Deepfake Detection Using Machine Learning Algorithms

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

Beddhu Murali
School of Computing Sciences and Computer Engineering
The University of Southern Mississippi
Hattiesburg, MS 39406, USA
beddhu.murali@usm.edu

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

***Abstract*—Deepfake, a new video manipulation technique, has drawn much attention recently. Among the unlawful or nefarious applications, Deepfake has been used for spreading misinformation, fomenting political discord, smearing opponents, or even blackmailing. As the technology becomes more sophisticated and the apps for creating them ever more available, detecting Deepfake has become a challenging task, and accordingly researchers have proposed various deep learning (DL) methods for detection. Though the DL-based approach can achieve good solutions, this paper presents the results of our study indicating that traditional machine learning (ML) techniques alone can obtain superior performance in detecting Deepfake. The ML-based approach is based on the standard methods of feature development and feature selection, followed by training, tuning, and testing an ML classifier. The advantage of the ML approach is that it allows better understandability and interpretability of the model with reduced computational cost. We present results on several Deepfake datasets that are obtained relatively fast with comparable or superior performance to the state-of-the-art DL-based methods: 99.84% accuracy on FaceForecics++, 99.38% accuracy on DFDC, 99.66% accuracy on VDFD, and 99.43% on Celeb-DF datasets. Our results suggest that an effective system for detecting Deepfakes can be built using traditional ML methods.**

Keywords— Deepfake, Deep Learning, Machine Learning, Face Manipulation.

I. Introduction

Deepfake manipulation involves swapping an individual's face and facial expressions with others' and creating photorealistic fake videos called Deepfakes. Recent advances in video editing tools, such as FaceApp [1] or FakeApp [2], artificial intelligence (AI), and social networking, make it possible to produce Deepfake videos easily. These counterfeit videos can be used for various malicious purposes; the detection and countermeasures of Deepfakes have thus become a fundamental and challenging problem.

Deep learning (DL)-based approaches minimize the human effort in feature engineering; however, as much is done automatically, it increases the complexity of understanding and interpreting the model. Interpretability is a significant disadvantage of DL-based methods as the model is too difficult to understand with its high nonlinearity and interactions between inputs. Typically, a tradeoff between accuracy and interpretability occurs when analyzing any classical machine learning (ML) method [3]. DL methods are languid to train and require many computing resources, making them resource-intensive because of their complexity, a large number of layers, and large volumes of data. On the other hand, ML methods are easier to evaluate and understand.

The above context motivated us to, in this paper, experiment with and evaluate a classical ML method in detecting Deepfakes. The process and the results are described below:

- Create a unique set of features by combining HOG, Haralic, Hu Moments, and Color Histogram features.
- Lessen the data's complexity and making patterns recognizable by splitting a task into two stages: object detection and object recognition.
  - ➢ The object detection phase scans an entire image and identifies all possible objects.
  - ➢ The object recognition step identifies relevant objects.
- The advantages of the ML method:
  - ➢ Provide better understandability and interpretability of the model.
  - ➢ Reduce training time and achieve the same level of performance. Using FaceForecics++, DFDC, Celeb-DF, and VDFD datasets with the same amount of train and test samples, the ML methods take between several seconds and a couple of hours, while the most advanced DL algorithm, for instance, ResNet, takes nearly two weeks.

The rest of the paper is organized as follows: Sect. 2 gives a literature review; Sect. 3 describes our proposed method; Sect. 4 presents results and discussions; Sect. 5 draws conclusions and outlines probable future work.

II. Literature Review

Guera and Delp [4] proposed two end-to-end trainable recurrent systems: a CNN that extracts the most critical features and an LSTM for sequential analysis. In [5], four distinct CNNs were trained using real and fake faces. In [6], the authors introduced another method based on eye-blinking detection, in which authors believe that eye-blinking is disappeared in analyzing Deepfakes. Afchar et al. [7] proposed a CNN model called MesoNet to classify original and Deepfakes. Zhou et al. [8] proposed a two-stream network: (i) the CNN-based face classification network that captures tampering artifact evidence and (ii) the steganalysis feature-based triplet network that captures local noise residual evidence. In [9], researchers used computer-generated masks to crop and adjust the face areas to detect the manipulation. ForensicTransfer (FT) [10] can efficiently differentiate original images from forgery. In [11], to simultaneously detect and segment manipulated images and videos, the authors developed a multi-tasking learning method. Besides, researchers introduced a CNN model called capsule network [12] to detect replay attacks and computer-generated photos or videos. Ciftci and Demir [13] presented the Deepfake Detector called FakeCatcher (FC) for detecting synthetic portrait videos by exploiting biological signals, whereas, in [14], missing reflections and missing details in the eye and teeth areas are being used. In [15], the authors proposed a deep ensemble learning technique, DeepfakeStack, for detecting Deepfake. Together with the previously trained base-learners, the principle of DeepfakeStack is to train meta-learner, providing an interface for the meta-learner according to predictions of base learners.

