ESTIMATING THE CONFIDENCE OF SPEECH SPOOFING COUNTERMEASURE

Xin Wang, Junichi Yamagishi

National Institute of Informatics, Japan

ABSTRACT

Conventional speech spoofing countermeasures (CMs) are designed to make a binary decision on an input trial. However, a CM trained on a closed-set database is theoretically not guaranteed to perform well on unknown spoofing attacks. In some scenarios, an alternative strategy is to let the CM defer a decision when it is not confident. The question is then how to estimate a CM's confidence regarding an input trial. We investigated a few confidence estimators that can be easily plugged into a neural-network-based CM. On the ASVspoof2019 logical access database, the results demonstrate that an energy-based estimator and a neural-network-based one achieved acceptable performance in identifying unknown attacks in the test set. On a test set with additional unknown attacks and bona fide trials from other databases, the confidence estimators performed moderately well, and the CMs better discriminated bona fide and spoofed trials that had a high confidence score. Additional results also revealed the difficulty in enhancing a confidence estimator by adding unknown attacks to the training set.

***Index Terms*—** anti-spoofing, presentation attack detection, countermeasure, logical access, deep learning

1. INTRODUCTION

Advanced voice conversion (VC) and text-to-speech (TTS) technologies make it easy to create a high-quality synthetic voice. However, synthetic voices can be misused to attack automatic speaker verification (ASV) systems [1], now referred to as a presentation attack (PA) by the ISO/IEC 30107-1 standard [2]. They can also be abused to fool humans and have lead to an issue known as deepfakes. These concerns call for reliable PA and deepfake detection methods.

Most PA and deepfake detection methods, or spoofing countermeasures (CM) in general, are based on a binary classification scheme. Given an input speech trial $\boldsymbol{x}_{1:T}$ of length T, the CM extracts N frames of acoustic features $\boldsymbol{a}_{1:N}$ and computes a score $s \in \mathbb{R}$ to indicate how likely the input trial is bona fide – a real human voice. It then makes a decision by comparing the score with an application-dependent threshold θ_s. Most CMs use deep neural networks (DNNs) to detect artifacts in input trials, and many of them have achieved impressive results on benchmark databases [3].

However, a CM well trained on a closed-set database is likely to misclassify unseen trials from unknown attacks and unseen bona fide trials from mismatched unknown domains [4, 5][1]. These trials are sometimes referred to as "known-unknown" and "unknown-unknown" in the machine learning field [6], and in this study, we simply refer to them as being *unknown* to the CM. While data augmentation techniques [7] can make a CM more robust, they cannot cover all unknown conditions. Rather than being forced to make a binary decision, a practical CM should abstain from making decisions on trials that are difficult to judge. Such a CM is illustrated in Fig. 1. The option to abstain is desired when a classification error incurs a high risk, regardless of being a false positive or false negative. If a CM abstains, the input trial can be scrutinized by other CMs or a human expert. Although it is not investigated in this study, an active learning strategy can be used to collect the trials annotated by the human expert and fine-tune the CM [8].

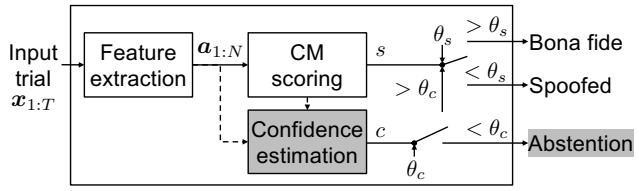


Fig. 1: CM with confidence estimator can opt for abstention. s and c denote CM and confidence scores, respectively. θ_s and θ_c denote the threshold for CM classification and abstention, respectively.

Classification with abstention is an established topic in the machine learning field, and most methods require a separately trained model to decide whether to abstain or not [9]. Other studies treat the task as outlier or out-of-distribution (OOD) data detection. They propose augmenting classifiers with a trainable confidence estimator [10, 11] or a non-trainable scoring module [12, 13]. OOD data, which is likely to be misclassified by a classifier, usually receives a low confidence score and can be identified.

