SearchAuth: Neural Architecture Search based Continuous Authentication Using Auto Augmentation Search

YANTAO LI, JIAXING LUO, and SHAOJIANG DENG, Chongqing University, China
GANG ZHOU, Department of Computer Science, William & Mary, USA

Mobile devices have been playing significant roles in our daily lives, which has made device security and privacy protection extremely important. These mobile devices storing user sensitive and private information, therefore, need rigorous user authentication mechanisms. In this paper, we present SearchAuth, a novel continuous authentication system on smartphones exploiting a neural architecture search (NAS) to find an optimal network architecture and an auto augmentation search (AAS) to more effectively train the optimal network along with the best data augmentation policies, by leveraging the accelerometer, gyroscope and magnetometer on smartphones to capture users' behavioral patterns. Specifically, SearchAuth consists of three stages, i.e. the offline stage, registration stage, and authentication stage. In the offline stage, we utilize the NAS on sensor data of the accelerometer, gyroscope and magnetometer to find an optimal network architecture based on the designed search space. With the optimal network architecture, namely NAS-based model, the AAS automatically optimizes the augmentation of the input data for more effectively training the model that is for feature extraction. In the registration stage, we use the trained NAS-based model to learn and extract deep features from the legitimate user's data, and train the LOF classifier with 55 features selected by the PCA. In the authentication stage, with the well-trained NAS-based model and LOF classifier, SearchAuth identifies the current user as a legitimate user or an impostor when the user starts operating a smartphone. Based on our dataset, we evaluate the performance of the proposed SearchAuth, and the experimental results demonstrate that SearchAuth surpasses the representative authentication schemes by achieving the best accuracy of 93.95%, F1-score of 94.30%, and EER of 5.30% on the LOF classifier with dataset size of 100.

CCS Concepts: • **Security and privacy** → **Biometrics**; • **Human-centered computing** → *Collaborative and social computing devices*; • **Computing methodologies** → *Neural networks.*

Additional Key Words and Phrases: Continuous authentication, Neural architecture search, Auto augmentation search, LOF classifier, Accuracy

ACM Reference Format:
Yantao Li, Jiaxing Luo, Shaojiang Deng, and Gang Zhou. 2023. SearchAuth: Neural Architecture Search based Continuous Authentication Using Auto Augmentation Search. *ACM Trans. Sensor Netw.* x, x, Article 1 (May 2023), 23 pages. https://doi.org/10.1145/3397179

1 INTRODUCTION

With the rapid development of communication technologies, mobile devices have played a significant role in our daily lives, which makes privacy protection in mobile devices extremely important, since we prefer to store a lot of sensitive and private information on them. Even since 2011, sales of smartphones have exceeded sales of personal computers [1]. However, due to the high-frequency

Authors' addresses: Yantao Li; Jiaxing Luo; Shaojiang Deng, Chongqing University, 174 Shapingba Central St., Chongqing, 400044, China, {yantaoli,luojiaxing,sj_deng}@cqu.edu.cn; Gang Zhou, Department of Computer Science, William & Mary, 251 Jamestown Rd., Williamsburg, VA, 23185, USA, gzhou@cs.wm.edu.

1550-4859/2023/5-ART1 $15.00
https://doi.org/10.1145/3397179

usage and information interaction among these devices (e.g. smartphones, smartwatches, and tablets), it is difficult to prevent personal information leakage and illegal access by the one-time authentication. It identifies users only at the time of initial logging-in, such as personal identification numbers (PINs), and fingerprints. The PIN is the most basic one-time authentication approach for mobile devices compared to others, which faces a serious threat of online guessing and even longer PIN only attains marginally improved security [2, 3]. Wang et al. 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 [4]. Biometric information cannot be acquired by direct covert observation, but once biological information is stolen, it is not naturally available to reissue [5]. For example, fingerprint recognition can be cracked by people with ulterior motives obtaining legitimate users' fingerprints left on the screen. Therefore, there is a severe security and privacy threat in one-time authentication mechanisms that impostors can easily gain access to a mobile device when the legitimate user leaves the supervision of the device after the initial authentication (e.g., the screen is unlocked).

Compared with traditional one-time authentication mechanisms, continuous or implicit authentication approaches can provide an additional line of defense by designing a non-intrusive and passive security countermeasure [6]. The current continuous authentication mechanisms essentially use built-in sensors and accessories to frequently collect physiological or behavioral biometrics to identify the legitimacy of users, such as voice [7], face patterns [8], touch gestures [9], typing motion [10], and gait dynamics [11]. There are two main phases for continuous authentication systems: user registration phase and continuous authentication phase. During the user registration phase, owners of mobile devices are usually required to perform some operations on the devices to collect biometrics to identify them. During the continuous authentication phase, the systems collect the users' sensor readings at their regular intervals to determine whether they are the device owners. If the system finds that the current user is an impostor, it will lock the device to prevent the owner's privacy from leaking. The accelerometer, gyroscope, and magnetometer are the most commonly used sensors for collecting behavioral biometrics without users' notice. Both the accelerometer and gyroscope are motion sensors that can monitor the users' motion on the devices, while the magnetometer records the users' general environment. However, how to design a lightweight and highly efficient continuous authentication model still faces challenges. On the one hand, although deep neural networks have shown superiority on behavioral biometrics by learning high-level representative features from input data and extracting discriminative features as the outputs [12, 13], they are increasingly deeper and larger, and require more computation resource with fixed architectures. Therefore, it is challenging to design a lightweight and accurate deep model architecture for feature extraction. On the other hand, data augmentation methods, such as flipping, cropping, color dithering [14, 15], and generative adversarial networks (GANs) [16, 17], are very common techniques in the field of image recognition, which help to disclose unexplored input space, prevent overfitting, and improve the generalization ability of classification models. However, currently there are few data augmentation methods dedicating to time-series sensor data because they are quite different from image data, and thus most of the current data augmentation methods cannot be used to create time-series data directly. Therefore, it is also challenging to collect a large amount of high-quality sensor data for model training, which costs lots of time and computing resources. Overall, the challenges for current continuous authentication systems are how to find an optimal model architecture for deep feature extraction, and how to search the best data augmentation policies for model training.

To address the above challenges, we are among the first to utilize neural architecture search to find an optimal network architecture for deep feature extraction and use the auto augmentation search to more effectively train the optimal model along with the best data augmentation strategies.

In this paper, we present SearchAuth, a neural architecture Search based continuous Authentication on smartphones using auto augmentation Search, as an extension of our previous work [18]. In SearchAuth, the user performs a wide operation based on typing gestures on a smartphone for user authentication, which leverages the accelerometer, gyroscope and magnetometer on smartphones to capture users' behavioral patterns. Specifically, SearchAuth consists of six modules: data collection, neural architecture search (NAS), auto augmentation search (AAS), feature extraction and selection, classifier training, and authentication. The process of SearchAuth includes three stages of the offline stage, registration stage, and authentication stage. Specifically, in the offline stage, SearchAuth collects sensor data of the accelerometer, gyroscope and magnetometer on smartphones for the NAS and AAS training. Using the preprocessed sensor data, SearchAuth exploits the NAS to search an optimal deep network architecture, namely NAS-based model, with the designed search space based on MobileNetV3 blocks, and utilizes the AAS to automatically optimize the augmentation of the input data for more effectively training the NAS-based model with the designed augmentation strategy search space. In the registration stage, SearchAuth utilizes the trained NAS-based model to learn and extract deep features from the legitimate user's data, and trains the local outlier factor (LOF) classifier after 55 deep features are selected by the principal component analysis (PCA). In the authentication stage, based on the sampled sensor data, SearchAuth uses the trained NAS-based model to learn and extract features and utilizes the trained LOF classifier to conduct the authentication based on the 55 PCA-selected features. Based on our dataset, we evaluate the effectiveness of SearchAuth in terms of the feature number and classifier parameter, NAS-based model, AAS, efficiency of optimal strategy, and comparison with representative schemes. The experimental results demonstrate that the NAS-based model outperforms representative network models by reaching the best accuracy of 92.08%, F1-score of 92.32%, and EER of 6.25% on the LOF classifier with dataset size of 100 based on the original dataset, and SearchAuth surpasses the representative authentication schemes by achieving the best accuracy of 93.95%, F1-score of 94.30%, and EER of 5.30% on the LOF classifier with dataset size of 100, respectively.

The main contributions of this work are summarized as follows:

- We present SearchAuth, a neural architecture search based continuous authentication on smartphones using auto augmentation search, leveraging the smartphone built-in accelerometer, gyroscope and magnetometer to capture users' behavioral patterns. SearchAuth consists of six modules, i.e. data collection, neural architecture search, auto augmentation search, feature extraction and selection, classifier training, and authentication.
- We utilize the neural architecture search to find an optimal network architecture for deep feature extraction, and use the auto augmentation search to more effectively train the optimal model along with the best data augmentation strategies. In addition, the NAS-based model is used as the network model for the auto augmentation search.
- We evaluate the effectiveness of NAS-based model and the performance of SearchAuth, and the experimental results illustrate that the trained NAS-based model reaches the highest accuracy of 92.08% on the original dataset, and SearchAuth achieves the best authentication accuracy of 93.95% on the LOF classifier with dataset size of 100 on our dataset, respectively.

The remainder of this work is organized as follows: Section 2 reviews the state-of-the-art on continuous authentication. We elaborate the SearchAuth design in terms of the system overview, data collection and preprocessing, the NAS and AAS, NAS-based feature extraction, and authentication with LOF classifier, respectively, in Section 3. In Section 4, we evaluate the performance of SearchAuth, and we conclude this work in Section 5.

