Cost Sensitive Optimization of Deepfake Detector

Ivan Kukanov*†‡, Janne Karttunen*†§, Hannu Sillanpää*†§, and Ville Hautamäki†§
†School of Computing, University of Eastern Finland, Finland
‡Institute for Infocomm Research, A*STAR, Singapore
E-mail: ivan@kukanov.com, {jannkar, hannusi, villeh}@uef.fi

***Abstract*—Since the invention of cinema, the manipulated videos have existed. But generating manipulated videos that can fool the viewer has been a time-consuming endeavor. With the dramatic improvements in the deep generative modeling, generating believable looking fake videos has become a reality. In the present work, we concentrate on the so-called deepfake videos, where the source face is swapped with the targets. We argue that deepfake detection task should be viewed as a screening task, where the user, such as the video streaming platform, will screen a large number of videos daily. It is clear then that only a small fraction of the uploaded videos are deepfakes, so the detection performance needs to be measured in a cost-sensitive way. Preferably, the model parameters also need to be estimated in the same way. This is precisely what we propose here.**

I. Introduction

In a just few years, the attention of the general public and research community has been raised to the dangers of the deepfakes [1]. Deepfakes, in general, are defined as face swapped videos, where the *source* individual's face is swapped to the *target* individuals face. This is also known as an identity swap [1]. It is easy to imagine socially disruptive applications of such a technology [2], such as video of a politician in a questionable activity before elections. In addition, it has been shown that human observers can be fooled by the deepfakes [3]. This raises a need to develop automatic methods for deepfake detection. Such methods could be then employed by streaming services, law enforcement personnel and individual citizens.

Multiple ways exist in generating deepfakes [4]: Face-swap [5] swaps the face of a person with another person frame-by-frame, lip-sync methods modify mouth movements in the video to match a swapped sample of speech, puppet-master [6] methods transfer movements from an actor to the target person. Generating swapped face images requires a high-quality generative model of face images. Such models are for example GAN models like StyleGAN [7], FS-GAN [8] and the few-shot method in [9]. The idea is that new face images are generated frame by frame. Then same face expressions and orientations of the target face would be automatically generated to the new face image.

Being generated frame by frame, deepfakes can be detected based on cues such as inconsistent head poses [10] and eye blinking [11]. Deepfakes can also be detected by training a deep classifier to focus on frame-by-frame artefacts [5], [12], [13] and considering temporal differences between frames [14]. As is expected, for known deepfake generation types low error rates are reported, but for the unseen attack type, collected from the Internet, the results are shown to be poor [12]. It is noteworthy that all previous studies consider equal costs for both miss classification rates (miss and false alarm).

Some of the datasets are available for the development of deepfake detectors. Publicly available datasets are DeepfakeTIMIT [15] and deepfakes subset from Faceforensics++ (FF++) [3] have been widely used for training and evaluation [3], [12], [16], [17]. In multiple studies datasets have been collected directly from online sources [5], [14], utilizing videos created by regular users. Just recently more datasets have been emerging, such as newly added extension for FF++ dataset [1] and Celeb-DF [12].

We trained our models using the pooled DeepfakeTIMIT and FF++. We also collected a number of deepfake videos made for entertainment purposes from the YouTube. This set works as a proxy for unseen deepfake condition. Both of these sets we released publicly. In addition, noting that it is expected that deepfakes are much more rarer than legitimate videos, we promote a cost sensitive measurement of deepfake detection performance. And finally, we finetune detection models to directly optimize the cost sensitive metric via *maximal figure-of-merit* (MFoM) framework [18].

II. Detection Metrics

A. Detection Cost Function

In this work, we use detection cost function (DCF) as the performance measure. It is the conventional performance measure in the speaker recognition domain for long time [19]. It serves as a unified measure for evaluation a performance of detection models and gives insights on the new advanced methods. DCF is defined as a weighted sum of two types of errors: *miss detection* P_{miss} and *false alarm* (acceptance) P_{fa}

$$C_{\text{DCF}}(t) = C_{\text{miss}} \cdot P_{\text{tar}} \cdot P_{\text{miss}}(t) + \\ + C_{\text{fa}} \cdot (1 - P_{\text{tar}}) \cdot P_{\text{fa}}(t), \quad (1)$$

* Equal contribution.

§ These authors were supported by the Academy of Finland (grant #313970) and Finnish Scientific Advisory Board for Defence (MATINE) project #2500M-0106. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp and V GPUs used for this research. We thank Marko Vahela for collecting the deepfake videos.

