Robot Mimicry Attack on Keystroke-Dynamics User Identification and Authentication System

Rongyu Yu[1], Burak Kizilkaya[1], Zhen Meng[1], Emma Li[2], Guodong Zhao[1], and Muhammad Imran[1]

***Abstract*— Future robots will be very advanced with high flexibility and accurate control performance. They will have the ability to mimic human behaviours or even perform better, which raises the significant risk of robot attack. In this work, we study the robot mimic attack on the current keystroke-dynamic user authentication system. Specifically, we proposed a robot mimicry attack framework for keystroke-dynamics systems. We collected keyboard logging data and acoustical signal data from real users and extracted the timing pattern of keystrokes to understand victim's behaviour for robot imitation attacks. Furthermore, we develop a deep Q-Network (DQN) algorithm to control the velocity of robot which is one of the key challenges of forging the human typing timing features. We tested and evaluated our approach on the real-life robotic testbed. We presented our results considering user identification and user authentication performance. We achieved a 90.3% user identification accuracy with genuine keyboard logging data samples and 89.6% accuracy with robot-forged data samples. Furthermore, we achieved 11.1%, and 36.6% EER for user authentication performance with zero-effort attack, and robot mimicry attack, respectively.**

I. INTRODUCTION

For network and information security, user identification and authentication are critical. In order to distinguish one user's identity from others, user identification necessitates the collection of data from all users who use or access the system. User authentication, on the other hand, occurs when the system verifies the user's identity, determining whether the user is truly who that person claims to be [1].

Recently, behavioural biometrics-based continuous user authentication has received a lot of attention, and it is expected to be a future trend to increase security level in case of credentials theft and overcome the drawbacks of conventional authentication schemes such as poor memorability for password use [2], potential facial image leakage risks for face recognition [3], and specialised sensor device requirement for fingerprint [4]. One of the well known behavioural biometrics-based continuous user authentication is keystroke dynamics [5], which analyzes the rhythm/patterns of typing considering keystroke events to differentiate between users. Many researchers in keystroke dynamics system field focus on developing and evaluating the verification algorithms pursuing for a high recognition accuracy [6], [7]. The attacks can be classified into three categories according to information the attacker has about the keystroke dynamics of the victim [8], [14], [15]. These are zero-effort attack, statistical attack and imitation attack. Early studies often evaluated the biometrics-based systems by zero-effort attack [9], which is a naive approach assuming the attacker is simply a 'casual' imposter, who does not put any effort to mimic the given victim's behaviour. Imitation attack, on the other hand, can be divided into two categories, namely, manual imitation attack and robot imitation attack [7]. In the scenario of manual imitation attack, the attacker obtains the victims' behavioural biometrics data in some means, then intend to adapt their own behaviour to match that of the victim. On the other hand, in robot imitation attack, the robot is trained to manipulate the attempt, whose process is autonomous with higher efficiency, and more likely to defeat any live detection of defence approach.

In existing literature, the manual imitation attack have been extensively studied [10], [11], [12]. For example, in [11], authors invited a group of 84 participants to perform human imitation attack. Their results show that the equal error rate (EER) increased from 0.24 to 0.63 for a weak password 'serndele' and from 0.2 to 0.42 for a stronger password 'ths.ouR2'. On the other hand, robot imitation attack on keystroke-dynamic system is not exploited in the current literature. To the best of our knowledge, there are only two studies toward touch-based biometric systems. In [14], authors utilized a simple Lego constructed robot to perform a user-specific attack on touch-dynamics based authentication system by mimicking the swiping pattern under the assumption that some user's samples were stolen. The effect of the attack increases the EER by about five times the benchmark in the zero-effort threat model test. On top of this, authors trained a Humanoid Robot Nao to more flexibly imitate the shape of target user's touch gestures in [15], due to the limitation in precision of execution of touch strokes.

Furthermore, in previous related attack designs, the perfect copy of victim's template is assumed to be available to the adversary, which is a strong assumption considering that most of the biometrics-based authentication systems employ powerful actions to secure user's sensitive data.