III. Methodology

A. Data Preprocessing and Feature Selection

We used FaceForensics++ (FF++) [16], DFDC [17], Celeb-DF [18] and VDFD (our newly proposed) datasets to conduct the experiment and evaluate our proposed technique. We applied the following preprocessing and analysis techniques to reach the most excellent performances for ML models.

- We use a total of 400 videos with 200 real faces and 200 fake faces.
- From each video, we randomly select 200 frames to reduce the cost of computing.
- Track and extract each image's face and feed them into the classifiers using the '***dlib***' Python package.

In ML, feature extraction and feature selection are significant problems. Accurate or discriminating features are mandatory to enhance the model's performance because adverse computation and accuracy loss may result from irrelevant features. A more comprehensive training dataset will usually get reasonable classification accuracy as the number of features increases. For a fixed data set, accuracy increases to a point with a variety of features, and the accuracy of classification tends to decline as the number of features increases. In image processing and pattern recognition, feature extraction involves obtaining new features with higher-level info of an image, for example, color, shape, texture, etc. And feature selection reduces the input data dimension to find out the most appropriate features. Also, feature selection reduces the time complexity of the training process [19]. To achieve these goals, we used the following feature extraction and selection techniques: Haralic Texture Features (HTF) [20], Histogram of Oriented Gradients (HOG) [21-23], Color Histogram (CH) [24], and Hu Moments (HM) [25-26]. A feature set defines a single image after applying these feature engineering techniques using various combinations of these features. Finally, we concatenated the individually extracted features from each dataset to form a unique feature vector, namely, Deepfake Feature (DFF) and used it to feed ML classifiers. Table 1 shows various feature set by combining features extracted using these feature engineering or descriptors.

TABLE I. Contribution of individual feature engineering technique to the construction of DFF

Feature Set	***#Elements per Feature Engineering Techniques***			
	HMs	***HTF***	***CH***	***HOG***
DFF-109	7	13	8	81
DFF-117	7	13	16	81
DFF-133	7	13	32	81
DFF-228	7	13	64	144
DFF-717	7	13	256	441
DFF-2296	7	13	512	1764

B. Model Selection and Validation

To train classifiers, we used support vector machine (SVM), random forest (RF), extremely randomized trees (ERT), decision tree (DT), multilayer perceptron network (MLP), Stochastic gradient boosting (SGB), K-Nearest Neighbor (KNN).

An omnibus test with a null hypothesis ($\alpha = 0.05$) is a popular approach to validating performance by different classifiers. This is a statistical test that helps to verify that arbitrary experiments vary from a null hypothesis. The so-called Analysis of Variance (ANOVA) [27] is a typical example of an omnibus test used to find the importance of this null hypothesis. Our experiment conducts an omnibus test to assess ML classifiers' significance, using Cochran's Q-test and F-test.

IV. Results and Discussion

This section presents our results and compares the performance of ML-based methods, and Table 2 shows the results of the Q-test and the F-test, which lead to perform a null hypothesis test.

TABLE II. Q-test and F-test

Dataset	***Feature Set***	***Q-test***		***F-test***	
		Q-stat	***p-value***	***F-stat***	***p-value***
FF++	DFF-109	9329.233	0.000002	1722.1539	0.000003
	DFF-117	9703.819	0.000002	1799.0783	0.000003
	DFF-133	8509.299	0.000002	1556.0731	0.000003
	DFF-228	7273.790	0.000002	1311.6142	0.000002
	DFF-717	6771.821	0.000001	1214.2285	0.000002
	DFF-2296	8290.169	0.000002	1512.2134	0.000003
DFDC	DFF-109	1934.997	0.000002	337.8089	0.000001
	DFF-117	2026.625	0.000002	354.6049	0.000003
	DFF-133	2255.071	0.000001	396.8126	0.000001
	DFF-228	1751.854	0.000000	304.4638	0.000002
	DFF-717	1263.535	0.000003	217.0004	0.000001
	DFF-2296	1285.951	0.000002	220.9699	0.000001
Celeb-DF	DFF-109	7549.388	0.000003	1366.3963	0.000003
	DFF-117	7050.527	0.000002	1268.8916	0.000002
	DFF-133	6499.617	0.000002	1162.4864	0.000002
	DFF-228	7112.409	0.000003	1280.9267	0.000002
	DFF-717	5084.560	0.000003	895.1325	0.000001
	DFF-2296	5152.525	0.000002	907.7816	0.000001
VDFD	DFF-109	7628.235	0.000002	1381.6975	0.000003

