2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)

DeepfakeStack: A Deep Ensemble-based Learning Technique for Deepfake Detection

Md. Shohel Rana
Computing Sciences and Computer Engineering
The University of Southern Mississippi
Hattiesburg, MS 39406, United States
md.rana@usm.edu

Andrew H. Sung
Computing Sciences and Computer Engineering
The University of Southern Mississippi
Hattiesburg, MS 39406, United States
andrew.sung@usm.edu

***Abstract*—Recent advances in technology have made the deep learning (DL) models available for use in a wide variety of novel applications; for example, generative adversarial network (GAN) models are capable of producing hyper-realistic images, speech, and even videos, such as the so-called "Deepfake" produced by GANs with manipulated audio and/or video clips, which are so realistic as to be indistinguishable from the real ones in human perception. Aside from innovative and legitimate applications, there are numerous nefarious or unlawful ways to use such counterfeit contents in propaganda, political campaigns, cybercrimes, extortion, etc. To meet the challenges posed by Deepfake multimedia, we propose a deep ensemble learning technique called DeepfakeStack for detecting such manipulated videos. The proposed technique combines a series of DL based state-of-art classification models and creates an improved composite classifier. Based on our experiments, it is shown that DeepfakeStack outperforms other classifiers by achieving an accuracy of 99.65% and AUROC of 1.0 score in detecting Deepfake. Therefore, our method provides a solid basis for building a Realtime Deepfake detector.**

Keywords-Deepfake; DeepfakeStack; GANs; Deep Ensemble Learning;, Greedy Layer-wise Pretraining.

I. INTRODUCTION

Recent progress in automated video processing applications (e.g., FaceApp [1], FakeApp [2]), artificial neural networks (ANN), ML algorithms and social media allow cybercriminals to create and spread high-quality manipulated video contents (aka. fake videos) through digital media that lead to the appearance of deliberate misinformation. The illustration of certain entities acting or stating things they never actually said or performed are becoming remarkably practical, even hard to recognize the proof of authenticity. The keyword "Deepfake manipulation" permits anyone to swap the face of an individual by another's face, including expressions and creates photorealistic fake images or videos that are known as Deepfakes. These videos are readily visible in malicious use whereas some of them could be harmful to individuals and society. In the past, video manipulation was an expensive task that required an extensive amount of workforce, time, and money, but now, only it needs a gaming laptop or desktop with an internet connection and a basic knowledge of ANN. Deepfakes became popular when fabricated porn videos of well-known faces; for example, celebrities or politicians are in progress of making it online. The term violates not only the rules of consent but the victim's privacy. Because creating Deepfakes without a person's approval is a form of abuse leading in another way of crime.

As presented in the annual report [3] under the broader name of Deepfake, Google searches provide several webpages for the keyword 'Deepfake' that expanded rapidly since 2017, as well as searches for webpages containing related videos (see Fig. 1). This report also presents.

- **1790+** of Deepfake videos hosted by the top 10 adult websites without considering *pornhub.com,* which has disabled searches for 'Deepfakes'.
- **6174** of Deepfake videos hosted by adult websites featuring fake video content only.
- **3** new sites dedicated to hosting Deepfake pornography.
- **902** of papers published on the arXiv, including '*GAN (Generative Adversarial Network)'* in titles or abstracts in 2018 only.
- **25** articles on the topic published, including non-peer, reviewed where DARPA funds 12 of them.

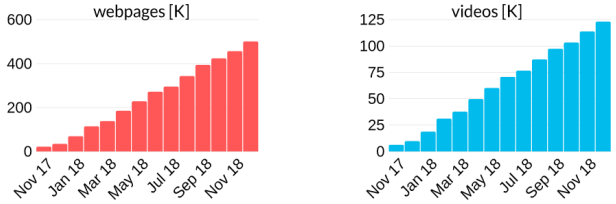


Figure 1. (a) Number of webpages returned by Google search for "Deepfake", (b) Number of searches for webpages containing Deepfake videos.