In this study, we exploit physical robot imitation attack and propose a robot mimicry attack framework for keystroke-dynamics system. We collect keyboard logging data, and acoustical signal data from real users and extracted the timing pattern of keystrokes to generate victim templates for the attack. This makes our design more feasible in real world since the victim template is not readily available. Furthermore, we

[1]Rongyu Yu, Burak Kizilkaya, Zhen Meng, Guodong Zhao and Muhammad Imran are with James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, UK 2658366y@student.gla.ac.uk, b.kizilkaya.1@research.gla.ac.uk, z.meng.1@research.gla.ac.uk, Guodong.Zhao@glasgow.ac.uk, muhammad.imran@glasgow.ac.uk

[2]Emma Li is with the School of Computing Science, University of Glasgow, Glasgow, G12 8RZ, UK liying.li@glasgow.ac.uk

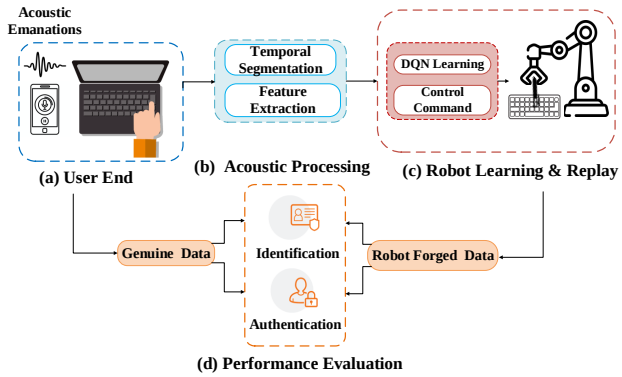

Fig. 1. Robot mimicry attack framework

develop a deep Q-Network (DQN) algorithm to precisely control the velocity of pressing and releasing keys which is one of the key challenges of forging the human typing timing features. The DQN algorithm can be applicable to different keyboard specifications.

The main contributions of the study can be summarized as follows.

- We propose a robot mimicry attack framework for keystroke-dynamics system by training physical robot to forge human typing keystroke timing patterns.
- We collect keyboard logging data and acoustical signal data from 10 participants and extracted keystrokes timing patterns. Extracted patterns are used to generate victim templates to be used in attacks. This way, we relaxed the assumption of template availability and demonstrate how to extract behavioral patterns from acoustical signal data.
- We design a robotic testbed using UR3e robot to perform attack experiments. Furthermore, we propose a DQN algorithm to dynamically control and refine robot's velocity for better attack performance.
- We demonstrate that the robot can learn typing timing behaviours and perform imitation attack with 36% EER.

The rest of the paper is organized as follows. In Section II, we propose a robot mimicry attack framework, where system overview, password timing features, feature extraction approach from the keystroke acoustic signal, and evaluation metrics are discussed. In Section III, we present the robotic testbed implementation along with DQN algorithm design. We discuss the experimental results in Section IV while we conclude the study in Section V.

II. Robot Mimicry Attack Framework

In this section, we propose the robot mimicry attack framework, where a physical robot tries to imitate users' password typing behaviour. Users' behaviour is extracted from keystroke acoustic signal data.

A. System Overview

The proposed attack framework is illustrated in Fig. 1. Mainly, there are four major components, including the (a) user-end, (b) keystroke acoustic signal processing, (c) robot learning & replay and (d) performance evaluation.

At user-end, we assume the victim is being eavesdropped by an audio recording device placed in their vicinity and the password is known to the attacker. The main objective is to learn victim's typing behaviour from acoustical signal. It is also assumed that the environment is designed with limited background noise to investigate the maximum likelihood of a successful attack since it become very hard for an adversary if the loudness of the background noise is comparable to signal. This assumption may be relaxed by introducing filters to filter out the noise from the acoustic signal data.

For acoustic signal data processing, we apply temporal segmentation and feature extraction to understand victim's typing behaviour for a potential attack. More explanation is provided for temporal segmentation and feature extraction in Section II-C.

At robot-end, extracted timing features are used to control the robot by applying trapezoidal trajectory planning. Furthermore, we apply DQN algorithm to dynamically control and refine the robot's velocity for better attack performance. The details of trapezoidal trajectory planning and the proposed DQN algorithm are given in Section III.