Dataset	*Feature Set*	*Q-test*		*F-test*	
		Q-stat	*p-value*	*F-stat*	*p-value*
	DFF-117	7647.576	0.000002	1385.5058	0.000003
	DFF-133	7184.748	0.000002	1294.8315	0.000003
	DFF-228	7497.939	0.000002	1356.0855	0.000003
	DFF-717	5430.538	0.000001	959.62038	0.000001
	DFF-2296	6705.140	0.000001	1201.8674	0.000002

Based on the results shown in Table 2, the p-value is less than α (α = 0.05) that leads to rejecting the null hypothesis. Based on the nested cross-validation test, Table 3 shows the outcomes of classical ML-based classifiers.

TABLE III. PERFORMNACE OF CLASSICAL ML-BASED ALGORITHMS

Data set	*Feature*	*Model*	*Prec.*	*Rec.*	*F1-score*	*ACC*	*AUC*
FF++	DFF-109	SVM	94.60	93.99	94.29	94.22	94.23
		RF	**99.28**	**99.04**	**99.16**	**99.15**	**99.15**
		ERT	**99.64**	**99.46**	**99.55**	**99.54**	**99.55**
		DT	92.57	92.32	92.44	92.33	92.34
		SGB	81.41	78.78	80.08	80.10	80.12
		MLP	95.97	97.59	96.77	96.69	96.68
		KNN	98.42	98.82	98.62	98.59	98.59
	DFF-117	SVM	94.35	92.76	93.55	93.51	93.52
		RF	**99.44**	**99.27**	**99.36**	**99.35**	**99.35**
		ERT	**99.58**	**99.50**	**99.54**	**99.52**	**99.53**
		DT	93.51	93.68	93.60	93.47	93.49
		SGB	81.96	78.17	80.02	80.19	80.22
		MLP	98.34	96.97	97.65	97.63	97.64
		KNN	98.41	98.82	98.61	98.59	98.58
	DFF-133	SVM	93.02	91.20	92.10	92.06	92.07
		RF	**99.49**	**99.32**	**99.41**	**99.40**	**99.39**
		ERT	**99.65**	**99.53**	**99.59**	**99.60**	**99.59**
		DT	92.83	92.89	92.86	92.76	92.75
		SGB	83.66	80.91	82.27	82.29	82.31
		MLP	98.83	98.08	98.45	98.44	98.44
		KNN	98.54	98.99	98.77	98.74	98.73
	DFF-228	SVM	98.41	97.5	97.95	97.93	97.94
		RF	**99.79**	**99.53**	**99.66**	**99.67**	**99.66**
		ERT	**99.80**	**99.75**	**99.78**	**99.77**	**99.78**
		DT	93.22	93,68	93.45	93.34	93.33
		SGB	87.34	87.67	87.51	87.30	87.29
		MLP	98.99	99.88	99.43	99.42	99.41
		KNN	99.63	99.68	99.66	99.65	99.64
	DFF-717	SVM	97.86	97.30	97.58	97.55	97.55
		RF	**99.60**	**99.26**	**99.43**	**99.42**	**99.43**
		ERT	**99.52**	**99.32**	**99.42**	**99.41**	**99.41**
		DT	89.29	88.73	89.01	88.90	88.88
		SGB	87.46	86.75	87.10	86.97	86.96
		MLP	99.56	99.22	99.39	99.38	99.38
		KNN	96.75	97.43	97.09	97.04	97.02
	DFF-2296	**SVM**	**99.42**	**99.52**	**99.47**	**99.46**	**99.47**
		RF	**99.28**	**99.42**	**99.85**	**99.84**	**98.84**
		ERT	**99.15**	**98.92**	**99.03**	**99.02**	**99.04**
		DT	83.32	84.26	84.04	83.76	83.75
		SGB	87.60	90.36	88.96	88.61	88.58
		MLP	99.23	98.60	98.90	98.91	98.90
		KNN	95.92	96.23	96.08	96.01	96.00
DFDC	DFF-109	SVM	93.25	88.59	90.86	91.46	91.34
