2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS) | 979-8-3315-1596-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICPADS63350.2024.00018

Chameleon: An Adaptive System for Overlapping Keystroke Signal Separation and Identification

Jiayi Zhao[1]*, Yongzhi Huang[1]†, Qipeng Xie*§, Weizheng Wang‡, Lu Wang[2]¶, Kaishun Wu*†, *Fellow, IEEE*

*Internet of Things (IoT)/†Data Science and Analytics (DSA) Thrust, Information Hub, Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

‡Computer Science Engineer, City University of Hong Kong, Hong Kong, China

§Hong Kong University of Science and Technology, Hong Kong, China

¶Shenzhen University, China

***Abstract*—Keystroke dynamics has proven to be highly effective, with its applications expanding significantly over the years in areas such as preventing transaction fraud, account takeovers, and identity theft. Key-positioning and feature-learning methods are commonly used to identify keystroke signals. However, the existing methods face challenges in detecting overlapping keystrokes and environmentally changed signals. We propose a solution called Chameleon to address these limitations. Unlike previous signal separation and deep learning methods that are ineffective in keystroke signals and computationally demanding, Chameleon employs a low-computation Ranking Model to separate overlapping keystroke signals. Moreover, our experiments demonstrate that Chameleon separated signals can be recognized with an average accuracy of 92.69%, surpassing the commonly used FastICA method, which only reaches 25% accuracy. To account for environmental changes, we utilize the Fréchet Inception Distance (FID) as a guiding metric for model migration. Additionally, we introduce the Inductive Vector, which enables our key-identifying model to adapt to altered environmental conditions such as environment, phone location, and user variety. The Inductive Vector adjusts the model parameters based on the shift in FID. In scenarios with various phone locations, the Inductive Vector significantly improves recognition accuracy from 61% to 98%, outperforming the best existing keystroke recognition algorithm. In other dynamic environmental conditions, our approach achieves an average accuracy rate of 81.7%, which is at least 1.6 times better than the current state-of-the-art keystroke recognition algorithm.**

I. Introduction

The proliferation of Internet of Things (IoT) devices has seamlessly integrated into various aspects of our lives, encompassing applications in sensing [1]–[7], dietary management [8], industrial automation [5], [9], and more [10]–[19].

Keystroke identification has been devised to address the challenges surrounding mobile technology interaction. In 2004, Asonov [20] demonstrated that a 10-minute analysis of Keystroke Acoustic Signals successfully recovered 96% of the text. This discovery brought to light the potential for decompiling text through Keystroke Acoustic Signals. Consequently, a significant amount of research has emerged in the field of Keystroke Acoustic Signals recognition.

[1]These authors contributed equally to this work.

[2]Corresponding authors.

Existing research in keystroke identification can be categorized into two main groups: Key-positioning and feature-learning methods. Key-positioning studies focus on using the propagation model of acoustics in physical space, with TDOA (Time Difference of Arrival) being the most commonly used method [21]–[23]. However, there is a significant challenge: the keystrokes are not always point-source sounds or Far-field acoustics, which can lead to substantial errors in the TDOA results.

Although much research has achieved remarkable results in text recovery from Keystroke Acoustic Signals, reaching theoretical results in field deployment is challenging. The reason is that the signal information includes overlapping unknown sources of Keystroke Acoustic Signals, the change of venue, and the movement of mobile phone placement introduces different multi-path superpositions the user variety that leads to the shift in tapping habits. All this unfavorable information introduces the original signal features into a new domain space, resulting in a sharp drop in accuracy.

Many studies have proposed techniques for separating these overlapping signals from unknown sources, called blind source separation techniques. These techniques mainly include Independent Component Analysis [24], Sparse Component Analysis [25]. Non-negative Matrix Factorization [26], Bounded Component Analysis [27], blind source separation neural networks [28]. However, these techniques require the signal to satisfy certain assumptions, such as particular distribution. Therefore, it isn't easy to deal with overlapping Keystroke Acoustic Signals, which is unknown distribution.

A promising but challenging solution is to design an adaptive model. Similar to a chameleon, this model will adjust its protective color in response to environmental variations. This model can discern the environmental conditions alterations and transition the initial model to suit the new domain.

However, we are confronted with three technical hurdles to realize this adaptive model. First, we must separate the overlapping Keystroke Acoustic Signal. We need to recognize and separate the overlapping signals without missing features. However, restoring each signal using characteristics without prior knowledge may produce many wrong combinations. Second, we need a basis for deciding whether the model needs

979-8-3315-1596-6/24/$31.00 ©2024 IEEE
DOI 10.1109/ICPADS63350.2024.00018

to be migrated. Because each signal may have a different source, locating the model of the signal is a critical step. According to the corresponding model, we need to develop the judgment basis. The third challenge is how we transfer the model to the correct domain. This way requires changing a lot of parameters. Therefore, we need to change these parameters to the correct vector direction and displacement magnitude.

In this paper, we innovatively utilize a Ranking Model to address the overlapping keystroke acoustic signal separation. Because the signal features that could contribute more than 80% to the classification results were located in the signal initiation stage, it is more efficient to obtain the overlapping keystrokes' starting points precisely than the traditional blind keystroke separation methods, like cleaning irrelevant features. The accuracy of signal separation is more than 90%, much higher than the 25% of the commonly used method FastICA. The key-identifying model can recognize the separated signals with 92.69% accuracy on average.