For performance evaluation, we evaluate the system in terms of identification and authentication performance. We obtain the genuine data from the user-end and the robot forged data from the robot-end. We use the split keyboard logging to train and test the classifier and detectors for identification and authentication purposes as our baseline. Then, we use the robot forged samples to compare our attack performance. Further details are provided in Section II-D.

B. Password Timing Features

Keystroke dynamics refer to detailed time information for keystroke events. The hold time, T_{h}, up-down time, T_{ud}, and down-down time, T_{dd}, are most frequently used features in the literature. Let's denote key pressing and releasing events as k_1 and k_2, respectively. Then, the keystroke pair (k_1, k_2) has four timings, which are (a) key-down time of k_1: $t_{k_1}^{\mathrm{down}}$, (b) key-up time of k_1: $t_{k_1}^{\mathrm{up}}$, (c) key-down time of k_2: $t_{k_2}^{\mathrm{down}}$, and (d) key-up time of k_2: $t_{k_2}^{\mathrm{up}}$. Then, based on four absolute timings, the feature vector, $\mathbf{V}$ is generated as $\mathbf{V} = (T_{\mathrm{h}}, T_{\mathrm{dd}}, T_{\mathrm{ud}})$. As a result, three relative timing features are given as:

- Hold time: $T_{\mathrm{h}} = t_{k_1}^{\mathrm{up}}$ - $t_{k_1}^{\mathrm{down}}$
- Up-Down time: $T_{\mathrm{ud}} = t_{k_2}^{\mathrm{down}}$ - $t_{k_1}^{\mathrm{up}}$
- Down-Down time: $T_{\mathrm{dd}} = t_{k_2}^{\mathrm{down}}$ - $t_{k_1}^{\mathrm{down}}$

C. Temporal Segmentation and Feature Extraction

In this section, we explain feature extraction from acoustic signal data. Temporal segmentation and short-term energy analysis are conducted to identify keystroke events in acoustic data.

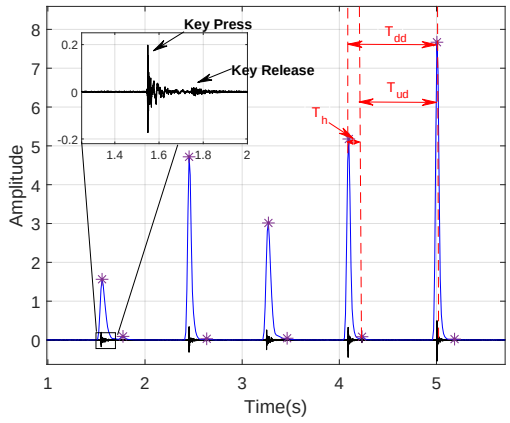

Fig. 2. Keystroke acoustic signal

Acoustic data are shown in Fig. 2 with extracted keystrokes events. As shown in the figure, the acoustic data of keystroke events contain two distinct peaks. The high peaks in every stroke show the key press event, and low peaks show the key release event. To identify the both events, we need to identify high and low peaks in the acoustic data. To achieve this, we apply short-term energy analysis to the keystroke acoustic data in the time domain by computing short-term energy of the signal using a sliding hamming window with 250 ms displacement and 44.1k Hz sampling frequency. As a result, we produce short-term energy curve of the signal as shown in Fig. 2. Then, we identify high and low peaks for every keystroke event by finding local maximums of the signal. Local maximums are used to calculate the hold time, T_{h}, up-down time, T_{ud}, and down-down time, T_{dd}, for our analysis. With this approach, we estimated the T_{h}, T_{ud}, and T_{dd} as follows.

- T_{h}: the time difference between two energy peaks of the same keystroke event
- T_{ud}: the time difference between last peak of the current keystroke and the first peak of the next keystroke.
- T_{dd}: the time difference between the first peaks of two consecutive keystroke events, i.e., $T_{\mathrm{h}} + T_{\mathrm{ud}}$.