In this paper, we apply a deep ensemble learning technique, namely, DeepfakeStack by evaluating several DL based state-of-art models. The idea behind the DeepfakeStack is based on training a meta-learner on top of pre-trained base-learners and offers an interface to fit the meta-learner on the predictions of the base-learners and shows how the ensemble technique performs the classification task. The architecture of the DeepfakeStack involves two or more base learners, called *level-0 models,* and a meta-learner called *level-1 model* that combines the

978-1-7281-6550-9/20/$31.00 ©2020 IEEE
DOI 10.1109/CSCloud-EdgeCom49738.2020.00021

predictions of these *level-0 models*. The *level-1 model* is trained on the predictions made by base models on out-of-sample data. That is, data not used to train the base models is fed to the base models, predictions are made, and these predictions, along with the expected outputs, provide the input and output pairs of the training dataset used to fit the meta-model. In this experiment, XceptionNet, ResNet101, InceptionResNetV2, MobileNet, InceptionV3, DenseNet121, and DenseNet169 are used as base-learners and a newly defined CNN model as a 2nd level meta learner which is also known as Deepfake Classifier (DFC). The experiment using these models shows that the DeepfakeStack achieves an accuracy of 99.65% and AUROC of 1.0 and the promising results are shown.

The rest of the paper has been formatted as Sect. 2 gives an overview of Deepfake and a brief about deep ensemble learning techniques; Sect. 3 describes related works; Sect. 4 presents methodology including dataset description, data preprocessing, proposed technique, the technology used; Sect. 5 presents results and analysis, and Sect. 6 gives conclusions and future work.

II. OVERVIEW

A. Definition of Deepfake

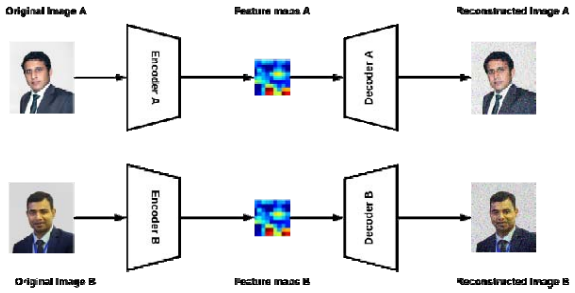
A combination of "*Deep Learning*" and "*Fake*" can be called Deepfake that refers to any photo-realistic audiovisual content created using the DL technology. The technique is initiated by analyzing plenty of photos or a video of one's face, training an AI algorithm to manipulate that face, and then using that algorithm to map the face onto a person in an image or video. In late 2017, the term "*Deepfake*" is named after a Reddit user known as Deepfakes, who used DL technology and attempted to replace a target actor's face with another's face in pornographic videos. In recent, two popular facial manipulation methods have attracted a lot to cybercriminals in doing video manipulation job.

- *Facial expression manipulation*: Face2Face [4] allows anyone to transfer the facial expressions of a person to a different person using standard applications in real-time. For example, "*Synthesizing Obama*" [5] can animate a person's look by transferring another person's facial expression based on an audio input sequence.
- *Facial identity manipulation*: In FaceSwap [6], the face of a person is replaced by any other person's face instead of changing expressions. For example, Snapchat. The same methodology is applied in Deepfake using DL technology.

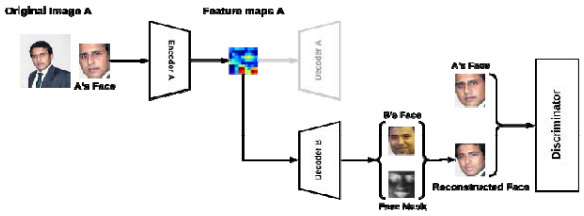
B. Deepfake Generation Pipeline

To create Deepfakes (Deepfake image or video), researchers apply the face-swapping technique using Generative Adversarial Networks (GANs) [7-8]. A combination of two neural networks builds GANs used in the Deepfakes generation. One is called a synthesizer or generative network, and another is called a detector or discriminative network. These two networks participate in generating realistic fake videos or images. The generative system is built with encoder and decoder and responsible for creating images, and the discriminative network determines whether the created representation is accurate and believable. The below example describes how it works (see figure 2).

1. After collecting many images for both actors (***A*** and ***B***), build an encoder for encoding all these images to extract essential features and then use a decoder to reconstruct the corresponding image.
2. Use different decoders for actor ***A*** and actor ***B*** for decoding the features. To do this, using the backpropagation algorithm train the generative network in such a way that the input is fitted tightly with the output.
3. After the training, it needs to deal with the video frame-by-frame to exchange the ***A***'s face with the ***B***'s face. To do this job, first, it needs to extract ***A***'s face using a standard face detection technique and feed it into the encoder. Then use the decoder of actor ***B*** to reconstruct the image instead of feeding to the decoder of ***A***. Finally, amalgamate the original image into the newly formed face image.
4. At the final step, feed original images to the discriminative network and train itself to identify originality better. It confirms the created images as human eyed indistinguishable to the authentic.



