A STACKING ENSEMBLE FOR ANOMALY BASED CLIENT-SPECIFIC FACE SPOOFING DETECTION

Soroush Fatemifar, Muhammad Awais, Ali Akbari, and Josef Kittler

Centre for Vision, Speech and Signal Processing, University of Surrey, UK

ABSTRACT

To counteract spoofing attacks, the majority of recent approaches to face spoofing attack detection formulate the problem as a binary classification task in which real data and attack-accseses are both used to train spoofing detectors. Although the classical training framework has been demonstrated to deliver satisfactory results, its robustness to unseen attacks is debatable. Inspired by the recent success of anomaly detection models in face spoofing detection, we propose an ensemble of one-class classifiers fused by a Stacking ensemble method to reduce the generalisation error in the more realistic unseen attack scenario. To be consistent with this scenario, anomalous samples are considered neither for training the component anomaly classifiers nor for the design of the Stacking ensemble. To achieve better face-anti spoofing results, we adopt client-specific information to build both constituent classifiers as well as the Stacking combiner. Besides, we propose a novel 2-stage Genetic Algorithm to further improve the generalisation performance of Stacking ensemble. We evaluate the effectiveness of the proposed systems on publicly available face anti-spoofing databases including Replay-Attack, Replay-Mobile and Rose-Youtu. The experimental results following the unseen attack evaluation protocol confirm the merits of the proposed model.

Index Terms— Spoofing Detection, Anomaly detection, Ensemble Pruning, CNNs, Fusion of Anomaly Classifiers

1. INTRODUCTION

Face spoofing detection has become an important issue in face recognition. A common form of spoofing attack is to use a fake biometric trait to access a service illegally. Over the last decade, a large number of approaches have been presented to protect biometric devices against a diverse range of Presentation Attacks (PA) including print photo, video replay and 3D mask. To prevent spoofing attacks, the majority of approaches adopt a multiclass framework to formulate the Presentation Attack Detection (PAD) problem. However, two-class systems have several drawbacks including (a) difficulty in finding an efficient decision boundary to distinguish between real and spoofing samples [1] (b) poor generalisation to novel attacks [2] (c) the risk of over-fitting, owing to the training dataset being of limited size [3]. To address the limitations of the binary face spoofing detection formulation, several authors [1, 4, 5, 3] proposed one-class (anomaly) based systems, which offer desirable properties compared to the two-class counterparts: (a) the training dataset can easily be extended as only real data is required to train anomaly classifiers (b) better generalisation due to their robustness to the new forms of spoofing attacks.

Recent studies have demonstrated the merits of using client-specific information in the design of anomaly-based face anti-spoofing systems [1, 4]. The main motivation for adopting a client-specific design is that the discriminative features extracted to facilitate the PAD problem tend to be dependent on the identities of individual clients [6]. Furthermore, to improve the generalisation ability of the face anti-spoofing systems several authors proposed ensemble approaches with a notable degree of success [7, 8, 4]. Combining both ideas in this paper, we propose a client-specific anomaly-based Stacking ensemble approach to address the PAD problem. To enhance the performance of the Stacking ensemble, we propose a novel Genetic Algorithm (GA) optimisation technique consisting of two stages. In the first stage, we measure the competency of a Pool of Classifiers (POC) by proposing a novel anomaly-based fitness function, called Proportion Above Unity (PAU), which does not need a-priory information about the spoofing class. To this end, we prune the constituent one-class classifiers (OCCs) of a GA Stacking ensemble according to the PAU criterion. In the second stage, we re-train the pruned GA Stacking (obtained from the first step) in an attempt to further improve the performance.

We evaluate the effectiveness of the proposed model on three publicly available spoofing datasets, namely, Replay-Attack [9], Replay-Mobile [10] and Rose-Youtu [11] following the unseen attack scenario. We build both client-specific and client-independent variants of the Stacking ensemble to investigate the merits of the client-specific designs.

The main contributions of this paper are as follows:

- We propose a solution for the PAD problem based on an ensemble of OCCs using stacking.
• We adopt client-specific thresholds on the output of the Stacking ensemble without relying on attack-accesses.

• We propose a novel 2-stage GA to improve the generalisation performance of the Stacking ensemble.

• We design a new anomaly-based fitness function to compute the competency of the POC.