Inspired by the aforementioned studies, this study investigates how and whether it is useful to introduce abstention to deep neural network (DNN)-based speech spoofing CMs. On the basis of two high-performance CMs, we compared a few confidence estimators on the ASVspoof 2019 logical access (LA) database [14] and an additional evaluation set with *unknown* trials from Voice Conversion Challenges (VCC) [15, 16]. The results demonstrate that simply using the probability from a softmax as the confidence score lead to overconfidence, a finding consistent with other studies [17, 18]. An energy-based confidence scoring method [12] and a confidence branch [10] achieved an acceptable performance, and both helped the CM to identity *unknown* trials from the VCC test set. Without making decisions on trials with a confidence score lower than the threshold, the CM achieved a better CM EER on the remaining trials.

The confidence estimators investigated in this study are explained in Sec. 2. The experiments are described in Sec. 3. This paper ends with a conclusion in Sec. 4. Codes and datasets used in this study will be available online[2].

This study is supported by JST CREST Grants (JPMJCR18A6 and JPMJCR20D3), MEXT KAKENHI Grants (21K17775, 21H04906, 18H04112, 16H06302), and Google AI for Japan program.

[1]Although benchmark databases such as those from the ASVspoof challenges intentionally keep unknown attacks in the evaluation set, the labels of the evaluation set are released to the public after the challenges. Network architectures and other hyper-parameters of CMs can *unintentionally overfit* the evaluation set.

[2]https://github.com/nii-yamagishilab/project-NN-Pytorch-scripts

2. CONFIDENCE ESTIMATORS

2.1. Max probability from CM

The first estimator is a simple plug-in to a pre-trained DNN-based CM, for example, a feedforward DNN followed by average pooling, affine transformation, and softmax [19]. Suppose the input $\boldsymbol{x}_{1:T}$ has been converted into an utterance-level vector $\boldsymbol{h} \in \mathbb{R}^h$ after average pooling. The output probability given by the affine transformation and softmax output layer can be written as $P(j|\boldsymbol{x}) = \frac{\exp(l_j)}{\sum_{k=1}^{2}\exp(l_k)} = \frac{\exp(\boldsymbol{w}_j^\top \boldsymbol{h}+b_j)}{\sum_{k=1}^{2}\exp(\boldsymbol{w}_k^\top \boldsymbol{h}+b_k)}$, where $P(j=1|\boldsymbol{x})$ and $P(j=2|\boldsymbol{x})$ denote the probability of $\boldsymbol{x}$ being bona fide and spoofed, respectively, and where $\boldsymbol{w}_j$ and b_j are the j-th row of the matrix $\boldsymbol{W}$ and the bias vector $\boldsymbol{b}$ of the affine transformation layer, respectively.

Given a well-trained CM, the confidence score can be estimated as $c = \max_j P(j|\boldsymbol{x})$ [13]. Note that, when the CM uses an angular softmax (e.g., [20]), the logit is computed as a cosine similarity $l_j = \frac{\boldsymbol{w}_j{}^\top \boldsymbol{h}}{||\boldsymbol{w}_j||\cdot||\boldsymbol{h}||}$, and an additional hyper-parameter α is needed to scale the logit and compute $P(j|\boldsymbol{x}) = \frac{\exp(\alpha l_j)}{\sum_{k=1}^{2}\exp(\alpha l_k)}$.

2.2. Energy-based confidence score

Given the logits $\boldsymbol{l} = [l_1, l_2]^\top$ from a pre-trained CM, the second estimator computes an energy-based confidence score as $c = \log\sum_{j=1}^{2}\exp(l_j)$ [12]. The value of c is argued to be proportional to the unconditional model likelihood $p_\phi(\boldsymbol{x})$, where ϕ is the parameter set of the CM. Therefore, *known* attacks are likely to receive a higher score c than *unknown* ones.

This estimator can be directly used on pre-trained CMs. When some *unknown* trials are available, it is also possible to re-train the CM with an energy-based training loss [12], which encourages the confidence scores of *unknown* and *known* trials to be separable.

