Re-evaluating Keystroke Dynamics for Continuous Authentication

Dilshan Senarath
Dept. of Computer Sci. and Eng.
University of Moratuwa
Moratuwa, Sri Lanka
dilshan.18@cse.mrt.ac.lk

Sanuja Tharinda
Dept. of Computer Sci. and Eng.
University of Moratuwa
Moratuwa, Sri Lanka
sanuja.18@cse.mrt.ac.lk

Maduka Vishvajith
Dept. of Computer Sci. and Eng.
University of Moratuwa
Moratuwa, Sri Lanka
maduka.18@cse.mrt.ac.lk

Sanka Rasnayaka
School of Computing
National University of Singapore
Kent Ridge, Singapore
sanka@nus.edu.sg

Sandareka Wickramanayake
Dept. of Computer Sci. and Eng.
University of Moratuwa
Moratuwa, Sri Lanka
sandarekaw@cse.mrt.ac.lk

Dulani Meedeniya
Dept. of Computer Sci. and Eng.
University of Moratuwa
Moratuwa, Sri Lanka
dulanim@cse.mrt.ac.lk

2023 3rd International Conference on Advanced Research in Computing (ICARC) | 979-8-3503-4737-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICARC57651.2023.10145743

***Abstract*—Today we use smartphones for banking, shopping, and monitoring our health. These applications store sensitive data in the smartphone, making reliable authentication a crucial element in mobile devices. However, current mobile authentication systems such as pin codes, passwords, pattern locks, fingerprints, and face IDs have security vulnerabilities as they are one-time authentication systems. In contrast, behavioural biometric-based Continuous Authentication (CA) focuses on continuously authenticating the user while using the device. Among behavioural biometrics, keystroke dynamics is an efficient and well-researched behavioural biometric. Keystrokes dynamics refers to the unique typing patterns of the user. However, despite many frameworks proposed for mobile CA using keystroke dynamics, they are only evaluated for their discriminative power using traditional metrics such as Equal Error Rate. However, these evaluations are unable to capture temporal performance. Hence, in this work, we evaluate the state-of-the-art keystroke dynamics systems using continuous evaluation metrics. Our analysis indicates that these systems perform differently with traditional and continuous evaluation metrics, stressing the importance of evaluating CA systems using traditional and continuous evaluation metrics for a holistic assessment. Further, our analysis highlights how different models perform differently when evaluated with CA evaluation metrics highlighting the need to carefully select appropriate evaluation schemes based on the requirements such as security and usability.**

***Keywords*—Mobile Continuous Authentication, Mobile Behavioural Biometric Authentication, Keystroke Dynamics, Continuous Evaluation**

I. Introduction

Today, we heavily rely on smartphones to carry out our day-to-day lives, from keeping in touch with others to banking and monitoring our health. While this increases people's efficiency, the sensitive data stored by these applications on the smartphone introduces many additional security vulnerabilities. Hence, reliable authentication has become a critical aspect of mobile phone development.

Current mobile authentication systems are primarily physiological biometrics-based, such as fingerprint and face, or knowledge-based, such as pin codes, passwords, and pattern locks. However, the knowledge-based authentication systems are vulnerable to guessing, sniffing, social engineering attacks, and shoulder surfing attacks [1]. Further, biometric authentication methods are prone to vulnerabilities such as presentation attacks (spoofing), replay attacks, and data simulation [2]. Furthermore, on average, 2.9% of the total usage time of a user is spent for knowledge-based authentication [3]. These vulnerabilities and problems motivated continuous authentication systems.

In contrast to traditional session-based authentication, CA monitors and verifies the user continuously without asking the user to carry out any specific authentication steps. Hence, CA solves most problems in traditional authentication methods. Continuous authentication is achieved through behavioural biometrics, which can be defined as particular activities unique to each user, such as typing, touch gestures, gait, and device holding. Among those, keystroke dynamics is one of the most prevalent behavioural biometrics used in CA systems.

