CNN-Based Continuous Authentication on Smartphones with Auto Augmentation Search

Shaojiang Deng, Jiaxing Luo, and Yantao Li(✉)

College of Computer Science, Chongqing University, Chongqing 400044, China
yantaoli@cqu.edu.cn

Abstract. In this paper, we present CAuSe, a **C**NN-based **C**ontinuous **Au**thentication on smartphones using **Au**to **Au**gmentation **Se**arch, where the CNN is specially designed for deep feature extraction and the auto augmentation search is exploited for CNN training data augmentation. Specifically, CAuSe consists of three stages of the offline stage, registration stage and authentication stage. In the offline stage, we utilize auto augmentation search on the collected data to find an optimal strategy for CNN training data augmentation. Then, we specially design a CNN to learn and extract deep features from the augmented data and train the LOF classifier after 95 features are selected by PCA in the registration stage. With the trained CNN and LOF classifier, CAuSe identifies the current user as a legitimate user or an impostor in the authentication stage. Based on our dataset, we evaluate the effectiveness of optimal strategy and the performance of CAuSe. The experimental results demonstrate that the strategy of Time-Warping(0.6)+Time-Warping(0.6) reaches the highest accuracy of 93.19% with data size 400 and CAuSe achieves the best authentication accuracy of 96.93%, respectively, comparing with other strategies and classifiers.

Keywords: Continuous authentication · Auto augmentation search · CNN · LOF classifier

1 Introduction

The mobile devices have played an essential role in our daily lives, which makes privacy protection in mobile devices extremely important, since they store a lot of private and sensitive information. Even since 2011, sales of smartphones have exceeded sales of personal computers [2]. However, due to the high-frequency usage and information interaction of these devices (e.g. smartphones), it is difficult to prevent personal information leakage and illegal access by the one-time authentication that identifies users only at the time of initial logging-in, such as personal identification numbers (PINs), passwords, voice-prints, fingerprints and face recognition. PINs face a much serious threat of online guessing and even longer PINs only attain marginally improved security [3,26]. Wang et al.

D. Gao et al. (Eds.): ICICS 2021, LNCS 12918, pp. 169–186, 2021.
https://doi.org/10.1007/978-3-030-86890-1_10

systematically characterized typical targeted online guessing attacks with seven sound mathematical models, each of which was based on varied kinds of data available to an attacker [27]. Biometric information cannot be acquired by direct covert observation, but once biological information is stolen, it is not naturally available to reissue [22]. For example, fingerprint recognition can be cracked by people with ulterior motives obtaining legitimate users' fingerprints left on the screen. In addition, there is a severe security and privacy threat in one-time authentication mechanisms that when a legitimate user leaves the supervision of the device after the initial authentication (the screen is unlocked), impostors can easily gain access to the device illegally.

Compared with the traditional one-time authentication mechanisms, continuous or implicit authentication approaches would provide an additional line of defense by designing a non-intrusive and passive security countermeasure [9]. The current continuous authentication mechanisms essentially use built-in sensors and accessories to frequently collect physiological or behavioral biometrics to identify the legitimacy of the user, such as voice [8], face patterns [1], touch gestures [28], typing motion [10] and gait dynamics [21]. There are two main stages for continuous authentication systems: user registration phase and continuous authentication phase. During the user registration phase, owners of mobile devices are usually asked to perform some operations to collect information to recognize the owners. During the continuous authentication phase, the system collects the user's sensor readings at regular intervals to determine whether the current user is the device owner. If the system finds that the current user is an illegal user, the system will lock the device to prevent the owner's privacy from leaking. The accelerometer, gyroscope, and magnetometer are the three most commonly used sensors for collecting behavioral biometrics without users' notice. Both accelerometer and gyroscope are motion sensors that can monitor the users' motion on the device. Magnetometer is a position sensor used to determine the physical position of the device in the true frame of reference. However, in order to obtain a high-performance continuous authentication model, it is often necessary to collect a large amount of high-quality data for training models, which costs lots of time and resources. Data augmentation methods, such as flipping, cropping, color dithering and generative adversarial networks (GANs), are very common techniques in the field of image recognition, which help cover unexplored input space, prevent overfitting and improve the generalization ability of classification model. However, there are currently few data augmentation methods specifically for time-series sensor data because time-series sensor data are quite different from image data and most of the current data augmentation methods cannot be used to create time-series data directly. Since the sufficient amount of sensor data collection needs lots of volunteers to participate, it is challenging to augment time-series sensor data. Moreover, for specific applications, artificially constructing features for time-series sensor data often requires a lot of prior expert knowledge. It is also challenging to extract features with high representation capacity on time-series sensor data.

To address the challenges of data shortage and feature contribution, we are among the first to utilize the auto augmentation search to find an optimal

data augmentation strategy for CNN training and design a CNN-based deep feature extraction method consisting of feature learning and feature selection. In this paper, we present CAuSe, a **C**NN-based **C**ontinuous **Au**thentication on smartphones using **Au**to **Au**gmentation **Se**arch. Specifically, CAuSe consists of five modules: data collection, auto augmentation search, feature extraction, classifier training and authentication. The process of CAuSe includes three stages of the offline stage, registration stage, and authentication stage. In the offline stage, CAuSe collects time-series sensor data of the accelerometer, gyroscope, and magnetometer, and then utilizes auto augmentation search on the collected sensor data to find an optimal data augmentation strategy. In the registration stage, CAuSe applies the optimal augmentation strategy on the collected sensor data, uses the designed CNN to learn and extract deep features from the augmented data, and trains the local outlier factor (LOF) classifier after 95 deep features are selected by principal component analysis (PCA). In the authentication stage, based on the sampled sensor data, CAuSe uses the trained CNN to learn and extract features and utilizes the trained LOF classifier to conduct the authentication based on the 95 PCA-selected features. Based on our dataset, we evaluate the effectiveness of auto augmentation search and the corresponding optimal strategy and the performance of CAuSe. The experimental results demonstrate that the augmentation strategy of Time-Warping(0.6)+Time-Warping(0.6) reaches the highest authentication performance with the 93.19% accuracy, 93.77% F_1-score, and 3.9% EER with data size 400, and CAuSe achieves the best accuracy of 96.93% with the LOF classifier on 95 PCA-selected features, respectively, comparing with other augmentation strategies and classifiers.