(a) Encode all images of both actors (**A** and **B**) using the generative network.



(b) Reconstruct the image of actor **A** by using the decoder of the actor **B** instead of feeding to the decoder of **A** and then merging it into the original image. Using the discriminative network confirm the created image is accurate and believable;

Figure 2. Deepfake generation pipeline.

C. Deep Ensemble Learning Technique

The ensemble technique is a way of combining a list of sub-models (also known as base-learners) to form an ideal perceptive model where each sub-model contributes to producing the final output. The newly generated combined model is called a meta-model. The two commonly used methods are [9]:

a. *Stacking ensemble (SE)*: The SE takes the output of the base-learners and used them as input to train the meta-learner in such a way that the model learns how to best map the base-learner's decisions into an enhanced output.
b. *Randomized weighted ensemble (WRE)*: In WRE technique, each of the base-learners is weighted by a value based on their performance that is evaluated on a hold-out validation dataset. The model receives a higher weight if it performs better than others. In other words, this technique is just optimizing weights that are used for weighting all the base-learners' output and taking the weighted average.

III. Related Works

A. Facial Manipulations

Guera and Delp [10] proposed a system that contains two essential components: (i) a CNN and (ii) an LSTM. For a given image sequence, the CNN generates a set of features for each frame and passes them to the LSTM for analysis after concatenating those features of multiple sequential frames. The proposed network was trained on 600 videos and achieved 97.1% of accuracy. Li and Lyu [11] proposed a method for detecting warping artifacts in Deepfake videos by training four different CNN on authentic and manipulated face images and obtained the accuracy up to 99%. In [12] authors proposed another plan that reveals fake faces using eye blinking detection technique, in which authors assume that function is missing in Deepfake analysis. Afchar et al. [13] trained a CNN, namely MesoNet, for classifying the real and Deepfake manipulated face. The network has two inception modules, namely Meso-4 and MesoInception-4, in conjunction with two convolution layers connected by max-pooling segments. Zhou et al. [14] proposed a two-stream network for detecting face manipulation in the video. In its first stream, a CNN based face classification network is trained to capture tampering artifact evidence, and in the second stream, a steganalysis feature-based triplet network is trained for controlling functions that capture local noise residual evidence. In [15], the authors proposed a two-phases method, where the first phase crops and adjusts the facial area that is taken from the video frames using computer-generated masks, and the second step is for detecting such manipulation using a recurrent convolutional network (RCNN). The experiment has improved the performance of detection accuracy in detecting such manipulation than the previously reported results up to 4.55%. In [16], Do et al. suggested a deep CNN model for classifying a real image or fake image generated by GANs. The main objectives of this article can be concise as (i) creation of training data sets that can be adapted to the test data set, (ii) building a deep learning network based on face recognition networks for extracting face features, and finally, (iii) performing a fine-tune to fit the face features to classify the real/fake face.

B. Digital Media Forensics

Much work has been done on digital media forensics, early papers include, e.g. [17, 18]; Cozzolino et al. [19] introduced an autoencoder-based architecture named *ForensicTransfer* (FT), which differentiates authentic images from counterfeit. The FT establishes a series of experiments and achieves up to 80-85% in terms of accuracy. Nguyen et al. [20] proposed multi-task learning based CNN for concurrently carrying out detection and segmentation of manipulated facial images and videos. The proposed system contains an encoder that encodes features used for the binary classification and a Y-shaped decoder where the output of one of its branches is used for segmenting the manipulated regions. In [21], the authors presented a deep CNN based model that uses a capsule network (CN) to detect Deepfake. In addition to this, it identifies replay attacks and computer-generated image. In [22], the authors proposed an approach for building a Deepfake detector called FakeCatcher (FC) to detect synthetic portrait videos where the proposed method exploits biological signals extracted from facial areas. In [23] method, missing reflections and missing details in the eye and teeth areas are being exploited and the texture features are extracted from the facial region based on facial landmarks and are fed them into the ML classifiers for classifying them as the Deepfakes or the real videos. Koopman et al. [24] explored a photo response non-uniformity (PRNU) analysis to detect Deepfake in video frames. In this PRNU analysis, a series of frames are generated from the input videos and are kept in sequentially labeled folders. To leave the portion of the PRNU pattern and to make it consistent, it crops each video frame by precisely identical pixels values. These frames are then split into 8 groups where each of them is equal size, and for each, a typical PRNU pattern is created using second order FSTV method and compare them to one another with calculating the normalized cross-correlation scores. For each video, it estimates variations in correlation scores and the average correlation score. Finally, it performs a t-test on these results for Deepfakes and real videos where the t-test produces statistical significances between the results for both Deepfakes and real videos.