Meanwhile, we found that FID (Fréchet Inception Distance) is highly correlated with environmental changes, so we let the FID serve as the commander in model migration. This information helps the model know when and where to migrate.

We also design Inductive Vector methods to adapt our key-identifying model to environment dynamics (such as venue, phone location, and user diversity). The Inductive Vector can adapt the model parameters according to the shift of FID. For various mobile phone locations, the Inductive Vector method improves recognition accuracy from 61% (the maximum achieved by other methods) to 98%, significantly surpassing the best existing keystroke recognition algorithm. In terms of environmental dynamics, the accuracy rate of the Inductive Vector method is 81.7%, making it at least 1.65 times and up to 71 times more accurate than the leading current keystroke recognition algorithm.

Through the system design, we make the following contributions:

- To the best of our knowledge, Chameleon is the first system that can adapt to the environment and recover mixed keystroke information from more than one keyboard.
- We introduce an innovative signal separation method that can be used in overlapping keystroke acoustic signals. Through experiments, we found some key factors that affect overlapping keystrokes.
- We introduce the Inductive Vector, which assists in adapting our key-identifying model to environmental dynamics, such as location, phone location, and different users, thus enhancing accuracy in new environmental conditions. Additionally, we present FID, an innovative method that guides adjustments to the model.

II. Overlapping Keystroke Analysis

A. Probability of Overlapping

Keystroke recognition rarely accounts for overlapping signals, which can pose challenges when implemented in real-world scenarios. In this section, we will demonstrate a straightforward model to demonstrate the significance of this issue.

In a typing scenario, such as in an office, it is essential to recognize that keystrokes occur randomly and independently of one another because each individual types as per their needs and requirements. Assuming everyone is typing, the probability of keystroke events can be seen as the average typing rate. We want to observe the overlapping of the signal, which would indicate the occurrence of multiple events within the same timeframe. We can reference the Poisson distribution to determine the probability of multiple random events occurring within a given period of time. The Poisson distribution is $P(X = k) = \frac{\lambda^k}{k!}e^{-\lambda}, k = 0, 1, ...$ where X is the overlap times X of keystrokes, and λ is the average number of keystrokes that occur per unit of time.

Considering the average typing speed is about 180 kpm (180 characters per minute = 3 kps), and the keystroke signal length is 250 ms, the average number of keystroke events is $\lambda = \frac{250ms}{1000ms} * 3kps = 0.75$. The probability of the keystroke overlapping means that two keystrokes occur simultaneously, so the $X > 1$, which is $P(X > 1) = 1 - P(X = 0) - P(X = 1) \approx 1 - 47.24\% - 35.43\% = 17.33\%$. That means the keystrokes have severe overlapping and amount to nearly one-fifth of the total keystrokes made in the actual scenario when only considering two people are typing.

B. The Influence of Overlapping Keystroke

It is essential to investigate the impact of overlapping signals. This section will analyze keystroke signals and demonstrate how overlapping signals affect the results.

1) The Contribution Distribution in each Keystroke: For our study, we gathered 23,200 labeled keystrokes (58 keys in the main keyboard area) and calculated their amplitude-spectrum values (ASV). We selected four commonly used machine learning models in keystroke recognition research: BP(Back Propagation), KNN (k-Nearest Neighbor), Linear Regression Classification, and Support Vector Machine (SVM). We aimed to identify which parts of the signals' features significantly influence machine learning.

Given that the underlying principles of these four models differ, it is crucial to develop tailored methodologies for each model. Unlike the other three models, KNN operates as a parameter-free model and does not factor in the weight of each feature. Instead, KNN is determined based on the labels of the k nearest neighbor instances. Therefore, to ascertain the impact of the features on the KNN model's predictive capacity, we will employ feature selection techniques. Subsequently, we will evaluate the effects of eliminating features on the model's accuracy.

The experimental findings revealed that over **80% of the contributions to the classification outcomes were concentrated within the initial 62.5ms in the 250 ms keystroke signal**. These outcomes indicate that the classification model's accuracy is significantly decreased when another keystroke signal disrupts the leading edge of a keystroke signal. It means if another keystroke signal disrupts the "head" of a keystroke signal, the classification model's accuracy will be significantly reduced. The proliferation of people engaging in typing activities will result in a significant surge in this

phenomenon. Consequently, classification models are likely to encounter a substantial influx of misjudgments.

2) The Interference Factors of the Overlapping Signal: In the preceding section, we observed that the trained model tends to display a bias towards the leading edge of the signal due to the uneven distribution of compelling keystroke features. In this section, we will discuss two factors that contribute to the shortcomings of traditional keystroke recognition methods: **"High-Weight Feature Interference" (HWFI)** and **"Segment Deviation"**.

Due to the front-heavy weight distribution of the model, it is evident that **introducing interference to the front of the signal would impact the model's accuracy**, which is a phenomenon we call **"High-Weight Feature Interference" (HWFI)**. However, the impact has not been discussed in past research. To explore this further, we conducted an experiment whereby we introduced noise like the keystroke signal, causing it to "permeate" into the signal from two directions, respectively. We focused on continuous signal segments rather than affecting a random sample point. We then observed how the classification effectiveness of four different models changed as the duration of noise "permeate" increased. To present the data comprehensively, we chose the data with the highest overall classification accuracy rate for representation. The results in Fig 1 revealed that the accuracy dropped below 50% at 27 ms. If the noise covers the last segment of the signal, there is no notable decline in the classification accuracy.