Please also note that keyboard specification is an important factor, affecting the the produced keystroke sound. In the literature, T_{h} is generally assumed to be constant (e.g., 100ms) for all keys. However, T_{h} cannot be constant for every type of keyboard. For example, standard laptop keyboard keys are close ($< 2mm$) to the keyboard plate which imposes short key travel time. On the other hand, mechanical keyboard has larger key travel. As a result, constant T_{h} assumption may result unreliable performance evaluation. Therefore, we extract all the features, including T_{h}, from the acoustic data instead of assuming constant value for the hold time.

D. Evaluation Criteria

In this study, the proposed system is evaluated considering user identification and user authentication.

For user identification, we utilize the Gradient-boosting decision tree classifier (GBDT). User identification accuracy is used to evaluate the system. Labelled genuine keyboard logging dataset is used to train the classifier. Then, identification accuracy is investigated using both genuine user data and the robot's forged data samples.

For user authentication, we use the Scaled Manhattan Distance Detector since it achieves the best average EER of 9% as discussed in [14]. We first test the authentication system with the zero-effort attack as baseline, where genuine user data samples are used to test the system. Furthermore, we test the system with robot imitation attack, where both genuine user samples and robot forged data are used. The main objective is to show how the system degrades under the proposed robot mimicry attack. We use the Receiver Operating Curve (ROC) to show the performance of the detector. Furthermore, we use the well-known EER threshold metric by adopting macro-averaging approach considering each class with the same contributing weight to evaluate the system performance under zero-effort and robot imitation attacks.

E. Attack Design Limitation

The core purpose of this work is to pursue the simplicity and feasibility of robotic attack to keystroke dynamics system. We assume the password is known by the imposter. Previous study [16] has been provided the acoustic emanation side-channel attack aiming to identify the text/password the victim is typing. We use a UR3e robotic arm to imitate human hunt & peck typing style using one finger. The participants is likely to behave slower than their natural typing speed in this scenario. Although the defined typing style narrows individual difference, the result shows their personal keystroke behavioural trait is still distinctive enough for recognition purposes.

III. Robotic Testbed Implementation

As seen in Fig. 3, we deploy *UR3e* robotic arm with *Robotiq 2F-85* gripper for our experiments. The robotic arm is equipped with a *stylus* to enable key pressing. A wireless keyboard with scissor-kick keys is utilized to collect the robotic forged data at robot-end. UR3e robot is controlled by control PC via Real-Time Data Exchange (RTDE) interface which provides a platform to synchronize external applications with UR3e controller over a TCP/IP connection. We also use the *ur-rtde* python API to control and receive data from the robot. For performing an attack, we decompose the continuous typing motion into multiple waypoints as pre-defined series of poses $\mathbf{P}(p, o)$, where $p = (x, y, z)$ is the position and $o = (rx, ry, rz)$ is the orientation of pose $\mathbf{P}$ with respect to UR3e base frame. Let's denote the pose that stylus touches the key without pressing as key up pose, P_{up}, and denote the pose that stylus pressing the key as key down pose, P_{down}. Then, we enable robot to press keys by moving from the key up pose to key down pose. The orientation of the robot is defined as a constant, i.e., $o = (0, 3.14, 0)$, so that the stylus vertically downward.