[1] https://github.com/ondyari/FaceForensics/tree/master/dataset/DeepFakeDetection

it depends on the decision threshold t, applied to the scores; parameters C_{miss} (cost of a miss detection) and C_{fa} (cost of a false alarm) are usually set to one; P_{tar} is *a prior* probability of the target class, P_{tar} takes value from $\{0.1, 0.05, 0.01\}$. Empirical probabilities of miss detection and false alarm are

$$P_{\text{miss}}(t) = \frac{FN\,(t)}{P} = \frac{\sum\limits_{y_i \in y_{\text{tar}}} \mathbf{1}\left(g\left(\mathbf{X}_i\right) < t\right)}{P}, \tag{2}$$

and

$$P_{\text{fa}}(t) = \frac{FP\,(t)}{N} = \frac{\sum\limits_{y_i \in y_{\text{non}}} \mathbf{1}\left(g\left(\mathbf{X}_i\right) \geq t\right)}{N}, \tag{3}$$

where the function $\mathbf{1}(\cdot)$ is the indicator function applied to the model scores $g(\mathbf{X}_i)$ on every sample $\mathbf{X}_i$, P and N are the total number of target and non-target samples.

B. Equal Error Rate

Another conventional detection performance measure is the equal error rate (EER). The EER is expressed using the same P_{miss} and P_{fa}, those are increasing and decreasing functions of the threshold t and the value of EER is defined on the intersection. The lower the value of EER the better the performance of a system. On the other hand, EER is defined as the equality

$$\text{EER}(t^*) = P_{\text{miss}}(t^*) = P_{\text{fa}}(t^*), \tag{4}$$

where an optimal threshold for the EER is t^* and any threshold $t \in [0, 1]$. Criteria for the optimal threshold is

$$t^* = \underset{t}{argmin}\,|P_{\text{miss}}(t) - P_{\text{fa}}(t)|. \tag{5}$$

III. Maximal Figure-of-Merit Solution

In this work we explore MFoM framework [18] for DCF and EER optimization. The goal is to develop an objective function which directly optimizes the measures of performance.

A. Discriminant Function

The inference of MFoM framework begins with the definition of the *discriminant function*. For a neural network, it is the activation scores $g(\mathbf{X}|\mathbb{W})$ of the output layer, which defines the confidence of a model on a particular input sample $\mathbf{X}$. The choice of a proper discriminant function depends on the nature of the classifier, and the task at hand. Discriminant functions are defined on the classifier parameters set $\mathbb{W}$.

B. Misclassification Measure

The next part of MFoM is a *misclassification measure* [20]. This approach, based on misclassification measures, allows us to define different strategies for decision rules based on discriminant scores. In the previous studies for phonetic feature detection [21], [22] and acoustic events detection [23], authors proposed the *units-vs-zeros* misclassification measure for each class C_k as

$$\psi_k = -g_k + \frac{1}{\eta} \ln \left(\frac{1}{|\mathbf{I}|} \sum_{j \in \mathbf{I}} e^{\eta g_j} \right), \tag{6}$$

$$\begin{cases} \text{if } C_k \text{ is } 1 \Rightarrow \mathbf{I} = \mathbf{y}_{\{0\}}, \\ \text{if } C_k \text{ is } 0 \Rightarrow \mathbf{I} = \mathbf{y}_{\{1\}}, \end{cases} \tag{7}$$

where ψ_k is defined for current sample $\mathbf{X}$ and its label $\mathbf{y}$; $\mathbf{I}$ is an index set, $\mathbf{y}_{\{1\}}$ is the set of unit indexes and $\mathbf{y}_{\{0\}}$ is the set of zero indexes in the label vector $\mathbf{y}$; g_k are the discriminant functions; η is a positive real-valued smoothing constant ($\eta = 1$ in our experiments).

The first term on the left-side of (6) is called the *target model* and the right-side is the *Kolmogorov mean* (generalised f-mean) [24] of the competing (*confusing*) models. The misclassification measure is the differences between the target class and the average of the confusing classes.

The sign of the misclassification measure indicates the correctness of classification: $\psi_k(\cdot) \leq 0$ indicates the predicted class is correct, and $\psi_k(\cdot) > 0$ implies incorrect decision. The absolute value of the ψ_k quantifies the separation between the correct and competing classes [25]. The equality $\psi_k(\cdot) = 0$ defines the decision boundary between a class k and the rest.