Segment deviation significantly impacts the model's classification performance. This refers to the bias in segmenting the signal. When applying Voice Activity Detection (VAD) to segment noise-overlapping signals, the bias caused by the noise in the starting position leads to a considerable decrease in the model's classification accuracy. This is depicted in Figure 2, where the accuracy drops below 20% across all four models.

3) Low accuracy in blind source separation: The available techniques for handling overlapping signal processing in signal analysis are to separate them into original signals, broadly classified into two categories. The first category comprises methods that rely on statistical analysis to estimate the source signal, such as FastICA, without prior knowledge. These methods assume that the source signals are statistically independent in the mixed signal and are a linear combination. However, due to the complex combination of keystrokes, these methods' classification accuracy could be higher than VAD depicted in Fig 2, but still lower than 30%.

On the other hand, the second category of methods involves using prior knowledge, such as Deep Learning, to address this issue. Deep Learning leverages prior knowledge to separate the signals. However, the computational complexity and practical feasibility of deploying deep learning methods for keystroke segmentation pose significant challenges. Therefore, the question arises: Is it necessary to separate overlapping keystrokes into their original signals?

The answer is negative. The model can be enhanced by tweaking the distribution of the model weights (More suitable for environmental attribute changes, and we'll cover this in section III) and implementing precise segment cuts (in the next section). This will reduce the computing power requirements.

4) Modifying the Model Weights: However, it is worth noting that we cannot simply amplify the weight of the second half of the signal in the model to improve the recognition accuracy. Previous experiments have demonstrated the accuracy achieved without Segment Deviation under noise interference conditions. We will introduce two random keystrokes for the upcoming experiment to better mimic real-world scenarios. The second signal will randomly overlap with the first signal by simulating the random occurrence of keystrokes and their subsequent overlapping. We measured the improvements in classification separately by adjusting the segmentation, modifying the model weights (amplifying the weight of the second half of the signal in the model), and combining both approaches. Fig 3 depicts the boosting effect that these three methods have on the four models. That means recognizing that the overlapping keystrokes may not need to be separated into original signals or modify the model weight. Compared with VAD, only adjusting the segmentation starting point can improve the keystroke recognition accuracy of the second signal to larger than 65%.

III. Features Transfer

The previous research has identified a challenge in the performance of Classification Models when confronted with changing conditions. Although there are a large number of transfer learning algorithms, these solutions are designed for images, and there is still a lack of a transfer method designed for short speech signals such as keystrokes. We present a new algorithm called Inductive Vector (IV) to address this issue. Its purpose is to enhance the resilience of keystroke recognition to some extent. Through experimentation, we investigate the detrimental effects of various attribute changes, such as different environments (Env), keyboard variations (Keyboard), keyboard or phone movement (Move), different desk materials (Medium), diverse phones (Phone), and the typing habits of different users (User).

A. Impact of Various Attribute Change

1) Experimental Settings: We conducted the following experiments to explore the negative impact of the change of scene.

Model: We take the multilayer perception composed of two fully connected layers as the keystroke recognition model M. The first fully connected layer of M uses the standard normal distribution to initialize the weights and bias values and freezes the first layer without changing in the later experiments. The data set calculates normalized ASD as the input of M.

Train Set and Test Set: We collected data from seven attributes, including five environments, five keyboards, six keyboard locations, three desk materials, three mobile phones, eight users, and fifty-six keys in the keyboard's main function area. We use the fixed, variable method for experiments. The data volume is the Cartesian product of all attributes. Moreover, each key has 100 labels. We randomly divided the data into a 1:1 ratio train set and test set.

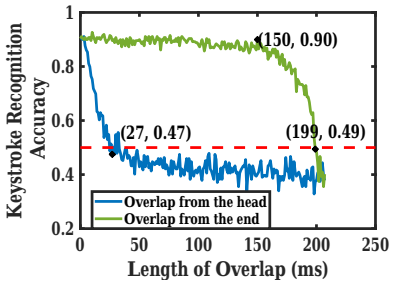


Fig. 1: The noise overlaps different numbers of sample points in the original keystroke signal and affects the classification performance.

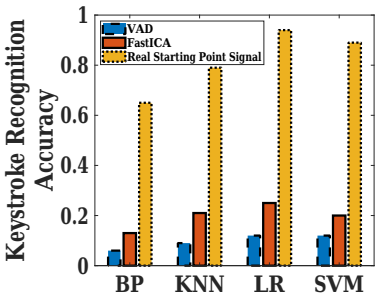


Fig. 2: The data shows the performance of the segment deviation.

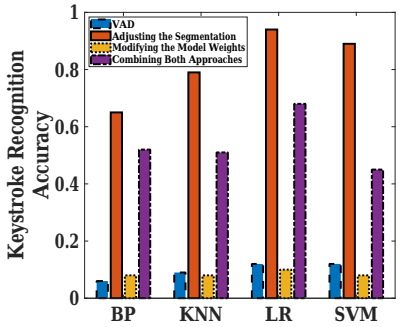


Fig. 3: Two randomly overlapping real-world keystroke signals classify performance using different methods under four models.