2.3. Negative Mahalanobis distance

Assuming that the utterance-level vectors $\boldsymbol{h}$ of trials from the same attack type follow a Gaussian distribution, we can use the negative Mahalanobis distance [11] as a confidence estimator. Given the $\boldsymbol{h}$ of an input trial, we compute

$$c = -\min_k (\boldsymbol{h}-\widehat{\boldsymbol{\mu}}_k)^\top \widehat{\boldsymbol{\Sigma}}_k^{-1}(\boldsymbol{h}-\widehat{\boldsymbol{\mu}}_k), \tag{1}$$

where $\widehat{\boldsymbol{\mu}}_k$ and $\widehat{\boldsymbol{\Sigma}}_k$ are the sample mean vector and the covariance matrix of the k-th class. Note that the class here can be bona fide or any *known* attack in the training set. This method has been used for OOD detection [11] and CM scoring [21]. Here, we assume that a trial far away from the *known* classes – and hence with a smaller score c – is likely to be *unknown* and misclassified by the CM.

When some *unknown* trials are available, we can fine-tune the CM to tighten the Gaussian distributions of the *known* classes. This is done in this work using an outlier exposure training loss [22].

2.4. Confidence branch

The fourth confidence estimator adds a trainable module $\mathcal{H}_\psi$ to the CM [10]. It learns to map the vector $\boldsymbol{h}$ of the input trial into a confidence score $c = \sigma(\mathcal{H}_\psi(\boldsymbol{h}))$, where $\sigma(\cdot)$ is the Sigmoid function. The $\mathcal{H}_\psi(\cdot)$ is jointly trained with the CM by minimizing the loss over the training data $\{\boldsymbol{x}, y\}$:

$$\mathcal{L}_{\psi\phi}(\boldsymbol{x}, y) = -\sum_{j=1}^{2}\delta(y=j)\log\tilde{P}_j - \lambda\log c, \tag{2}$$

Table 1: Summary of confidence estimators compared in this study.

	Score range	Trainable?	Use of *unknown* data
Max prob.	$c \in (0.5, 1)$	No	Not applicable
Energy score	$c \in \mathbb{R}$	No	Usable for CM training
Neg. M-dist.	$c \in (-\infty, 0)$	Trained after CM	Usable for CM training
Conf. branch	$c \in (0, 1)$	Jointly trained with CM	Not applicable
Supervised	$c \in (0, 1)$	Separately trained	Required

where $\tilde{P}_j = cP(j|\boldsymbol{x}) + (1-c)\delta(y=j)$, $\delta(\cdot)$ is an indicator function, $y \in \{1, 2\}$ is the target label, and ϕ denotes the parameter set of the CM scoring module.

When the CM predicts a small confidence score c, the value of $\tilde{P}_y$ for the target class is increased, while that of $\tilde{P}_{j\neq y}$ is decreased. This means that the CM can ask for more hints from the target label y when it is less confident to classify the input. However, the regularization term $-\lambda\log c$ prevents the CM from predicting a small c for all trials. Therefore, a well-trained CM is expected to predict a small c only for trials that are difficult to classify.

We implemented $\mathcal{H}_\psi(\boldsymbol{h})$ using two linear layers, where the first layer used 128 hidden units and the Tanh activation function. We followed the official implementation and used a budget mechanism to tune the hyper-parameter λ. We also observed that it is essential to balance the ratio of bona fide and spoofed trials in each mini-batch.

2.5. Supervised binary *known-unknown* classifier

The last estimator in this study is a standalone DNN that predicts the confidence score $\boldsymbol{c}$ from input acoustic features. It has the same network structure as the CM but the target class is either *known* or *unknown*. Bona fide and spoofed trials in the original training set are treated as *known*, and trials from other databases are *unknown*.