2 RELATED WORK

In this section, we review the state-of-the-art of the CNN-based continuous authentication systems and data augmentation approaches in continuous authentication systems, respectively.

2.1 CNN-based Continuous Authentication Systems

In the field of continuous authentication, high-precision identification is often inseparable from efficient and effective system architectures. Well-performed continuous authentication systems based on specially designed deep neural networks have been creatively proposed. They can automatically learn high-level representative features from input data and extract discriminative features as the outputs. The main continuous authentication architectures are broadly composed of two phases: the registration phase and authentication phase. During the registration phase, these systems utilize CNNs with fixed architectures to extract deep features from the collected data and then train classifiers with labeled features. During the authentication phase, they exploit the trained classifiers to classify current CNN-extracted features as the legitimate user or impostors. The authors in [19] used a Siamese CNN that learned a distance metric from a large dataset to extract deep features for new user authentication in a mobile based continuous authentication system. In [12], the authors exploited a deep learning autoencoder in a continuous biometric authentication system that relied on user-specific motion patterns while interacting with the smartphone. The authors in [20] used a deep recurrent neural network to capture the subtle motion signatures during password input in a smartwatch authentication framework leveraging the unique motion patterns when users entering passwords as behavioural biometrics. In [21], the authors proposed a reliable biometric recognition schemes for mobile devices based on keystroke dynamics, by applying deep learning technique of CNN to extract discriminative information from the typing behaviors of different users. The authors in [22] utilized the LSTM-based architectures to process sequential sensory data records for capturing the behavioral patterns of users when holding their smartphones in an implicit continuous authentication system. In [23], the authors proposed a two-stream CNN for feature learning in the continuous authentication system which was based on bottleneck structure of MobileNetV2, with both time domain data and frequency domain data of the accelerometer and gyroscope as the network inputs. The authors in [13] specially designed a lightweight CNN based on the basic block of a bottleneck unit with depthwise convolution and down block of a basic block for spatial downsampling to learn and extract discriminative deep features in a continuous authentication system. In [18], the authors designed a CNN-based deep feature extraction method consisting of feature learning and feature selection in a continuous authentication system on smartphones with auto augmentation search.

Different from the aforementioned contributions, we utilize the neural architecture search to automatically find an optimal network architecture for deep feature extraction in a continuous authentication system.

2.2 Data Augmentation in Continuous Authentication Systems

In the field of data augmentation, high-quality and substantial sensor data can significantly improve the accuracy of the continuous authentication systems. Data augmentation based on geometric transformations has been widely used in the field of image recognition, such as permutation, sampling, scaling, cropping, jittering, flipping and rotation. The authors in [14] were the first to use geometric transformations of permutation, sampling, scaling, cropping and jittering as sensor data augmentation approaches for Parkinson's disease monitoring. However, the authors in [15] were among the first to exploit five data augmentation approaches of permutation, sampling, scaling, cropping and jittering in a continuous authentication system that created additional data by applying

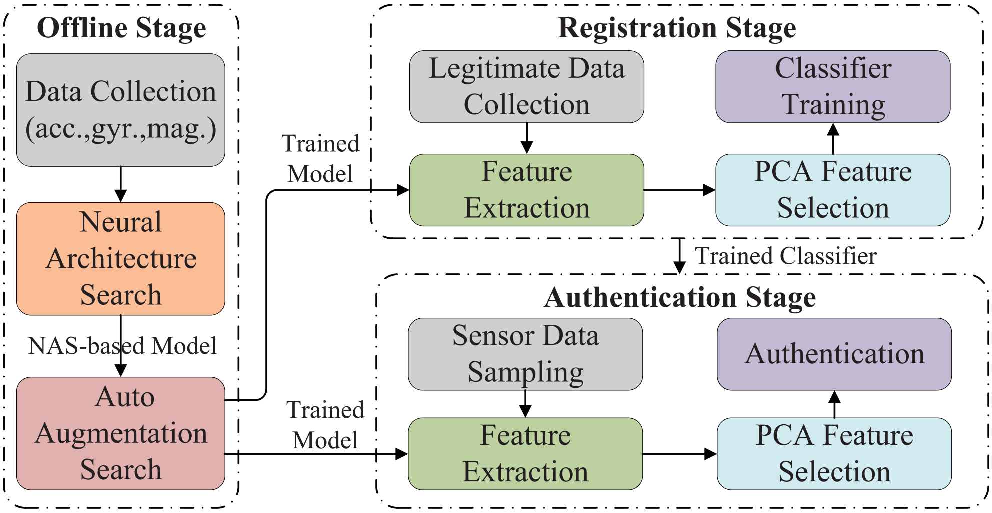

Fig. 1. SearchAuth Architecture.

the transformations on the training data. In [24], the authors exploited rotation data augmentation to create additional data in a sensor-based user authentication system for continuously monitoring users' behavior patterns.

Data augmentation based on generative adversarial networks (GANs) has been used in the area of time-series sensor data. The authors in [16] proposed an emotion classification system using data augmentation with a cycle-consistent adversarial network (CycleGAN). In [17], the authors trained a conditional Wasserstein generative adversarial network (CWGAN) on the electroencephalography (EEG) data to generate additional data for data augmentation. The authors in [25] investigated the possibility of using GANs to augment time-series Internet of Things (IoT) data. In [26], the authors 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. The authors in [27] employed a CWGAN to generate additional sensor data for data augmentation in a smartphone-based continuous authentication system. In [28], the authors utilized a multiscale and multidirection GAN consisting of multiple fully convolutional GANs, each of which is responsible for learning the patch distribution within an image at a different scale and at a different direction for a single-sample-per-person palm-vein identification.

Although these data augmentation approaches have been used in these representative continuous authentication systems, we differ in that we are among the first to exploit the auto augmentation search to find the best data augmentation policies along with a network model training in a continuous authentication system.

3 SEARCHAUTH DESIGN

We elaborate the design of the neural architecture search based continuous authentication system using auto augmentation search, SearchAuth, by leveraging the accelerometer, gyroscope and magnetometer on smartphones to capture users' behavioral patterns. In this section, we begin with the system overview, then describe the data collection and preprocessing, detail the NAS and

AAS, elaborate NAS-based feature extraction, and finally introduce the authentication with LOF classifier.

3.1 System Overview

The key idea underlying our authentication system SearchAuth is to continuously authenticate smartphone users leveraging the accelerometer, gyroscope and magnetometer to sense users' operation on smartphones, where the neural architecture search (NAS) is exploited for finding an optimal network architecture and the auto augmentation search (AAS) is utilized for more effectively training the model along with the best data augmentation policies.

As illustrated in Fig. 1, SearchAuth consists of six modules, i.e. data collection, neural architecture search, auto augmentation search, feature extraction and selection, classifier training, and authentication. The process of SearchAuth includes three stages, i.e. the offline stage, registration stage, and authentication stage. Specifically, in the offline stage, SearchAuth collects sensor data of the accelerometer, gyroscope and magnetometer for the NAS and AAS training, where we recruit volunteers to operate smartphones with sensor data collection tools. With the preprocessed sensor data, SearchAuth exploits the NAS to find an optimal deep network architecture with the designed architecture search space based on MobileNetV3 blocks. Then, SearchAuth utilizes the AAS to automatically optimize the augmentation of the input data for more effectively training the NAS-based model with the designed augmentation strategy search space. The AAS process is embodied as a hyper-parameter learning problem while formulating the augmentation strategy as a parameterized probability distribution. After AAS, the NAS-based model has been well-trained as the network model for the AAS while the probability distribution converges. The trained NAS-based model is used for the deep feature extraction in the registration stage and the authentication stage, respectively. In the registration stage, SearchAuth uses the trained NAS-based model to extract deep features after the same data preprocessing on a legitimate user's samples, and trains the LOF classifier after 55 deep features are selected by the PCA. In particular, the legitimate user is required to operate on a smartphone to collect sensor data of the accelerometer, gyroscope and magnetometer. Then, with the well-trained NAS-based model, SearchAuth performs data preprocessing, learns and extracts deep features from the legitimate user's sensor data and 55 deep features selected by the PCA are then used to train the LOF classifier. In the authentication stage, with the sampled sensor data for the current user, SearchAuth uses the trained NAS-based model to learn and extract features and utilizes the trained LOF classifier to conduct the user authentication based on the 55 features selected by the PCA. If the user is a legitimate user, SearchAuth will allow the continuous usage of the smartphone and meanwhile continuously authenticate the user; otherwise, it will require the initial login inputs.

In addition, sensor-based continuous authentication on smartphones is an implicit process to identify a user by extracting behavioral patterns, such as touch gesture and gait. However, the above sensor-based continuous authentication is threatened by mimic attack. In this attack, an adversary first observes the way that a legitimate user performs to pass the authentication, and then practices to mimic the user's behaviors for conducting the attack. Note that sensor-based continuous authentication works on condition that user behavior can be sensed.

3.2 Data Collection and Preprocessing

In this section, we first introduce the data collection and then detail the data preprocessing.

3.2.1 Data Collection.

The accelerometer captures coarse-gained motion patterns by measuring the acceleration force. The gyroscope captures fine-grained motion patterns by measuring a device's rate of rotation. The magnetometer records a user's general environment by measuring the ambient geomagnetic field [29]. The three sensors are widely equipped on the modern mobile devices. Considering the above advantages, we select the accelerometer, gyroscope, and magnetometer to collect the sensor data for our continuous authentication system.