IV. Methodology

A. Dataset Description

To conduct the experiment and evaluate the proposed technique, we have used the FaceForensics++ (FF++) dataset [25]. The dataset has been collected by Visual Computing Group (VGG) which is an active research group on computer vision, computer graphics, and machine learning. There are 1000 real videos included in this dataset that was downloaded from YouTube. Then, manipulated them by using 3 popular state-of-art manipulation techniques (i.e., Deepfake, FaceSwap, and Face2Face).

B. Data Analysis and Preprocessing

To achieve the best performance of the used ML/DL models, we need to preprocess the dataset by applying some data analysis. Below the idea of how this dataset is organized as follows:

- ✓ In this experiment, we have used 49 real and 51 fake to make the balanced dataset. After then, we

separate each of the videos based on its category under the directories (e.g., *Original*, *Deepfakes*).

- ✓ For each video, each folder is created that contains all extracted image sequences. For example, if the video file's name is '485.mp4' then we create a directory with the same name '485' where it contains all the frames of '485.mp4' for each of the original videos and we have followed the same procedure for Deepfakes data.
- ✓ We don't consider the entire video sequence instead; we take only 101 frames from each video to reduce the computational time.

As the main objective is to detect the manipulated face images, we are concerned only in the face area. So, it is a good idea to ignore all others like the body, background, etc. Therefore, we track the face in each of the images and feed them into the classifier.

C. DeepfakeStack

The DeepfakeStack provides a way of combining a set of k base-learners, $C_1, C_2, \ldots, C_k$, to produce an enhanced classifier C_*. For a given dataset, D, it splits it into k training sets, $D_1, D_2, \ldots, D_k$, and uses D_i to build the base-learner, C_i. For classifying a new (unseen) data tuple, the DeepfakeStack returns a class prediction based on the votes of these base-learners. Simply, for a given tuple X to classify, it accumulates the class label predictions obtained from the base-learners and yields the class in the majority. The algorithm can be summarized in figure 3.

Algorithm DeepfakeStack Classifier (DFSC)

Input: Training data, $D = \{x_i, y_i\}_{i=1}^{m} (x_i \in \mathbb{R}^n, y_i \in Y)$

Output: Ensemble-based classifier, DFSC

1. Generate base learners/classifiers, $c_1, c_2, \ldots, c_T$
2. **foreach** base learner in C
3. Learn base learner c_t based on D
4. **end foreach**
5. Create new dataset from D
6. **for** i ← 1 to m **do**
7. Create new dataset, D' that contains $\{x_i', y_i\}, where\ x_i' = \{c_1(x_i), c_2(x_i), \ldots, c_T(x_i)\}$
8. **end for**
9. Step 3: Learn meta learner (2nd level classifier)
10. Learn a new classifier C' based on the newly created dataset
11. **return** $\mathrm{DFSC}(x) = C'(c_1(x), c_2(x), \ldots, c_T(x))$

Figure 3. The algorithm for the DeepfakeStack classifier.

The working procedure of DeepfakeStack is split into two sections: (i) Base-Learners Creation, (ii) Stack Generalization.