2.6. Remarks

The confidence estimators used in this study are summarized in Tab. 1. We included them because they cover various application scenarios. When the CM scoring module has been trained and cannot be fine-tuned, the max-probability, energy-based score, or M-distance can be used. If it is possible to train the CM while only *known* data is available, the confidence branch can be used. If we collect new *unknown* training data, we can choose to build a standalone confidence estimator or update the Gaussian statistics for the M-distance after tuning the CM.

3. EXPERIMENT

3.1. Databases and protocols

We used three databases: the ASVspoof 2019 LA database [14], bona fide and TTS trials collected from Blizzard Challenge 2019 (BC19) [23] and ESPNet [24], and bona fide and VC trials from Voice Conversion Challenge (VCC) 2018 and 2020 [15, 16]. The BC19, ESPNet, and VCC datasets contain more types of spoofed trials than LA, and the sets of speakers are disjoint in all databases.

To simulate real application scenarios, we prepared different training and test sets, which are listed in Tab. 3. The LA test set was split into two subsets. *LA test kn.* contains bona fide trials and four spoofing attacks (A08, A09, A16, and A19), and *LA test unk.* contains the rest of the spoofed data in the test set. Note that A16 and A19 are *known* because they used exactly the same TTS/VC algorithms as two attackers in the training set. A08 and A09 are

Table 2: Experiment results for CM and confidence scoring. Training and test sets are defined in Tab. 3. CM scoring components are based on LCNN-LSTM (Section 3.2) but have different softmax functions. Symbol ↓ indicates that lower EER, C_{llr}, and FPR are better, and ↑ suggests that higher AUROC and AUPR are better. For visualization, **cells with better performance have lighter background color**.

Train set	CM scoring	Confidence scoring	Test set E1						Test set E2					
			C_{llr} ↓	EER ↓	At TPR=95% EER ↓	At TPR=95% FPR ↓	AUROC ↑	AUPR ↑	C_{llr} ↓	EER ↓	At TPR=95% EER ↓	At TPR=95% FPR ↓	AUROC ↑	AUPR ↑
T1	AM softmax	Max prob.	0.64	4.64	-	-	0.51	0.38	0.65	5.50	-	-	0.51	0.35
		Energy	0.64	4.64	4.25	80.39	0.65	0.63	0.65	5.50	3.52	83.34	0.62	0.57
		Conf. branch	0.61	3.72	3.98	88.05	0.75	0.70	0.61	6.05	5.50	89.66	0.72	0.68
	plain softmax	Max prob.	0.45	3.33	3.43	70.98	0.78	0.64	0.41	5.55	2.82	72.91	0.70	0.49
		M-distance	0.45	3.33	3.28	98.69	0.55	0.43	0.41	5.55	4.14	88.57	0.63	0.52
		Energy	0.45	3.33	3.47	71.14	0.79	0.70	0.41	5.55	2.85	72.79	0.70	0.49
		Conf. branch	0.64	3.60	4.07	73.86	0.76	0.70	0.61	6.39	4.95	81.44	0.72	0.61
T2	plain softmax	Supervised	0.45	3.33	3.22	97.59	0.35	0.29	0.41	5.55	5.97	98.17	0.34	0.27
		M-distance	0.30	4.34	4.53	91.71	0.52	0.38	0.33	6.52	6.21	94.50	0.53	0.38
		Energy	0.35	5.06	3.98	75.20	0.73	0.65	0.62	9.10	6.47	74.38	0.70	0.49
T3	plain softmax	Supervised	0.45	3.33	3.39	98.44	0.42	0.40	0.41	5.55	5.62	91.35	0.55	0.51
		M-distance	0.33	7.14	7.25	91.71	0.52	0.38	0.39	11.35	11.26	94.54	0.52	0.38
		Energy	0.34	8.74	8.12	88.68	0.58	0.47	0.34	9.42	7.42	83.71	0.57	0.45

Table 3: List of configurations of training and test sets. Numbers of bona fide and spoofed trials are separated by /.