C. Smooth Error Counter

The third block of the MFoM framework is the *smooth error counter*, which plays the key role for the approximation of discrete performance measures based on discrete error counts (i.e., false positive and false negative statistics)

$$l_k = \frac{1}{1 + \exp\left[-\alpha_k \psi_k - \beta_k\right]}, \tag{8}$$

where $k = \overline{1, M}$ is the class index, α_k and β_k are real valued parameters of the scale and shift transformation. From deep learning point of view, we can interpret the linear transformation (α_k and β_k) of the *misclassification measure* as an additional layer of a network. In this work, we propose the optimization of those parameters similar to the batch normalization technique, when the error of the objective function E is backpropagated through α_k and β_k as well

$$\frac{\partial E}{\partial \alpha_k} = -\frac{\partial E}{\partial l_k} \cdot \psi_k, \tag{9}$$

$$\frac{\partial E}{\partial \beta_k} = -\frac{\partial E}{\partial l_k}. \tag{10}$$

D. Approximation of DCF Objective

The key ingredients of the proposed MFoM framework are: a) discriminant function, which in our case are output scores of a network model, b) misclassification measure (6), and c) smoothed error counter (8). Now that these components have been introduced, we can express the DCF in terms of those three entities within the deep neural network paradigm. We introduce a smooth approximation of discrete error rates $P_{\text{miss}}(t)$ and $P_{\text{fa}}(t)$, for this purpose we apply the smooth error counter from (8)

$$\hat{P}_{\text{miss}} \triangleq \frac{\sum\limits_{k=1}^{M} FN_k}{P} = \frac{\sum\limits_{k=1}^{M} \sum\limits_{\mathbf{X} \in \mathbb{T}} l_k \cdot y_k}{P}, \tag{11}$$

$$\hat{P}_{\text{fa}} \triangleq \frac{\sum_{k=1}^{M} FP_k}{N} = \frac{\sum_{k=1}^{M} \sum_{\mathbf{X} \in \mathbb{T}} (1 - l_k) \cdot \overline{y}_k}{N}, \quad (12)$$

where y_k and $\overline{y}_k$ are the binary labels and their inverse, assigning sample $\mathbf{X}$ to class k. Eventually, the MFoM-DCF objective function is obtained and applied for DNN parameters ($\mathbb{W}$), optimized on a training set $\mathbb{T}$

$$E_{\text{DFC}}(\mathbb{W}|\mathbb{T}) = P_{\text{tar}} \cdot \hat{P}_{\text{miss}}(\mathbb{W}|\mathbb{T}) + + (1 - P_{\text{tar}}) \cdot \hat{P}_{\text{fa}}(\mathbb{W}|\mathbb{T}), \quad (13)$$

where P_{tar} is *a prior* probability from (1).

E. Approximation of EER Objective

Similar to the DCF approximation using the MFoM framework, we can embed the EER into the objective function for DNN optimization. Using two properties of the discrete EER (4) and (5), we infer smoothed MFoM-EER objective, which is transformed to unconstrained function

$$E_{\text{EER}}(\mathbb{W}|\mathbb{T}) = \hat{P}_{\text{fa}}(\mathbb{W}|\mathbb{T}) + + \lambda \left| \hat{P}_{\text{miss}}(\mathbb{W}|\mathbb{T}) - \hat{P}_{\text{fa}}(\mathbb{W}|\mathbb{T}) \right|, \quad (14)$$

where $\lambda \neq 0$ is a Lagrange multiplier, in the experiments we assign $\lambda = 0.5$.

IV. EXPERIMENTS

A. Detection methods

For the baseline methods we use CNN-based approach and recurrent *long short-term memory* (LSTM) network. While CNN implementations for deepfake detection exist [5], we chose to train one from scratch, using the ready implementation of MobileNet [26]. For the LSTM, we implemented a network based on the description in [14], using similarly pretrained InceptionV3 for feature extraction, and then LSTM for the temporal analysis. Subsequences of 20 frames were used as an input for network. So, network produced score after 20 frames. For a single video with length more than 20 frames, subsequence scores were averaged.

B. Dataset

For training data we have used the FaceForensics++ and Deepfake-TIMIT datasets merged together. FaceForensics++ includes data for both real and fake classes. For Deepfake-TIMIT, corresponding pristine videos from VidTIMIT [27] are used for data in the real class. Deepfake-TIMIT includes face swaps made using a higher and lower quality model, of which both are included in our merged data. Two versions of all videos were included with different values for the Constant Rate Factor (CRF) compression parameter: high quality (CRF=23) and low quality (CRF=40). Facial images were cropped and aligned from the video frames at resolution 256x256 using DeepFaceLab software[2]

To additionally evaluate our methods in a scenario more accurate to real-life, we have collected a dataset from YouTube including deepfakes generated by regular users. These deepfakes were originally created for entertainment purposes, which is why they are more fine-tuned and polished than our initial training set. The generation algorithms of these deepfakes are unknown, and unseen to our methods, which is why the detection rate is expected to be lower, similarly as in previous studies [12].