2) Result: The experiment result is shown in Fig 4. It shows that the change of attributes dramatically lowers the recognition accuracy, even making the model unable to recognize keystrokes. Also, when the keyboard, environment, phone, medium, user, or keyboard location changes, the accuracy drops, even with subtle changes. Therefore, it is necessary to have a new algorithm to improve the robustness of the model.

B. Feature Transfer Distance

To migrate features, we first need to know the migration distance. There are many ways to measure feature changes, and we chose The Fréchet Inception Distance (FID) as a reference. The Fréchet Inception Distance (FID) $d^2(F,G)$ is used to calculate the statistical distance in feature Space between the train set and the test set [29]–[31]. We evaluate the distance between keystroke data sets in different domains using FID. The formula is shown below, $d^2(F,G) = |\mu_X - \mu_Y|^2 + tr\left[C_X + C_Y - 2(C_X C_Y)^{1/2}\right]$ Random variables X and Y belong to the distributions F and G. And C_X and C_Y represent the covariance of X and Y, respectively. It is assumed that the abstract features extracted from the middle layer of the trained model M constitute the feature space R^n. The probability distribution of the data set on the feature space R^n constitutes the discrete multivariate distributions F and G. Since F and G need to belong to the same distribution before calculating $d^2(F,G)$, normalization processing is performed on the data set on R^n to make F, and G conform to multivariate Gaussian distribution.

TABLE I: Pearson Correlation Coefficient between FID and Accuracy

	Environment	Keyboard	Move	Medium	Phone	User
Pearson	-0.970	-0.685	-0.925	-0.976	-0.955	-0.933

C. Correlation between FID and Accuracy

We use the first fully connected layer to form the feature space. For each scene, we calculate FID between two data sets by the training set in the source domain and test set in the target set while getting keystroke recognition accuracy between two data sets by the model in the source domain and test set in the target domain. Then Pearson correlation coefficient is used to evaluate the relationship between FID and accuracy (shown in Table I). And the binary one-order equation is used to fit FID and accuracy (shown in Fig 5).

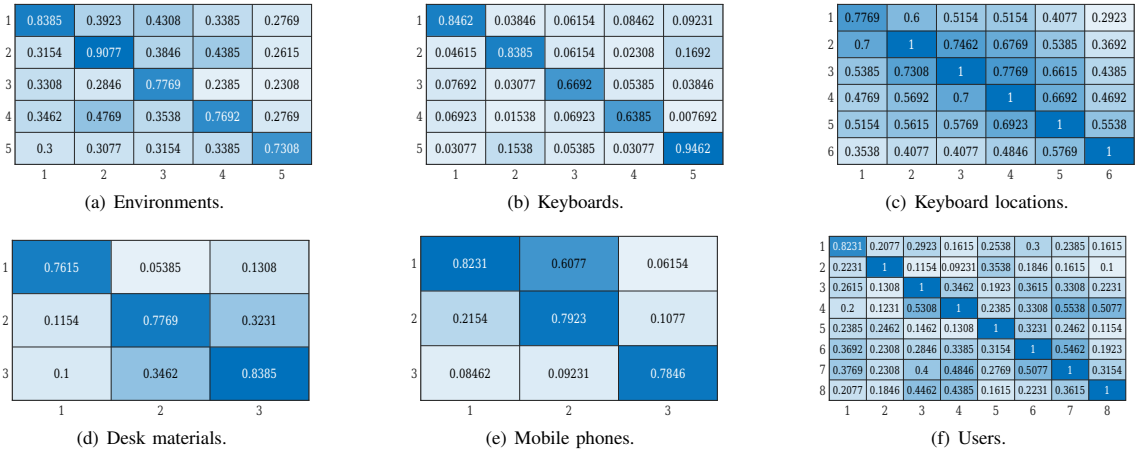
From Table I, the Pearson correlation coefficient between FID and accuracy was lower than -0.9, showing a strong negative correlation. Fig 5 was obtained using binary one-order equation fitting FID and accuracy. Most data fall near the line, and an obvious negative correlation exists between FID and key recognition accuracy. When FID is less than 236, the key recognition accuracy is more than 0.3.

Our experiments have found that the problem in recognizing aliased keystroke signals stems from variations in the feature space, as indicated by the large modulus of FID. However, we observed that introducing a certain degree of offset can notably enhance identification accuracy for most signals.

D. Inductive Vector

According to the above experiments, the key accuracy rate will decrease when the keyboard position and environment change, making the key recognition system challenging to deploy because users are likely to move the keyboard or even change the environment when they are typing for a long time. Subtle changes exist all the time and affect the model recognition effect. At the same time, obtaining a large amount of training data by changing the keyboard position and other factors is not practical. Therefore, we need a model that can improve the robustness of key recognition under limited data. Fortunately, from the above experiments, it can be seen that FID is strongly correlated with key recognition accuracy.

Meanwhile, FID represents the distance between two data sets in the feature space, while the Normal Vector divides and classifies data in the feature space. Therefore, we put



(a) Environments. (b) Keyboards. (c) Keyboard locations.

(d) Desk materials. (e) Mobile phones. (f) Users.

Fig. 4: Confusion matrix of classification accuracy for different attributes.