		Known trials	*Unknown* trials
Train set	T1	LA trn. (2,580 / 22,800)	-
	T2	LA trn. (2,580 / 22,800)	ESPNet (250 / 2,000)
	T3	LA trn. (2,580 / 22,800)	BC19 (100 / 7,525)
Test set	E1	LA test kn. (7,355 / 19,656)	LA test unk. (0 / 44,226)
	E2	LA test kn. (7,355 / 19,656)	VCC (770 / 49,467)

treated as *known* in this study because they use a TTS framework similar to some spoofing attacks in the training set.

Using T1 and E1 is equivalent to the official protocol of ASVspoof 2019 LA. Using E2 simulates a scenario in which some test trials are *unknown*. The use of T2 and T3 simulates a scenario in which a small amount of *unknown* trials can be used to train the confidence estimator. However, these *unknown* trials are disjoint from those in the test set.

3.2. Model configurations and training recipes

We followed our previous study to configure the CMs since they performed well on the ASVspoof 2019 LA database [19]. The acoustic features were linear frequency cepstrum coefficients (LFCC) extracted with a frame length of 20ms, a frame shift of 10ms, and a 512-point FFT. The LFCC vector per frame had 60 dimensions, including static, delta, and delta-delta components. We compared two back-end classifiers in the experiment. While both are based on the light CNN (LCNN) [25] with two bi-directional LSTM layers and an average pooling layer, one uses an **plain softmax**, and the other uses an additive margin-softmax (**AM-softmax**) [20] with the hyper-parameter set from [19]. These CMs were combined with the confidence estimators for the experiments.

The training recipe was borrowed from our previous study: the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ [26], a mini-batch size of 64, and a learning rate initialized to 3×10^{-4} and halved every ten epochs. Each model was trained on an Nvidia Tesla A100 card for three rounds, and the result was averaged. Voice activity detection and feature normalization were not applied.

3.3. Evaluation metrics

The following evaluation metrics were used. The first set, including EER and C_{llr} [27], was used to evaluate the CMs in the conventional scenario without discriminating *known* and *unknown* trials. C_{llr} was used because it is a measure of both discrimination and calibration.

The second set of metrics were for the confidence estimators. By treating *known* and *unknown* as positive and negative classes, respectively, we computed the false positive rate (FPR) given the threshold θ_c for which the true positive (TPR) rate is 95%. This is used in other studies [13, 22]. We also computed the area under ROC (AUROC) and the area under the precision-recall curve (AUPR).

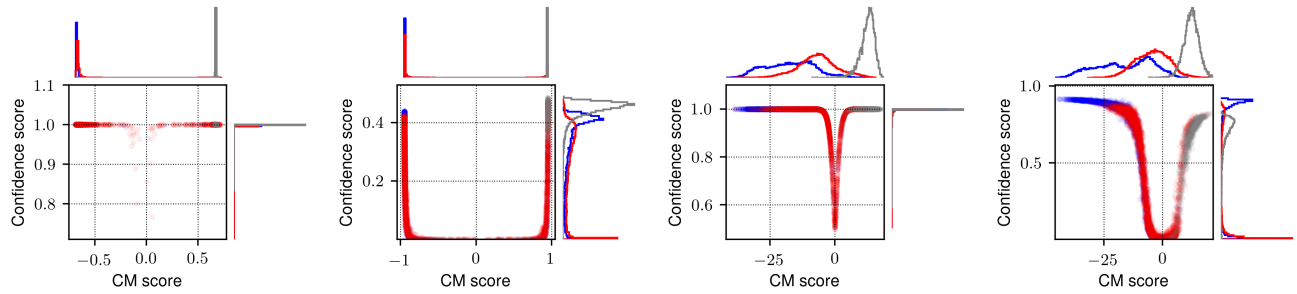
Finally, another EER for the CMs was computed for trials whose confidence score was larger than θ_c at TPR=95%. This EER measures how well a CM discriminates bona fide and spoofed trials that the CM is confident about.