In order to collect the sensor data for SearchAuth, 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 $100Hz$ [30].

3.2.2 Data preprocessing. Since the collected raw sensor data are long time-series streams, we use a sliding window to conduct non-repetitive sampling, each containing 2-second 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 deep network with $shape = (C, H, W)$, we adaptively adjust 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 $H = 200$.

To simulate the data flow during the system running in a real environment, we divide the 88 volunteers into 68 legitimate users with 3000 windows $LU^{3000\times68}$ for the registration and offline stages and 20 impostors with 3000 windows $IU^{3000\times20}$ for the authentication stage. In our experiments, one of the legitimate users will be randomly-selected as the legitimate user $LU^{3000}_{legitimate}$ in the registration stage while the rest 67 users will be regarded as pre-collected data in the offline stage. For the 68 legitimate users, the selected legitimate user with 1000 windows $LU^{1000}_{legitimate}$ (from $LU^{3000}_{legitimate}$) is used for feature extraction, classifier training and testing while the rest 67 users with 100 windows $LU^{100\times67}_{learning}$ (from $LU^{2000\times67}_{learning}$ of $LU^{3000\times67}$) are used for the NAS training. Specifically, $LU^{900}_{legitimate}$ (from $LU^{1000}_{legitimate}$) are exploited for deep feature extraction and then the classifier is trained by the extracted features, and the rest $LU^{100}_{legitimate}$ are used for classifier testing. We repeat the above experiments until all the 68 users are selected as the legitimate user once for the generality and generalization of our system. For impostors, 20 users with 1000 windows $IU^{1000\times20}_{testing}$ (from $IU^{3000\times20}$) are chosen as impostors' testing dataset for feature extraction and classifier testing. We calculate the mean value of evaluation metrics in all cases as the reported result. Next, $LU^{100\times67}_{learning}$ are fitted and transformed by *RobustScaler* in Python library *sklearn.preprocessing*, which ignores outliers in the dataset. $LU^{1000}_{legitimate}$ and $IU^{1000\times20}_{testing}$ are transformed by the same *RobustScaler*, so that the three groups of data can be consistently normalized for data augmentation.

3.3 Neural Architecture Search

Different from architectures manually developed by human experts, NAS, the process of automating architecture engineering, has been widely applied in image classification [31], object detection [32] or semantic segmentation [33]. We are among the first to use NAS in model architecture search for time-series data classification. In the NAS, a recurrent neural network (RNN) based controller is trained in a loop with many iterations on a validation set. In each loop, the NAS first samples a child model as a candidate architecture, then trains it with a related optimizer to converge, and finally measures the validation accuracy as the reward. The controller parameters are updated by maximizing the expected reward using Proximal Policy Optimization [34]. The reward is computed on the validation set rather than the training set. The loop is repeated until the

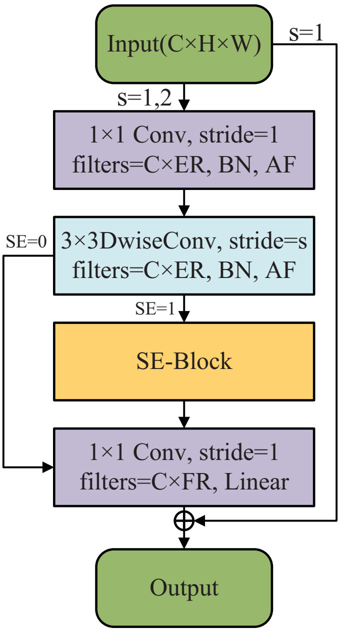

Fig. 2. Bottleneck Architecture.

convergence of controller parameters or the maximum number of steps. The challenges of applying NAS in SearchAuth are: 1) how to reduce search time, and 2) how to design search space. To address the first challenge, we force all child models to share weights to eschew training each child model from scratch to convergence, inspired by [35]. To solve the second challenge, we design a model architecture search space based on MobileNetV3 blocks [36]: bottleneck number, expansion ratio, filter ratio, squeeze and excitation, and activation function.

We detail the architecture search space as follows.

1) **Bottleneck number**: The inverted residual with linear bottleneck is used as the basic building block. As illustrated in Fig. 2, the bottleneck structure basically consists of a 1×1 Conv2D, a 3×3 DwiseConv2D, a squeeze-and-excitation block (SE-Block), and another 1×1 Conv2D. The filters of both Conv2D ($C \times ER$ for the first, $C \times FR$ for the second), the stride of DwiseConv2D (stride=s), and the activation function (AF) are customized, where if stride $= 1$ at the input, the shortcut will apply. Whether SE-Block is included depends on the parameter SE. If $SE = 0$, SE-Block will be ignored. In order to ensure that the input resolution decreases gradually, we provide 4 bottlenecks with stride of 2 and each is followed by the number of 0, 1, 2, or 3 bottlenecks with stride of 1.

2) **Expansion ratio (ER)**: Each bottleneck is followed by expansion to a much higher dimensional space and projection to the output. The expansion ratio (1.0, 1.5 or 2.0) is the ratio between the expansion filter number and the input filter number.

3) **Filter ratio (FR)**: Filter ratio (1.0, or 1.5) is the ratio between the number of the output filters over that of input filters, so that the channel number increases gradually.

4) **Squeeze and excitation (SE-Block)**: Squeeze and excitation block can adaptively recalibrate channel-wise feature responses by explicitly modelling interdependencies between channels, which has been integrated into ResNet-based modules [37] and applied in the residual layer of MobileNetV3 [36]. Whether SE-block is included depends on the parameter value of SE.

Table 1. NAS-based Model Architecture

Layer	Output	ER	FR	SE	AF	Stride	# Para
Sensor	$3 \times 200 \times 3$	-	-	-	-	-	0
Conv2d+BN	$16 \times 100 \times 2$	-	-	-	HS	2	512
Bottleneck1	$24 \times 50 \times 1$	2.0	1.5	1	RE	2	3291
Bottleneck2	$24 \times 25 \times 1$	1.5	1.0	1	HS	2	2714
Bottleneck3	$36 \times 25 \times 1$	1.0	1.5	1	HS	1	3206
Bottleneck4	$36 \times 13 \times 1$	1.5	1.0	0	HS	2	5094
Bottleneck5	$54 \times 13 \times 1$	1.0	1.5	0	HS	1	6156
Bottleneck6	$54 \times 13 \times 1$	1.5	1.0	1	RE	1	11453
Bottleneck7	$54 \times 7 \times 1$	1.0	1.0	1	HS	2	7509
Conv2d+BN	$54 \times 7 \times 1$	-	-	-	HS	1	3186
GlobalAveragePooling2d	$54 \times 1 \times 1$	-	-	-	-	1	0
Dense	$CN \times 1 \times 1$	-	-	-	-	-	3740

5) **Activation function (AF)**: Activation function can be ReLU6 or h-swish, where $h{-}swish[x] = x\frac{ReLU6(x+3)}{6}$ [36, 38].

Based on the NAS, we search a network based on neural architecture search, namely NAS-based model, from the aforementioned search space on our dataset, as illustrated in Table 1. Compared to pre-determined layers, layers between the two Conv2d+BN layers are searched architectures based on the NAS, where the convolution kernel size is fixed as 3 × 3. Note that “CN” represents the class number for training and “-” indicates unavailable parameter in the table.

3.4 Auto Augmentation Search

Varying from manually designed data augmentation approaches, the AAS can automatically search for the best data augmentation polices from image dataset [39]. For images, there is spatial correlation among the pixels and other pixels around them, while for sensor data, there is temporal correlation among samples. We utilize the AAS to train the optimal model along with the best augmentation strategies. The challenges of deploying the AAS in SearchAuth are: 1) how to design search space, and 2) how to perform strategy search. To solve the first challenge, by regarding the problem of searching the best augmentation strategy as a discrete search problem, we specially design a data augmentation strategy space that considers the possible invariant geometric transformations of sensor data in time series: rotation, jittering, scaling, permutation, magnitude-warping, time-warping, and cropping. To address the second challenge, by formulating the augmentation strategy as a parameterized probability distribution, we sample an augmentation strategy from the search space and apply to it for each training sensor data input, and the strategy with the highest probability can be the optimal one. In the following, we elaborate the search space and search pipeline, respectively.

3.4.1 Search Space. We design the candidate augmentation methods for sensor data, and list the ranges of magnitude for the candidates in Table 2:

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 ($-\pi/3$, $-\pi/6$, $-\pi/12$, $\pi/3$, $\pi/6$, $\pi/12$).

Table 2. Candidate Augmentation Methods.

Method	Range of Magnitude
Rotation	{-12, -6, -3, 3, 6, 12}
Jittering	{0.05, 0.25, 0.5}
Scaling	{0.05, 0.1, 0.2}
Permutation	{4, 5, 8}
Magnitude-Warping	{0.2, 0.4, 0.6}
Time-Warping	{0.2, 0.4, 0.6}
Cropping	{10, 20, 30}

2) **Jittering**: Noise can be introduced in the process of sensor data collection which might be caused by the environmental disturbance. Jittering function adds a noise matrix generated by a normal distribution with standard deviations (SDs) of 0.05, 0.25, and 0.5 to the sampled sensor data. Note that we ignore the injection attacks in jittering augmentation [40].

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 SDs 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 SDs of 0.2, 0.4, 0.6, then 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 a 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 repeatably. In other words, there are 24^2 strategies in the augmentation strategy search space in total.