The main contributions of this work are summarized as follows:

- We present CAuSe, a CNN-based continuous authentication on smartphones using auto augmentation search, leveraging the smartphone built-in accelerometer, gyroscope and magnetometer.
- We specially design a CNN for deep feature extraction and utilize the auto augmentation search to find an optimal data augmentation strategy for CNN training.
- We evaluate the effectiveness of auto augmentation search and the performance of CAuSe, and the experimental results illustrate that the searched augmentation strategy reaches the highest accuracy (93.19%) with data size 400, and CAuSe achieves the best authentication accuracy (96.93%), respectively.

The remainder of this work is organized as follows: Sect. 2 reviews the state-of-the-art on continuous authentication. We elaborate the architecture of CAuSe in Sect. 3 and evaluate the performance of the optimal strategy and CAuSe in Sect. 4. Section 5 concludes this work.

2 Related Work

In this section, we review the state-of-the-art of the continuous authentication systems, time-series data augmentation methods and auto augmentation methods, respectively.

2.1 Continuous Authentication System

In the field of continuous authentication, high-precision discrimination results are often inseparable from an efficient system framework. In recent years, researchers have creatively designed well-performed continuous authentication systems based on different data sources [20]. The mainstream continuous authentication solutions are broadly composed of two phases: registration phase and authentication phase. During the registration phase, these systems extract features from the collected datasets and train classifiers with labeled features. During the authentication phase, these systems utilize the trained classifiers to classify features that are extracted from unidentified users' data. Considering that different types of touch operations may contain quite different characteristics, the authors in [28] designed specific features for different touch operations, and then adopted the trained classifiers for authentication. Z. Sitová et al. [23] designed hand movement, orientation and grasp behavioral features based on sensor readings from smartphones, then trained and tested one-class classifiers after feature selection. Mahbub et al. [19] trained a linear SVM with statistical features obtained from face proposals that were derived from the estimated faces in their designed system. In [5], the authors proposed a continuous motion recognition system that was based on motion data from the accelerometer, gyroscope and magnetometer. They used a Siamese convolutional neural network to learn deep features, and then trained the one-class SVM with learned features of the legitimate user, to predict new observations. In [13], Li et al. proposed a two-stream convolutional neural network for feature learning in the continuous authentication system which was based on bottleneck structure of Mobilenet v2, with both time domain data and frequency domain data of the accelerometer and gyroscope as the network inputs.

Inspired by the above contributions, we design an efficient CNN-based continuous authentication system which can achieve very close performance with few sampled sensor data for training using time-series data auto Augmentation technology.

2.2 Time-Series Data Augmentation Method

In the image recognition field, data augmentation can be implemented by labeling the same labels for images obtained by performing operations, such as scaling, cropping, jittering and flipping on raw images. However, in the time-series data field, such as sensor data, there are few data augmentation approaches proposed. In [25], the authors were among the first to exploit geometric transformation, such as permutation, sampling, scaling, cropping and jittering, as sensor data

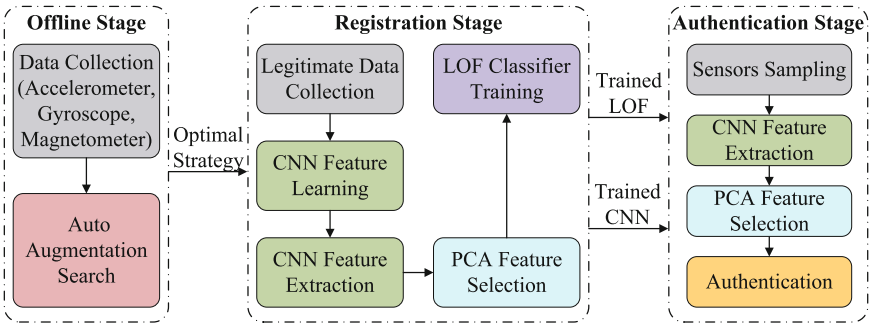

Fig. 1. CAuSe architecture.

augmentation approaches, which were different to those in image augmentation. DeVries et al. [7] used a sequence autoencoder to project data into feature space and investigated augmentation techniques in the feature space.

Data augmentation with generative adversarial networks (GANs) has attracted some researchers' attentions recently. Zhu et al. [31] proposed an emotion classification system using data augmentation with a cycle-consistent adversarial network (CycleGAN) and Luo et al. [17] trained a conditional Wasserstein generative adversarial network (WGAN) with electroencephalography (EEG) data to generate additional data for data augmentation. In [24], the authors investigated the possibility of using GANs to augment time-series Internet of Things (IoT) data. In [12], the author investigated five sequential data augmentation techniques (additional Gaussian noise, masking noise, signal translation, amplitude shifting, and time stretching) including sample-based and dataset-based methods to improve the intelligent fault diagnosis accuracy.

2.3 Auto Augmentation Method

Since the current data augmentation implementations are almost manually designed [7,25], researchers prefer to apply one or several fixed data augmentation methods based on their experience for most datasets, although there theoretically exists an optimal data augmentation method for a specific dataset. Cubuk et al. [6] first proposed the concept of auto augmentation, which automatically searched optimal augmentation policies from data to improve validation accuracy. Their search algorithm (implemented as a RNN controller based on Reinforcement) sampled thousands of policies to train a child model to measure the performance of the generalization improvement, and then updated the augmentation policy distribution with a reward signal. Despite its promising empirical performance, this scheme was difficult to apply because it was very expensive with time-consuming calculation in the whole process. Lin et al. [16] formulated the augmentation policy as a parameterized probability distribution, thus allowing the augmentation policy probability distribution parameters to be optimized along with the network parameters simultaneously. Based on a bilevel

framework, this solution eliminated the need of re-training model after optimal augmentation policy search and achieved comparable performance with dozens of times faster than [6]. In [15], the authors proposed a fast auto augmentation algorithm to find effective augmentation policies via a more efficient search strategy based on density matching. Moreover, [29] proposed effective optimization algorithms to reduce the computational burden and time consumption of auto augmentation.