- ✓ *Base-Learners Creation*: As we defined this work to solve the binary classification problem, we need to fix the label ***0*** for real and ***1*** for Deepfake and, measure both accuracy and categorical log loss. Once we are done with data analysis it is very crucial to decide what kind of models might work for this data. It is a good idea to prefer any CNN-based architecture as we have an image dataset. In addition to this, selecting picture-perfect factors is a huge challenge, which may include the number of layers, number of units, dropout rates, activations, learning rates, etc. Contemplating all, we can adjust to fine-tune any architecture relevant to the model that has already been trained and tested on a similar dataset. For example, the CNN-based networks that have already been trained and tested on the ImageNet dataset [26]. To adapt to any of these architectures, the dataset needs to be preprocessed and establish an environment accordingly. These CNN-based networks were trained with normalized images of equal size (224x224) on RGB images. Therefore, before feeding into the model, we must look after that the dataset should be normalized and preprocessed into the same size. In this experiment, we initialize 7 DL models (e.g., XceptionNet, MobileNet, ResNet101, InceptionV3, DensNet121, InceptionReseNetV2, DenseNet169) with ImageNet weights and apply the transfer learning by replacing only the topmost layer with 2 outputs with *SoftMax* activation. We consider these architectures as base-learners, and to train these models, we follow Greedy Layer-wise Pretraining (GLP) [27] technique. The GLP uninterruptedly adds a new hidden layer to a model and refit the model. Besides, it permits the newly added model to learn the inputs from the existing hidden layer, while keeping the weights for the existing hidden layers fixed. This procedure is called "*layer-wise*" as the model is trained one layer at a time and is referred to as "*greedy*" because of this layer-wise method can resolve the problem of training a deep network.
- ✓ *Stack Generalization*: Once the base-learners are ready, we need to define the meta-learner. In the case of meta-learner, we create a CNN based classifier, namely, *DeepfakeStackClassifier* (DFC), and embed in a larger multi-headed neural network to learn to obtain the best combination of the predictions from each input base-learner. This approach permits the stacking ensemble to be treated as a single large model and the benefit is that the outputs of these base-learners are provided directly to the meta-learner. In addition to this, it makes it possible to update the weights of the base-learners as well as the meta-learner model (see figure 4). The input layer of each base-learner is used as an individual input head to the DFC model. This means k copies of input data are fed to the DFC model, where k represents the number of input models (base-learners) and merge the output of each of these models. In this experiment, a simple concatenation merge has been used, where a single 14-element vector is formed from the two class-probabilities predicted by each of the 7 base-learners. To interpret this "*input*" to the meta-learner, we define a hidden layer in conjunction with an output layer that makes its probabilistic prediction.

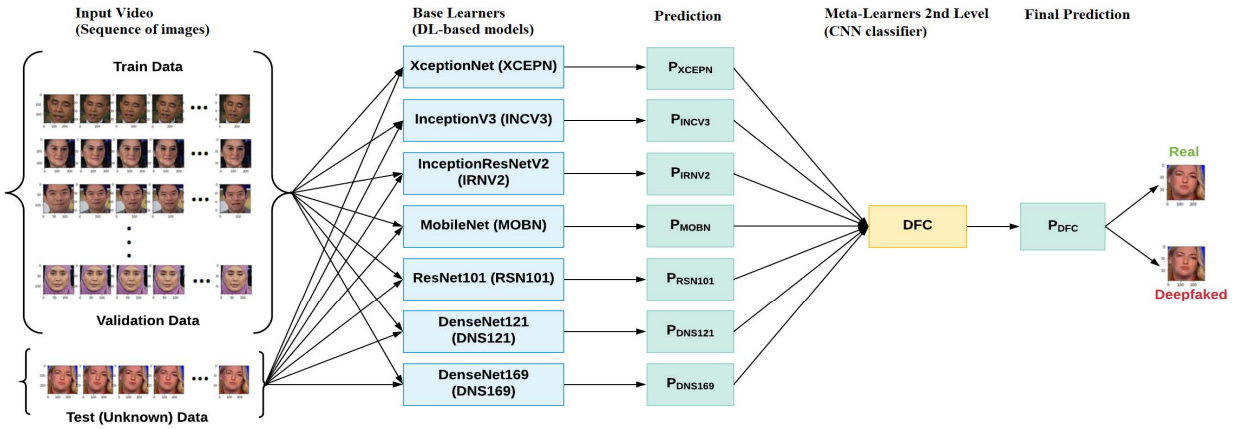


Figure 4. Overview of DeepfakeStack.

V. Results and Analysis

After defining the DFC model, we fit it directly on the holdout test dataset for 300 epochs. Note that, the weights of the base-learners will not be updated during the training since their trainable property is set to *False* (i.e., not trainable) while defining them. Only the weights of the new hidden and output layer will be updated. After successful fitting, we use this DFC model to predict unseen data and expect the DFC to perform better compared to any individual sub-model (base-learner). For comparison, the performances are depicted in Table 1.