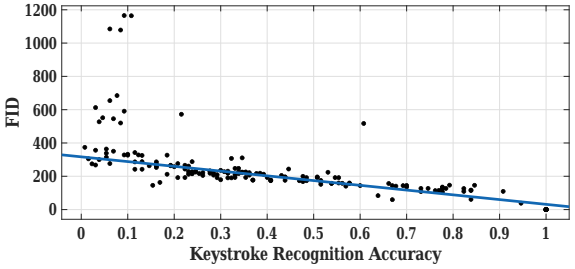


Fig. 5: The Fitting Lines for FID and Accuracy

forward a question of whether FID can assist the adaptive adjustment of the model Normal Vector between different domains, that is, when a certain factor changes (such as the keyboard position moving 8cm), can adjust the Normal Vector of the original model according to the FID of the new data on the original model, to improve the key recognition accuracy without retraining. Based on this, we conduct the following experiments and propose an adaptive algorithm for the Normal Vector adjustment in different domains, the Inductive Vector (IV).

1) Algorithm Description: Assume that the model obtained from the original training set U_0 is M_0, the Normal Vector of M_0 (i.e., the weight and bias value of the last layer) is W_0, the current data set is U_1. The distribution of U_0 and U_1 on the feature space of M_0 is F and G, respectively. When $FID(U_0, U_1) = d^2(F, G) > \lambda$(Set $\lambda = 236$), according to $FID(U_0, U_1)$ and W_0 obtain new Normal Vector $W_1^{'}$, then use $W_1^{'}$ to replace W_0 and get new model $M_1^{'}$, thus improve the model accuracy in U_1. The model of obtaining new Normal Vector $W_1^{'}$ according to $FID(U_0, U_1)$ and W is the proposed Inductive Vector (IV), IV models.

The IV model consists of a fully connected layer, the number of nodes is the amount of data contained in W_0, and the sigmoid activation function and Adam optimizer are adopted. MSE is used as a loss function to calculate the distance between the predicted Normal Vector and the target Normal Vector, and the training is 1000 rounds. The input vector of the IV model is the FID of U_0 and U_1 on the feature space of M_0, and the Normal Vector W_0 of M_0. The true value is the Normal Vector W_1 trained on the training set of U_1 to obtain the model M_1. It is worth noting that, in order to control variables, the Normal Vector is only affected by the distribution of data in the feature space, and the previous experiments show that high key recognition accuracy can be obtained with the same feature space, so the feature space of model M_0 and M_1 is consistent.

TABLE II: Performance comparison of various methods across different categories.

Method	Env	Keyboard	Move	Medium	Phone	User
Inductive Vector	**0.85**	**0.71**	**0.98**	**0.72**	**0.71**	**0.93**
BP	0.21	≤ 0.1	0.05	≤ 0.2	≤ 0.1	0.35
BP-STFT	≈ 0	≈ 0	0.43	≤ 0.1	≤ 0.1	≤ 0.2
SVM-MFCCs	0.21	≈ 0	0.61	≤ 0.1	≤ 0.1	0.32
SVM-ASD	0.53	≈ 0	0.54	≤ 0.1	≤ 0.1	0.32
CNN +ASD+MFCCs	0.23	≤ 0.1	0.35	≤ 0.1	≤ 0.1	≤ 0.2
LR-ASD	0.48	≈ 0	0.55	≤ 0.1	≤ 0.1	0.35
Bayesian-ASD	0.22	≈ 0	0.58	≤ 0.1	≤ 0.1	0.3
Decision Tree -ASD	0.22	≈ 0	0.3	≤ 0.1	≤ 0.1	≤ 0.2
KNN-ASD	0.38	≈ 0	0.41	≤ 0.1	≤ 0.1	0.33
Baseline (max ac)	0.53	≈ 0.1	0.61	≈ 0.2	≈ 0.1	0.35
Times	1.6x	71x	1.6x	36x	71x	2.7x

2) Experimental results: In this experiment, for each scene, we take the last data set in the scene as the test set of the Inductive Vector model, namely, environment 5, keyboard 5, keyboard position 6, medium under the keyboard 3, the mobile phone 3, or user 8, and the remaining data sets in the scene as the training set of the IV model. We set the baseline as the accuracy of the original model M_0 on target test set U_1. The experimental results of different scenarios are shown in Table II. The Inductive Vector model consistently enhances the key recognition accuracy of the original model, achieving an accuracy that surpasses the baseline by up to 71 times.

IV. System Overview

A. System Workflow

We propose a new keystroke recognition system (Fig 6) to solve key recognition problems caused by overlapped multi-key acoustic and environmental condition changes. In the preparation stage, we obtain the keystrokes from target keyboards in the room at different distances, but each keyboard

TABLE III: Comparison Object for Single Keystroke Identification

Paper	Model	Feature
[20]	*BP neural network*	*STFT*
[32]–[34]	*SVM*	*MFCCs* or *ASD*
[35]	*CNN*	*ASD+MFCCs*

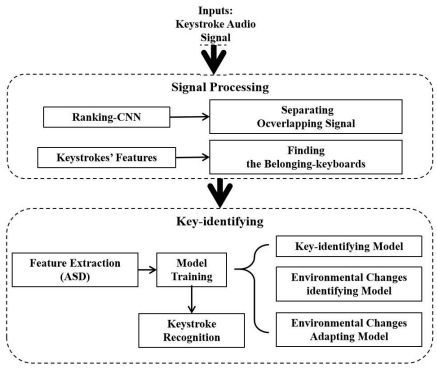


Fig. 6: System Overview

will only have one position as a training set. Then, in the feature extraction step, Chameleon will get three things: a key-identifying model, an environmental factors perception model to identify the changes and an Inductive Vector which can conduct the key-identifying model to adapt to the environmental changes. The system, like Chameleon, can sense changes in the color of the environment and automatically make changes.