3.4. Results and discussions

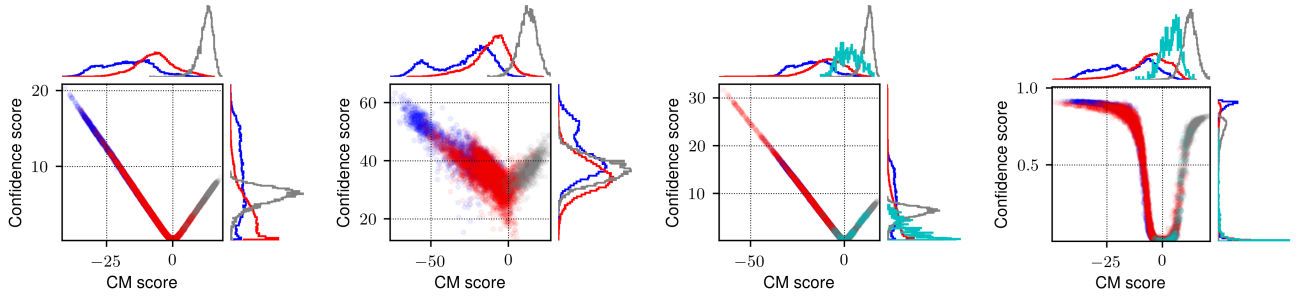
The evaluation results are listed in Tab. 2. Note that the CMs with a confidence branch were trained with the loss in Eq. (2), and their C_{llr} and EER were hence different from the others that used a non-trainable confidence estimator. The CMs with the energy-based confidence estimator were also re-trained when using T2 or T3.

Which confidence estimator is effective? To answer this question, we first focus on the condition using the training set T1, test set E1, and the CMs with the AM softmax. Among the three confidence estimators, the max-probability-based one performed poorly. As shown in Fig. 2a, the confidence score was close to the maximum value of 1.0 for most of the *unknown* test trials (in red color), and it was impossible to compute FPR at TPR=95%. This indicates that the CM was overconfident, which is consistent with the findings in other studies [17, 18]. In comparison, the energy-based method and the confidence branch produced relatively useful confidence scores. Although the FPR at TPR 95% was higher than 80% for both methods, the AUROC and AUPR improved. Particularly, as shown in Fig. 2b, the confidence branch produced confidence scores that varied for the three types of test trials even though most of the CM scores were either -1 or 1.

For T1-E1, the max probability, energy-based scoring, and confidence branch improved the AUROC values when the CM used



(a) T1-E1, AM, Max prob. (b) T1-E1, AM, Conf. branch (c) T1-E1, plain, Max prob. (d) T1-E1, plain, Conf. branch



(e) T1-E1, plain, Energy (f) T2-E1, plain, Energy (g) T1-E2, plain, Energy (h) T1-E2, plain, Conf. branch

Fig. 2: Scatter plot and histogram of CM and confidence scores. Each sub-caption lists training-test set, CM with AM or plain softmax, and confidence estimator. *Known* bona fide, *unknown* bona fide, *known* spoofed, and *unknown* spoofed trials are in different colors.

the plain softmax rather than the AM one. However, although the results of the max-probability method were close to the other two methods, Fig. 2c shows that many of the *unknown* spoofed trials (in red color) still received a confidence score close to 1.0. In contrast, the confidence scores from the other two methods were more dispersed as Figs. 2d and 2e show. The M-distance-based method was not competitive as the results demonstrate.

The energy-based score and confidence branch were good candidates for the task. Interestingly, the energy-based confidence scores were highly correlated with the CM scores. This is partially due to the large numeric difference between the logits $\{l_1, l_2\}$. The confidence score becomes $c \approx \max_j(l_j)$ and correlates with the CM score $s = l_1 - l_2$. This also indicates that the logits from the plain softmax contain useful information for confidence estimation.