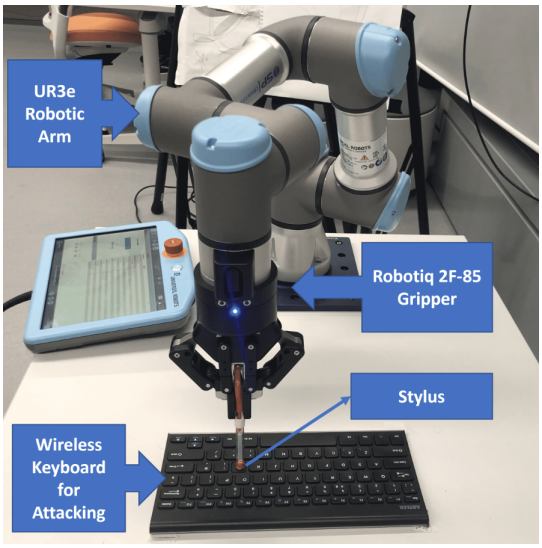

Fig. 3. The robotic arm typing on the keyboard

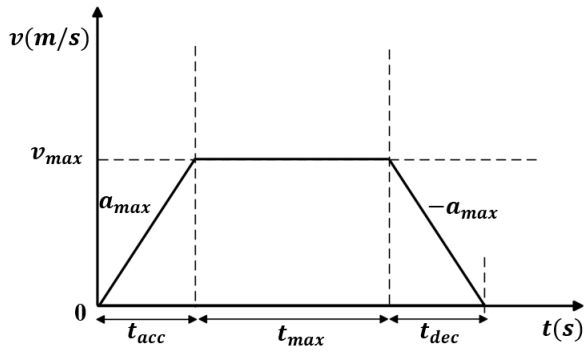

Fig. 4. Trapezoid velocity profile

Only the position is changed to define different poses. Robot movement follows *trapezoidal trajectory planning* for a given maximum velocity and acceleration with zero initial and final velocities.

A. Trapezoidal Trajectory Planning

Robotic arm is controlled considering trapezoidal trajectory planning which follows trapezoidal velocity profile as seen in Fig. 4. It has three main components, namely, acceleration, constant velocity, and deceleration. In acceleration, velocity starts from zero and increases to the maximum value $w_{\max}$ with a defined positive maximum acceleration $a_{\max}$. In constant velocity, velocity is kept at $w_{\max}$ for a specific time or distance. In deceleration, velocity decrease to zero with a deceleration $-a_{\max}$.

According to the characteristic of trapezoidal velocity profile, for a given time period $T = t_{\text{acc}} + t_{\max} + t_{\text{dec}}$ and distance between waypoints s, the maximum velocity can be estimated as

$$v_{\max 1} = \frac{T a_{\max}}{2} + \frac{\sqrt{T^2 a_{\max}^2 - 4s^2}}{2}, \tag{1}$$

$$v_{\max 2} = \frac{T a_{\max}}{2} - \frac{\sqrt{T^2 a_{\max}^2 - 4s^2}}{2}, \tag{2}$$

$$v_{\max} = \begin{cases} \min(v_{\max 1}, v_{\max 2}), & v_{\max 1} \geq 0, v_{\max 2} \geq 0, \\ v_{\max 1}, & v_{\max 1} \geq 0, v_{\max 2} \leq 0, \\ v_{\max 2}, & v_{\max 1} \leq 0, v_{\max 2} \geq 0. \end{cases} \tag{3}$$

B. Velocity Adjustment via DQN Learning

As mentioned in previous sections, T_{dd}, i.e., down-down time, is the addition of T_{ud} and T_{h}, thus our trajectory planning strategy will focus on the path in up-down and hold segments. Let's assume that there is a keystroke pair (k_1,k_2), then the movement can be decomposed into three segments: (1) $p_{k_1}^{\text{up}} \rightarrow p_{k_1}^{\text{down}}$ (2) $p_{k_1}^{\text{down}} \rightarrow p_{k_1}^{\text{up}}$ (3) $p_{k_1}^{\text{up}} \rightarrow p_{k_2}^{\text{up}}$. For segment (a), we firstly make an initial estimation of the $v_{\max}$ in the previous subsection to move between (3) referring to the given time and predefined acceleration. It may even be different for different keys for the same keyboard. The key challenge is the velocity of stylus moving at the segment (1) and (2) which will affect the both T_{h} and T_{ud}. As a result, this will affect the behaviour of typing. Furthermore, we cannot simply define the amount of time spent in segment (1) and (2) as T_{h} since the activation and deactivation points of keys are different for different keyboards

Motivated by these issues, we turn to design velocity adjustment algorithm. Since this is a sequential decision problem, it can be formulated as a Markov Decision Process (MDP).

*1) **States**:* The state, S is target acoustic extracted data which includes the hold time, T_{h}, and up-down time, T_{ud}. For each password input, we define a state chain with 25 timing features.