When keystrokes come, Chameleon will input the collected keystroke signals mixing with several unknown sources to the Keystroke Segment Extractor. The Keystroke Segment Extractor will then use a Ranking Model to separate these overlapping signals by identifying the location and span. Next, the Keystroke Segment Extractor will find the belonging keyboards according to the keystrokes' features. Finally, Chameleon used the three prepared models to identify the changes in environmental factors in keystrokes and guided the Inductive Vector to adjust the key-identifying model. So the Chameleon will recognize the eavesdropped character by the adaptive key-identifying model.

B. Key-identifying Model

1) Feature Extraction: Because of the feasibility of keystroke recognition by ASD, we calculate the ASD of keystroke signal segment s and normalize it. Then, we concatenate the normalized ASD from 2 microphones as the feature x and input it to the corresponding keystroke recognition model according to ΔA.

2) Model Training: We use the multilayer perceptron M containing two fully connected layers as the keystroke recognition model. The first fully connected layer contains 1000 nodes, while the second contains 26 nodes with softmax as its activation function. Then, the epochs of the model are 100, and the loss function is sparse categorical cross-entropy.

3) Keystroke Recognition: We reconstruct the information by using M from step 4 and x from step 3. When getting a large enough data set U_1 and $FID(D, U_1) > \delta$, we use Inductive Vector (IV) to adjust the model M and get a new model $M_1^{'}$ to improve the accuracy. The details of the Inductive Vector are described in Section 8.

V. Evalution

A. Real-world Overlapping Signal

1) Data Set: We collect single keystroke data sets R_1 and R_2 from two keyboards. The data in the data set $R = R_1, R_2$ is the single keystroke signal recording of double microphones with a sampling rate of 96kHz. Each keystroke operation does not overlap, and the number of labels is balanced.

We divide R into the training set and verification set and use the train set and verification set to form an overlapping data set for evaluating the recognition performance of overlapping signals.

Train Set and Validation Set: Each subset in data set R is randomly divided into a training subset (520 samples) and a validation subset (260 samples) in a 2:1 ratio. The number of labels in each subset is balanced.

Overlapping Data Set: We collected an overlapping acoustic data set of 10000 samples. We randomly select data from the verification set with putting back (it is uncertain whether they come from the same keyboard and whether they come from the same key) and then randomly select the overlapping beginning position of the second keystroke (i.e., $\Delta t = t_a - t_b$ randomly, where t_a is the beginning position of the first keystroke, t_b is the beginning position of the second keystroke. Then, the two signals are linearly super-placed together in accordance with Δt to create an overlapping signal, which is labeled with the keys, keyboards, and Δt corresponding to the two sets of data. Each sample in the overlapping acoustic data set is synthesized as described above.

Mixed Data Set: We use the validation set and overlapping data set in a 1:1 ratio to synthesize a mixed acoustic data set used to evaluate the overall effect of the system.

2) Overall Performance: We use a mixed data set to evaluate the system's overall performance. The keystroke recognition **accuracy is 87.31%**, proving the model's feasibility in dual keyboard mixed key recognition.

3) Performance in Details: We also use mixed data set to evaluate the keystroke beginning estimation algorithm, keystroke number determination algorithm, and keystroke-keyboard matching algorithm. Then, an ablation experiment was conducted for each step to explore the influence of each part on keystroke recognition.

Keystroke Beginning Estimation: We take the threshold-based algorithm as the baseline of keystroke beginning estimation. The algorithm based on the threshold takes the fixed threshold ρ(ρ=0.1) as the cutting standard. When a signal's absolute value of time t is more significant than ρ, the signal fragment from t-5.21ms to t+244.79 ms is intercepted as the key signal.

We use the recognition accuracy on different keyboards as the evaluation standard of keystroke beginning estimation. Specifically, we use VAD or baseline to extract keystroke signal segments from R_1 or R_2. Keystroke recognition accuracy was obtained as shown in Fig 7.

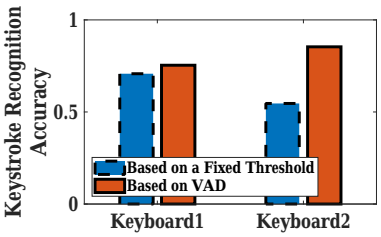


Fig. 7: Keystroke Beginning Estimation Performance Compared with Baseline

TABLE IV: The performance of Each Step

	performance
Keystroke Beginning Estimation	80.4%
Keystroke Number Estimation	87.5%
Overlap Beginning Estimation	5.32e+05
Keystroke-Keyboard Matching	87.3%

Due to the different energy of sound waves emitted by different keyboards, the baseline cannot adapt to the diversity of keystrokes with a fixed threshold. So the beginning position is unstable, leading to a decrease in the recognition rate.

Keystroke Number Estimation: We take the number of keystrokes fixed at two as the baseline, so the accuracy of keystroke number estimation is 87.5%.

Keystroke-Keyboard Matching: In our cognition, this topic is the first work for multi-keyboard mixed keystroke recognition. Therefore, we assume that all keystrokes come from a single keyboard. As the baseline of this algorithm, the correct rate of keystroke recognition achieved by the system is 87.3%.