The essence of keystroke dynamics is to analyze the user's typing behaviour, primarily considering the time between consecutive key events and the typed keys to identify the user uniquely. There are two types of keystroke dynamics: fixed text and free text [4]. In the fixed text keystroke dynamics scenario, users should type the same pre-defined text in both the enrollment and verification phases. In contrast, in the free text keystroke dynamics scenario, users can type any text in both the enrollment and verification phases. Among those, the most challenging method is free text analysis.

According to previous work [3], [4], [5], [6], [7] keystroke dynamics were evaluated with traditional evaluation metrics such as Accuracy, Precision, F1 Score, Recall, Equal Error Rate (EER), False Acceptance Rate (FAR), False Rejection Rate (FRR), and Area Under the Receiver Operating Char-

acteristic (ROC) Curve (AUC). However, to the best of our knowledge, no study has been conducted to evaluate the existing keystroke dynamics-based CA frameworks using CA evaluation metrics to evaluate their behaviour through time.

In this paper, we evaluate a set of traditional Machine Learning (ML) models and state-of-the-art CA frameworks based on keystroke dynamics by using the continuous authentication evaluation metrics (time-based evaluation metrics) proposed for computer continuous authentication systems in [8], and [9] (See Fig. 1). The evaluation metrics used are Usability, Time to Correct Reject (TCR), False Reject Worse Interval (FRWI), and False Accept Worse Interval (FAWI). The results indicate that under the two evaluation schemes, traditional and continuous, different CA models produce the best results. Hence, our research emphasizes the importance of evaluating CA models utilizing continuous evaluation metrics.

II. Related Work

In the field of keystroke dynamics-based authentication systems, initial work was carried out using traditional ML classifiers such as Random Forest (RF), k-nearest neighbour (kNN), and Support Vector Machine (SVM) [6], [10]. The feature extraction process required significant effort to develop these traditional models. With the emergence of large public datasets such as AaltoDB [11], deep learning approaches were introduced for continuous authentication [3], [5], [7]. These deep learning approaches include Multi-Layer Perceptron, Recurrent Neural Networks, Long Short Term Memory (LSTM), and most recently, Transformers. Raw data usage rather than following a feature extraction process also became popular with these deep learning approaches.

De-Marcos et al. in [6] compared seven ML classifiers for continuous authentication. The classifiers they compared are Random Forest Classifier (RFC), Extra Trees Classifier (ETC), Gradient Boosting Classifier (GBC), kNN, SVM, Classification and Regression Tree (CART), and Naive Bayes (NB). Their work investigated the discriminative power for the smallest number of key events (two consecutive keys). The highest results were achieved with ensemble models. From those also, the best result was obtained from GBC. In both [5] and [3], multimodal architectures have been followed by incorporating a separate LSTM model for a specific touch task or background sensor data. The final score in both these systems was calculated with a score-level fusion.

Stragapede et al. in [5] introduce a novel dataset named BehavePassDB. HuMINet proposed in [3] has been trained using HuMIdb [12] dataset. The TypeNet introduced in [7] is based on LSTM. They trained TypeNet using three loss functions, and the best results were achieved with the triplet loss. They obtained results for different lengths of keystroke input sequences and a different number of enrollment sequences. Further, the result-obtaining process was done for identification as well as authentication. A transformer-based system TypeFormer was proposed in [4], achieving the best EER for keystroke-based authentication systems.

Finally, while computer CA systems have been evaluated with CA evaluation metrics [8], [9], to the best of our knowledge, no study has evaluated mobile CA systems with CA evaluation metrics.

III. Experimental Protocol

This paper evaluates the state-of-the-art CA systems using EER, a traditional bio-metric evaluation metric, and a set of continuous authentication evaluation metrics adopted from existing work. First, based on our literature review, we identify a set of datasets and models for our analysis. Out of those datasets, we select AaltoDB [11] for our study. Next, we use a set of traditional ML and state-of-the-art deep learning models to conduct our analysis. Following [6], we chose RFC [13], kNN [14], and GBC [15] as the traditional ML models, whereas HuMINet, TypeNet, and TypeFormer were selected as the state-of-the-art deep learning models. Since the public code bases for selected deep learning models are unavailable, we implement them ourselves for this study. Finally, the above models are evaluated using EER and continuous evaluation metrics [8], [9]. Fig. 1 shows the overview of our methodology.