		RF	**99.08**	**97.94**	**98.51**	**98.58**	**98.55**
		ERT	**99.08**	**97.85**	**98.46**	**98.53**	**98.51**
		DT	93.11	92.59	92.85	93.16	93.14
		SGB	91.04	80.68	85.54	86.93	86.69
		MLP	95.79	97.00	96.39	96.53	96.54
		KNN	96.93	96.47	96.70	96.84	96.83
	DFF-117	SVM	93.27	87.65	90.37	91.05	90.91
		RF	**99.11**	**98.06**	**98.58**	**98.65**	**98.62**
		ERT	**99.17**	**98.44**	**98.80**	**98.86**	**98.84**
		DT	94.01	94.15	94.08	94.32	94.13
		SGB	91.29	81.74	86.25	87.51	87.28
		MLP	97.01	97.29	97.15	97.27	97.26
		KNN	96.88	96.68	96.78	96.91	96.90
	DFF-133	SVM	93.02	84.32	88.46	89.46	89.25
		RF	**99.08**	**98.32**	**98.70**	**98.76**	**98.74**
		ERT	**99.35**	**98.59**	**98.97**	**99.01**	**98.99**
		DT	94.35	94.29	94.32	94.56	94.55
		SGB	91.92	83.32	87.41	88.50	88.29
		MLP	**97.89**	**98.41**	**98.15**	**98.22**	**98.23**
		KNN	97.20	96.82	97.01	97.14	97.13
	DFF-228	SVM	97.02	93.91	95.44	95.70	95.63
		RF	**99.35**	**98.71**	**99.03**	**99.07**	**99.06**
		ERT	**99.44**	**99.26**	**99.35**	**99.38**	**99.37**
		DT	94.35	94.38	94.37	94.60	94.59
		SGB	93.97	85.68	89.63	90.50	90.31
		MLP	99.00	99.29	99.15	99.18	99.19
		KNN	98.64	98.12	98.38	98.45	98.44
	DFF-717	SVM	96.60	93.71	95.13	95.41	95.34
		RF	**98.99**	**97.76**	**98.37**	**98.45**	**98.42**
		ERT	**99.07**	**97.53**	**98.30**	**98.38**	**98.35**
		DT	94.57	93.79	94.18	94.45	94.42
		SGB	93.79	86.65	90.08	90.85	90.67
		MLP	**99.32**	**98.97**	**99.15**	**99.18**	**99.17**
		KNN	96.58	96.21	96.39	96.55	96.53
	DFF-2296	**SVM**	**99.15**	**99.38**	**99.27**	**99.30**	**99.29**
		RF	**99.20**	**99.00**	**99.10**	**99.14**	**99.02**
		ERT	**99.23**	**98.94**	**99.09**	**99.15**	**99.14**
		DT	94.63	94.88	94.76	94.97	95.26
		SGB	95.80	89.32	92.45	93.01	92.86
		MLP	99.44	99.76	99.60	99.62	99.19
		KNN	97.83	98.06	97.94	98.03	98.01
Celeb-DF	DFF-109	SVM	92.26	90.33	91.28	91.27	91.28
		RF	**98.95**	**98.19**	**98.57**	**98.56**	**98.56**
		ERT	**98.95**	**98.49**	**98.72**	**98.70**	**98.71**
		DT	92.18	91.53	91.86	91.80	91.79
		SGB	81.54	77.43	79.43	79.72	79.75
		MLP	95.35	94.91	95.13	95.08	95.08
		KNN	96.39	95.85	96.12	96.09	96.10
	DFF-117	SVM	91.79	88.82	90.28	90.33	90.35
		RF	**99.10**	**98.59**	**98.85**	**98.84**	**98.83**
		ERT	**99.07**	**98.52**	**98.80**	**98.78**	**98.80**
		DT	92.95	93.34	93.14	93.05	93.04
		SGB	83.92	79.24	81.51	81.42	81.85
		MLP	97.01	96.94	96.97	96.94	96.94
		KNN	96.46	95.53	96.19	96.17	96.16
	DFF-133	SVM	91.13	88.63	89.86	89.89	89.90
		RF	**99.19**	**98.74**	**98.96**	**98.95**	**98.96**
		ERT	**99.24**	**98.89**	**99.06**	**99.08**	**99.06**