TABLE I. Performance of the DeepfakeStack model and individual DL model (base-learner)

Model	Precision		Recall		F1-score		Accuracy	AUC ROC
	0	*1*	*0*	*1*	*0*	*1*		
XCEPN	0.94	1.00	1.00	0.94	0.97	0.97	96.88	0.976
INCV3	0.88	0.95	0.85	0.88	0.86	0.87	86.49	0.866
MOBN	0.84	1.00	1.00	0.81	0.92	0.90	90.74	0.911
RSN101	0.94	0.96	0.96	0.94	0.95	0.95	94.95	0.954
IRNV2	0.82	1.00	1.00	0.79	0.90	0.88	89.26	0.899
DNS121	0.93	1.00	1.00	0.93	0.96	0.96	96.34	0.969
DNS169	0.95	1.00	1.00	0.94	0.97	0.97	97.13	0.971
DFC	**0.99**	**1.00**	**1.00**	**0.99**	**1.00**	**1.00**	**99.65**	**1.000**

The main difference of performances among various learners or classifiers is based on their model size; for DeepfakeStack, the model size is very large and if not carefully built, it results in overfitting. The results of the overall accuracy of each DL model using the same parameters are summarized in figure 5, where it is seen that the best performance is obtained by the DeepfakeStack (DFC) model. The DFC achieves an accuracy of 99.65%. Based on the experiment, we can say that the DFC model now learned to detect the manipulated videos/images and perform very well when the video or image contents are manipulated by Deepfake.

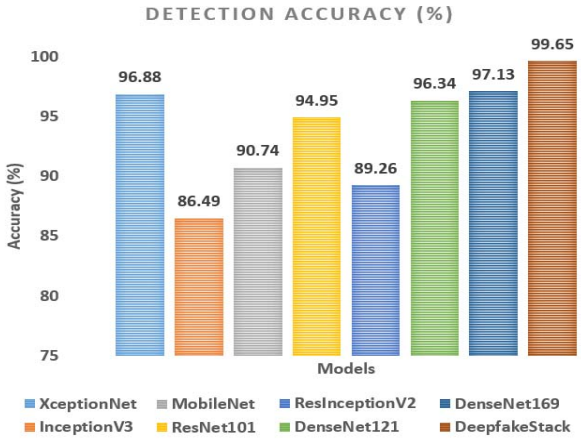


Figure 5. Accuracy.

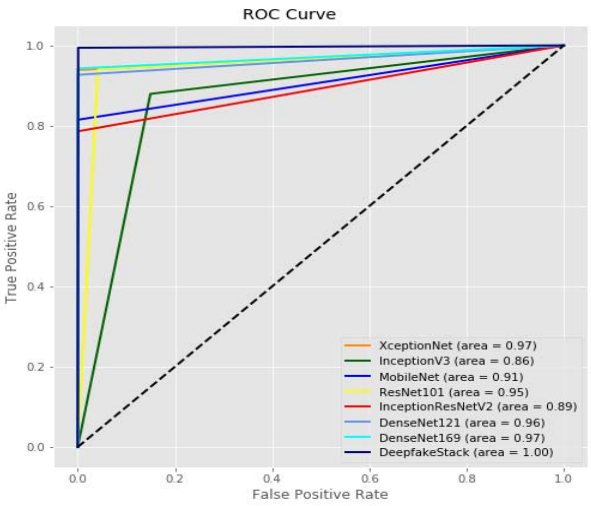


Figure 6. ROC curve.

To summarize, a ROC curve is produced per model by varying the threshold values from 0 to 1 which helps to visualize the tradeoff between *sensitivity* and *specificity* and recognize how well-separated our data classes are. As shown

in figure 6, the ROC tells us how good the model is for classifying the two classes: *Original* and *Deepfake*. The area covered by the curve is the area between the colored line and the axis where each color line represents an individual model/classifier (i.e., the blue line represents the DFC model). The bigger the area covered, the better the models are at classifying the given classes. In other words, the closer the AUCROC is to 1, the better. Based on the experiment, we can see that the DFC achieves an AUROC of 1.0 which indicates that the positive and negative data classes are perfectly separated, and the model is as efficient as it can get.

VI. Conclusions and Future Works

Detecting Deepfakes has become a significant challenge because even though many such manipulated videos are intended for entertainment, still many of them could be harmful to individuals and society. Based on the research needs, a few datasets of Deepfake manipulation have been made available. In this paper, we propose a deep ensemble learning technique, DeepfakeStack, by experimenting with various DL-based models on the FF++ dataset. The experiment shows that a larger stacking ensemble neural network (called DFC) model is defined and fit on the test (unseen) dataset, then the new model is used to predict the test dataset. Evaluating the results, we see that the proposed DFC model achieves an accuracy of 99.65% and AUROC 1.0, outperforming the DL-based models, thereby provides a strong basis for developing an effective Deepfake detector.