This paper addresses the following research questions:

1) RQ1: How does the state-of-the-art CA models perform with respect to the CA evaluation metrics?
2) RQ2: What are the impacts of sequence length and the number of enrollment sequences on CA evaluation?

A. Dataset

We conduct experiments using the AaltoDB dataset developed by Palin et al. in [11]. It contains only keystroke data which was collected from 260,000 participants. The data collection process has been conducted through a mobile web application in a completely unsupervised way. Participants were required to type a random English sentence picked from a pool of 1,525 sentences. In the data collection process, the participants had to memorize the selected sentence and type it quickly and accurately. Each sentence contains, at most, 70 characters. However, when participants were typing the sentences, they could type more than 70 characters due to correcting mistakes and adding new characters. Therefore, each session may contain more than 70 samples. While typing the sentence, the key press timestamp, key release timestamp, and key code were recorded as raw data. But the AaltoDB dataset contains only 23% of participants who completed at least 15 sessions [11]. Following [7], we filter out the participants who had at least 15 sessions during the preprocessing process. After that, the first 15 sessions from the filtered participants are selected as the preprocessed dataset. Finally, we select 31400 user data out of all preprocessed data. From that subset, we use the first 30000 participants' data for training, the next 1000 participants' data for testing, and the next 400 participants' data for validation. We collect five features from the dataset following [4]. Those features are *hold latency, inter-key latency, press latency, release latency, key pressed* (See Fig. 2). In addition to these five features,

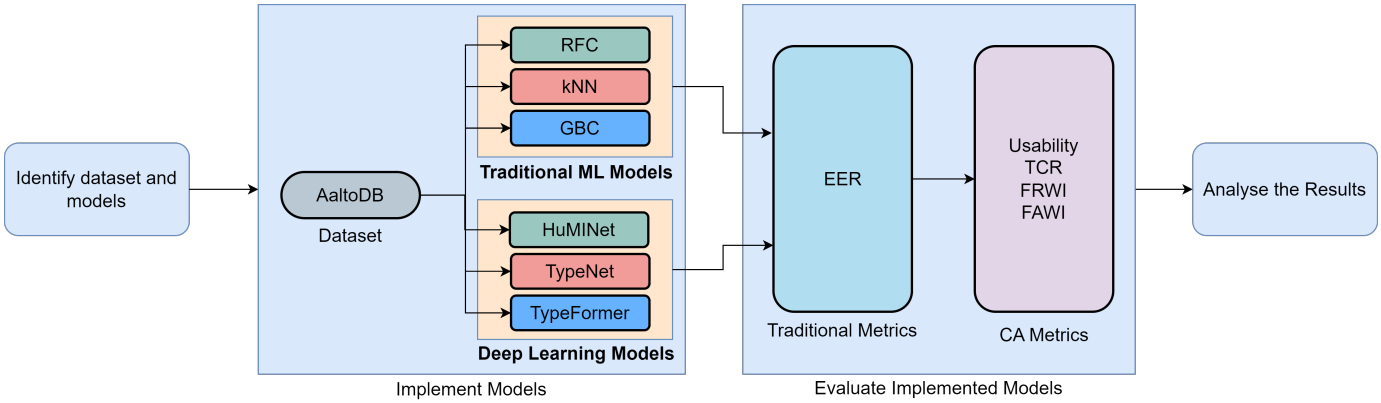

Fig. 1: The overview of the process followed in this study.

we feed another feature,*next key*, which is used in [6] to the traditional ML models.

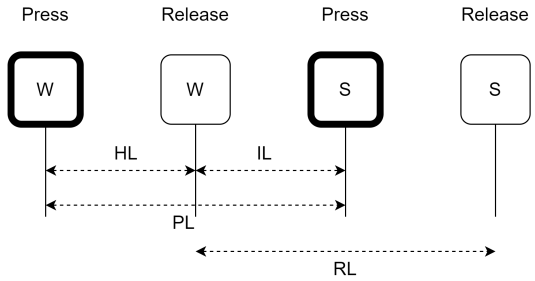

Fig. 2: Example of the keystroke features extracted from the AaltoDB [11]. HL: Hold Latency; IL: Inter-key Latency; PL: Press Latency; RL: Release Latency; ASCII: Key Pressed.