When is the confidence estimation useful for the CM? In the condition T1-E1, the EER at TPR=95% was not lower than the original EER in most cases. The CM cannot better discriminate bona fide and spoofed trials on which the CM is confident. One possible reason is that the CM has been unintentionally overfitted to the attackers in the LA test set (see the footnote on the 1st page). Therefore, to reveal the usefulness of confidence estimation, we need to examine the performance on real *unknown* trials in the test set E2. From the results for T1-E2, we observed that the EER at TPR=95% was lower than the original EER for all confidence estimators except the max-probability-based one for the AM softmax CM. As Fig. 2g on the energy-based method shows, the bona fide (in green color) and spoofed (in red color) trials from VCC had smaller confidence scores than those from LA (in grey and blue color). This is expected since the trials from VCC were quite different from those in the CM's training set. Avoiding making decisions on these low-confidence trials is a reasonable strategy for the CM.

Can we improve the confidence estimator if we have some *unknown* spoofing data? A comparison between the metrics across T1, T2, and T3 indicate that it was not effective to use *unknown* training data to fine-tune the CM with an M-distance- or energy-based confidence estimator. In the case of using the energy-based estimator on T2, a comparison between Fig. 2e and 2f shows that the confidence score of *unknown* spoofed test trials was pushed towards the *known* ones. Although the M-distance's FPR was slightly improved when evaluating on E1, it was higher than 90%. Note that the M-distance achieved similar FPR, AUROC, and AUPR values for T2-E1 and T3-E1 because the confidence scores were similar.

Last but not least, the estimator trained in a supervised manner produced an AUROC smaller than or around 0.5. One possible reason is that the trials for T2 and T3 were too different from those for T1 and the test set. The confidence estimator learned a simple decision boundary that separated T2 and T3 from T1 but did not generalize to the spoofing attacks in the test set.

4. CONCLUSION

This study investigated speech spoofing CMs that can opt for abstention. This is implemented by augmenting the CMs with a confidence estimator and comparing the confidence score of an input trial against a decision threshold. We compared various methods to estimate the confidence score and conducted experiments on a mix of speech databases. The results on the ASVspoof 2019 LA database demonstrated that the energy-based confidence score can be a convenient method for estimating the confidence for pre-trained CMs. The confidence branch is also a potential candidate. Another experiment with *unknown* spoofed trials from the VCC database showed that the CM can reduce misclassification rate if it can refrain from classifying low-confidence trials.

Future work will investigate other confidence estimation methods, including using calibrated CM score [27]. Another direction is to borrow the idea of active learning and add the 'unknown' trials with low confidence scores to the CM training set.

5. REFERENCES

[1] Nicholas
Evans, Tomi Kinnunen, and Junichi Yamagishi, "Spoofing and countermeasures for automatic speaker verification," in *Proc. Interspeech*, 2013, pp. 925–929.

[2] ISO/IEC JTC1 SC37 Biometrics, *ISO/IEC 30107-1. Information Technology - Biometric presentation attack detection - Part 1: Framework*, 2016.

[3] Andreas Nautsch, Xin Wang, Nicholas Evans, Tomi H. Kinnunen, Ville Vestman, Massimiliano Todisco, Hector Delgado, Md Sahidullah, Junichi Yamagishi, and Kong Aik Lee, "ASVspoof 2019: Spoofing Countermeasures for the Detection of Synthesized, Converted and Replayed Speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 252–265, apr 2021.

[4] Dipjyoti Paul, Md Sahidullah, and Goutam Saha, "Generalization of spoofing countermeasures: A case study with ASVspoof 2015 and BTAS 2016 corpora," in *Proc. ICASSP*. IEEE, 2017, pp. 2047–2051.

[5] Rohan Kumar Das, Jichen Yang, and Haizhou Li, "Assessing the scope of generalized countermeasures for anti-spoofing," in *Proc. ICASSP*. IEEE, 2020, pp. 6589–6593.

[6] Joshua Attenberg, Panos Ipeirotis, and Foster Provost, "Beat the machine: Challenging humans to find a predictive model's "unknown unknowns"," *Journal of Data and Information Quality (JDIQ)*, vol. 6, no. 1, pp. 1–17, 2015.

[7] Rohan Kumar Das, "Known-unknown Data Augmentation Strategies for Detection of Logical Access, Physical Access and Speech Deepfake Attacks: ASVspoof 2021," in *Proc. ASVspoof Challenge workshop*, 2021, pp. 29–36.