*2) **Actions**:* Action A includes the value of minimum velocity, v_{lower}, the value of maximum velocity, v_{upper}, and the value of steps between maximum and minimum velocities, v_{step}.

*3) **Reward**:* Given the states and actions, the reward R is defined as the negative of the difference between the genuine T_{h}, and T_{ud} and observed t_{h}, and t_{ud}.

The algorithm for the proposed DQN-based velocity adjustment is given in Algorithm 1. We train the Q-network to find the most appropriate velocity adjustment considering target timings T_{h} or T_{ud}. During the training process, the agent generates actions from the action space and transforms them into robotic control commands. UR3e robot receives the control commands via python API and executes them to press the desired keys. Then, the reward is obtained by measuring the negative of the difference between the genuine and observed timings.

C. Data Collection

We utilized a standard laptop keyboard and a cellphone to record the keyboard logging and acoustic signal data. We collected keystroke data from 10 subjects (5 female, 5 male, average age is 24, all of whom are students at our university).

Every participant typed the password **'.tie5Roanl'** 150 times in six sessions. The password is chosen considering the related studies in the literature [6] which is a typically strong password. Each sample contains 13 key press events and total 37 features including two key presses of 'caps locks' for capital letter of 'r' and a 'Return' key. We collected 120 genuine keyboard logging data to train the both classifier and detector. Furthermore, we collected 30 acoustic signal data for feature extraction.

Algorithm 1 Robot Control Algorithm via DQN Learning

Input parameters: Initialize parameters of Q-network, θ with random weights. Obtained the value of $v_{\max}$, T_{h}, T_{ud}, v_{upper}, v_{lower}, v_{step}, a, α, γ from the calculation of trapezoidal trajectory planning.
for episode = 1, M **do**
  for t=1, T **do**
    With probability ε select a random action a_t
    Otherwise select $a_t = \max_a Q^*(s_t, a; \theta)$
    Execute action a_t by the robotic arm and obverse the reward r_t and state s_{t+1}
    if $done_i$ **then**
      $y_i = r_i$
    else
      $y_i = r_i + \gamma \max_{a'} Q(s_{t+1}, a'; \theta)$
    end if
    Performing a gradient descent step on $(y_j - Q(s_t, a_j; \theta))^2$
  end for
end for

IV. Performance Evaluation

In this section, we evaluate the proposed robot mimicry attack framework in terms of user identification and user authentication[1].

To illustrate the collected data, we present average keyboard logging data, average acoustic extracted data, and average robot forged data in Fig. 5 for User 2 with 30 samples. Acoustic extracted data are similar to the keyboard logging data. The main difference is that keyboard logging data comes from keystroke logs of the keyboard, whereas acoustic extracted data are derived from the keystroke acoustic signal data. As seen from the figure, the acoustic extracted data have larger hold time, T_{h} which is around 0.01-0.1s and have slightly smaller up-down time, T_{ud}, (0.01-0.05) attributing to the short key travel of scissor switch. Furthermore, the time difference between the keyboard logging data and the robot forged data is around 0.3s for typing whole password for User 2.

A. User Identification Performance

To evaluate the performance of user identification, we present accuracy results of the classifier for both genuine keyboard logging data and robot forged data. To provide a baseline, we used the genuine keyboard logging dataset to train and test the model for identification accuracy. Furthermore, we use robot forged data, which are extracted from acoustic data, to show the performance of the proposed robot mimicry attack. As seen in Fig. 6, user identification accuracy is around 90.3% if the genuine keyboard logging data samples are used to test the classifiers. On the other hand, 89.6% user identification accuracy is achieved when the UR3e forged data are used (see Fig. 7). The user identification accuracy results are very close to each other, showing that real user and robot are almost indistinguishable.

[1]We will release our data set and source code along with the publication of this paper.