3 CAuSe Architecture

In this section, we present the architecture of CAuSe, the CNN-based continuous authentication on smartphones with auto augmentation search, as illustrated in Fig. 1. As shown in Fig. 1, CAuSe consists of three stages: the offline stage, registration stage, and authentication stage.

In the offline stage, CAuSe collects time-series sensor data and then utilizes auto augmentation search on the collected sensor data to find an optimal data augmentation strategy for CNN training data augmentation in the registration stage. First, we recruit volunteers to use smartphones equipped with sensor data collection tools to collect sensor data of the accelerometer, gyroscope and magnetometer. Then, we perform preprocessing operations on the collected time-series sensor data, and based on the preprocessed data, we conduct the auto augmentation search to obtain an optimal augmentation strategy.

In the registration stage, CAuSe applies the optimal augmentation strategy on the collected sensor data, uses the designed CNN to learn and extract deep features from the augmented data, and trains the local outlier factor (LOF) classifier after 95 deep features are selected by PCA. Specifically, the owner (the legitimate user) is required to operate on the smartphone to collect data of the accelerometer, gyroscope and magnetometer. Then, we use the optimal augmentation strategy to augment the collected sensor data including the legitimate user's for feature extraction. We specially design a CNN based on Shufflenet V2 [16] to learn and extract deep features from the augmented sensor data. 95 deep features are selected by PCA and then used to train the LOF classifier.

In the authentication stage, based on the sampled sensor data, CAuSe uses the trained CNN to learn and extract features and utilizes the trained LOF classifier to conduct the authentication based on 95 features selected by PCA. If the user is a legitimate user, CAuSe will allow the continuous usage of the smartphone and meanwhile continuously authenticate the user; otherwise, it will require the initial login inputs.

3.1 Data Collection and Preprocessing

Data Collection. The accelerometer and gyroscope are motion sensors, and they can capture the motion patterns of the devices. The magnetometer is a position sensor that records changes in the physical position of the devices. The

three sensors are widely equipped on the modern smart devices. Considering the above advantages, we select the accelerometer, gyroscope, and magnetometer to collect the data for user continuous authentication.

In order to collect the sensor data for CAuSe, we recruited 88 volunteers (44 male and 44 female) to operate on 10 Samsung Galaxy S4 smartphones, each of which was installed a designed virtual keyboard. They were required to participate in 8 sessions, and they used the virtual keyboard to answer 3 questions in each session. For each answer, they entered 250 characters at least. During their operations, we collected data on the three axes of the accelerometer, gyroscope and magnetometer with a sampling rate of 100 Hz.

Data Preprocessing. Since the collected raw sensor data are long time-series streams, we use a sliding window to perform non-repetitive sampling, each containing 2 s-sensor data. In a sliding window, each row represents the sampled sensor data, and each column indicates the x, y, and z axes of a sensor. In order to enable the time-series sensor data to be used as the inputs of a CNN with $shape = (H, W, C)$, we adaptively change the shape of the collected data. Specifically, the three sensor data are regarded as three channels (C), and the rows and columns of the sliding window correspond to H and W, respectively. Ignoring the error in the sampling process and according to the sampling frequency, it can be inferred that $H = 200$.

We divide the 88 volunteers' data into three groups (88 users with 3000 windows): 68 users with 2000 windows $D_{learning}$ for CNN training, 68 users with 1000 windows $D_{positive}$ as legitimate users' testing dataset for feature extraction and classifier training, and 20 users with 3000 windows $D_{negative}$ as impostors' testing dataset for feature extraction and classifier training. $D_{learning}$ are fitted and transformed by *RobustScaler* in Python library *sklearn.preprocessing*, which ignores outliers in the dataset. $D_{positive}$ and $D_{negative}$ are transformed by the same *RobustScaler*, so that the three groups of data can be consistently normalized for data augmentation.

3.2 Auto Augmentation Search

Search Space. For images, there is spatial correlation among the pixels and other pixels around them, while for sensor data, there is temporal correlation among samples. Therefore, we design specific data augmentation strategies that consider the possible invariant geometric transformation of sensor data in time series. For each input of CNN training sensor data, we sample an augmentation strategy from the search space and apply. Each augmentation strategy is composed of two augmentation methods.

We design the candidate augmentation methods for sensor data:

1) **Rotation:** When users operate on mobile devices, the devices are likely to be flipped or rotated at a certain angle. Accordingly, the x, y, and z axes of the sensors on the devices rotate at the same angle corresponding to the

Cartesian coordinate system. In order to simulate this, we design a rotation method, which rotates the x, y, and z axes of the sampled sensor data by multiplying a rotation matrix to obtain angles of ($-\pi/3$, $-\pi/6$, $-\pi/12$, $\pi/3$, $\pi/6$, $\pi/12$).