Data set	*Feature*	*Model*	*Prec.*	*Rec.*	*F1-score*	*ACC*	*AUC*
		DT	93.35	92.85	93.10	93.04	93.05
		SGB	83.99	79.82	81.85	82.10	82.13
		MLP	91.75	99.17	95.32	95.07	95.02
		KNN	96.52	95.98	96.25	96.22	96.22
	DFF-228	SVM	96.27	95.56	95.91	95.88	95.89
		RF	**99.69**	**99.18**	**99.43**	**99.43**	**99.43**
		ERT	**99.56**	**99.18**	**99.37**	**99.36**	**99.37**
		DT	93.79	93.40	93.59	93.53	93.54
		SGB	86.44	83.00	84.68	84.81	84.84
		MLP	**99.23**	**99.27**	**99.25**	**99.24**	**99.23**
		KNN	97.85	97.47	97.66	97.64	97.64
	DFF-717	SVM	95.98	95.79	95.89	95.84	95.85
		RF	**99.06**	**98.56**	**98.81**	**98.82**	**98.80**
		ERT	**98.88**	**98.07**	**98.48**	**98.46**	**98.47**
		DT	92.74	92.10	92.42	92.36	92.35
		SGB	87.16	84.74	85.93	85.97	85.98
		MLP	**99.34**	**98.64**	**98.99**	**98.98**	**98.98**
		KNN	94.62	94.57	94.60	94.54	94.53
	DFF-2296	SVM	98.15	97.81	97.98	97.96	97.96
		RF	98.36	97.47	97.86	97.85	97.85
		ERT	97.91	97.29	97.60	97.58	97.58
		DT	89.18	89.14	89.16	89.07	89.04
		SGB	86.76	86.21	86.48	86.37	86.37
		MLP	98.85	98.69	98.77	98.78	98.76
		KNN	93.92	93.76	93.84	97.79	93.77
VDFD	DFF-109	SVM	93.63	94.13	93.88	93.87	93.87
		RF	98.91	98.65	98.78	98.79	98.79
		ERT	**99.22**	**99.23**	**99.23**	**99.23**	**99.22**
		DT	91.62	91.14	91.38	91.40	91.41
		SGB	80.29	84.80	82.48	82.01	82.16
		MLP	95.88	96.71	96.30	96.29	96.27
		KNN	98.27	98.70	98.49	98.51	98.49
	DFF-117	SVM	93.28	93.54	93.40	93.42	93.41
		RF	**99.05**	**98.82**	**98.94**	**98.94**	**98.93**
		ERT	**99.31**	**99.30**	**99.28**	**99.31**	**99.31**
		DT	92.71	91.87	92.29	92.33	92.34
		SGB	81.07	85.46	83.21	82.78	82.79
		MLP	98.59	96.63	97.60	97.63	97.62
		KNN	**98.42**	**98.83**	**98.62**	**98.63**	**98.62**
	DFF-133	SVM	92.33	92.03	92.18	92.20	92.21
		RF	**99.23**	**99.03**	**99.13**	**99.13**	**99.13**
		ERT	**99.40**	**99.38**	**99.37**	**99.38**	**99.39**
		DT	92.53	91.72	92.12	92.17	92.16
		SGB	81.89	85.92	83.85	83.48	83.49
		MLP	99.54	95.44	97.45	97.52	97.50
		KNN	98.39	98.73	98.56	98.58	98.56
	DFF-228	SVM	97.40	96.94	97.17	97.18	97.17
		RF	**99.36**	**99.33**	**99.35**	**99.36**	**99.35**
		ERT	**99.53**	**99.62**	**99.58**	**99.58**	**99.58**
		DT	93.95	92.98	93.46	93.51	93.50
		SGB	84.23	88.41	86.27	85.95	85.95
		MLP	**99.66**	**99.37**	**99.51**	**99.52**	**99.52**
		KNN	**99.18**	**99.21**	**99.19**	**99.20**	**99.20**
	DFF-717	SVM	98.66	98.40	98.53	98.54	98.53
		RF	99.21	98.55	98.88	98.89	98.89
		ERT	**99.32**	**99.27**	**99.29**	**99.31**	**99.30**
		DT	91.21	90.43	90.82	90.89	90.87
		SGB	87.89	90.76	89.30	89.16	89.14
		MLP	**99.72**	**99.57**	**99.66**	**99.65**	**99.64**
		KNN	97.00	97.21	97.10	97.12	97.10
	DFF-2296	**SVM**	**99.71**	**99.62**	**99.67**	**99.66**	**99.67**
		RF	99.16	97.55	98.34	98.36	98.35
		ERT	98.90	98.68	98.79	98.79	98.79
		DT	86.44	84.82	85.62	85.79	85.78
		SGB	91.60	93.12	92.35	92.32	92.30
		MLP	**99.46**	**99.63**	**99.55**	**99.55**	**99.54**
		KNN	95.96	96.29	96.12	96.13	96.12