2. PRIOR WORK

The existing solutions to PAD problem can be divided into three broad categories: (a) motion-based (b) texture-based (c) deep learning-based approaches. In the motion analysis category, the aim is to distinguish between genuine-accesses and multi-modal spoofing samples according to some physiological sign of life [12]. In the texture-based methods [2], it is assumed that real face images and attack-accesses are different from each other in terms of the skin colour, texture and reflectance characteristics. Among the three aforementioned categories, deep learning approaches have received notable attention due to their propensity to improve the performance of face spoofing detection algorithms. The work of [13] proposed a deep learning framework to conduct an end-to-end face spoofing analysis. The authors also designed two novel descriptors to characterise the local appearance and the stereo structures of bonafide and spoofing samples. A more detailed review of the recent face anti-spoofing approaches can be found in [14].

Despite the recent success of anomaly detection approaches across a broad range of domains [15], only a few studies have adopted the anomaly formulation for PAD problem. The pioneering work in [2] demonstrated that anomaly-based approaches can be as good as two-class solutions. Later, the authors in [1] exploited client-specific information to further improve the performance of one-class experts. The results obtained using client-specific anomaly classifiers were better than those achieved with client-independent counterparts.

Ensemble approaches have been empirically and theoretically proven to be superior to the best single classifier in terms of generalisation capability and predictive performance [16, 17]. The work of [7] investigated the potential impact of auxiliary supervision on deep learning approaches in PAD problem. The proposed model incorporates CNN and RNN networks to estimate the face depth and Remote Photoplethysmography Pulse (rPPG) signal of the face video. The fusion of the estimated depth and rPPG signal is performed to distinguish real accesses from attack-accesses. In another study [18], the authors trained a specific ensemble for each type of spoofing attack with the view of enhancing the generalisation capability of their framework to previously unseen attacks. The proposed model achieved state-of-the-art results.

3. PRELIMINARIES

It has been theoretically and empirically proven that the accuracy and diversity of POC play an important role in enhancing the recognition performance of ensemble techniques [19]. We inject diversity by using seven different regions of the face, suggested in [4], three OCCs including one-class SVM (OCSSVM), Gaussian Mixture Model (GMM), and Mahalanobis Distance (MD) and three different pre-trained CNNs such as GoogLeNet[20], ResNet[21] and VGG16[22]. The combination of each CNN, facial region and anomaly detector creates a single learner which together results in a POC consisting of 63 different spoofing detectors. A brief introduction of OCCs adopted in this work is as follows:

Mahalanobis Distance We assume that the model representation obtained from real samples induces a single-mode Gaussian distribution. Once the parameters determining the normal distribution are determined using the genuine-access samples of the training set, the normality checking includes measuring the Mahalanobis Distance (MD) of a test sample to the mean of the normal class.

Gaussian Mixture Model A GMM is a parametric probability density function producing a weighted sum of Gaussian component densities. The model parameters of GMM are estimated using the Expectation Maximisation (EM) algorithm [23]. Once the GMM is fitted to the training set, anomalous samples are identified by measuring the mean MD to the respective mixture components.

One-Class SVM The Support Vector Data Description (SVDD) [24] is an extension of the OCSSVM expert adopted in our experiments. The main goal of SVDD is to estimate the smallest hypersphere containing all training observations. To detect outliers using the SVDD model, any test observations falling outside the sphere are considered as anomalies.

4. ANOMALY BASED CLIENT-SPECIFIC STACKING ENSEMBLE

The main assumption behind the anomaly-based approaches is that spoofing samples would generate scores that differ from real-accesses, and could be recognised as outliers of the distribution of normal scores. As classifier scores computed for different clients follow different score distributions, client-specific thresholds are required to separate real from spoofing attempts. This procedure contrasts with client-independent approaches that set a single global threshold based on the combined score distribution of all the clients. Inspired by the recent success of deploying client-specific thresholds in anomaly detection [1, 4], we also make use of client-specific information for decision making, as described in Section 4.1.
4.1. Stacking

A Stacking ensemble [16] is a 2-stage procedure where base-classifiers output their own classification scores, which are fused in the next stage by a meta-classifier to produce the final decision. Here, to implement Stacking, we use all 63 anomaly detectors as the base-classifiers. We select GMM as the meta-classifier since it stood out as the most competent OCC in recent studies [1, 4]. We adopt the client-specific formulation for the PAD problem to design Stacking ensembles by training the base-classifiers and the meta-classifier for each client, specifically. Hence, if a database contains $n$ clients, we build $n$ client-specific Stacking ensembles, each one being trained for a particular client. To train the base-classifiers and obtain the first-stage scores, we use the enrolment data corresponding to the queried client with $10 - fold$ cross-validation. It has been demonstrated empirically that $10 - fold$ produces test error estimates that will not suffer from excessively high bias or variance [25]. Once the scores from OCCs in the first-stage are determined, we normalise the scores using the confidence level score normalisation method, suggested in [4]. To this end, we set confidence level $p$ to 0.97 which means that for each base-classifier, the proportion of real data lying under the tail of the score distribution above unity equals $3\%$. Finally, we set a client-specific threshold based on the obtained normalised scores and train the meta-classifier.