3.4.2 Search Pipeline. Inspired by Lin *et al.*'s work [41], we adopt probability distribution optimization to the field of continuous authentication to search an optimal data augmentation strategy for time-series sensor data. As mentioned, since each augmentation strategy consists of two augmentation methods and the total number of augmentation methods is 24, there are $K = 24^2$ strategies in the designed augmentation strategy search space. Thus, we first initialize a 24 × 24 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 Eq. (1):

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

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

Then, we perform the auto augmentation strategy search. We take an epoch t of total T epochs in model training process as an example. 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 NAS-based 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 $acc(w')$, and record the network parameters;

2) Make gradient back propagation for θ, update the value of θ, and then clear the gradient of θ;

3) Save the network parameters with the highest $acc(w')$ 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 Eq. (2):

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

where N denotes the total times of network training (e.g. $N = 4$), 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 NAS-based model is used as the network model in the auto augmentation search.

3.5 Feature Extraction

We utilize the trained NAS-based model to learn and extract deep features, where the feature extraction consists of feature learning and feature selection.

3.5.1 Feature Learning. When users register in SearchAuth, the NAS-based model will be well trained based on 67 legitimate users with 100 windows $LU_{learning}^{100\times67}$, and the well-trained NAS-based model is then used as the feature extractor and the corresponding outputs are regarded as deep features. As illustrated in Table 1, there are 1,800 (3 sensors × 2 seconds × 100 Hz × 3 axes) samples in a 2s-sliding window. The first Conv2d layer with 16 filters of 3×3 and stride of 2 followed by a batch normalization and a h-swish, aims to make down sampling and increase channels. Then, 7 bottlenecks with different configurations (expansion ratio, filter ratio, squeeze and excitation, ReLU or h-swish) are applied to learn high dimensional features. Next, there is another Conv2d layer with 54 filters, kernel size of 1×1 and stride of 1 followed by an adaptive GlobalAveragePooling2d layer and a dense layer. The total parameters of the NAS-based model are 46,861 (44,705 trainable parameters and 2,156 non-trainable parameters), where the 6th bottleneck contributes the most parameters of 11,453. The outputs of the dense layer are deep features learned from the sensors of the accelerometer, gyroscope and magnetometer.

3.5.2 Feature Selection. We use the principal component analysis (PCA) to select appropriate number of deep features for the classifier based on the trained NAS model-extracted features. Based on the experiments in Sec. 4.2, the PCA selects 55 deep features for the LOF classifier to conduct the authentication.

3.6 Authentication with LOF Classifier

With the 55 PCA-selected deep features, SearchAuth 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 [42].

In the registration stage, SearchAuth generates the legitimate user's profile 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 the usage of the device, SearchAuth authenticates the current user as a legitimate user or an impostor. If the user is a legitimate user, SearchAuth 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 evaluate the performance of SearchAuth based on the collected 88 volunteers' dataset, where the randomly-selected 68 users are used as legitimate users for legitimate user registration in the registration stage and network model training in the offline stage while the rest 20 users are regarded as impostors for classifier testing in the authentication stage. For the 68 legitimate users, one of them is randomly-selected as the legitimate user in the registration stage while the rest 67 users are used as pre-collected data for network model training in the offline stage until all the 68 users are selected as the legitimate user once. We calculate the mean value of evaluation metrics in all cases as the reported result. To evaluate the authentication performance, we start with experimental settings, and then investigate the performance of SearchAuth in terms of the feature number and classifier parameter, NAS-based model, AAS, efficiency of optimal strategy, impact of dataset division, and comparison with representative schemes.

4.1 Experimental Settings

4.1.1 Network model training. Based on the 67 legitimate users' data $LU_{learning}^{100\times67}$, 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 0.01, 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.

4.1.2 Auto augmentation strategy search. The parameters of the augmentation probability distribution are initialized 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 to update the matrix. The distribution parameters are updated 300 times in total.

4.1.3 Classifier training and testing. We use ten-fold cross-validation to train the LOF classifier. For one selected legitimate user in the registration stage, we first randomly choose $LU_{legitimate}^{1000}$ from the legitimate user's data $LU_{legitimate}^{3000}$, and then randomly divide $LU_{legitimate}^{1000}$ into ten equal-size subsets $LU_{legitimate}^{100\times10}$, where nine subsets $LU_{legitimate}^{100\times9}$ are used as training data and the rest one $LU_{legitimate}^{100}$ is used as the positive testing data. Then, we randomly select one subset $IU_{testing}^{100}$ from the 20 impostors' data $IU_{testing}^{1000\times20}$ as the negative testing data, which is repeated 20 times to acquire the mean result of the classifier testing. Finally, we repeat the above process until all the legitimate

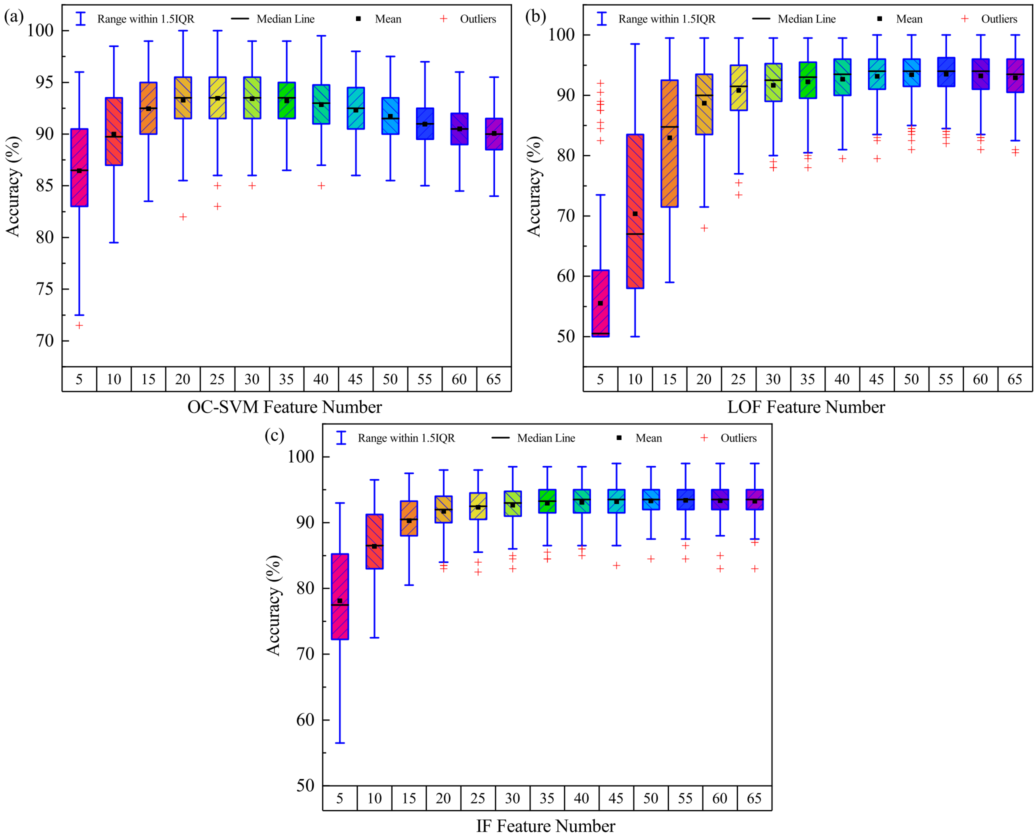

Fig. 3. Accuracy for the OC-SVM, LOF, and IF Classifiers with Varying Feature Numbers.

users are selected once. The training time for each user including data selection, data division, and ten-fold cross-validation is about 15 seconds according to our experiments.

4.1.4 Evaluation metric. We utilize three evaluation metrics: accuracy, F1-score, EER to evaluate the effectiveness of SearchAuth. Accuracy is the percentage ratio of the total number of the correct authentications against the total number of authentications, defined as: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$. F1-score is the harmonic mean of the precision and recall indicating the comprehensive performance and is defined as: F1-score = $\frac{2TP}{2TP+FP+FN}$. EER is the point that the false acceptance rate (FAR) equals to the false rejection rate (FRR), where the FAR measures the proportion of imposters who gain access while the FRR measures the proportion of legitimate users who are denied for access [43].

4.2 Feature Number and Classifier Parameter

We conduct experiments to investigate classifier selection and optimal feature number selected by the PCA. We consider three classifiers, i.e. OC-SVM (one-class SVM), IF (isolation forest), and LOF, for classifier selection and vary feature numbers to find the optimal feature number. We compute the accuracy and standard deviation (SD) of SearchAuth with the three classifiers as the feature number increases from 5 to 65, as tabulated in Table 3. As shown in Table 3, the accuracy gradually increases with the feature number growing until an optimal number and then slightly decreases for all the classifiers. For OC-SVM, 25 features selected by the PCA reach the best accuracy of 93.46%

Table 3. Accuracy (SD) % for Different Classifiers with Varying Feature Numbers

Classifier	5	10	15	20	25	30	35	40	45	50	55	60	65
OC-SVM	86.45 (5.07)	90.00 (4.22)	92.46 (3.42)	93.29 (3.11)	**93.46** (2.93)	93.43 (2.63)	93.20 (2.50)	92.86 (2.44)	92.32 (2.46)	91.73 (2.30)	90.97 (2.36)	90.50 (2.36)	90.05 (2.39)
IF	78.10 (7.86)	86.39 (5.76)	90.31 (3.97)	91.70 (3.31)	92.33 (2.97)	92.67 (2.85)	92.96 (2.75)	93.08 (2.69)	93.21 (2.55)	93.31 (2.37)	**93.38** (2.39)	93.32 (2.54)	93.27 (2.55)
LOF	55.53 (9.51)	70.37 (14.77)	82.95 (11.65)	88.70 (7.13)	90.79 (5.70)	91.66 (5.14)	92.23 (4.81)	92.67 (4.44)	93.14 (4.12)	93.41 (3.89)	**93.52** (3.85)	93.20 (4.14)	92.90 (4.39)

Table 4. Optimal Parameter Combinations

Classifier	# Feature	Optimal Parameter Combination
OC-SVM	25	$\mu = 0.0001, \gamma = 0.015625$
IF	55	`n_estimators` = 900
LOF	55	`n_neighbors` = 500, $p = 1$

and for IF, 55 features achieve 93.38% accuracy. However, LOF with 55 features selected by the PCA reaches the highest accuracy of 93.52%. Therefore, we use the PCA to select 55 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 4. As shown in Table 4, the LOF classifier with 55 deep features obtains the optimal parameters of $n\_neighbors = 500$ and $p = 1$.