B. Implementation Details

In this study, we investigate three ML models for CA: RFC, kNN, and GBC, which are built using the six features mentioned in the previous section. In RFC implementation, as hyper-parameters we use n_estimators as 100 and criterion as gini. In kNN implementation, as hyper-parameters we use the number of neighbors as 10 and weights as uniform. In GBC implementation, as hyper-parameters we use 2000 estimators and validation_fraction as 0.2.

We implement HuMINet and TypeNet using TensorFlow-Keras, whereas TypeFormer is implemented using Pytorch. All three models were trained using Adam Optimizer. We set the learning rate to 0.0001, 0.001, and 0.001 for HuMINet, TypeNet, and TypeFormer, respectively. The batch size for HuMINet and TypeNet is 512, whereas it is 1024 for TypeFormer. Further, HuMiNet, TypeNet, and TypeFormer are trained for 10, 200, and 1000 epochs, respectively.

The original HuMINet consists of six LSTM-based models, each composed of two LSTM layers because it was proposed to use with multimodal data. However, since we only consider keystroke dynamics in this study, our implementation of HuMINet consists of a single LSTM-based model. For regularization, we use batch normalization layers, a dropout layer with a dropout rate of 0.5, and a recurrent dropout with a rate of 0.2. The model is trained using the triplet loss function with a margin value of 1.0. The sequence length of the input to HuMINet is 70. Raw data is used without a feature extraction following the original work.

For TypeNet, we use the same architecture proposed in [7]. The model contains two LSTM layers with 128 units, tanh activation function, and 0.2 recurrent dropout rate. Both LSTMs have Batch Normalisation layers. A Dropout layer is added between two LSTM layers with a rate of 0.5. The model is trained using three loss functions: cross-entropy loss, contrastive loss with a 1.5 margin, and triplet loss with a 1.5 margin. To analyze the impact of the input sequence, we train TypeNet with different input sequences such as 30, 70, 50, 100, and 150.

When implementing the TypeFormer model, we use the same transformer model proposed in [4]. 20 Gaussian distributions are used in the Gaussian range positional encoder. 10 and 5 heads are used in multi-headed self-attention of the temporal module and the channel module, respectively. 10 temporal layers and one channel layer are also used in the model. The multi-scale keystroke CNN contains three layers with 1, 3, and 5 kernel sizes followed by dropout layers with a rate of 0.1 with ReLU activation. Outer two CNN layers with max-pooling in the temporal and channel modules contain 128 and 32 kernel sizes with dropout layers of 0.5 rate with ReLU activation. The final size of the feature embedding is 64. The sequence length of the input is 50, and the model is trained using the euclidean triplet loss function with a 1.0 margin.

C. Evaluations

In the evaluation process, we use the enrollment-verification method [7] to test and evaluate the different models. Models are trained with a different set of user data and tested with the remaining user data from which also a fraction of each user is used for enrollment, and the remaining data of that user is used for verification (See Fig. 3). For each user, the other testing users are considered as imposters. When evaluating the models, we calculate each metric for every user in the test set