We manually downloaded and annotated 79 deepfake videos from YouTube, extracting total of 98 face swaps. For the real data, we similarly downloaded 98 samples annotated in VoxCeleb2 dataset [28]. All the videos were further divided to scenes, to make temporal analysis possible. Our collected dataset will be available for download[3].

C. Results

TABLE I
Performance results on test dataset and self collected evaluation dataset, denoted by Eval. MEER and MDCF signifies MFoM with soft EER and DCF objectives. Results are shown in EER (%) and DCF with corresponding P_{tar}.

Method	EER	minDCF with P_{tar} 0.1	0.05	0.01	Eval EER
LSTM [14]	24.1	0.88	0.92	0.96	38.90
CNN	8.07	0.37	0.43	0.60	30.80
CNN+MEER	7.16	0.44	0.50	1.00	32.32
CNN+MDCF_0.1	**6.03**	0.33	0.46	0.88	32.49
CNN+MDCF_0.05	6.67	0.28	**0.32**	0.46	**30.56**
CNN+MDCF_0.01	6.72	0.35	0.43	0.56	32.09

The results of detection methods are shown in Table I. We did not reach similar accuracy with LSTM as reported in [14], which can be explained by the different dataset or implementation differences. With the CNN, we obtained 8,07% EER, on which we applied MFoM-based objectives for the further fine-tuning. We performed fine-tuning for 5 epochs with MFoM-EER and MFoM-DCF, i.e. MEER, MDCF_0.1, MDCF_0.05 and MDCF_0.01, respectively. Results are shown in Table I. Consistent improvement was obtained. The best performance in terms of EER, was obtained by optimizing CNN with MFoM-DCF and prior of 0.1 (CNN+MDCF_0.1). These methods are additionally evaluated on the collected dataset (Table I, rightmost column). As expected, the results were significantly worse due to better deepfake quality. The best detection method was CNN with MFoM-DCF tuning (CNN+MDCF_0.05), it scored only 30.56% EER. Analysis of results showed that videos with lower deepfake quality were classified correctly as deepfakes. However, most of the deepfakes in evaluation set surpassed our training set in terms of quality, which is why tampered videos were not caught with our detection methods.

V. CONCLUSIONS

In this work, we defined the deepfake detection task as a cost-sensitive objective. We borrowed the measurement technology from the NIST SRE campaigns. The idea being that

[2]https://github.com/iperov/DeepFaceLab

[3]http://cs.uef.fi/deepfake_dataset/

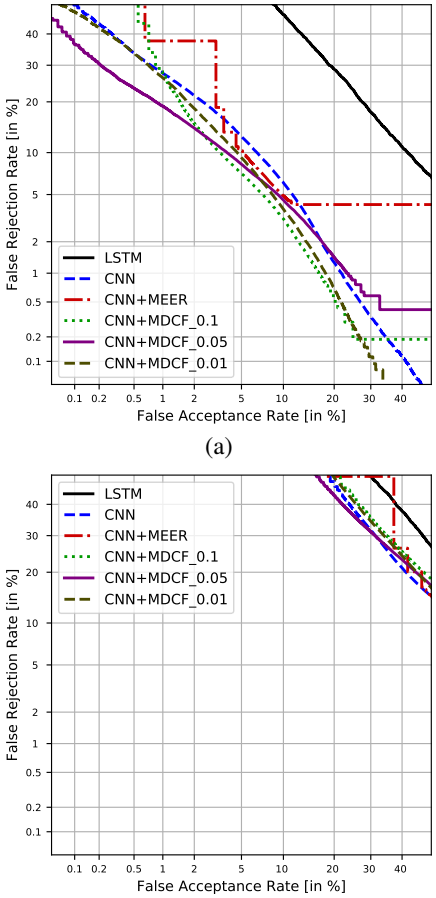

Fig. 1. Detection error tradeoff (DET) curves on the test (a) and evaluation sets (b) for the evaluated deepfake detectors.

essentially both tasks, NIST SRE and deepfake detection, are screening tasks. We then defined a cost-sensitive optimization technique, utilizing the MFoM theory. We showed that fine tuning the CNN model with MFoM improves the EER from the 8.07% to 6.03%.