4.3 NAS-based Model

Using the LOF classifier (n_neighbors=500, $p = 1$) proposed in Section 4.2, we evaluate the performance of the NAS-based model on dataset $LU_{learning}^{100\times 67}$. To show the superiority, we compare the NAS-based model with the existing representative network models, such as ResNet50 [44], MobileNetV3-small [36] and ShuffleNetV2 [45]. To adapt these network models to SearchAuth, we slightly adjust ResNet50, MobileNetV3-small, and ShuffleNetV2, respectively. (1) ResNet50: replace the kernel size $(7, 7)$ of the first conv2d with $(3, 3)$; replace the kernel size $(2, 2)$ of the last AveragePool layer with $(7, 1)$ and the following Flatten layer is used as the feature extraction layer. (2) MobileNetV3-small: replace all the layers after the Bottleneck layer with one Conv2d layer (filters = 1024, kernel size = $(1, 1)$, and stride = $(1, 1)$), one GlobalAveragePool layer as the feature extraction layer, and one Dense layer. (3) ShuffleNetV2: $0.5\times$ output channel; stride = $(2, 1)$ for downsampling; match the output channel of the full connection layer to class number, and the full connection layer is used as the feature extraction layer. We plot the boxes of the accuracy, F1-score, and EER for ResNet50, MobileNetV3-small, ShuffleNetV2, and NAS-based model, respectively, in Fig. 4. As demonstrated in Fig. 4, the proposed NAS-based model outperforms the representative network models with the highest accuracy of 92.08%, the highest F1-score of 92.32%, and the lowest EER of 6.25 while ResNet50 shows the worst performance with 70.64% accuracy, 77.65% F1-score, and 11.76% EER. In addition, we list the accuracy, F1-score, EER, parameter number, and FLOPs for the representative network models in Table 5. As shown in Table 5, the proposed NAS-based model surpasses the representative network models with margins of 6.99% accuracy, 5.08% F1-score, 3.18% EER at least, respectively. As for parameters and FLOPs, the proposed NAS-based model is the lightest model with the least parameters of 46,861 and FLOPs of 82,568, both of which are approximately one tenth of the ShuffleNetV2's.

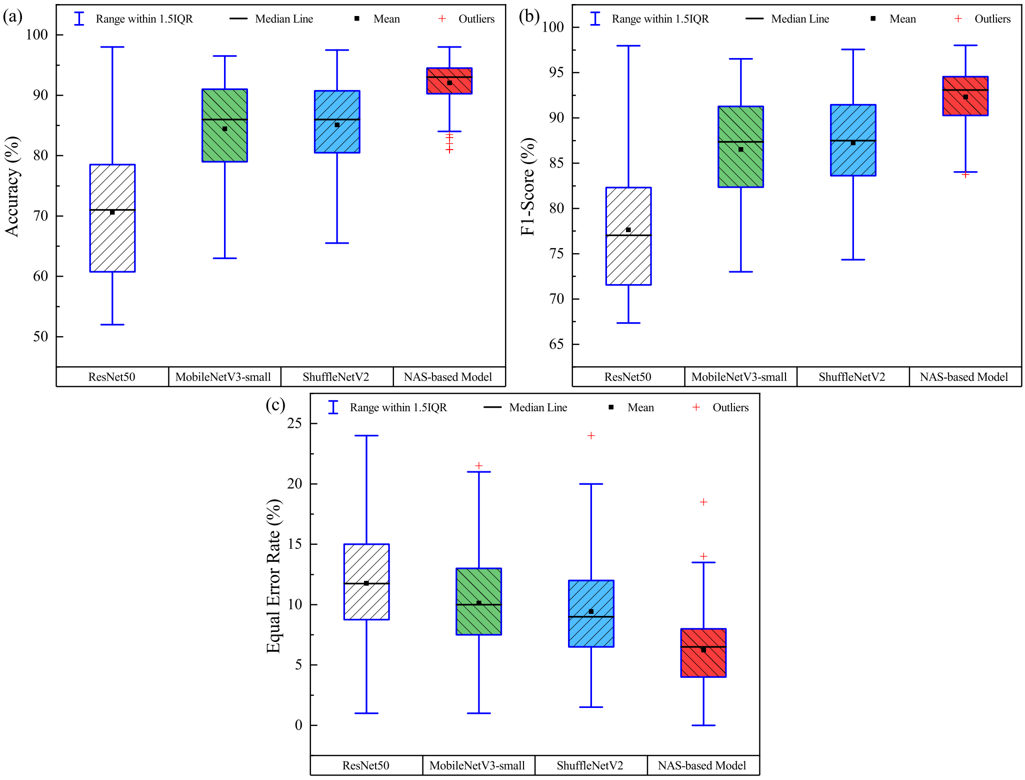

Fig. 4. Accuracy, F1-score and EER for Different Network Models.

Table 5. Performance Comparison with Different Network Models

Network	Accuracy (SD) %	$F1$-score (SD) %	EER (SD) %	# Para	# FLOPs
ResNet50	70.64 (10.71)	77.65 (6.78)	11.76 (4.62)	23,719,364	47,279,392
MobileNetV3-small	84.45 (7.98)	86.53 (5.76)	10.13 (4.17)	2,411,924	4,779,626
ShuffleNetV2	85.09 (7.36)	87.24 (5.43)	9.43 (4.17)	419,228	818,821
NAS-based Model	**92.08 (3.48)**	**92.32 (3.09)**	**6.25 (3.19)**	**46,861**	**82,586**

4.4 Auto Augmentation Search

We evaluate the performance of the auto augmentation search by utilizing the NAS-based model as the network architecture based on dataset $LU^{100\times67}_{learning}$. In the auto augmentation search, we instantiate the augmentation probability distribution parameters as a 24 × 24 matrix and save the corresponding matrix for each of the 300 epochs. Based on the saved matrices, we sum the rows of each matrix, normalize all rows for each epoch, and visualize rows varying with the increase of epochs. We calculate the marginal distribution of parameters of the first augmentation method of each strategy, as illustrated in Fig. 5. As illustrated in Fig. 5, the darker the red, the closer the probability of the method is to 1, while the darker the blue, the closer the probability is to 0. As the auto augmentation search progresses, the marginal distribution of each matrix row converges to 0 or 1. It can be seen that during random sampling training, the parameters of some augmentation methods gradually increase while others gradually decrease, which indicates that

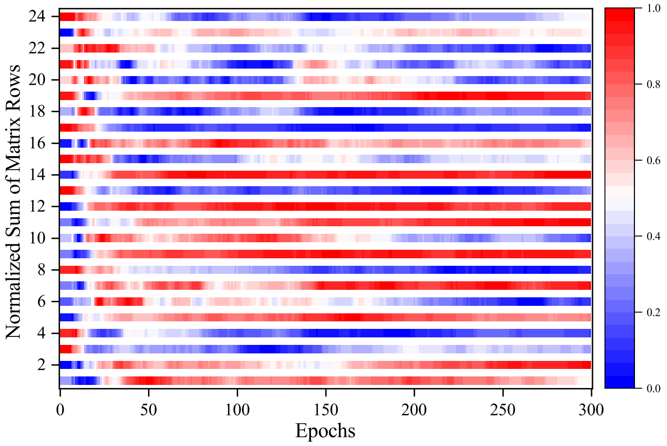

Fig. 5. Marginal Distribution of Augmentation Methods.

Table 6. Row and Column Corresponding to the Optimal Augmentation Strategy

Epoch	0 ~ 9	10	11 ~ 12	13
(row,column)	(7,5)	(23,2)	(1,12)	(12,2)
Epoch	14 ~ 112	**113 ~ 121**	122 ~ 126	**127 ~ 299**
(row,column)	(10,5)	**(15,8)**	(10,5)	**(15,8)**

some augmentation methods are abandoned while other augmentation methods may be better. In the last epoch of the process, rows of 2, 7, 11, 14 converge to 1 and rows of 8, 17 converge to 0 explicitly, which indicate that the whole process converges gradually and reaches a local optimal state finally.

In addition, after updating the parameters of the auto augmentation probability distribution matrix 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 6. We can see that during the training process, with the update of the probability distribution parameters, the optimal strategy constantly changes for the first 126 epochs and remains for the rest epochs. At the end of the training, row 15 and column 8 of the optimal strategy for local convergence can be obtained. It can be considered that Magnitude-Warping (0.2) + Jittering (0.05) is a relatively good augmentation strategy found on our dataset in the entire auto augmentation search space with the NAS-based model in Table 1 trained to converge.