2) **Jittering:** Noise can be introduced in the process of sensor data collection which might be caused by environmental disturbance. Jittering function adds a noise matrix generated by a normal distribution with standard deviations of 0.05, 0.25, and 0.5 to the sampled sensor data. Note that we ignore the injection attacks in jittering augmentation [11].
3) **Scaling:** Scaling function multiplies the x, y, and z axes of the sampled sensor data separately by scale factors generated by a normal distribution with standard deviations of 0.05, 0.1, and 0.2.
4) **Permutation:** Since the segmentation position of the fixed window is arbitrary for sensor data collected in a period of time, the position of the event implied in the sub-window in the whole window is meaningless. Permutation function segments the whole sample window to 4, 5, or 8 sub-windows by rows to perturb the temporal location of within-window events.
5) **Magnitude-Warping:** We sample values from a normal distribution with standard deviations of 0.2, 0.4, 0.6, feed them to *scipy.interpolate.cubicSpline* to generate three random smooth curves corresponding to x, y, and z axes, and finally convolute them with the sampled sensor data.
6) **Time-Warping:** Time-Warping function utilizes the aforementioned smooth curves and one dimensional linear interpolation to perturb the temporal location smoothly.
7) **Cropping:** Cropping can diminish the dependency on event locations. In the cropping function, we randomly select different numbers of window rows (e.g. 10, 20, or 30) and set values of these selected window rows to 0.

Seven augmentation functions with specific magnitude parameters make up a total of 24 augmentation methods. In our designed augmentation strategy search space, each augmentation strategy consists of 2 augmentation methods orderly and repeatable. In other words, there are totally 24^2 strategies in the augmentation strategy search space.

Search Pipeline. Inspired by Lin et al.'s work [16], we adapt distribution optimization to the continuous authentication area to search an optimal data augmentation strategy for time-series sensor data. As mentioned, since each augmentation strategy consists of two augmentation methods and there are 24 augmentation methods in total, there are 24^2 strategies in the designed augmentation strategy search space. Thus, we first initialize a 24^2 matrix sampled from a uniform distribution as the augmentation probability distribution θ. The probability of the kth augmentation strategy p_θ can be formulated as:

$$p_\theta(S_k) = \frac{\frac{1}{1+e^{-\theta_k}}}{\sum_{i=1}^{K} \frac{1}{1+e^{-\theta_k}}} \tag{1}$$

where $\theta \in R^K$, and S_k indicates the kth data augmentation strategy candidate.

Next, we perform the auto augmentation strategy search. We take an epoch t of total T epochs in model training process. Each input will be applied with a randomly chosen augmentation strategy for each batch b of total B batches. Since the validation accuracy $acc(w^*)$ of the network model is only decided by the optimal network model parameters w^* and the model training process is only influenced by the augmentation strategies applied to each input, the augmentation probability distribution matrix θ is defined as a variable matrix with gradient about the network model parameters w^*. However, it is a tricky problem to calculate the gradient of validation accuracy $acc(w^*)$ with respect to θ. To approximate the gradient, we execute the following steps four times for epoch t:

1) Sample and apply an augmentation strategy for each input, train the network model with augmented inputs, obtain the validation accuracy w', and record the network parameters;
2) Make gradient back propagation for θ, update values of θ, and then clear the gradient of θ;
3) Save the network parameters with the highest w' as the initial network parameters for next epoch.

Based on the reinforcement learning and Monte-Carlo sampling, at the end of epoch t, the cumulative gradient can be approximately formulated as:

$$\nabla_\theta \Gamma(\theta) \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{I \times B} \nabla_\theta log(p_\theta(S_{k(j),n}))acc(w,n) \tag{2}$$

where N denotes the total times of network training and $acc(w,n)$ indicates the validation accuracy of the nth network. Network parameters with the highest validation accuracy will be broadcast to the network before the next epoch. After sufficient epochs of parameters updates, the augmentation probability distribution converges. The augmentation strategy with the highest probability is the optimal augmentation strategy we search. Note that the network model architecture is the same to the designed CNN architecture.

3.3 Feature Extraction

In this section, we design a CNN-based deep feature extraction method, which consists of feature learning and feature selection. In the following, we first elaborate the design of the CNN and then detail the CNN-based feature extraction.

CNN Design. We design the architecture of the CNN inspired by Shufflenet V2 [18], as illustrated in Table 1, for auto augmentation search, feature learning and extraction. As demonstrated in Table 1, the designed CNN is composed of a 2D convolutional layer (Conv2d), a 2D max pooling (MaxPooling2d), a stack of Shufflenet V2 units grouped into three stages (Stage 1, Stage 2, and Stage 3),

Table 1. CNN architecture.

Layer	Output	# Kernel	KSize	Stride	Parameter	Repeat
Sensor	$200 \times 3 \times 3$	–	–	–	–	–
Conv2d (BN+ReLU)	$100 \times 3 \times 24$	24	3×3	(2,1)	672	1
MaxPooling2d	$50 \times 3 \times 24$	–	3×3	(2,1)	–	1
Stage 1	$25 \times 3 \times 48$	48	–	(2,1)	2760	1
	$25 \times 3 \times 48$	48	–	(1,1)	1728×3	3
Stage 2	$13 \times 3 \times 96$	96	–	(2,1)	8976	1
	$13 \times 3 \times 96$	96	–	(1,1)	5760×7	7
Stage 3	$7 \times 3 \times 96$	192	–	(2,1)	31776	1
	$7 \times 3 \times 96$	192	–	(1,1)	20736×3	3
Conv2d (BN+ReLU)	$7 \times 3 \times 1024$	1024	1×1	(1,1)	197632	1
GlobalAveragePooling2d	$1 \times 1 \times 1024$	–	7×3	–	–	1
Dense	$CN \times 1$	–	–	–	69700	1

another Conv2d, a 2D global average pooling (GlobalAveragePooling2d), and a dense layer. We adopt BN and ReLu right after each Conv2d. In addition, Stages 1, 2, and 3 are composed of the building blocks of a basic unit followed by several basic units for spatial down sampling. 'CN' represents class number for CNN training (class_num).