4.2. GA-Stacking

Ensemble pruning has been successfully used to find the optimal configuration for Stacking ensemble [19]. To this end, we propose a novel 2-stage GA optimisation to determine an optimal configuration for Stacking. The initial chromosomes in GA are obtained by assigning random weights corresponding to each facial region, OCC and CNN as shown in Fig.1. Then, each base-classifier score $s$ is weighted by the corresponding gene in a chromosome to obtain the weighted score $\tilde{s}$ as follows:

$$\tilde{s}_{r,c,n} = (w^C_r + w^N_n)s_{r,c,n}$$

where, $s_{r,c,n}$ is the score of the given frame $x$ obtained from the $r$-th facial region (R), $c$-th anomaly classifier (C), and $n$-th CNN (N). Different from work of [4], we consider both negative and positive weights, ranging from $[-8, 8]$. Moreover, in contrast to the binary GA fitness function used in [4], we define a novel anomaly-based fitness function to measure the competency of the resultant meta-classifier from GA. In this respect, the Proportion Above Unity (PAU) function computes the ratio of real samples that are greater than unity after fusion by the meta-classifier. Suppose $\{b_i, i = 1, 2, 3, ..., B\}$ is a set of chromosomes, then the best chromosome among the set will be:

$$b^* = \arg\min_{b_i} (PAU(b_i))$$

After the final iteration of GA in the first stage of optimisation, we discard those base-classifiers receiving the negative weights from elite chromosomes (top 10% of the final population) and re-start the GA for the second time. Once GA terminated (no further improvement for PAU), we use the best chromosome $b^*$ and build the GA-Stacking ensemble accordingly. In Fig. 1, we present an example of the proposed GA optimisation process. As seen in Fig. 1, once negative weights are discarded from the first stage, we re-train GA using a new population. In this respect, we add the elite chromosomes with some new randomly initialised chromosomes to build the (first) population of GA in the second stage using only positive weights from the range of $(0, 8]$.

![Fig. 1](image-url) The proposed GA optimisation for Stacking

5. EXPERIMENTAL EVALUATION

In this section, we conduct extensive experiments and evaluate our proposed model. We first introduce the benchmark databases used in Section 5.1. We then describe the implementation details and evaluation metrics of the current work in Section 5.2. Finally, we report the results and compare them with the state-of-the-art methods in Section 5.3.

5.1. Datasets

The availability of the enrolment set is an important pre-requisite to building client-specific models. The enrolment set is available in the Replay-Mobile and Replay-Attack datasets. However, the most existing face spoofing datasets lack an enrolment set. In theory, we could define new protocols to experiment on these datasets. However, some of these impose strict protocol restrictions on the use of their data. We, therefore, confine ourselves to experiment on the Rose-Youtu dataset. To this end, we adopt the protocol defined in [1], which allows us to make a comprehensive comparison between the clients-specific and client-independent models. We provide a brief introduction of the experimental datasets in Table 1.
5.2. Implementation Details

We normalise video frames photometrically by the Retinex method [26] to reduce the undesirable effect of the various background and illumination conditions. We pass the normalised videos to CNNs to extract deep features using the pre-ultimate layer of each network. This results in feature vectors of 1024 elements for the GoogLeNet, 2048-element feature vectors for ResNet50 and 4096 elements for VGG16 per each frame. We then pass the feature vectors to OCCs as input data. All OCCs are built in the PCA space by retaining 99% of the variance of the feature vectors. To implement the OCsvMs, we adopt the SVDD implementation provided in [27] with a linear kernel (default settings). For GMMs, we use different $M$ Gaussian components: (a) in client-independent settings, $M$ is equal to the total number of clients in a dataset (b) in client-specific settings, $M$ is set to 20 for each client. The MD classifier is implemented as a Euclidean distance between a test sample and the mean of training data. To use GA in this paper, we set the value of the initialisation population, $M$ Gaussian components: (a) in client-independent settings, $M$ is equal to the total number of clients in a dataset (b) in client-specific settings, $M$ is set to 20 for each client. The MD classifier is implemented as a Euclidean distance between a test sample and the mean of training data. To use GA in this paper, we set the value of the initialisation population, $M$ is set to 20 for each client.