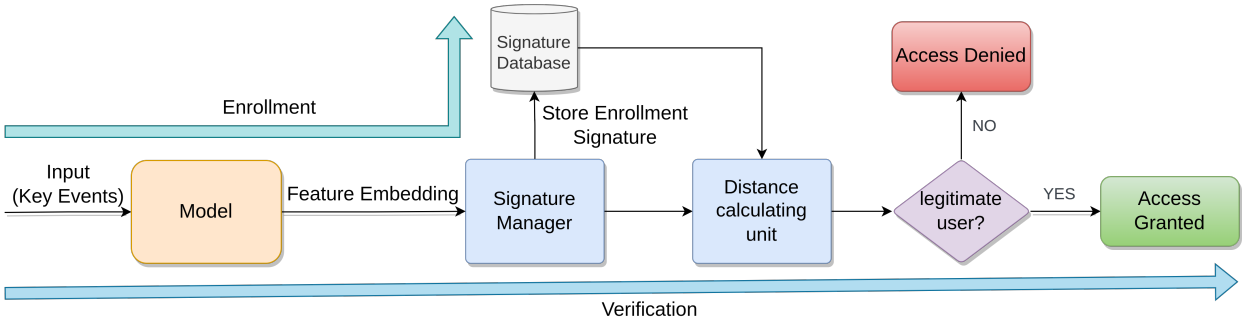

Fig. 3: Enrollment and Verification Process.

and get the average for all users. We evaluate all six systems using EER, Usability, TCR, FAWI, and FRWI. In TypeNet, we use a different number of enrollment sequences (1, 2, 5, 7, 10) when evaluating the model.

IV. Traditional Evaluation

The first evaluation phase covers the evaluations with Equal Error Rate (EER), a traditional evaluation metric.

- Equal Error Rate (EER)
  EER refers to the point where the False Acceptance Rate (the frequency that the CA model grants access to an unauthorized person) and False Rejection Rate (the frequency that the CA model denies access to an authorized person) are equal [16]. EER measures how well a biometric-based authentication model performs: a lower EER indicates a higher accuracy.

TABLE I, 2nd column shows the traditional evaluation results of the CA systems. The best EER value of 6.08 has been achieved by RFC, followed by kNN, TypeNet, GBC, TypeFormer, and HuMINet, respectively.

TABLE II shows the traditional evaluation results of TypeNet considering a different number of keys per sequence and enrollment sequences per subject. In this table, "keys per sequence" refers to the number of key events considered for one sequence of inputs, and "enrollment sequences per subject" refers to the number of sequences that have been used when the legitimate user registered to the system. The best EER value of 6.99 has been achieved with a sequence length of 70 and enrollment sequences per subject of 10. EER decreases as the sequence length increases until the sequence length is 70, and after that, EER starts to increase. Graphs for all the enrollment sequences display this same behaviour. Further, for each sequence length, the EER value decreases with the increasing number of enrollment sequences per subject.

V. Continuous Evaluation

Next, we evaluate all models against the continuous evaluation metrics. This section describes the continuous evaluation metrics used in this analysis and the results.

- Usability (U)
  Usability measures the amount of time the legitimate user has access to the protected resources out of total usage time. Given that a legitimate user spends **T** time on the system in which they were only granted access to the protected resources for **t** amount of time, then the usability **U** for the legitimate user is calculated as in (1).

$$U = \frac{t}{T} \tag{1}$$

  Each time a legitimate user is rejected from the system, they has to re-authenticate which can be inconvenient to the user. Therefore, Inconvenience can be defined as,

$$Inconvenience = 1 - Usability \tag{2}$$

  For an ideal system, the usability should be 1 and the inconvenience should be 0. Therefore, for a better continuous authentication system, the usability should be closer to 1 and the inconvenience should be closer to 0 [8].
- Time to Correct Reject (TCR)
  Time to Correct Reject measures the time interval between the first action (t_{start}) of the imposter and the time instance he is rejected (t_{reject}) by the system. TCR finds how quickly the imposter is rejected by the system as given by (3). TCR is measured in seconds [8].

$$TCR = t_{reject} - t_{start} \tag{3}$$

  TCR should be less than a system-specific time window W. W can be defined as the minimum time taken for an imposter to cause considerable damage. As an example, in [8], W has been defined as W=3 for a linux system, considering the time taken for "rm -rf *" command to be executed in the terminal.
- False Reject Worse Interval (FRWI)
  False Reject Worse Interval measures the longest time interval over which a legitimate user is falsely rejected by the system and labelled as an imposter as in Fig. 4. This is a worst-case analysis of usability [9].

