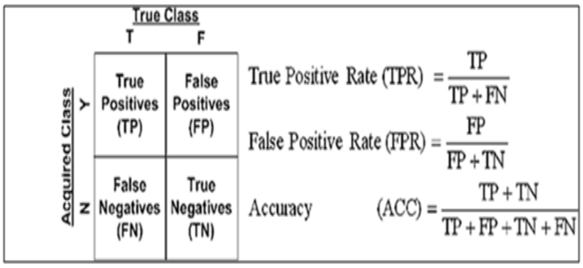
False Positive (FP): This happens when the model predicts that the claimed person is the real one but actually, he is not [20].

False Positive Rate = False Positives / (False Positives + True Negatives)

Equation [2] How to calculate The False positive rate term [20]

ROC Curve Score: the area under the curve can be calculated to give a single score for a classifier model across all threshold values. This is called the ROC area under curve or ROC AUC or sometimes ROCAUC. The score is a value between 0.0 and 1.0 for a perfect classifier [20].

Accuracy: As shown in figure 7, It is the ratio between the correct predictions and the total predictions made [21].



Acquired Class	True Class T	True Class F
Y	True Positives (TP)	False Positives (FP)
N	False Negatives (FN)	True Negatives (TN)

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP+FN}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP+TN}$$

$$\text{Accuracy} \quad (ACC) = \frac{TP+TN}{TP+FP+TN+FN}$$

Fig 7: equations of True positive Rate, False Positive Rate and Accuracy [21]

Experimental Results

Results and discussion are split in two sections; the first is for work done with the fixed-text dataset while the second is for the free-text dataset.

Fixed-text Dataset

The dataset instances isdivided by the percentages 40% for training and 60% for testing.Different classifiers have been used such as RF, SVMs, BN, and KNN. Moreover, we have investigated employing different combinations of these classifiers along with different combination rules which such as Average of Probabilities (AoP) and Major Voting (MV).

Table 2shows the results of True Prediction Rate, False Prediction Rate, Average Roc Area and Accuracy when we used different classifiers such as SVM, KNN, BN and RF, we calculated the Combination Rate, Average Roc Area and Accuracy on the easy password (**kicsikutyatarka**) part of the dataset.

Table 2: Results of using different classifiers with the easy password “kicsikutyatarka”.

Method	TP Rate (%)	FP Rate (%)	Average Roc Area (%)	Accuracy (%)
RF	98.8	0.1	100	**98.83**
SVM	97.2	0. 3	99.6	97.16
BN	96.8	0. 4	99.9	96.83
KNN (k=8)	93.8	0. 7	96.4	93.82

It is obvious that RF classifier gives the highest accuracy of 98.83% while KNN results are the lowest in accuracy. However, they are all do a great job!

Table 3shows the results of combining different classifiers which are SVM, KNN, BN and RF, we calculated the Combination Rate, Average Roc Area and Accuracy for the easy password (**kicsikutyatarka**) as shown.

Table 3: Results of applying different combinations of classifiers with the combination rules Average of Probabilities (AoP) and Major Voting (MV) on the easy password "kicsikutyatarka"

Method	Combination Rule	Average Roc Area (%)	Accuracy (%)
SVM-KNN-BN	AoP	99.9	97.50
SVM-KNN-BN-RF	AoP	100	**98.66**
SVM-KNN-BN-RF	MV	99.3	**98.66**

Accuracy has increased slightly to 98.66% after combining the four classifiers all together with either one of the combination rules (AoP or MV). Major Voting has not been used in the case of combining the three classifiers (SVM-KNN-BN) as it always gives the same result that is the highest accuracy among the three classifiers when used individually as presented before in Table 2. Results are higher than some individual classifiers such as KNN but RF classifier is still accomplishing the highest Accuracy.