Our experiment used four different datasets, including FF++, DFDC, Celeb-DF, and VDFD. In this paper, we conducted a series of experiments for six feature sets, for example, *DFF-109, DFF-117, DFF-133, DFF-228, DFF-717*, and *DFF-2296*. Based on the experiments using FF++, we see that RF and ERT achieved the best performances for all the feature sets and obtained more than 99% accuracy, while the SVM only achieved the same performance for the DFF-2296 feature set. MLP and KNN achieved 98% accuracy, where SGB and DT achieved about 95% accuracy. For the DFDC dataset, using the DFF-2296 feature set, SVM, RF, and ERT achieved 99% accuracy, where MLP achieved the best performance for all feature sets. Also, it is noticeable that the SVM decreased its performance in detecting Deepfake while decreasing the size of the feature set, for example, its accuracy decreased to 89% for DFF-133 and 91% for DFF-117. For the same feature sets, the SVM again produced the same level performance for Celeb-DF Dataset. In VDFD dataset-based experiments, it is seen that MLP obtained better performance than RF and ERT for any feature set and obtained more than 99% accuracy. Also, it was noticed that using VDFD, the KNN improved its performance for a certain feature set, for example, DFF-117 and DFF-228.

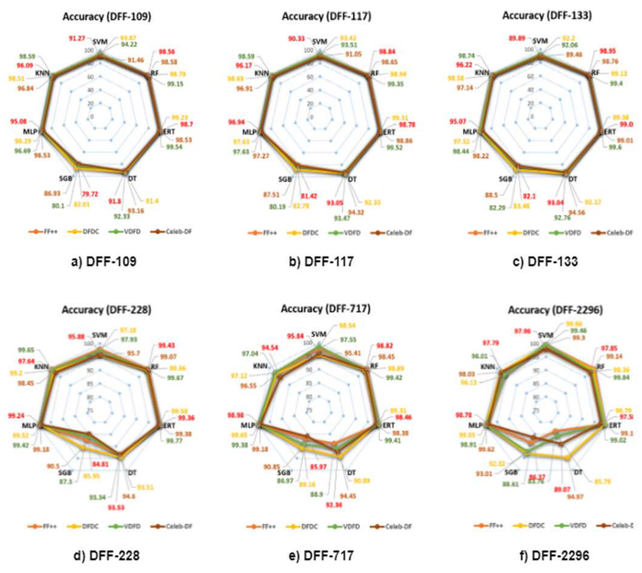
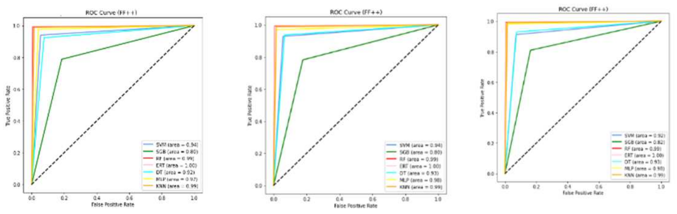


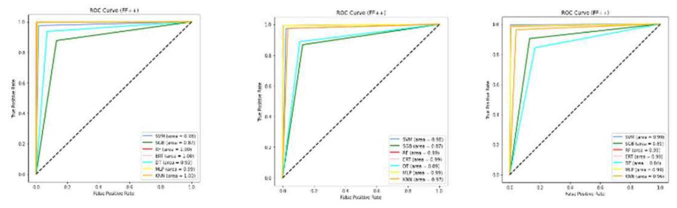
Fig. 1. Accuracy of ML classifiers using various feature sets.