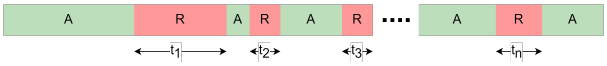

Fig. 4: Time sequence of a legitimate user where A and R refer to Acceptance and Rejection.

TABLE I: Traditional and Continuous Evaluation results of CA systems

Model	Traditional Evaluation	Continuous Evaluation			
	EER	*Usability*	*TCR (s)*	*FRWI (min)*	*FAWI (min)*
RFC	**6.08**	**0.97**	17.95	**0.03**	0.84
kNN	6.90	0.96	18.30	**0.03**	**0.82**
GBC	7.89	0.94	**15.97**	0.06	0.83
HuMINet	22.99	0.85	53.70	0.04	3.15
TypeNet	6.99	0.93	16.16	0.07	1.01
TypeFormer	15.88	0.95	22.58	0.05	2.04

TABLE II: Traditional Evaluation results (EER) of TypeNet

Keys per sequence	enrollment sequences per subject				
	1	*2*	*5*	*7*	*10*
30	15.87	14.41	13.28	13.00	12.55
50	13.22	11.58	10.09	9.54	9.21
70	11.32	9.45	7.70	7.32	**6.99**
100	11.49	9.80	8.15	7.65	7.30
150	11.90	10.31	8.57	8.24	7.90

$$FRWI = max(t_1, t_2, t_3, \ldots, t_n) \tag{4}$$

- False Acceptance Worse Interval (FAWI)
  False Acceptance Worse Interval measures the longest time interval where an imposter is falsely accepted by the system and labelled as a legitimate user as stated in Fig. 5. This is a worst-case analysis of security [9].

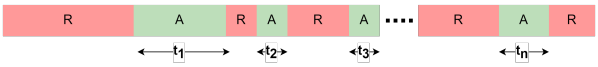

Fig. 5: Time sequence of a imposter where R and A refers as Rejection and Acceptance.

$$FAWI = max(t_1, t_2, t_3, \ldots, t_n) \tag{5}$$

TABLE I shows the continuous evaluation results of the CA systems. TABLE III shows the continuous evaluation results of TypeNet considering a different number of keys per sequence and enrollment sequences per subject.

As shown in TABLE I, the best Usability value of 0.97 has been achieved with RFC. From the other models, Usability results vary from best to worst as kNN, TypeFormer, GBC, TypeNet, and HuMINet respectively. The best TCR value of 15.97s has been achieved with GBC. From the other models, TCR results vary from best to worst as TypeNet, RFC, kNN, TypeFormer, and HuMINet respectively. The best FRWI value of 0.03min has been achieved with RFC and kNN. From the other models, FRWI results vary from best to worst as HuMINet, TypeFormer, GBC and TypeNet respectively. The best FAWI value of 0.82min has been achieved with kNN. From the other models, FAWI results vary from best to worst as GBC, RFC, TypeNet, TypeFormer, and HuMINet respectively.

As illustrated in TABLE III, for TypeNet, the best Usability value of 0.92 has been achieved with the sequence length of 70 (For 5, 7, and 10 number of enrollment sequences), 100 (For 7 and 10 number of enrollment sequences), and 150 (For 10 number of enrollment sequences). The best TCR value of 16.16s has been achieved with a sequence length of 100 (For 10 enrollment sequences). The best FRWI value of 0.07min has been achieved with the sequence length of 70 (For 5, 7, and 10 enrollment sequences) and 100 (For 7 and 10 enrollment sequences). The best FAWI value of 1.01min has been achieved with a sequence length of 70 (For 10 enrollment sequences) and 100 (For 10 enrollment sequences).

With the increasing sequence length, for each different number of enrollment sequences, Usability, TCR, FRWI and FAWI have shown increasing results up to the sequence length of 70. After the sequence length of 70, the results for these metrics decrease (The results are similar for sequence lengths of 70 and 100 in some cases). For each different number of keys per sequence, all CA evaluation metric results have increased with the increase of enrollment sequence per subject.