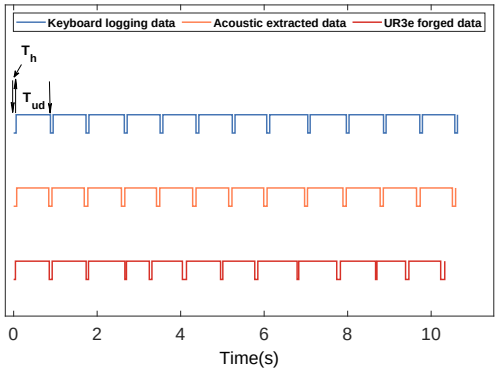

Fig. 5. An example comparison of timing feature pattern for User 2

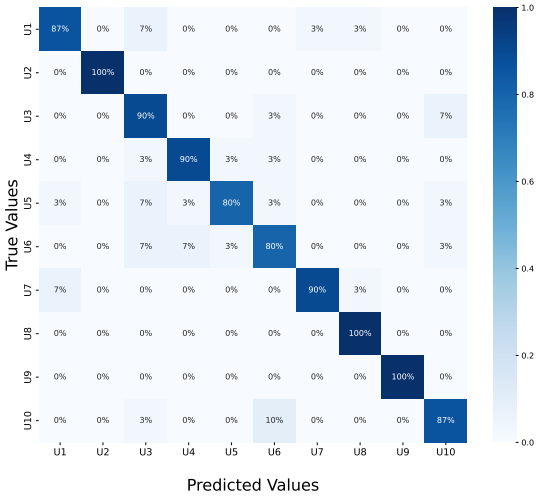

Fig. 6. The confusion matrix using genuine samples

B. User Authentication Performance

To evaluate the user authentication performance, we perform zero-effort and robot imitation attacks. Zero-effort attack is performed as a baseline, where 30 valid users' feature vectors and randomly selected 30 other users' samples are used as genuine and anomalous data, respectively. On the other hand, robot imitation attack is performed by using the robot forged users' samples as the anomalous data to test the system. The main objective is to show whether the

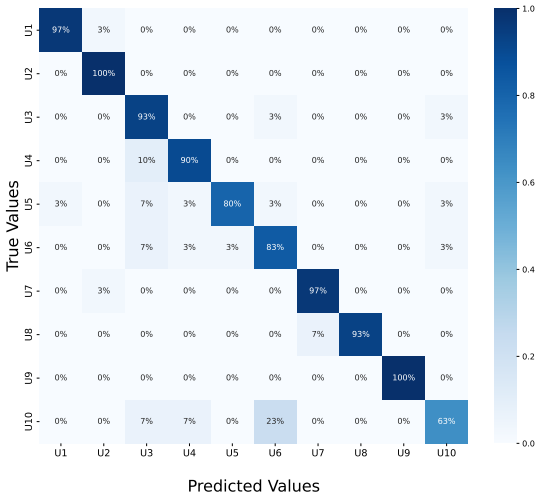

Fig. 7. The confusion matrix using UR3e samples

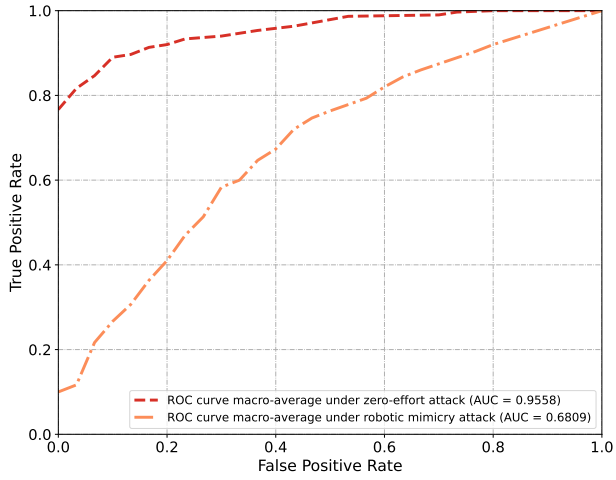

Fig. 8. The ROC curve under different attacks, i.e., zero-knowledge human attack and robot mimicry attack

effectiveness of the system is degraded under the proposed robot mimicry attack. As seen in Fig. 8, we achieve 0.9558 area under curve (AUC) value with zero-effort attack when EER is 11% and 0.6809 with robot mimicry attack when EER is 36%. Comparing with the similar study in literature [11], where AUC is 0.6 with human attack when EER is 24%, our zero-effort attack and robot mimicry attack performances are quite promising and comparable.