5.3. Results

The main objective of the experiments conducted in this paper is to demonstrate the merits of the client-specific information in the context of anomaly detection in building a Stacking ensemble. We also investigate the proposed GA optimisation method in connection with the client-specific Stacking. To perform the experiments, we follow the unseen attack protocol, where no multi-modal spoofing samples were used to train and set the thresholds of the base-classifiers as well as the Stacking ensemble. The performance of the proposed GA-Stacking is compared with the client-specific and client-independent Stacking, and the best single classifiers of a POC in Table 5.3. As shown in Table 5.3, our client-specific Stacking outperforms its constituent classifiers as well as client-independent Stacking significantly. In Replay-Mobile, the client-specific Stacking has the HTER of 5.06% compared to 8.58% and 7.14% obtained by the best single classifier and client-independent Stacking, respectively. In Rose-YouTu, a large performance gap between the client-specific Stacking (HTER=4.12%) and client-independent counterpart (HTER=6.25%) confirms the merits of using client-specific information in designing ensemble learners. In addition, GA-Stacking outperforms traditional client-specific Stacking in all three datasets which demonstrates the effectiveness of our proposed GA optimisation technique. Finally, we compare the proposed model with the state-of-the-art multiclass as well as anomaly-based methods in Table 3. The methods which comply with the unseen attack scenario are marked with (*). According to Table 3, our GA-Stacking outperforms the state-of-the-art anomaly and multi-class approaches in both seen and unseen attack scenarios.

6. CONCLUSIONS

In this paper, we present a novel client-specific anomaly-based Stacking ensemble for face spoofing detection to address the limitations of two-class learners in unseen attack scenarios. To construct the pool of classifiers for Stacking, we adopt seven facial regions, three deep CNNs and three anomaly experts which makes 63 spoofing detectors. To further increase the performance of the proposed Stacking ensemble, we adopt the client-specific information to set the final threshold on the output of the stacking ensemble. Besides, we propose a novel GA optimisation approach to improve the generalisation capability of Stacking. To assess the predictive strength of the proposed model in the unseen attack scenario, we conducted experiments on three publicly available datasets, namely, Replay-Attack, Replay-Mobile, and Rose-YouTu. Although the main aim of the study was to assess the potential of anomaly-based methods, rather than outperforming two-class approaches, surprisingly, the experiments showed that the proposed client-specific anomaly-based Stacking method fared extremely well in comparison.

### Table 1. A brief description of datasets

<table>
<thead>
<tr>
<th>Devices</th>
<th>Num.</th>
<th>Client Specific</th>
<th>Processed Features</th>
<th>Illumination Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replay-Attack</td>
<td>50</td>
<td>1500</td>
<td>2D attacks: print, replay</td>
<td>1, 2</td>
</tr>
<tr>
<td>Replay-Mobile</td>
<td>40</td>
<td>1200</td>
<td>2D attacks: print, replay</td>
<td>2, 5</td>
</tr>
<tr>
<td>Rose-YouTu</td>
<td>20</td>
<td>350</td>
<td>3D attacks: print, replay</td>
<td>5, 5</td>
</tr>
</tbody>
</table>

### Table 2. The HTER(%) and ACER(%) results of Client-Specific (CS) approaches vs. Client-Independent (CI) counterparts

<table>
<thead>
<tr>
<th>Devices</th>
<th>HTER (%)</th>
<th>ACER (%)</th>
<th>CS-Stacking</th>
<th>GA-Stacking</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replay-Attack</td>
<td>0.3</td>
<td>0.06[13]</td>
<td>2.82[1]^*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 3. Comparison between the performance of the proposed countermeasure and state-of-the-arts.

<table>
<thead>
<tr>
<th>Devices</th>
<th>HTER (%)</th>
<th>GA-Stacking</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replay-Attack</td>
<td>0</td>
<td>2.82[1]^*</td>
<td></td>
</tr>
<tr>
<td>Replay-Mobile</td>
<td>4.02</td>
<td>9.30[4]^*</td>
<td></td>
</tr>
</tbody>
</table>

* HTER stands for Half Total Error Rate, ACER for Average Classification Error Rate.
7. REFERENCES