We also compare the impact of the AAS on SearchAuth with that on CAuSe [18] by evaluating their performance with/without AAS. CAuSe utilizes the ShuffleNetV2 model as the CNN network structure and the optimal classifier of LOF with parameters of n_neighbors=800, $p = 1$ and 95 PCA-selected features while SearchAuth uses the trained NAS-based model as the deep network and the optimal classifier of LOF with parameters of n_neighbors=500, $p = 1$ and 55 PCA-selected features. Based on these settings and the same dataset, we evaluate the performance of CAuSe and SearchAuth. We plot boxes of the accuracy, F1-score, and EER of CAuSe and SearchAuth with/without AAS in Fig. 6. As illustrated in Fig. 6, both CAuSe and SearchAuth with (w) AAS outperform themselves without (w/o) AAS, and SearchAuth outperforms CAuSe in terms of both with AAS and both without AAS. That is, SearchAuth with AAS shows the best performance. Moreover, we list the impact of the AAS on CAuSe and SearchAuth in terms of the accuracy,

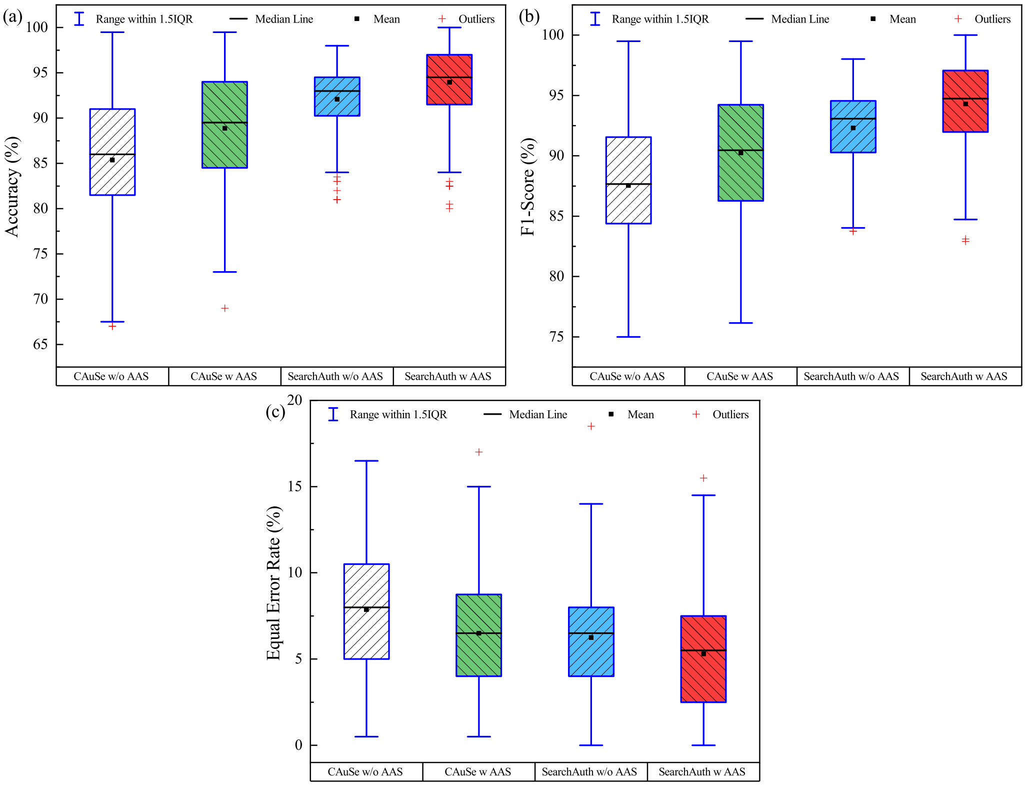

Fig. 6. Accuracy, F1-score and EER for CAuSe and SearchAuth with/without AAS.

Table 7. Impact of AAS on Network Models

Network	Accuracy (SD) %	F1-score (SD) %	EER (SD) %
CAuSe w/o AAS	85.37 (7.61)	87.54 (5.71)	7.87 (3.59)
SearchAuth w/o AAS	92.08 (3.48)	92.32 (3.09)	6.25 (3.19)
CAuSe w AAS	88.88 (6.64)	90.24 (5.29)	6.50 (3.38)
SearchAuth w AAS	**93.95** (4.17)	**94.30** (3.68)	**5.30** (3.32)

F1-score, and EER in Table 7. As shown in Table 7, the AAS improves the accuracy (3.51% on CAuSe and 1.87% on SearchAuth) and F1-score (2.7% on CAuSe and 1.98% on SearchAuth) and decreases the EER (1.37% on CAuSe and 0.95% on SearchAuth) for both CAuSe and SearchAuth, and SearchAuth with AAS achieves the best performance with 93.95% accuracy, 94.30% F1-score and 5.30% EER, respectively.

4.5 Efficiency of Optimal Strategy

We evaluate the efficiency of the optimal strategies searched by the AAS. Specifically, based on the NAS-based model and dataset $LU_{learning}^{100\times67}$, we randomly select four augmentation strategies (Rotation (-3) + Magnitude-Warping (0.2), Time-Warping (0.6) + Time-Warping (0.6), Permutation (8) + Rotation (0.6), and Time-Warping (0.6) + Permutation (2)) and the optimal strategy (Magnitude-Warping (0.2) + Jittering (0.05)) to augment the dataset, respectively, re-train the NAS-based model

Table 8. Accuracy, F1-score, and EER (SD) (%) for Different Strategies

Strategy	Accuracy (SD)	F1-score (SD)	EER (SD)	(Row, Column)
Rotation(-3)+MagnWarp(0.2)	84.42 (3.60)	83.73 (3.47)	13.53 (5.30)	(0,15)
TimeWarp(0.6)+TimeWarp(0.6)	85.05 (3.67)	84.09 (3.98)	11.80 (5.01)	(20,20)
Perm(8)+Rotation(0.6)	90.56 (5.07)	91.34 (4.11)	7.01 (3.12)	(14,5)
TimeWarp(0.6)+Perm(2)	91.64 (3.96)	91.99 (3.42)	6.52 (3.49)	(20,12)
NAS-based model w/o AAS	**92.08** (3.48)	**92.32** (3.09)	**6.25** (3.19)	-
MagnWarp(0.2)+Jitter(0.05)	92.99 (3.42)	93.15 (3.12)	5.71 (3.43)	(15,8)
NAS-based model w AAS	**93.95** (4.17)	**94.30** (3.68)	**5.30** (3.32)	AAS

for feature extraction, and use the LOF classifier to conduct the classification. Moreover, we also include the NAS-based model without AAS and NAS-based model with AAS for comparison (in Table 7). We compare the proposed NAS-based model with AAS with the representative augmentation strategies in terms of the accuracy, F1-score, and EER, as tabulated in Table 8 and as illustrated in Fig. 7. As depicted in Fig. 7, the NAS-based model without AAS outperforms the randomly selected four augmentation strategies in the accuracy, F1-score, and EER, but is inferior to the optimal strategy with (15,8), and the NAS-based model with AAS outperforms the optimal strategy. The reason is that the augmented data are not the optimal match to the searched network architecture under the original data, even augmented by the optimal strategy. However, the AAS randomly samples augmentation strategy for each epoch to train the model, and thus the trained model completely learns the space distribution of the data and shows better generalization capability. Thus, the proposed NAS-based model with AAS shows the best performance. As listed in Table 8, we also include the corresponding rows and columns for the augmentation strategies. Specifically, the NAS-based model without AAS reaches the performance of 92.8% accuracy, 92.32% F1-score and 6.25% EER, and the NAS-based model with AAS achieves the best performance of 93.95% accuracy, 94.30% F1-score and 5.30% EER, by margins of 0.96%, 1.15%, 0.41%, respectively, compared with the optimal strategy of Magnitude-Warping (0.2) + Jittering (0.05) with row 15 and column 8.

4.6 Impact of Dataset Division

We investigate the impact of the dataset division on SearchAuth by segmenting the 88 volunteers into different numbers of legitimate users and imposters. Based on the 88 volunteers' dataset, we randomly select n users as legitimate users for legitimate user registration and network model training and the rest $88 - n$ as impostors for classifier testing, where we set n as 28, 38, 48, 58, and 68, respectively. For each dataset division, we re-conduct the experiments based on the experimental settings in Section 4.1. Table 9 lists the the accuracy, F1-score, and EER with standard deviation (SD) for SearchAuth on different dataset divisions. As shown in Table 9, with the growth of legitimate users, the performance of SearchAuth slightly increases (except $(LU = 58, IU = 30)$). This is because the more legitimate user data the network model is trained on, the better results the classifier can reach. For example, SearchAuth reaches the worst performance of 91.78% accuracy, 92.19% F1-score and 6.41% EER on data division $(LU = 28, IU = 60)$ while achieves the best performance of 93.95% accuracy, 94.30% F1-score and 5.30% EER on data division $(LU = 68, IU = 20)$. From the results, we can obtain that different dataset divisions pose less impact on the performance of SearchAuth, which also indicates the generality of the proposed SearchAuth.