Table 4 shows the results of True Prediction Rate, False Prediction Rate, Average Roc Area and Accuracy when we used different classifiers such as SVM, KNN, BN and RF, We calculated the Combination Rate, Average Roc Area and Accuracy on the strong password **(.tie5Roanl)** part of the dataset

Table 4: Results of using different classifiers with the strong password ".tie5Roanl"

Method	TP Rate (%)	FP Rate (%)	Average Roc Area (%)	Accuracy (%)
RF	97.5	0.3	100	**97.50**
SVM	95.2	0.6	99.0	95.16
BN	98.2	0.5	99.8	95.16
KNN (k=8)	98.0	1.2	93.8	88.98

We can notice an overall decrease in the accuracy level for all classifiers when used with the strong password. This is expected because users tend to spend more and unexpected time usual with such diverse set of letters (symbols, digits, small and capital letters) in the strong password. RF tree algorithm is still at the top of achieved accuracy while KNN was again the lowest level of accuracy.

The results of our proposed methods were compared to others obtained by S. Krishnamoorthy et al. [10] where the same dataset was used in their work. Comparison is shown in Table 5.

Table 5 Shows The difference between our results and S. Krishnamoorthy et al. [1] results when using classifiers such as RF, SVM{Normal} and SVM(RBF) in term of Accuracy on the Strong password **(.tie5Roanl)**

Table 5: Comparison between our results and S. Krishnamoorthy et al. [1] using different classifiers: RF and SVM with two kernels (Normal and RBF).

Method	Proposed System Accuracy (%)	Krishnamoorthy's Accuracy (%)
RF	***97.50***	***98.44***
SVM (Normal)	95.16	97.40
SVM (RBF)	94.62	97.27

Table 6 shows the results of combining different classifiers which are SVM, KNN, BN and RF, We calculated the Combination Rate, Average Roc Area and Accuracy on the Strong password **(.tie5Roanl)** as shown in Table 6.

Table 6: Results of applying different combinations of classifiers with the combination rules Average of Probabilities (AoP) and Major Voting (MV) on the strong password ".tie5Roan""

Method	Combination Rule	Average Roc Area (%)	Accuracy (%)
SVM-KNN-BN	AoP	99.8	95.83
SVM-KNN-BN-RF	AoP	98.0	***98.56***
SVM-KNN-BN-RF	MV	97.7	95.83

We can see that we have achieved a higher accuracy rate of 98.56% when using a combination of the four classifiers and Average of Probabilities than Result of Krishnamoorthy dataset which was (**98.44)** when they used RF as shown in Table 7.

Table 7:Comparison between the highest accuracy Result on the strong password between the proposed system and the “Krishnamoorthy” dataset

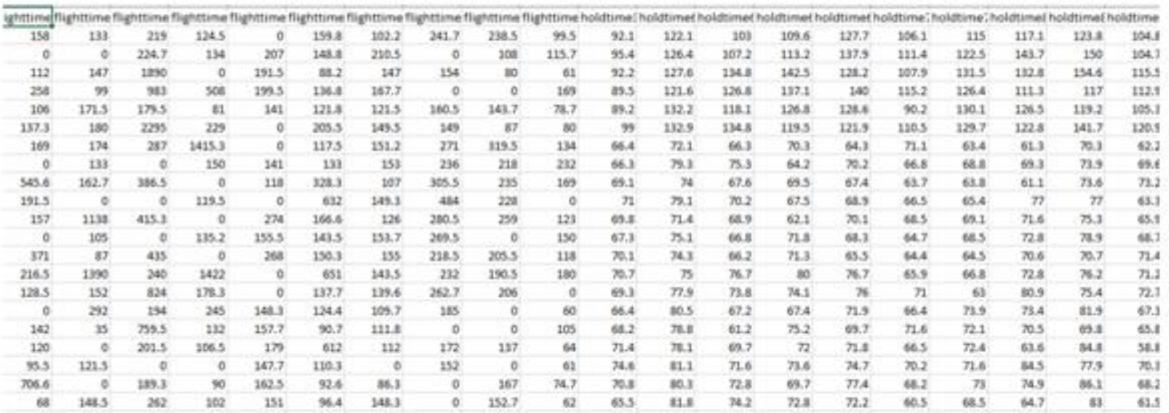
	Method	Highest Accuracy
Proposed System	SVM-KNN-BN-RF	**98.56**
Krishnamoorthy	RF	**98.44**

Free-text Dataset

Data Preprocessing:

Because the data provided in the dataset is very simple and in its raw format, we had to go through a lot of preprocessing actions to prepare the features.