VI. Discussion

By analyzing the results, we observe that CA evaluation results and traditional evaluation results do not agree on a single best-performing model. Rather, we see different models performing well on different evaluation matrices. The CA evaluation metric Usability has shown the most similar performance to traditional evaluation results.

According to [7], the best performance with respect to traditional evaluation metrics was obtained using a sequence length of 70 and a higher number of enrollment sequences. In our work, we corroborate this result and also highlight that the same sequence length of 70 and a higher number of enrollment sequences have shown better performance for CA evaluation metrics as well. While higher sequence lengths might achieve slightly better performance on some metrics, there is no significant improvement beyond that point (See Fig. 6).

Our analysis highlights that a higher window size and a longer enrollment phase achieve better performance as expected. However, this comes at the cost of usability since the inconvenience to the user is increased. Therefore, a trade-off between the window size and the required security level must be reached.

While traditional evaluation metrics would have us believe that the RFC model outperforms all other approaches, looking at the CA evaluation results highlights a different perspective. While RFC offers the best Usability, GBC provides a better TRC. This highlights the need to use these CA evaluation criteria for evaluating behavioural biometrics for the use of

TABLE III: Continuous Evaluation results of TypeNet

Keys per Sequence	Continuous Evaluation																			
	Usability					*TCR (s)*					*FRWI (min)*					*FAWI (min)*				
	1ES	*2ES*	*5ES*	*7ES*	*10ES*	*1ES*	*2ES*	*5ES*	*7ES*	*10ES*	*1ES*	*2ES*	*5ES*	*7ES*	*10ES*	*1ES*	*2ES*	*5ES*	*7ES*	*10ES*
30	0.46	0.86	0.87	0.87	0.88	19.41	19.00	18.62	18.46	18.39	0.16	0.15	0.13	0.13	0.12	1.53	1.44	1.39	1.38	1.36
50	0.88	0.89	0.91	0.91	0.92	18.29	17.66	17.31	17.12	16.94	0.16	0.11	0.09	0.09	0.09	1.40	1.29	1.22	1.19	1.16
70	0.89	0.91	**0.93**	**0.93**	**0.93**	17.62	17.14	16.60	16.39	16.20	0.11	0.09	**0.07**	**0.07**	**0.07**	1.25	1.16	1.07	1.03	**1.01**
100	0.89	0.91	0.92	**0.93**	**0.93**	18.14	17.15	16.57	16.39	**16.16**	0.11	0.10	0.08	**0.07**	**0.07**	1.29	1.16	1.07	1.04	**1.01**
150	0.88	0.90	0.92	0.92	**0.93**	18.09	17.40	16.95	16.73	16.63	0.12	0.10	0.08	0.08	0.08	1.28	1.18	1.09	1.07	1.05

[ES - Enrollment Sequences per subject]

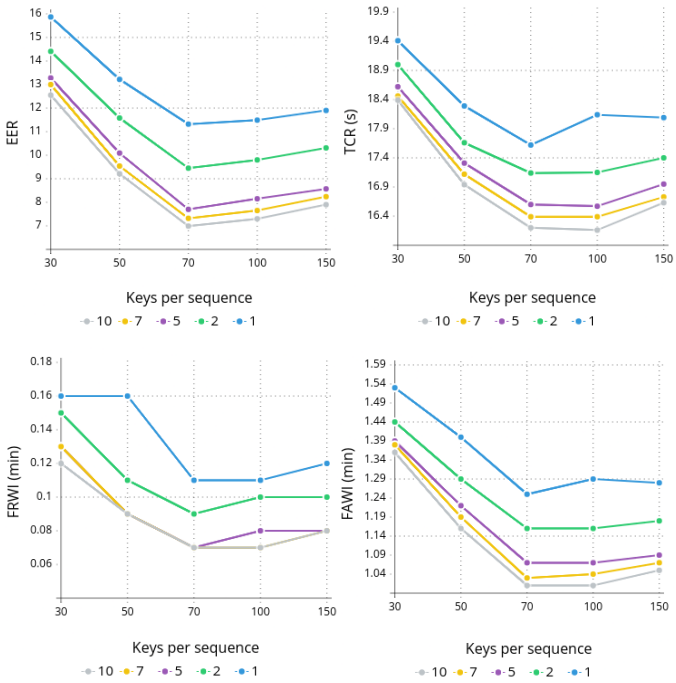

Fig. 6: TypeNet results with different number of enrollment sequences (10, 7, 5, 2, 1) per subject

CA. Based on the application requirement, we can select the appropriate model to prioritize usability over security or vice-versa based on these evaluations.