Feature Learning. Based on the optimal strategy obtained from the offline stage, $D_{learning}$ are augmented in the registration stage. As illustrated in Table 1, with the augmented data, there are 1800 (3 sensors $\times$ 2 s $\times$ $100Hz$ $\times$ 3 axes) samples in a 2s-sliding window. The first Conv2d layer with 24 filters of 3×3 and stride of (2,1) followed by a MaxPooling2d with kernel size of 3×3 and stride of (2,1), aims to make down sampling and increase channels. Then, three stages of a basic unit with stride (2,1), and several units for spatial down sampling with stride (1,1) are applied, where Stage 1 repeats 3 times of the unit for spatial down sampling, Stage 2 repeats 7 times, and Stage 3 repeats 3 times. Next, there is another Conv2d layer with 1024 filters of 1×1 and stride of (1,1) followed by a GlobalAveragePooling2d layer and a dense layer. The total parameters of the designed CNN are 419,228 and the second Conv2d layer contributes the most parameters (19,7632 parameters). The outputs of the GlobalAveragePooling2d are deep features learned from the sensors of the accelerometer, gyroscope and magnetometer.

Feature Selection. We use the principal component analysis (PCA) to select appropriate number of deep features for the classifier based on the CNN-extracted features. Based on the experiments in Sect. 4.2, PCA selects 95 deep features for the LOF classifier to conduct the authentication.

3.4 Authentication with LOF Classifier

With the 95 PCA-selected deep features, CAuSe utilizes the local outlier factor (LOF) classifier to identify users. LOF measures the local deviation of the data point to its neighbors, which decides whether a data point is an outlier using the anomaly score estimated by k-nearest neighbors based on a given distance metric. A data point with a substantially lower density than its neighbors will be regarded as an outlier [4].

In the registration stage, CAuSe generates the legitimate user's profile from the training data and the LOF classifier is trained by PCA-selected deep features. In the authentication stage, the trained LOF classifier classifies the PCA-selected deep features from the sampled sensor data. Based on the trained classifier and the sampled data while using the device, CAuSe authenticates the current user as a legitimate user or an impostor. If the user is a legitimate user, CAuSe will allow the continuous usage of the smartphone and meanwhile continuously authenticate the user; otherwise, it will require the initial login inputs.

4 Performance Evaluation

In this section, we start with experimental settings, then investigate the performance of CAuSe in terms of optimal feature number, and evaluate the effectiveness of auto augmentation search and optimal strategy, respectively.

4.1 Experimental Settings

Network Model Training. With the inputs of $D_{learning}$, 80% of the data are used for training and the rest 20% for testing, with a batch size of 128. We use the cross entropy as the loss function and the stochastic gradient descent (SGD) optimizer to update the learning rate. The initial learning rate is 0.2, and it complies with an exponential decay of decay_step = 1000 and decay_rate = 0.96. If the lowest validation loss remains for 10 continuous epochs or the network training process exceeds 150 epochs, the training process stops. The network with the lowest loss is used as the trained model.

Auto Augmentation Strategy Search. The parameters of the augmentation distribution initialize as a 24×24 matrix with initial values from a uniform distribution. We use Adam optimizer with learning rate 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight_decay = 0. The distribution parameters are updated 150 times in total.

Table 2. Accuracy (SD) % for different classifiers with varying feature numbers

Classifier\Number	5	35	55	75	95	115	135	155	175	195
OC-SVM	91.27	**94.58**	93.45	90.57	86.55	81.97	77.24	72.90	69.14	66.00
	(4.10)	(2.53)	(1.98)	(1.95)	(2.13)	(2.58)	(2.78)	(3.26)	(3.68)	(4.11)
IF	87.25	93.26	94.71	95.28	95.68	95.95	**96.03**	95.96	95.80	95.58
	(7.17)	(4.07)	(3.32)	(2.90)	(2.49)	(2.15)	(1.92)	(1.80)	(1.78)	(1.77)
LOF	80.69	92.51	94.45	95.40	**96.93**	96.79	96.66	96.38	95.97	95.77
	(11.44)	(5.91)	(3.93)	(2.84)	**(1.80)**	(1.92)	(2.05)	(2.10)	(2.38)	(2.48)

Classifier Training. To train the LOF classifier, we randomly select 1 legitimate user from $D_{positive}$ for 20 times. With the 1000-window data, we use 10-fold cross validation to obtain 900-window training dataset and 100-window positive testing dataset. We also randomly select 100-window from $D_{negative}$ as the negative testing dataset.

Evaluation Metric. We utilize three evaluation metrics: accuracy, F_1-score, EER to evaluate the effectiveness of CAuSe. Accuracy is the percentage ratio of the total number of correct authentication against the total number of authentication, defined as: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$. F_1-score is defined as: $F_1 = \frac{2TP}{TP+FP+FN}$. EER is the point where FAR equals to FRR.

4.2 Feature Number and Classifier Parameter

We conduct experiments to investigate classifier selection and optimal feature number selected by PCA. We consider three classifiers of OC-SVM, IF, and LOF for classifier selection and vary feature numbers for optimal feature number. We compute the accuracy (standard deviation) of CAuSe with the three classifiers as the feature number increases from 5 to 195, as tabulated in Table 2. As shown in Table 2, the accuracy gradually increases with the feature number growing until an optimal number and then slightly decreases for all the classifiers. For OC-SVM, 35 features selected by PCA reach the best accuracy of 94.58% and for IF, 135 features achieve 96.03% accuracy. However, LOF with 95 features selected by PCA reaches the highest accuracy of 96.93% and the lowest SD of 1.80%. Therefore, we use PCA to select 95 deep features for the LOF classifier.

In addition, based on the optimal numbers of features, we utilize the grid search to seek the best parameter combinations for classifiers of the OC-SVM, IF, and LOF. We list the classifiers, number of features, and optimal parameter combination in Table 3. As shown in Table 3, the LOF classifier with 95 deep features obtains the optimal parameters of $n\_neighbors = 800$ and $p = 1$.