1. Breaking each user’s data into parts from which we can generate equal-sized samples for all users (400 chunk).
2. Calculating hold time, flight time for each chunk.
3. Repeated presses of the same key before being released were neglected at all.
4. There were 79 different keys used by all users, so we selected the most prominent features as shown in figure 8 (Keys which have values in all users).



ighttime	flighttime	flighttime	flighttime	flighttime	flighttime	flighttime	flighttime	flighttime	flighttime	holdtime	holdtime	holdtime	holdtime	holdtime	holdtime	holdtime	holdtime	holdtime	holdtime
158	133	219	124.5	0	159.8	102.2	241.7	238.5	99.5	92.1	122.1	103	109.6	127.7	106.1	115	117.1	123.8	104.8
0	0	224.7	134	207	148.8	210.5	0	108	115.7	95.4	126.4	107.2	113.2	137.9	111.4	122.5	143.7	150	104.7
112	147	1890	0	191.5	88.2	147	154	80	61	92.2	127.6	134.8	142.5	128.2	107.9	131.5	132.8	154.6	115.5
258	99	983	508	199.5	136.8	167.7	0	0	169	89.5	121.6	126.8	137.1	140	115.2	126.4	111.3	117	112.9
106	171.5	179.5	81	141	121.8	121.5	160.5	143.7	78.7	89.2	132.2	118.1	126.8	128.6	90.2	130.1	126.5	119.2	105.1
137.3	180	2295	229	0	205.5	149.5	149	87	80	99	132.9	134.8	119.5	121.9	110.5	129.7	122.8	141.7	120.9
169	174	287	1415.3	0	117.5	151.2	271	319.5	134	66.4	72.1	66.3	70.3	64.3	71.1	63.4	61.3	70.3	62.2
0	133	0	150	141	133	153	236	218	232	66.3	79.3	75.3	64.2	70.2	66.8	68.8	69.3	73.9	69.6
545.6	162.7	386.5	0	118	328.3	107	305.5	235	169	69.1	74	67.6	69.5	67.4	63.7	63.8	61.1	73.6	73.2
191.5	0	0	119.5	0	632	149.3	484	228	0	71	79.1	70.2	67.5	68.9	66.5	65.4	77	77	63.1
157	1138	415.3	0	274	166.6	126	280.5	259	123	69.8	71.4	68.9	62.1	70.1	68.5	69.1	71.6	75.3	65.9
0	105	0	135.2	155.5	143.5	153.7	269.5	0	150	67.3	75.1	66.8	71.8	68.3	64.7	68.5	72.8	78.9	68.7
371	87	435	0	268	150.3	155	218.5	205.5	118	70.1	74.3	66.2	71.3	65.5	64.4	64.5	70.6	70.7	71.4
216.5	1390	240	1422	0	651	143.5	232	190.5	180	70.7	75	76.7	80	76.7	65.9	66.8	72.8	76.2	71.2
128.5	152	824	178.3	0	137.7	139.6	262.7	206	0	69.3	77.9	73.8	74.1	76	71	63	80.9	75.4	72.7
0	292	194	245	148.3	124.4	109.7	185	0	60	66.4	80.5	67.2	67.4	71.9	66.4	73.9	73.4	81.9	67.3
142	35	759.5	132	157.7	90.7	111.8	0	0	105	68.2	78.8	61.2	75.2	69.7	71.6	72.1	70.5	69.8	65.8
120	0	201.5	106.5	179	612	112	172	137	64	71.4	78.1	69.7	72	71.8	66.5	72.4	63.6	84.8	58.8
95.5	121.5	0	0	147.7	110.3	0	152	0	61	74.6	81.1	71.6	73.6	74.7	70.2	71.6	84.5	77.9	70.1
706.6	0	189.3	90	162.5	92.6	86.3	0	167	74.7	70.8	80.3	72.8	69.7	77.4	68.2	73	74.9	86.1	68.2
68	148.5	262	102	151	96.4	148.3	0	152.7	62	65.5	81.8	74.2	72.8	72.2	60.5	68.5	64.7	83	61.5