In future work, the authors intend to use the proposed method, in conjunction with Blockchain technology, to build a Deepfake detection and prevention system.

References

[1] FaceApp, https://www.faceapp.com/, last accessed 2020/06/07.

[2] FakeApp, https://www.fakeapp.org/, last accessed 2020/06/07.

[3] G. Patrini, F. Cavalli, and H. Ajder, "The state of Deepfakes: reality under attack," Annual Report v.2.3, January 2019.

[4] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-Time Face Capture and Reenactment of RGB Videos," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, pp. 2387–2395, November 2016.

[5] S. Suwajanakorn, S. M. Seitz, and I. K. Shlizerman, "Synthesizing Obama: learning lip sync from audio," ACM Transactions on Graphics, 36(4), July 2017.

[6] Faceswap, https://github.com/MarekKowalski/FaceSwap/, last accessed: 2020/06/07.

[7] Exploring DeepFakes, https://goberoi.com/exploring-deepfakes-20c9947c22d9, last accessed: 2020/06/07.

[8] How deep learning fakes videos (Deepfake) and how to detect it?, https://medium.com/@jonathan_hui/how-deep-learning-fakes-videos-deepfakes-and-how-to-detect-it-c0b50fbf7cb9, last accessed: 2020/06/07

[9] The Power of Ensembles in Deep Learning, https://towardsdata-science.com/the-power-of-ensembles-in-deep-learning-a8900ff42be9, last accessed: 2020/06/07.

[10] D. Guera, and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, pp. 1–6, November 2018.

[11] Y. Li, and S. Lyu, "Exposing DeepFake Videos by Detecting Face Warping Artifacts," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 46–52, 2019.

[12] Y. Li, M. Chang, and S. Lyu, "In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking," 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, pp. 1–7, December 2018.

[13] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: a Compact Facial Video Forgery Detection Network," 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, pp. 1–7, December 2018.

[14] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-Stream Neural Networks for Tampered Face Detection," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, pp. 1831–1839, July 2017.

[15] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos," Workshop on Applications of Computer Vision and Pattern Recognition to Media Forensics with CVPR, pp. 80–87, 2019.

[16] N. T. Do, I. S. Na, and S. H. Kim, "DeepFakes: Forensics Face Detection from GANs Using Convolutional Neural Network," International Symposium on Information Technology Convergence (ISITC 2018), South Korea 2018.

[17] Q. Liu, A. H. Sung, et al. "Feature Mining and Pattern Recognition in Steganalysis and Digital Forensics," Pattern Recognition, Machine Intelligence and Biometrics (Editor Patrick S.P. Wang), High Education Press and Springer, pp. 561–604, December 2011.

[18] Q. Liu, P. Cooper, et. al. "Detection of JPEG Double Compression and Identification of Smartphone Image Source and Post-Capture Manipulation," Applied Intelligence, 39(4), pp. 705–726, 2013.

[19] D. Cozzolino, J. Thies, A. Rossler, R. Christian, M. Nießner, and L. Verdoliva, "ForensicTransfer: Weakly-supervised domain adaptation for forgery detection," arXiv:1812.02510, December 2018.

[20] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task Learning for Detecting and Segmenting Manipulated Facial Images and Videos," arXiv:1906.06876, June 2019.

[21] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using Capsule Networks to Detect Forged Images and Videos," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, United Kingdom, pp. 2307–2311.

[22] U. A. Ciftci, and I. Demir, "FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals," ArXiv: abs/1901.02212, January 2019.

[23] F. Matern, C. Riess, and M. Stamminger, "Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations," 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), Waikoloa Village, HI, USA, pp. 83–92, January 2019.

[24] M. Koopman, A. M. Rodriguez, and Z. Geradts, "Detection of Deepfake Video Manipulation," 20th Irish Machine Vision and Image Processing conference (IMVIP 2018), Northern Ireland, United Kingdom, August 2018.

[25] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to Detect Manipulated Facial Images," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, South Korea, pp. 1–11, October-November 2019.

[26] ImageNet, http://www.image-net.org/, last accessed: 2020/06/07.

[27] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," Proceedings of the 19th International Conference on Neural Information Processing Systems (NIPS'06), Cambridge, MA, USA, pp. 153–160, December 2006.