Summary: The recognition effect of each step is shown in Table IV. The accuracy of single key recognition is about 80.4% by using the keystroke signal after VAD is used. The accuracy of keystroke number estimation is about 87.5%; The MSE of the overlapping beginning position estimation is about 5.32e+05; The accuracy of keystroke-keyboard matching is 87.3%.

VI. Related Work

A. Keystroke Recognition

Previous studies can be broadly divided into two categories based on Model-driven, like TDOA (Time Difference Of Arrival) and Data-driven. The keystroke recognition algorithms in model-driven are based on TDOA [21], [22]. They utilized multiple microphones and used the hyperbolic functions between more than two microphones to localize the keys' position. However, this method requires prior knowledge of phone and keyboard layout because it is tough to satisfy the deployment of such prior knowledge, especially the keyboard snooping. **Chameleon** do not require prior knowledge of the layout and can place phones arbitrarily.

Machine Learning has been used in keystroke recognition for a long time. In detail, using neural network (BP neural network [20], support vector machine [32], [33], RNN [36]) trained a large amount of data to obtain a single keystroke recognition model. The input of the model is ASD (amplitude-spectrum diagram) [33], MFCC (Mel Frequency Cepstral Coefficient) [32], STFT (Short-Time Fourier Transform) [20], [36], DTW (Dynamic Time Warping) [37], Cross-Correlation [37], frequency-based distance [37] or other common signals Physical characteristics. However, the accuracy of these methods could be more robust to the environment. The location change of the phone easily misjudges the recognition and the interference of other external key sounds. **Chameleon** can adapt to these changes and has high accuracy in the same environment. In our cognition, we are the first to focus on a fixed keyboard or a single keystroke and the overlapping signals from the same or different keyboard.

B. Blind Source Separation

The technique of decomposing a mixed signal composed of multiple source signals is called blind source separation. Traditionally, blind source separation techniques based on mathematical models have assumptions on the mixed and source signals. Related algorithms have also been proposed under these assumptions, which mainly include Independent Component Analysis [24] (like assuming that the source signal satisfies independence), Sparse Component Analysis [25] (like assuming that the source signal has sparsity in some domains). Non-negative Matrix Factorization [26] (like assuming that both the source signal and the mixing coefficients are non-negative) and Bounded Component Analysis [27] (like assuming that the source signal and noise have bounded properties). Nevertheless, in practice, these assumptions may need to be revised. At the same time, neural networks are also used in blind source separation [28], using frequency domain or time domain data as input to the neural network. However, poor separation results are obtained when the noise or the SNR is not matched.

VII. Conclusion and Limitation

We have demonstrated that the change of scene dramatically lowers the keystroke recognition accuracy, and keystroke signal overlapping occurs with high probability ($0.1302 \cdot n$, n is the number of keyboards), leading to signal overlap and alignment problems. To solve these problems, we propose an inductive vector to improve the robustness of the model, improving the accuracy when the environment, keyboard location, or user changes. Meanwhile, we use the first 42ms of the keystroke signal to solve the signal overlap problem and introduce Ranking-CNN to solve signal alignment. Finally, we implement a system to reconstruct information from two keyboards.

Although this approach can partially compensate for environmental interference, it is only sometimes suitable for some conditions, particularly when it comes to the keyboard, desktop material, and mobile phone changes, where its effectiveness is limited. The primary reason for this is the non-linear transformation of signal characteristics. Changes in keyboard and desktop materials result in alterations to signal properties, while mobile phones exhibit differences in microphone position and frequency response curves. These issues remain unresolved. Additionally, while the precise cutting method has shown effective enhancements, Ranking-CNN still necessitates substantial computational resources. Developing a lower-computation method would be highly valuable in this regard.

ACKNOWLEDGMENT

The research was supported in part by the China NSFC Grant (No. 62372307, No. U2001207), Guangdong NSF (No. 2024A1515011691), Shenzhen Science and Technology Program (No. RCYX20231211090129039), Shenzhen Science and Technology Foundation (No. JCYJ20230808105906014), Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things(No.2023B1212010007), the Project of DEGP (No.2023KCXTD042). Lu WANG is the corresponding author.

REFERENCES

[1] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "Csi-based indoor localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, 2012.

[2] Q. Dai, Y. Huang, L. Wang, R. Ruby, and K. Wu, "mm-humidity: Fine-grained humidity sensing with millimeter wave signals," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 204–211, IEEE, 2018.

[3] S. Zhong, Y. Huang, R. Ruby, L. Wang, Y.-X. Qiu, and K. Wu, "Wifire: Device-free fire detection using wifi networks," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2017.

[4] Y. Huang, S. Cai, L. Wang, and K. Wu, "Oinput: A bone-conductive qwerty keyboard recognition for wearable device," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 946–953, IEEE, 2018.

[5] Y. Huang, K. Chen, J. Zhao, L. Wang, and K. Wu, "Beverage deterioration monitoring based on surface tension dynamics and absorption spectrum analysis," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 3722–3740, 2023.

[6] K. Chen, L. Wang, Y. Huang, K. Wu, and L. Wang, "Lit: Fine-grained toothbrushing monitoring with commercial led toothbrush," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pp. 1–16, 2023.