Fig 8: the most prominent features

A total number of 21 features (11 hold-time, 10 flight-time) were selected and used later in the classification process.

The dataset instances is divided by the percentages 40% for training and 60% for testing.The selected features engineered from the free-text dataset were fed into four different classifiers used before: RF, SVMs, BN, and KNN. We have also attempted combining different classifiers together with different combination rules: Average of Probabilities (AoP) and Major Voting (MV).

Table 8shows the results of True Prediction Rate, False Prediction Rate,Average Roc Area and Accuracy when we used different classifiers such as SVM, KNN, BN and RF, we calculated the Combination Rate, Average Roc Area and Accuracy on the free-text dataset.

Table 8: Results of using different classifiers with the free-text dataset [11]

Method	TP Rate (%)	FP Rate (%)	Average Roc Area (%)	Accuracy (%)
RF	87.6	0.7	99	***87.58***
BN	78.4	1.2	98.1	78.43
KNN	65.0	2.0	82.1	65.36
SVM	59.0	2.2	95.4	59.48

We can see that the accuracy level has decreased than it was with the fixed-text data. That should be expected because of the large variation of keys and features found in this kind of dataset.RF classifier is still having the highest accuracy level, while SVM here is the lowest one.

Table 9 shows the results of combining different classifiers which are SVM, KNN, BN and RF, we calculated the Combination Rule, Average Roc Area and Accuracy on the free-text dataset .

Table 9: Results of applying different combinations of classifiers with the combination rules Average of Probabilities (AoP) and Major Voting (MV) on the free-text dataset [11]

Method	Combination Rule	Average Roc Area (%)	Accuracy (%)
SVM-KNN-BN	AoP	98.3	73.86
SVM-KNN-BN-RF	AoP	97.8	81.05
SVM-KNN-BN-RF	MV	90.3	***81.70***

Despite having the highest achieved accuracy (81.70 %) after combining the classifiers in Table 9 less than before the combination, it is still higher than using 3 individual classifiers results in Table 8 The Highest Accuracy Rate was 87.58 using Random Forest classifier

We did not make a comparison table for the free-text dataset results because it is a new dataset and no published work done on it up till now.

Conclusion:

Keystroke biometric systems have two main scenarios which are: fixed-text, in which the user types a prefix text like a predefined password, and free-text in which the user is able to write any thing like writing an email or any free sentences.

The fixed dataset obviously accomplished accuracy higher than the free-text, actually we can not depend on fixed-text all the time even it has higher accuracy.This is because we need to be sure that the user is the same person during the session in which he is using the system not on the login time only.

We applied four classifiers which are BN classifier, SVM, KNN, and RF on both fixed dataset that is “The MOBIKEY Keystroke Dynamics Password Dataset” and free dataset that is “The Politehnica University Timisoara keystroke dataset”. Both of the fixed-text and free-text had the highest accuracy with RF classifier.

We have combined these classifiers in order to find the best results. Combining classifiers raised the accuracy of the strong password in the fixed-text dataset only. On the other hand, It decreased the accuracy on the easy password in the fixed -text dataset and the free-text dataset.

References

[1] J. Bonneau, C. Herley, P. C. V. Oorschot and F. Stajano, "Passwords and the evolution of imperfect authentication," in *Communications of the ACM 58, no. 7*, 2015.