Based on the overall experiments using FF++, VDFD, Celeb-DF, DFDC datasets, it is seen that RF, ERT, and MLP obtain state-of-the-art performances by achieving more than 99% accuracy. Besides, DFF-133 and DFF-228 provide the best accuracy. Based on the results presented above, the classical ML models now learned to detect the Deepfake videos.

In summary, each model generates a ROC curve that shows us how good the model is for classifying the original and the Deepfake classes, as seen in Figures 2-5. The curve fills the area between the colored line and the x-axis. A single model/classifier is specified as each color line. The bigger the size covered, the better the models are at classifying the given classes. In other words, the closer the AUC is to 1.0, the better. Based on the experiments, we can see that ERT achieves an AUC of 1.0 using any feature sets of the FF++ dataset, but using other datasets, it achieves 0.99. MLP obtains an AUC of 0.99 for any dataset with any feature set. Overall, ERT, RF bring an AUC of 0.99 for any dataset with any feature set. The AUC value varies for KNN and SVM for various feature sets. Based on the overall results, it is seen that the classical ML-based methods can separate the positive and negative data classes correctly.

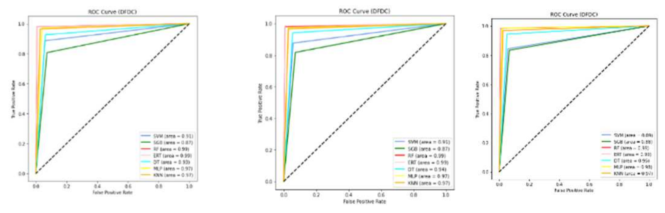


a) DFF-109 b) DFF-117 c) DFF-133

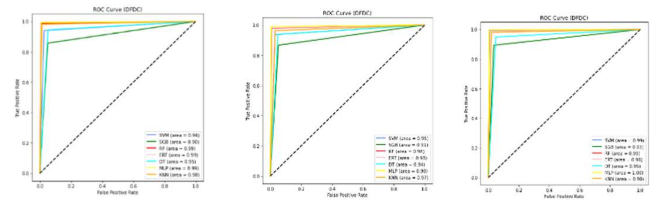


d) DFF-228 e) DFF-717 f) DFF-2296

Fig. 2. AUC of ML classifiers using all feature sets on the FF++ dataset.

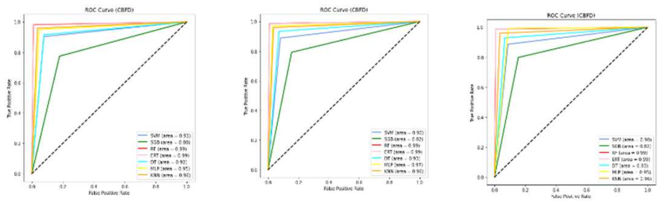


a) DFF-109 b) DFF-117 c) DFF-133

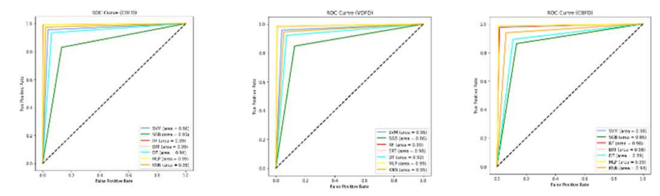


d) DFF-228 e) DFF-717 f) DFF-2296

Fig. 3. AUC of ML classifiers using all feature sets on the DFDC dataset.

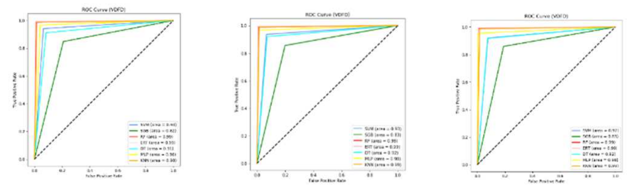


a) DFF-109 b) DFF-117 c) DFF-133

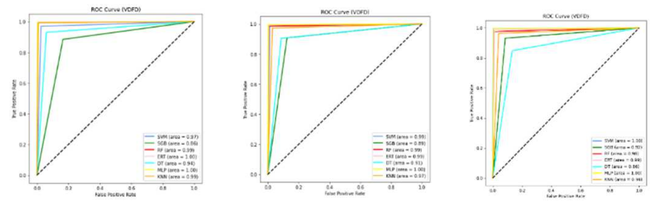


d) DFF-228 e) DFF-717 f) DFF-2296

Fig. 4. AUC of ML classifiers using all feature sets on Celeb-DF dataset.



a) DFF-109 b) DFF-117 c) DFF-133



d) DFF-228 e) DFF-717 f) DFF-2296

Fig. 5. AUC of ML classifiers using all feature sets on the VDFD dataset.