[8] Burr Settles, "Active Learning Literature Survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[9] Heinrich Jiang, Been Kim, Melody Y Guan, and Maya Gupta, "To trust or not to trust a classifier," *Proc. NIPS*, pp. 5546–5557, 2018.

[10] Terrance DeVries and Graham W Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[11] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin, "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks," in *Proc. NIPS*, 2018, pp. 7167–7177.

[12] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li, "Energy-based Out-of-distribution Detection," in *Proc. NIPS*, 2020, vol. 33, pp. 21464–21475.

[13] Dan Hendrycks and Kevin Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *Proc. ICLR*, 2017.

[14] Xin Wang, Junichi Yamagishi, Massimiliano Todisco, Héctor Delgado, Andreas Nautsch, Nicholas Evans, Md Sahidullah, Ville Vestman, Tomi Kinnunen, Kong Aik Lee, Lauri Juvela, Paavo Alku, Yu-Huai Peng, Hsin-Te Hwang, Yu Tsao, Hsin-Min Wang, Sébastien Le Maguer, Markus Becker, Fergus Henderson, Rob Clark, Yu Zhang, Quan Wang, Ye Jia, Kai Onuma, Koji Mushika, Takashi Kaneda, Yuan Jiang, Li-Juan Liu, Yi-Chiao Wu, Wen-Chin Huang, Tomoki Toda, Kou Tanaka, Hirokazu Kameoka, Ingmar Steiner, Driss Matrouf, Jean-François Bonastre, Avashna Govender, Srikanth Ronanki, Jing-Xuan Zhang, and Zhen-Hua Ling, "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, pp. 101114, nov 2020.

[15] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, Tomi Kinnunen, and Zhenhua Ling, "The Voice Conversion Challenge 2018: Promoting Development of Parallel and Nonparallel Methods," in *Proc. Odyssey*, 2018, pp. 195–202.

[16] Zhao Yi, Wen-Chin Huang, Xiaohai Tian, Junichi Yamagishi, Rohan Kumar Das, Tomi Kinnunen, Zhen-Hua Ling, and Tomoki Toda, "Voice Conversion Challenge 2020 — Intra-lingual semi-parallel and cross-lingual voice conversion —," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 80–98.

[17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger, "On calibration of modern neural networks," in *Proc. ICML*. PMLR, 2017, pp. 1321–1330.

[18] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic, "Revisiting the Calibration of Modern Neural Networks," *arXiv preprint arXiv:2106.07998*, 2021.

[19] Xin Wang and Junich Yamagishi, "A comparative study on recent neural spoofing countermeasures for synthetic speech detection," *Proc. Interspeech*, pp. 4259–4263, 2021.

[20] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[21] Nanxin Chen, Yanmin Qian, Heinrich Dinkel, Bo Chen, and Kai Yu, "Robust deep feature for spoofing detection—The SJTU system for ASVspoof 2015 challenge," in *Proc. Interspeech*, 2015, pp. 2097–2101.

[22] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich, "Deep anomaly detection with outlier exposure," *Proc. ICLR*, 2019.

[23] Zhizheng Wu, Zhihang Xie, and Simon King, "The blizzard challenge 2019," in *Proc. Blizzard Challenge Workshop*, 2019.

[24] Tomoki Hayashi, Ryuichi Yamamoto, Katsuki Inoue, Takenori Yoshimura, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Yu Zhang, and Xu Tan, "Espnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *Proc. ICASSP*. IEEE, 2020, pp. 7654–7658.

[25] Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and Alexandr Kozlov, "STC Antispoofing Systems for the ASVspoof2019 Challenge," in *Proc. Interspeech*, 2019, pp. 1033–1037.

[26] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014.

[27] David A Van Leeuwen and Niko Brümmer, "An introduction to application-independent evaluation of speaker recognition systems," in *Speaker classification I*, pp. 330–353. Springer, 2007.