[2] Shekhawat, Kirty and D. P. Bhatt, "Recent Advances and Applications of Keystroke Dynamics," in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, 2019.

[3] Ahmad, Nasir, A. Szymkowiak and P. Campbell, "Keystroke dynamics in the pre-touchscreen era.," in *Frontiers in human neuroscience 7*, 2013.

[4] Gedikli, A. Melih and M. Ö. Efe, "A simple authentication method with multilayer feedforward neural network using keystroke dynamics," in *Mediterranean Conference on Pattern Recognition and Artificial Intelligence*, 2019.

[5] Alsultan, Arwa and K. Warwick, "Keystroke dynamics authentication: a survey of free-text methods," in *International Journal of Computer Science Issues (IJCSI) 10, no. 4*, 2013.

[6] Antal, Margit and L. Nemes, "The mobikey keystroke dynamics password database: Benchmark results," in *Computer Science On-line Conference, 2013*

[7] Iapa, Augustin-Catalin and V.-I. Cretu., "Shared dataset for Free-Text Keystroke Dynamics Authentication Algorithms," in *Preprints*, 2021.

[8] Martin, Alvin, G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, "The DET curve in assessment of detection task performance," in *National Inst of Standards and Technology Gaithersburg MD*, 1997.

[9] Mondal, Soumik, P. Bours and S. S. Idrus, "Complexity measurement of a password for keystroke dynamics: Preliminary study.," in *Proceedings of the 6th International Conference on Security of Information and Networks*, 2013 .

[10] Hassan, S. I., M. M. Selim and H. H. Zayed, "User Authentication with Adaptive Keystroke Dynamics," in *International Journal of Computer Science Issues (IJCSI) 10, no. 4*, 2013.

[11] Cockell, Robert and B. Halak, "On the design and analysis of a biometric authentication system using keystroke dynamics," in *Cryptography 4, no. 2*, 2020.

[12] S. Krishnamoorthy, L. Rueda, S. Saad and H. Elmiligi, "Identification of user behavioral biometrics for authentication using keystroke dynamics and machine learning," in *Proceedings of the 2018 2nd International Conference on Biometric Engineering and Applications*, 2018.

[13] Jianwei Li, Han-Chih Chang and Mark Stamp, "Free-Text Keystroke Dynamics for User Authentication" in arXiv:2107.07009v1 [cs.LG] 1 Jul 2021

[14] C. Iapa and V. -I. Cretu, "Modified Distance Metric That Generates Better Performance For The Authentication Algorithm Based On Free-Text Keystroke Dynamics," IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), 2021

[15] Liu, S., Zhu, M. & Yang, Y. A bayesian classifier learning algorithm based on Optimization Model. *Mathematical Problems in Engineering*, pp.1–9, 2013.

[16] Escobar Grisales, D., Vásquez-Correa, J. C., Vargas-Bonilla, J. F., & Orozco-Arroyave, J. R. Identity verification in virtual education using biometric analysis based on keystroke dynamics. TecnoLógicas, 23(47), 197–211, 2020.

[17] Salama, K.M. & Freitas, A.A. Learning bayesian network classifiers using ant colony optimization. *Swarm Intelligence*, 7(2-3), pp.229–254, 2013.

[18] Gandhi, R. *SVM - introduction to machine learning algorithms*. Medium. From https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47, 2018.

[19] Allibhai, E. *Building a K-nearest-neighbors (K-nn) model with Scikit-Learn*. Medium. From https://towardsdatascience.com/building-a-k-nearest-neighbors-k-nn-model-with-scikit-learn-51209555453a, 2018.

[20] Brownlee, J. ROC curves and precision-recall curves. Available at: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/ , 2020.

[21] Amin, Md Ashraful & Yan, Hong. High speed detection of retinal blood vessels in fundus image using phase congruency. Soft Comput. 15. 1217-1230. 10.1007/s00500-010-0574-2, 2011