V. Conclusions and Future Works

While modern deep learning (DL) based methods have achieved highly accurate results in detecting Deepfakes, they carry with them disadvantages that include understandability, interpretability, data complexity, and high computational cost. The DL model is highly complicated and cannot easily be interpreted [3]. Further, they are precise to their context and, being very deep, tend to extract the underlying semantics from the images without having a single fingerprint [28]. In contrast, traditional machine learning (ML) methods allow better interpretability; for example, tree-based ML methods illustrate the detection process in the form of a decision tree. For an objective evaluation, therefore, we propose in this paper a classical ML-based method to detect Deepfakes, using the standard method of feature development, extraction, and classifier training and testing. The experimental outcomes prove that the ML techniques alone can obtain state-of-the-art performance in detecting Deepfakes. We believe that the proposed method can lay a basis for developing an effective solution for detecting Deepfakes and other facial manipulations. Our ongoing work includes further research on feature development and selection and ensemble detection for enhanced performance.

References

[1] FaceApp, https://www.faceapp.com/, last accessed: 2021/6/4.

[2] FakeApp, https://www.fakeapp.org/, last accessed: 2021/6/4.

[3] N. Koleva, "When and When Not to Use Deep Learning," https://bit.ly/3isoZdd, last accessed: 2021/6/4.

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

[5] 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.

[6] 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, Hong Kong, 2018, pp. 1–7.

[7] 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, 2018, pp. 1–7.

[8] 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, 2017, pp. 1831–1839.

[9] 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.

[10] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner and L. Verdoliva, "ForensicTransfer: Weakly-supervised domain adaptation for forgery detection," arXiv:1812.02510, 2018.

[11] 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, 2019.

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

[13] U. A. Ciftci, I. Demir and L. Yin, "FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals," IEEE Transactions on Pattern Analysis and Machine Intelligence.

[14] 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), 2019, pp. 83–92.

[15] M. S. Rana and A. H. Sung, "DeepfakeStack: A Deep Ensemble-based Learning Technique for Deepfake Detection," 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud), New York, USA, 2020, pp. 70–75.

[16] A. Rössler, 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, Korea (South), 2019, pp. 1–11.

[17] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The DeepFake Detection Challenge (DFDC) Dataset," arXiv: 2006.07397v4, 2020.

[18] Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, "Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 3204–3213, DOI: 10.1109/CVPR42600.2020.00327.

[19] M. N. Murti and V. S. Devi, "Feature Extraction and Feature Selection, Introduction to Pattern Recognition and Machine Learning," IISc Lecture Notes Series, June 2015, pp. 75–110.

[20] R. M. Haralick, "Statistical and structural approaches to texture," Proceedings of the IEEE, Vol. 67, No. 5, pp. 786–804, 1979.

[21] L. Weng, "Object Detection for Dummies Part 1: Gradient Vector, HOG, and SS", https://bit.ly/3oRFtxJ, last accessed: 2021/6/4.

[22] R. Ahmed, "A Take on HOG Feature Descriptor," https://bit.ly/2LutMyU, last accessed: 2021/6/4.

[23] A. Singh, "Feature Engineering for Images: A Valuable Introduction to the HOG Feature Descriptor," https://bit.ly/2LtjVt9, last accessed: 2021/6/4.

[24] F. Alamdar and M. R. Keyvanpour, "A New Color Feature Extraction Method Based on QuadHistogram," Procedia Environmental Sciences, Volume 10, 2011, pp. 777–783.

[25] J. Žunić, K. Hirota and P. L. Rosin, "A Hu moment invariant as a shape circularity measure, Pattern Recognition," Volume 43, Issue 1, 2010, pp. 47–57.

[26] Z. Huang and J. Leng, "Analysis of Hu's moment invariants on image scaling and rotation," 2010 2nd International Conference on Computer Engineering and Technology, Chengdu, 2010.

[27] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning," arXiv: 1811.12808, 2020.

[28] L. Guarnera, O. Giudice and S. Battiato, "Fighting Deepfake by Exposing the Convolutional Traces on Images," arXiv: 2008.04095v1, 2020.