In our self collected evaluation set, we noticed that the best test set EER of 6.03% is increased to more than 30%. The set is based on the deepfakes generated for entertainment purposes and uploaded to YouTube. We take these deepfakes to be a worst cases a user of the detector would encounter in practice. As a future work, we plan to investigate deepfake detector models that perform more robustly on the unseen conditions. In addition, we will expand the use of MFoM technique to our attentive pooling models, with the hope of further improvement on the detector performance.

REFERENCES

[1] J. Stehouwer, H. Dang, F. Liu *et al.*, "On the detection of digital face manipulation," *arXiv:1910.01717*, 2019.
[2] R. Chesney and D. Citron, "Deep fakes: A looming challenge for privacy, democracy, and national security," *107 California Law Review*, 2019.
[3] A. Rössler, D. Cozzolino, L. Verdoliva *et al.*, "Faceforensics++: Learning to detect manipulated facial images," *CoRR*, vol. abs/1901.08971, 2019.
[4] S. Agarwal, H. Farid, Y. Gu *et al.*, "Protecting world leaders against deep fakes," in *Proceedings of the IEEE Conference on CVPR Workshops*, 2019, pp. 38–45.
[5] D. Afchar, V. Nozick, J. Yamagishi *et al.*, "Mesonet: a compact facial video forgery detection network," *CoRR*, vol. abs/1809.00888, 2018.
[6] K. Nagano, J. Seo, J. Xing *et al.*, "paGAN: real-time avatars using dynamic textures." *ACM Transactions on Graphics*, no. 6, pp. 258–1, 2018.
[7] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *arXiv:1812.04948*, 2018.
[8] Y. Nirkin, Y. Keller, and T. Hassner, "Fsgan: Subject agnostic face swapping and reenactment," *arXiv:1908.05932*, 2019.
[9] E. Zakharov, A. Shysheya, E. Burkov *et al.*, "Few-shot adversarial learning of realistic neural talking head models," *CoRR*, vol. abs/1905.08233, 2019.
[10] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," *ICASSP 2019*, pp. 8261–8265, 2018.
[11] Y. Li, M. Chang, and S. Lyu, "In ictu oculi: Exposing AI generated fake face videos by detecting eye blinking," *CoRR*, vol. abs/1806.02877, 2018.
[12] Y. Li, X. Yang, P. Sun, H. Qi *et al.*, "Celeb-df: A new dataset for deepfake forensics," *arXiv:1909.12962*, 2019.
[13] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," *CoRR*, vol. abs/1810.11215, 2018.
[14] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *AVSS*, 2018.
[15] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *CoRR*, vol. abs/1812.08685, 2018.
[16] H. H. Nguyen, F. Fang, J. Yamagishi *et al.*, "Multi-task learning for detecting and segmenting manipulated facial images and videos," *CoRR*, vol. abs/1906.06876, 2019.
[17] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *WACVW*, 2019.
[18] S. Gao, W. Wu, C.-H. Lee *et al.*, "A maximal figure-of-merit (mfom)-learning approach to robust classifier design for text categorization," *ACM Trans. Inf. Syst.*, vol. 24, no. 2, pp. 190–218, Apr. 2006.
[19] S. O. Sadjadi, T. Kheyrkhah, A. Tong *et al.*, "The 2016 nist speaker recognition evaluation," in *Interspeech*, 2017.
[20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[21] I. Kukanov, V. Hautamäki, S. M. Siniscalchi *et al.*, "Deep learning with maximal figure-of-merit cost to advance multi-label speech attribute detection," in *SLT*, 2016.
[22] I. Kukanov, T. N. Trong, V. Hautamäki, S. M. Siniscalchi, V. M. Salerno, and K. A. Lee, "Maximal figure-of-merit framework to detect multi-label phonetic features for spoken language recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 682–695, 2020.
[23] I. Kukanov, V. Hautamäki, and K. A. Lee, "Maximal figure-of-merit embedding for multi-label audio classification," in *ICASSP*, 2018.
[24] V. M. Tikhomirov, "On the notion of mean," in *Selected Works of A. N. Kolmogorov*. Springer Netherlands, 1991, pp. 144–146.
[25] S. Katagiri, C.-H. Lee, and B.-H. Juang, "New discriminative training algorithms based on the generalized probabilistic descent method," in *Neural Networks for Signal Processing Proceedings IEEE Workshop*, 1991.
[26] A. Howard, M. Zhu, B. Chen *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv*, 2017.
[27] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," in *ICB*, 2009.
[28] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.