VII. Conclusion

This study considered existing keystroke dynamics-based authentication models and extended their evaluation processes with continuous evaluation metrics. Our work shows that when utilizing CA evaluation and traditional evaluation metrics to evaluate and rank different models, the two evaluation schemes give different performance rankings.

The analysis carried out highlighted the glaring issue of relying solely on traditional evaluation metrics for continuous authentication applications. While traditional methods are sufficient for static authentication systems, the innate differences in authenticating over time are not captured by these traditional evaluation methods. Therefore, we emphasize the importance of evaluating continuous authentication systems using continuous evaluation metrics. We highlight the trade-off between usability and TCR as well as FRWI and FAWI, similar to the trade-off between FAR and FRR in traditional authentication systems. We believe this study will motivate future research in continuous authentication to carry out more holistic evaluations using relevant metrics for more robust systems.

References

[1] F. Towhidi, A. A. Manaf, S. M. Daud, and A. H. Lashkari, "The knowledge based authentication attacks," in *Proceedings of the International Conference on Security and Management (SAM)*, Las Vegas, USA, 2011, pp. 1–5.

[2] Q. Xiao, "Security issues in biometric authentication," in *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, New York, USA, 2005, pp. 8–13.

[3] G. Stragapede, R. Vera-Rodriguez, R. Tolosana, A. Morales, A. Acien, and G. L. Lan, "Mobile behavioral biometrics for passive authentication," *Pattern Recognition Letters*, vol. 157, pp. 35–41, 2022.

[4] G. Stragapede, P. Delgado-Santos, R. Tolosana, R. Vera-Rodriguez, R. Guest, and A. Morales, "Mobile Keystroke Biometrics Using Transformers," *arXiv preprint arXiv:2207.07596*, 2022.

[5] G. Stragapede, R. Vera-Rodriguez, R. Tolosana, and A. Morales, "BehavePassDB: public database for mobile behavioral biometrics and benchmark evaluation," *Pattern Recognition*, vol. 134, p. 109089, 2023.

[6] L. de Marcos, J.-J. Martínez-Herráiz, J. Junquera-Sánchez, C. Cilleruelo, and C. Pages-Arévalo, "Comparing machine learning classifiers for continuous authentication on mobile devices by keystroke dynamics," *Electronics*, vol. 10, no. 14, p. 1622, 2021.

[7] A. Acien, A. Morales, J. V. Monaco, R. Vera-Rodriguez, and J. Fierrez, "TypeNet: Deep learning keystroke biometrics," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 1, pp. 57–70, 2022.

[8] T. Sim, S. Zhang, R. Janakiraman, and S. Kumar, "Continuous verification using multimodal biometrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, p. 687–700, 2007.

[9] A. Mosenia, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "CABA: Continuous authentication based on BioAura," *IEEE Trans. Comput.*, vol. 66, no. 5, p. 759–772, 2017.

[10] 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*, New York, USA, 2018, p. 50–57.

[11] K. Palin, A. M. Feit, S. Kim, P. O. Kristensson, and A. Oulasvirta, "How do people type on mobile devices? observations from a study with 37,000 volunteers," in *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, Taipei, Taiwan, 2019, pp. 1–12.

[12] A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez, and O. Delgado-Mohatar, "Becaptcha: Behavioral bot detection using touchscreen and mobile sensors benchmarked on humidb," *Engineering Applications of Artificial Intelligence*, vol. 98, p. 104058, 2021.

[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[14] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[15] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[16] J. D. Cook, "Biometric security and hypothesis testing," Sep 2020. [Online]. Available: https://www.johndcook.com/blog/2018/10/31/biometric-security-error/