Table 3. Optimal parameter combinations

Classifier	# Feature	Optimal parameter combination
OC-SVM	35	$\mu = 0.0001$, $\gamma = 0.015625$
IF	135	`n_estimators` $= 900$
LOF	95	`n_neighbors` $= 800, p = 1$

Table 4. Row and column corresponding to the optimal augmentation strategy

Epoch	0–2	3–4	5	6	7	8–10	11–16
(Row, column)	(7,5)	(13,22)	(6,22)	(5,16)	(7,14)	(17,2)	(7,14)
Epoch	17–19	20–26	27–45	46–92	93–107	108–130	131–149
(Row, column)	(5,1)	(7,14)	(16,14)	(13,22)	(20,20)	(13,22)	(20,20)

4.3 Auto Augmentation Search

We select dataset $D^{100}_{learning}$ with 100-window per user from $D_{learning}$ to conduct the evaluation of the auto augmentation search, due to the limitations of computer memory and GPU. In the auto augmentation search, we instantiate the augmentation distribution parameters as a 24×24 matrix and save the corresponding matrix for each epoch. Based on the saved matrices, we sum the rows of each matrix, normalize all rows for each epoch, and visualize rows varying with the epoch grows. We calculate the marginal distribution of parameters of the first augmentation method of each strategy, as illustrate in Fig. 2. As illustrated in Fig. 2, the deeper the red, the closer the probability of the method is to 1, and the deeper the blue, the closer the probability is to 0. As the search progresses, the edge probability of each method either converges to 0 or 1. When the search is complete, the edge probability of the method in rows of 4, 6, 10, 17, 18, and 21 is higher. From Fig. 2, it can be seen that during random training, the parameter values of some augmentation methods gradually increase while others gradually decrease, which indicates that some augmentation methods are abandoned while the probability of other augmentation methods is increasing.

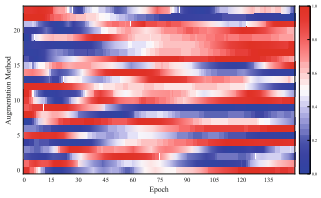

Fig. 2. Marginal distribution of augmentation operations. (Color figure online)

In addition, after updating the parameters of the augmentation probability distribution at the end of each epoch, we calculate the probability for each augmentation strategy by Eq. (1) and record the row and column of the corresponding optimal augmentation strategy, as shown in Table 4.

It can be seen that during the training process, with the update of the probability distribution parameters, the optimal strategy (the strategy with the highest probability) is also constantly changing, and at the end of the training, a row and a column $(20, 20)$ of the optimal strategy for local convergence can be

Table 5. Optimal parameter combinations

Network	Accuracy	F_1-score	EER
Network without augmentation	85.37 (7.61)	87.54 (5.71)	7.87 (3.59)
Network searched by auto augmentation	88.88 (6.64)	90.24 (5.29)	6.50 (3.38)

Table 6. Accuracy (SD) % on Different Strategies with Varying Data Sizes

Strategy\Data size	60	80	100	200	400
No augmentation	56.77 (6.33)	54.67 (3.90)	85.37 (7.61)	90.06 (5.95)	92.14 (5.31)
Rota-3+MagnWarp0.2	79.32 (8.18)	81.45 (8.10)	82.61 (7.53)	85.10 (8.20)	90.11 (6.02)
Perm8+Rotate12	84.34 (7.65)	86.75(6.93)	85.79 (7.39)	87.49 (7.36)	89.50 (7.02)
TimeWarp0.6+Perm2	87.99 (6.98)	88.05 (6.43)	89.59 (6.15)	90.76 (6.58)	92.47 (5.49)
Our strategy	88.65 (7.40)	88.70 (7.51)	91.12 (5.69)	91.89 (5.33)	93.19 (4.85)

Table 7. F_1 Score (SD) % on different strategies with varying data sizes

Strategy\data size	60	80	100	200	400
No augmentation	68.89 (3.34)	68.43 (2.09)	87.54 (5.71)	91.16 (4.80)	92.87 (4.37)
Rota-3+MagnWarp0.2	83.18 (5.74)	84.68 (5.82)	85.46 (5.52)	87.40 (5.98)	91.23 (4.84)
Perm8+Rota12	86.73 (5.50)	88.55 (5.27)	87.82 (5.50)	89.18 (5.70)	90.79 (5.48)
TimeWarp0.6+Permu2	89.55 (5.44)	89.52 (5.02)	90.76 (4.94)	91.78 (5.27)	93.19 (4.53)
Our strategy	90.12 (5.78)	90.16 (5.70)	92.00 (4.70)	92.64 (4.39)	93.77 (4.09)

Table 8. EER (SD) % on different strategies with varying data sizes

Strategy\data size	60	80	100	200	400
No Augmentation	39.12 (10.61)	37.79 (10.32)	7.87 (3.59)	5.62 (2.82)	4.65 (2.66)
Rota-3+MagnWarp0.2	10.06 (3.77)	9.72 (4.26)	8.62 (3.82)	6.65 (3.24)	5.34 (3.00)
Perm8+Rota12	9.07 (4.29)	7.43 (3.34)	7.90 (3.58)	6.39 (3.31)	5.51 (2.87)
TimeWarp0.6+Perm2	7.21 (3.80)	6.73 (3.12)	6.53 (3.28)	5.24 (2.98)	4.35 (2.73)
Our Strategy	6.67 (3.60)	6.64 (3.71)	5.68 (3.34)	4.99 (2.74)	3.90 (2.48)

obtained. It can be considered that Time-Warping (0.6) + Time-Warping (0.6) is a relatively good augmentation strategy found in our dataset in the entire search space with a CNN structure in Table 2 trained to converge. We also illustrate the continuous authentication performance of the network model trained by auto augmentation search and the network model obtained by training the same network structure without augmentation in Table 5.