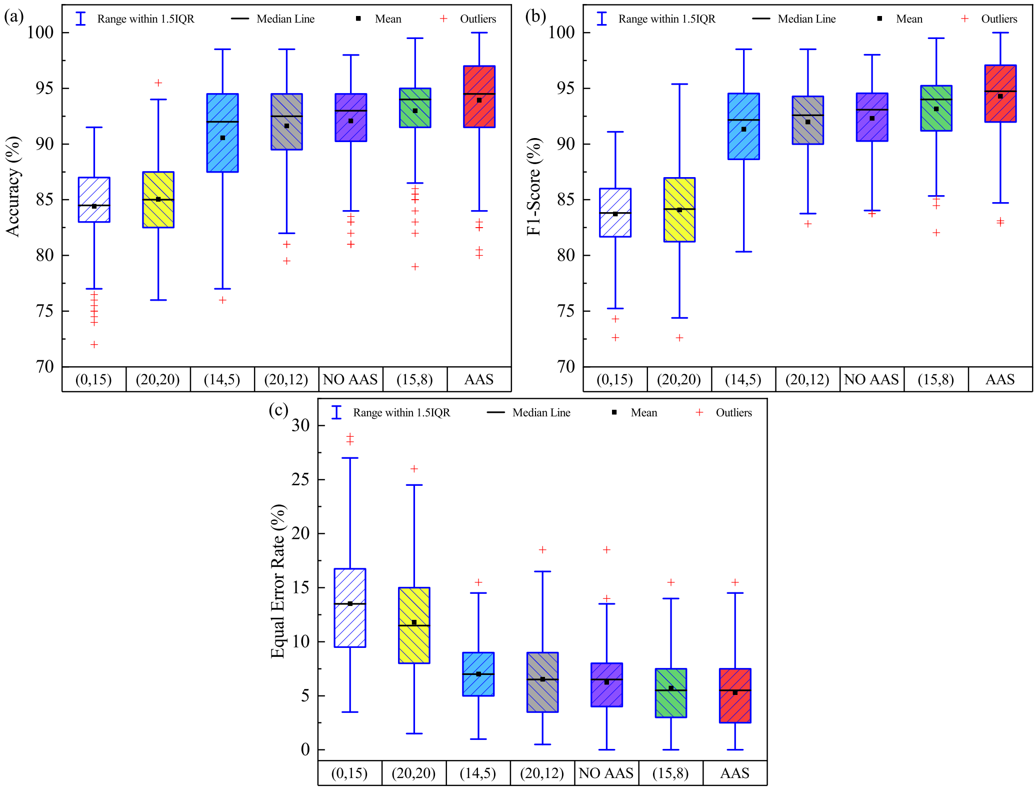

Fig. 7. Accuracy, F1-score and EER for Different Strategies.

Table 9. Accuracy, $F1$-score, and EER (SD) (%) for Different Dataset Divisions

LU / IU	28 / 60	38 / 50	48 / 40	58 / 30	**68 / 20**
Accuracy (SD)	91.78 (4.29)	91.81 (4.31)	93.11 (3.58)	92.25 (4.02)	93.95 (4.17)
$F1$-score (SD)	92.19 (3.62)	92.34 (3.67)	93.37 (3.13)	92.67 (3.40)	94.30 (3.68)
EER (SD)	6.41 (3.87)	7.16 (3.52)	6.00 (3.33)	6.68 (3.30)	5.30 (3.32)

4.7 Comparison with Representative Schemes

In this section, we first compare SearchAuth with data augmentation based continuous authentication schemes, and then with sensor-based continuous authentication approaches on our dataset.

We compare SearchAuth to eight data augmentation based continuous authentication schemes, i.e. SensorAuth [15], SensorCA [24], EmCL [16], CAGANet [27], EchoPrint [46], HMOG [47], EmRe [48], and CAuSe [18], as listed in Table 10. As illustrated in Table 10, 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 19.04% with dataset size of 100 on OC-SVM classifier by combining the five approaches [15]. SensorCA applies matrix rotation on accelerometer, gyroscope and magnetometer data to reach an EER of 4.2% with dataset size of 2000 on the SVM-RBF classifier [24]. EmCL utilizes

Table 10. Comparison with Data Augmentation based Continuous Authentication Schemes

Scheme	Data Source	Data Augmentation Approach	Accuracy (%)
SensorAuth [15]	Acc., Gyr.	Perm., sample, scale, crop, jitter	EER: 19.04 (OC-SVM, 100)
SensorCA [24]	Acc., Gyr., Mag.	Rotation	EER: 4.20 (SVM-RBF, 2000)
EmCL [16]	EEG	CycleGAN	Acc: 90.77 (CNN)
CAGANet [27]	Acc., Gyr., Mag.	CWGAN	Acc: 90.08; EER: 8.78 (LOF, 100)
EchoPrint [46]	Face image	Rotation	BAC: 81.78 (vision features)
HMOG [47]	Acc., Gyr., Mag.	HMOG with tap characteristics	EER: 7.16 (walking)
EmRe [48]	EEG	sWGAN	Acc: 65.20 (SVM, 200)
CAuSe [18]	Acc., Gyr., Mag.	AAS-based optimal strategy	Acc: 91.12; EER: 5.68 (LOF, 100)
SearchAuth	Acc., Gyr., Mag.	Auto Augmentation Search	Acc: 93.95; EER: 5.30 (LOF, 100)

Table 11. Comparison with Sensor-based Continuous Authentication Approaches on Our Dataset

Approach	Sensor	Classifier	Result (%)	
			FAR (SD)	FRR (SD)
SensorAuth [15]	Acc., Gyr.	OC-SVM	7.65 (4.59)	9.01 (5.05)
SensorCA [24]	Acc., Gyr., Mag.	SVM-RBF	3.16 (1.57)	7.35 (2.52)
CAGANet [27]	Acc., Gyr., Mag.	LOF	9.32 (5.12)	10.09 (4.52)
HMOG [47]	Acc., Gyr., Mag.	Scaled Manhattan	12.93 (6.57)	15.67 (7.24)
MultiSensorSVM [29]	Acc., Mag., Ori.	SVM	8.07 (4.54)	9.97 (4.93)
TwoSensorHMM [49]	Acc., Gyr.	HMM	10.12 (5.97)	12.58 (6.28)
MultiSensorHMM [50]	Acc., Gyr., Mag., Ori.	HMM	5.13 (3.01)	6.74 (3.58)
CAuSe [18]	Acc., Gyr., Mag.	LOF	4.87 (3.62)	8.01 (3.93)
SearchAuth	Acc., Gyr., Mag.	LOF	3.27 (3.29)	6.11 (3.78)

a cycle-consistent adversarial networks (CycleGAN) to generate EEG sensor data and obtains an accuracy of 90.77% with a CNN model as the classifier [16]. CAGANet exploits a conditional Wasserstein GAN (CWGAN) to generate acccelerometer, gyroscope, and magnetometer data and reaches an accuracy of 90.08% and an EER of 8.78% with dataset size of 100 on the LOF classifier [27]. 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 [46]. HMOG augments HMOG features with tap characteristics (e.g. tap duration and contact size) to obtain 7.16% EER for walking [47]. EmRe uses the selective WGAN (sWGAN) to augment EEG sensor and receives 65.20% accuracy with dataset size of 200 on the SVM classifier [48]. CAuSe exploits the AAS-based optimal strategy for data augmentation of the accelerometer, gyroscope and magnetometer, and achieves the best accuracy of 91.12% on the LOF classifier [18]. Comparing to these data augmentation based continuous authentication schemes, SearchAuth utilizes AAS to train the NAS-based model along with updating the augmentation probability distribution on the accelerometer, gyroscope and magnetometer sensor data and achieves the best accuracy of 93.95% and EER of 5.30% with dataset size of 100 on LOF classifier (except an EER of 4.20% for SensorCA with a long dataset size of 2000). Note that all the schemes illustrate their best accuracy in Table 10 since they are based on the most suitable classifiers, datasets and dataset sizes.

In addition, we compare SearchAuth with eight sensor-based continuous authentication approaches on our dataset, i.e. SensorAuth [15], SensorCA [24], CAGANet [27], HMOG [47], MultiSensorSVM [29], TwoSensorHMM [49], MultiSensorHMM [50], and CAuSe [18], as tabulated in Table 11. As demonstrated in Table 11, we show the sensors, classifiers, and corresponding results of FAR and FRR with their standard deviations (SDs) for the sensor-based approaches on our dataset. Concretely, SensorAuth explores the OC-SVM classifier on the accelerometer and gyroscope data to reach 7.65% FAR and 9.01% FRR [15]. With sensor data of the accelerometer, gyroscope and magnetometer, SensorCA [24], CAGANet [27] and HMOG [47] achieve 3.16% FAR and 7.35% FRR with SVM-RBF classifier on dataset size of 2000, 9.32% FAR and 10.09% FRR with LOF classifier on dataset size 200, and 12.93% FAR and 15.67% FRR with scaled Manhattan, respectively. MutiSensorSVM utilizes SVM classifier on the accelerometer, magnetometer and orientation data to obtain 8.07% FAR and 9.97% FRR [29]. Based on the HMM classifier, TwoSensorHMM [49] and MultiSensorHMM [50] receive 10.12% FAR and 12.58% FRR with the accelerometer and gyroscope data, and 5.13% FAR and 6.74% FRR with additional magnetometer and orientation data, respectively. With the accelerometer, gyroscope, and magnetometer data, CAuSe achieves 4.87% FAR and 8.01% FRR with LOF classifier [18]. Comparing with these sensor-based continuous authentication approaches on our dataset, SearchAuth explores the LOF classifier on the accelerometer, gyroscope, and magnetometer data to reach the best 3.27% FAR and 6.11% FRR with dataset size of 100 (except a FAR of 3.16% for SensorCA on the long dataset size of 2000).