V. Conclusions

In this study, we investigated physical robot imitation attack and proposed a robot mimicry attack framework for keystroke-dynamics systems. We collected keyboard logging data and acoustical signal data from real users and extracted the timing pattern of keystrokes to generate victim templates for the robot imitation attack. Furthermore, we developed a DQN algorithm to precisely control the velocity of pressing and releasing keys which is one of the key challenges of forging the human typing timing features. We tested and evaluated our approach on real robotic testbed. We presented our results in terms of user identification and user authentication performance. Considering user identification performance, we achieved 90.3% accuracy with genuine keyboard logging data samples and 89.6% accuracy with robot forged data samples. On the other hand, we achieved 11% EER for user authentication performance with zero-effort attack, whereas 36% EER is achieved with robot imitation attack which are quite promising.

References

[1] D. Todorov, "Mechanics of user identification and authentication: Fundamentals of identity management," Auerbach Publications, pp.4-5, 2007.

[2] B. Ives, K. R. Walsh, and H. Schneider, "The domino effect of password reuse," Communications of the ACM, vol. 47, no. 4, pp. 75–78, 2004.

[3] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis, "Face/off: Preventing privacy leakage from photos in social networks," in *Proceedings of the 22nd ACM SIGSAC Conference on computer and communications security*, 2015, pp. 781–792.

[4] D. Bhattacharyya, R. Ranjan, F. Alisherov, and M. Choi, "Biometric authentication: A review," International Journal of u-and e-Service, Science and Technology, vol. 2, no. 3, pp. 13–28, 2009.

[5] S. P. Banerjee and D. L. Woodard, "Biometric authentication and identification using keystroke dynamics: A survey," Journal of Pattern Recognition Research, vol. 7, no. 1, pp. 116–139, 2012.

[6] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems & Networks*, 2009, pp. 125-134.

[7] R. Giot, M. El-Abed, and C. Rosenberger, "Greyc keystroke: a benchmark for keystroke dynamics biometric systems," in *Proceedings of the IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*, 2009, pp. 1–6.

[8] I. Hazan, O. Margalit, and L. Rokach, "Securing keystroke dynamics from replay attacks," Applied Soft Computing, vol. 85, p. 105798, 2019.

[9] A. J. Mansfield and J. L. Wayman, "Best practices in testing and reporting performance of biometric devices," CESG, Nat. Phys. Lab., Teddington, U.K., NPL Tech. Rep. CMSC 14/02, 2002.

[10] S. Eberz, G. Lovisotto, A. Patane, M. Kwiatkowska, V. Lenders, and I. Martinovic, "When your fitness tracker betrays you: Quantifying the predictability of biometric features across contexts," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 889–905.

[11] T. C. Meng, P. Gupta, and D. Gao, "I can be you: Questioning the use of keystroke dynamics as biometrics," in Proceedings of the Annual Network and Distributed System Security Symposium, 2013, pp. 1–16.

[12] H. Khan, U. Hengartner, and D. Vogel, "Targeted mimicry attacks on touch input based implicit authentication schemes," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 2016, pp. 387–398.

[13] R. Kumar, V. V. Phoha, and A. Jain, "Treadmill attack on gait-based authentication systems," in *Proceedings of the IEEE 7th International Conference on Biometrics Theory, Applications and Systems*, 2015, pp. 1–7.

[14] A. Serwadda and V. V. Phoha, "When kids' toys breach mobile phone security," in *Proceedings of the ACM SIGSAC conference on Computer & Communications Security*, 2013, pp. 599–610.

[15] S. Poudel, A. Serwadda and V. V. Phoha, "On humanoid robots imitating human touch gestures on the smart phone," in *Proceedings of the IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2015, pp. 1-7.

[16] J. Roth, X. Liu, A. Ross, and D. Metaxas, "Investigating the discriminative power of keystroke sound," IEEE Transactions on Information Forensics and Security, vol. 10, no. 2, pp. 333–345, 2014.