4.4 Optimal Strategy

In the above experiments, we searched for an optimal strategy that located in the 20th row and 20th column of the probability distribution parameter matrix. The optimal strategy is a strategy composed of two identical augmentation operations Time-Warping(0.6)+Time-Warping(0.6). In order to demonstrate the superiority of the strategy, we randomly select 3 strategies from the search space to augment different size of data and compute the accuracy, F_1-score and EER,

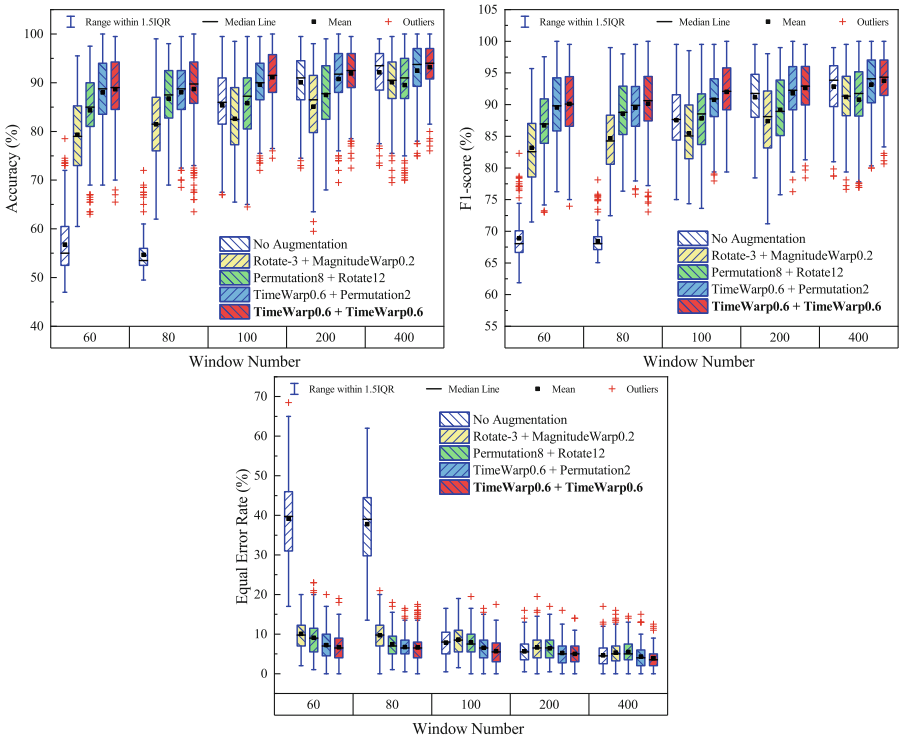

Fig. 3. Accuracy, F_1 score, and EER for different strategies with varying data sizes.

respectively. The corresponding results are tabulated in Tables 6, 7, and 8, and are plotted in Fig. 3.

We can obtain observations from Tables 6, 7, and 8, and Fig. 3:

1) When there is no data augmentation, as the data size increases, the authentication performance gradually improves, which indicates that the amount of real data is positively correlated with the authentication performance.

2) When the data size comes to 100, the EERs for the strategies of the Rotate(-3)+MagnitudeWarp(0.2) and Permutation(8)+Rotate(12) are even higher than that without data augmentation strategy, which indicates that the two strategies are relatively worse augmentation strategies.

3) On all data sizes, the strategy of Time-Warping(0.6)+Time-Warping(0.6) achieves the best authentication performance on the accuracy (93.19%), F_1 score (93.77%), and EER (3.9%), which proves that the optimal strategy searched by the proposed auto augmentation is optimal on different data sizes.

4.5 Comparison with Representative Schemes

We compare CAuSe to four representative continuous authentication schemes with data augmentation approaches, as listed in Table 9. As illustrated in Table 9,

Table 9. Comparison with representative schemes

Scheme	Data source	Data augmentation approach	Accuracy
SensorAuth [13]	Acc., Gyr.	Perm., sample, scale, crop, jitter	EER: 6.29% (dataset size 200)
EchoPrint [30]	Face image	Rotation	BAC: 81.78% (vision features)
SensorCA [14]	Acc., Gyr., Mag.	Rotation	EER: 3.7% (SVM-RBF)
HMOG [23]	Acc., Gyr., Mag., Tou.	HMOG with tap characteristics	EER: 7.16% (walking)
CAuSe	Acc., Gyr., Mag.	Auto Augmentation Search	Accuracy: 96.93% (LOF)

we show the data source, data augmentation approaches, and accuracy for all the schemes with data augmentation. Specifically, SensorAuth explores five data augmentation approaches of permutation, sampling, scaling, cropping, and jittering to create additional acccelerometer and gyroscope data and achieves an EER of 6.29% with dataset size 200 by combining the five approaches [13]. EchoPrint uses the projection matrix rotation imitating different camera poses to augment new face images and obtains 81.78% balanced accuracy (BAC) with vision features [30]. SensorCA applies matrix rotation on accelerometer, gyroscope and magnetometer data to reach an EER of 3.7% on the SVM-RBF classifier [14]. HMOG augments HMOG features with tap characteristics (e.g. tap duration and contact size) to obtain 7.16% EER for walking [23]. Different from these continuous authentication schemes with data augmentation, CAuSe exploits the auto augmentation search to find an optimal strategy for data augmentation of the accelerometer, gyroscope and magnetometer, and achieves the best accuracy of 96.93% on the LOF classifier.

5 Conclusion

To address the shortage of training data and improve the feature discriminability, we propose CAuSe, a CNN-based continuous authentication on smartphones using auto augmentation search, where the CNN is specially designed for deep feature extraction and the auto augmentation search is exploited for finding the optimal augmentation strategy. Although we take significant efforts to validate the effectiveness of CAuSe, there are some limitations in this work: 1) power consumption of CAuSe on smartphones, 2) impact of various attacks on CAuSe, and 3) privacy concerns on dataset collection and transportation. In future, we will consider issues of the energy, privacy and security for continuous authentication approaches.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China under Grant 62072061 and by the Fundamental Research Funds for the Central Universities under Grant 2021CDJQY-026.