5 CONCLUSION

To address the challenge of finding an optimal model architecture along with the best data augmentation policies for model training for the current continuous authentication systems, we propose SearchAuth, a NAS-based continuous authentication on smartphones using the AAS, where the NAS is specially designed for searching an optimal network architecture and the AAS is exploited for more effectively training the NAS-based model along with the updating the augmentation probability distribution. The extensive experiments show that SearchAuth surpasses the most of the representative authentication schemes by achieving the best accuracy of 93.95%, F1-score of 94.30%, and EER of 5.30% on the LOF classifier with dataset size of 100 on our dataset, respectively.

ACKNOWLEDGMENTS

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] Nawfal Al-Hadithy, Panagiotis D Gikas, and Shafic Said Al-Nammari. Smartphones in orthopaedics. *International orthopaedics*, 36(8):1543–1547, 2012.

[2] Ding Wang, Qianchen Gu, Xinyi Huang, and Ping Wang. Understanding human-chosen pins: Characteristics, distribution and security. In *2017 ACM on Asia Conference on Computer and Communications Security (ASIA CCS'17)*, page 372–385, 2017.

[3] Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. Passwords and the evolution of imperfect authentication. *Commun. ACM*, 58(7):78–87, June 2015.

[4] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. Targeted online password guessing: An underestimated threat. In *2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*, page 1242–1254, 2016.

[5] Feng Quan, Su Fei, Cai Anni, and Zhao Feifei. Cracking cancelable fingerprint template of ratha. In *2008 International Symposium on Computer Science and Computational Technology (ISCSCT'08)*, pages 572–575, 2008.

[6] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE transactions on information forensics and security*, 8(1):136–148, 2012.

[7] Huan Feng, Kassem Fawaz, and Kang G Shin. Continuous authentication for voice assistants. In *the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom'17)*, pages 343–355, 2017.

[8] Paolo Abeni, Madalina Baltatu, and Rosalia D'Alessandro. Nis03-4: Implementing biometrics-based authentication for mobile devices. In *IEEE Globecom 2006*, pages 1–5, 2006.

[9] Hui Xu, Yangfan Zhou, and Michael R Lyu. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *10th Symposium On Usable Privacy and Security (SOUPS'14)*, pages 187–198,

2014.

[10] Hugo Gascon, Sebastian Uellenbeck, Christopher Wolf, and Konrad Rieck. Continuous authentication on mobile devices by analysis of typing motion behavior. In *Sicherheit 2014*, pages 1–12, 2014.

[11] Muhammad Muaaz and René Mayrhofer. An analysis of different approaches to gait recognition using cell phone based accelerometers. In *Proceedings of International Conference on Advances in Mobile Computing & Multimedia (MoMM'13)*, pages 293–300, 2013.

[12] Mario Parreño Centeno, Aad van Moorsel, and Stefano Castruccio. Smartphone continuous authentication using deep learning autoencoders. In *the 15th Annual Conference on Privacy, Security and Trust (PST'17)*, pages 147–1478, 2017.

[13] Yantao Li, Peng Tao, Shaojiang Deng, and Gang Zhou. Deffusion: Cnn-based continuous authentication using deep feature fusion. *ACM Trans. Sen. Netw.*, 18(2), October 2021.

[14] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 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*, pages 216–220, 2017.

[15] Yantao Li, Hailong Hu, and Gang Zhou. Using data augmentation in continuous authentication on smartphones. *IEEE Internet of Things Journal*, 6(1):628–640, 2018.

[16] Xinyue Zhu, Yifan Liu, Zengchang Qin, and Jiahong Li. Data augmentation in emotion classification using generative adversarial networks. *arXiv preprint arXiv:1711.00648*, 2017.

[17] Yun Luo and Bao-Liang Lu. Eeg data augmentation for emotion recognition using a conditional wasserstein gan. In *the 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'18)*, pages 2535–2538, 2018.

[18] Shaojiang Deng, Jiaxing Luo, and Yantao Li. Cnn-based continuous authentication on smartphones with auto augmentation search. In *International Conference on Information and Communications Security (ICICS'21)*, pages 169–186, 2021.

[19] Upal Mahbub, Vishal M Patel, Deepak Chandra, Brandon Barbello, and Rama Chellappa. Partial face detection for continuous authentication. In *IEEE International Conference on Image Processing (ICIP'16)*, pages 2991–2995, 2016.

[20] Chris Xiaoxuan Lu, Bowen Du, Peijun Zhao, Hongkai Wen, Yiran Shen, Andrew Markham, and Niki Trigoni. Deepauth: In-situ authentication for smartwatches via deeply learned behavioural biometrics. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers (ISWC'18)*, page 204–207, 2018.

[21] Emanuele Maiorana, Himanka Kalita, and Patrizio Campisi. Deepkey: Keystroke dynamics and cnn for biometric recognition on mobile devices. In *the 8th European Workshop on Visual Information Processing (EUVIP'19)*, pages 181–186, 2019.

[22] Mohammed Abuhamad, Tamer Abuhmed, David Mohaisen, and DaeHun Nyang. Autosen: Deep-learning-based implicit continuous authentication using smartphone sensors. *IEEE Internet of Things Journal*, 7(6):5008–5020, 2020.

[23] Yantao Li, Hailong Hu, Zhangqian Zhu, and Gang Zhou. Scanet: Sensor-based continuous authentication with two-stream convolutional neural networks. *ACM Trans. Sen. Netw.*, 16(3), July 2020.

[24] Yantao Li, Hailong Hu, Gang Zhou, and Shaojiang Deng. Sensor-based continuous authentication using cost-effective kernel ridge regression. *IEEE Access*, 6:32554–32565, 2018.

[25] Maximilian Ernst Tschuchnig, Cornelia Ferner, and Stefan Wegenkittl. Sequential iot data augmentation using generative adversarial networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'20)*, pages 4212–4216, 2020.

[26] Xiang Li, Wei Zhang, Qian Ding, and Jian-Qiao Sun. Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. *Journal of Intelligent Manufacturing*, 31(2):433–452, 2020.

[27] Yantao Li, Jiaxing Luo, Shaojiang Deng, and Gang Zhou. Cnn-based continuous authentication on smartphones with conditional wasserstein generative adversarial network. *IEEE Internet of Things Journal*, 9(7):5447–5460, 2022.

[28] Huafeng Qin, Mounim A. El-Yacoubi, Yantao Li, and Chongwen Liu. Multi-scale and multi-direction gan for cnn-based single palm-vein identification. *IEEE Transactions on Information Forensics and Security*, 16:2652–2666, 2021.

[29] Yantao Li, Li Liu, Huafeng Qin, Shaojiang Deng, Mounim A. El-Yacoubi, and Gang Zhou. Adaptive deep feature fusion for continuous authentication with data augmentation. *IEEE Transactions on Mobile Computing*, pages 1–16, 2022.

[30] Qing Yang, Ge Peng, David T. Nguyen, Xin Qi, Gang Zhou, Zdeňka Sitová, Paolo Gasti, and Kiran S. Balagani. A multimodal data set for evaluating continuous authentication performance in smartphones. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys'14)*, page 358–359, 2014.

[31] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI'19)*, pages 4780–4789, 2019.

[32] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[33] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *5th International Conference on Learning Representations (ICLR'17)*, pages 1–16, 2017.

[34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[35] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *Proceedings of the 35th International Conference on Machine Learning (PMLR'18)*, pages 4095–4104, 2018.

[36] Andrew Howard et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'19)*, pages 1314–1324, 2019.

[37] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'18)*, pages 7132–7141, 2018.

[38] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'18)*, pages 4510–4520, 2018.

[39] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*, pages 113–123, 2019.

[40] Lorena Gonzalez-Manzano, Upal Mahbub, Jose M. de Fuentes, and Rama Chellappa. Impact of injection attacks on sensor-based continuous authentication for smartphones. *Computer Communications*, 163:150–161, 2020.

[41] Chen Lin, Minghao Guo, Chuming Li, Xin Yuan, Wei Wu, Junjie Yan, Dahua Lin, and Wanli Ouyang. Online hyper-parameter learning for auto-augmentation strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'19)*, pages 6579–6588, 2019.

[42] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, 2000.

[43] Cong Wu, Kun He, Jing Chen, Ziming Zhao, and Ruiying Du. Liveness is not enough: Enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2219–2236, 2020.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'16)*, pages 770–778, 2016.

[45] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV'18)*, pages 116–131, 2018.

[46] Bing Zhou, Jay Lohokare, Ruipeng Gao, and Fan Ye. Echoprint: Two-factor authentication using acoustics and vision on smartphones. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom'18)*, pages 321–336, 2018.

[47] Zdeňka Sitová, Jaroslav Šeděnka, Qing Yang, Ge Peng, Gang Zhou, Paolo Gasti, and Kiran S Balagani. Hmog: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security*, 11(5):877–892, 2015.

[48] Yun Luo, Li-Zhen Zhu, Zi-Yu Wan, and Bao-Liang Lu. Data augmentation for enhancing eeg-based emotion recognition with deep generative models. *Journal of Neural Engineering*, 17(5):056021, 2020.

[49] Aditi Roy, Tzipora Halevi, and Nasir Memon. An hmm-based multi-sensor approach for continuous mobile authentication. In *IEEE Military Communications Conference (MILCOM'15)*, pages 1311–1316, 2015.

[50] Chao Shen, Yuanxun Li, Yufei Chen, Xiaohong Guan, and Roy A Maxion. Performance analysis of multi-motion sensor behavior for active smartphone authentication. *IEEE Transactions on Information Forensics and Security*, 13(1):48–62, 2017.