[7] K. Chen, Y. Huang, Y. Chen, H. Zhong, L. Lin, L. Wang, and K. Wu, "Lisee: A headphone that provides all-day assistance for blind and low-vision users to reach surrounding objects," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 3, pp. 1–30, 2022.

[8] Y. Huang, K. Chen, Y. Huang, L. Wang, and K. Wu, "Vi-liquid: unknown liquid identification with your smartphone vibration," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pp. 174–187, 2021.

[9] Y. Huang, K. Chen, L. Wang, Y. Dong, Q. Huang, and K. Wu, "Lili: liquor quality monitoring based on light signals," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pp. 256–268, 2021.

[10] Q. Xie, S. Jiang, L. Jiang, Y. Huang, Z. Zhao, S. Khan, W. Dai, Z. Liu, and K. Wu, "Efficiency optimization techniques in privacy-preserving federated learning with homomorphic encryption: A brief survey," *IEEE Internet of Things Journal*, vol. 11, no. 14, pp. 24569–24580, 2024.

[11] S. Jiang, W. Ding, H.-W. Chen, and M.-S. Chen, "Pgada: perturbation-guided adversarial alignment for few-shot learning under the support-query shift," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 3–15, Springer, 2022.

[12] S. Jiang, R. Fang, H. Chen, and W. Ding, "Dual adversarial alignment for realistic support-query shift few-shot learning," *arXiv preprint arXiv:2309.02088*, 2023.

[13] S. Jiang, H. Chen, and M. Chen, "Dataflow systolic array implementations of exploring dual-triangular structure in qr decomposition using high-level synthesis," in *2021 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, pp. 1–8, IEEE, 2021.

[14] S. Jiang, X. Shuai, and G. Xing, "Artfl: Exploiting data resolution in federated learning for dynamic runtime inference via multi-scale training," in *23rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 1–12, IEEE, 2024.

[15] B. Fan, S. Jiang, X. Su, and P. Hui, "Model-heterogeneous federated learning for internet of things: Enabling technologies and future directions," *arXiv preprint arXiv:2312.12091*, 2023.

[16] X. Shuai, Y. Shen, S. Jiang, and Z. Zhao, "Balancefl: Addressing class imbalance in long-tail federated learning," in *21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 1–10, IEEE, 2022.

[17] K. Wu, Y. Huang, L. Wang, and C. Kaixin, "Auxiliary sensing method and system based on sensory substitution," Apr. 19 2022. US Patent 11,310,620.

[18] Q. Liao, Y. Huang, Y. Huang, Y. Zhong, H. Jin, and K. Wu, "Magear: Eavesdropping via audio recovery using magnetic side channel," in *MobiSys*, 2022.

[19] Q. Liao, Y. Huang, Y. Huang, and K. Wu, "An eavesdropping system based on magnetic side-channel signals leaked by speakers," *ACM Transactions on Sensor Networks*, 2024.

[20] D. Asonov and R. Agrawal, "Keyboard acoustic emanations," in *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, pp. 3–11, IEEE, 2004.

[21] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using keyboard acoustic emanations," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 453–464, 2014.

[22] J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang, and M. Gruteser, "Snooping keystrokes with mm-level audio ranging on a single phone," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pp. 142–154, 2015.

[23] L. Lu, J. Yu, Y. Chen, Y. Zhu, X. Xu, G. Xue, and M. Li, "Keylistener: Inferring keystrokes on qwerty keyboard of touch screen through acoustic signals," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 775–783, IEEE, 2019.

[24] J. V. Stone, "Independent component analysis: a tutorial introduction," 2004.

[25] R. Gribonval and S. Lesage, "A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges," in *ESANN'06 proceedings-14th European Symposium on Artificial Neural Networks*, pp. 323–330, d-side publi., 2006.

[26] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[27] S. Cruces, "Bounded component analysis of linear mixtures: A criterion of minimum convex perimeter," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2141–2154, 2010.

[28] C.-Y. Chiu, W.-Y. Hsiao, Y.-C. Yeh, Y.-H. Yang, and A. W.-Y. Su, "Mixing-specific data augmentation techniques for improved blind violin/piano source separation," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, IEEE, 2020.

[29] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[30] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[32] J. Wang, R. Ruby, L. Wang, and K. Wu, "Accurate combined keystrokes detection using acoustic signals," in *2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, pp. 9–14, IEEE, 2016.

[33] L. Zhuang, F. Zhou, and J. D. Tygar, "Keyboard acoustic emanations revisited," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 1, pp. 1–26, 2009.

[34] J.-X. Bai, B. Liu, and L. Song, "I know your keyboard input: a robust keystroke eavesdropper based-on acoustic signals," in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1239–1247, 2021.

[35] T. Giallanza, T. Siems, E. Smith, E. Gabrielsen, I. Johnson, M. A. Thornton, and E. C. Larson, "Keyboard snooping from mobile phone arrays with mixed convolutional and recurrent neural networks," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, pp. 1–22, 2019.

[36] D. Slater, S. Novotney, J. Moore, S. Morgan, and S. Tenaglia, "Robust keystroke transcription from the acoustic side-channel," in *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 776–787, 2019.

[37] T. Halevi and N. Saxena, "Keyboard acoustic side channel attacks: exploring realistic and security-sensitive scenarios," *International Journal of Information Security*, vol. 14, no. 5, pp. 443–456, 2015.