References

1. Abeni, P., Baltatu, M., D'Alessandro, R.: Nis03-4: Implementing biometrics-based authentication for mobile devices. In: IEEE Globecom 2006. pp. 1–5. IEEE (2006)
2. Al-Hadithy, N., Gikas, P.D., Al-Nammari, S.S.: Smartphones in orthopaedics. Int. Orthop. **36**(8), 1543–1547 (2012)
3. Bonneau, J., Herley, C., van Oorschot, P.C., Stajano, F.: Passwords and the evolution of imperfect authentication. Commun. ACM **58**(7), 78–87 (2015)
4. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. SIGMOD Rec. **29**(2), 93–104 (2000)
5. Centeno, M.P., Guan, Y., van Moorsel, A.: Mobile based continuous authentication using deep features. In: Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning, pp. 19–24 (2018)
6. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: learning augmentation strategies from data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 113–123 (2019)
7. DeVries, T., Taylor, G.W.: Dataset augmentation in feature space. arXiv preprint arXiv:1702.05538 (2017)
8. Feng, H., Fawaz, K., Shin, K.G.: Continuous authentication for voice assistants. In: Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, pp. 343–355 (2017)
9. Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D.: Touchalytics: on the applicability of touchscreen input as a behavioral biometric for continuous authentication. IEEE Trans. Inf. Forensics Secur. **8**(1), 136–148 (2012)
10. Gascon, H., Uellenbeck, S., Wolf, C., Rieck, K.: Continuous authentication on mobile devices by analysis of typing motion behavior. Sicherheit 2014-Sicherheit, Schutz und Zuverlässigkeit (2014)
11. Gonzalez-Manzano, L., Mahbub, U., de Fuentes, J.M., Chellappa, R.: Impact of injection attacks on sensor-based continuous authentication for smartphones. Comput. Commun. **163**, 150–161 (2020)
12. Li, X., Zhang, W., Ding, Q., Sun, J.-Q.: Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. J. Intell. Manuf. **31**(2), 433–452 (2018). https://doi.org/10.1007/s10845-018-1456-1
13. Li, Y., Hu, H., Zhou, G.: Using data augmentation in continuous authentication on smartphones. IEEE Internet Things J. **6**(1), 628–640 (2018)
14. Li, Y., Hu, H., Zhou, G., Deng, S.: Sensor-based continuous authentication using cost-effective kernel ridge regression. IEEE Access **6**, 32554–32565 (2018)
15. Lim, S., Kim, I., Kim, T., Kim, C., Kim, S.: Fast autoaugment. arXiv preprint arXiv:1905.00397 (2019)
16. Lin, C., et al.: Online hyper-parameter learning for auto-augmentation strategy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6579–6588 (2019)
17. Luo, Y., Lu, B.L.: Eeg data augmentation for emotion recognition using a conditional wasserstein gan. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2535–2538. IEEE (2018)
18. Ma, N., Zhang, X., Zheng, H.-T., Sun, J.: ShuffleNet V2: practical guidelines for efficient CNN architecture design. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 122–138. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_8

19. Mahbub, U., Patel, V.M., Chandra, D., Barbello, B., Chellappa, R.: Partial face detection for continuous authentication. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 2991–2995. IEEE (2016)
20. Mosenia, A., Sur-Kolay, S., Raghunathan, A., Jha, N.K.: Caba: continuous authentication based on bioaura. IEEE Trans. Comput. **66**(5), 759–772 (2016)
21. Muaaz, M., Mayrhofer, R.: An analysis of different approaches to gait recognition using cell phone based accelerometers. In: Proceedings of International Conference on Advances in Mobile Computing & Multimedia, pp. 293–300 (2013)
22. Quan, F., Fei, S., Anni, C., Feifei, Z.: Cracking cancelable fingerprint template of ratha. In: 2008 International Symposium on Computer Science and Computational Technology, vol. 2, pp. 572–575. IEEE (2008)
23. Sitová, Z., Šeděnka, J., Yang, Q., Peng, G., Zhou, G., Gasti, P., Balagani, K.S.: HMOG: new behavioral biometric features for continuous authentication of smartphone users. IEEE Trans. Inf. Forensics Secur. **11**(5), 877–892 (2015)
24. Tschuchnig, M.E., Ferner, C., Wegenkittl, S.: Sequential IoT data augmentation using generative adversarial networks. In: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4212–4216. IEEE (2020)
25. Um, T.T., et al.: Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In: Proceedings of the 19th ACM International Conference on Multimodal Interaction, pp. 216–220 (2017)
26. Wang, D., Gu, Q., Huang, X., Wang, P.: Understanding human-chosen pins: characteristics, distribution and security. In: 2017 ACM on Asia Conference on Computer and Communications Security (ASIA CCS 2017), pp. 372–385. ACM (2017)
27. Wang, D., Zhang, Z., Wang, P., Yan, J., Huang, X.: Targeted online password guessing: an underestimated threat. In: 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS 2016), pp. 1242–1254. ACM (2016)
28. Xu, H., Zhou, Y., Lyu, M.R.: Towards continuous and passive authentication via touch biometrics: an experimental study on smartphones. In: 10th Symposium On Usable Privacy and Security (SOUPS 2014), pp. 187–198 (2014)
29. Zhang, X., Wang, Q., Zhang, J., Zhong, Z.: Adversarial autoaugment. arXiv preprint arXiv:1912.11188 (2019)
30. Zhou, B., Lohokare, J., Gao, R., Ye, F.: Echoprint: two-factor authentication using acoustics and vision on smartphones. In: MobiCom, pp. 321–336. ACM (2018)
31. Zhu, X., Liu, Y., Qin, Z., Li, J.: Data augmentation in emotion classification using generative adversarial networks. arXiv preprint arXiv:1711.00648 (2017)