Data Mining, SID306, SD-A1, 2023

Deepfake Audio Detection

Anisah Aunillah[a], Dhafina Nadhira[b], Salsabila Dwi S.[c], Salsabilla Alya P.I[d]
Ziyan Iffatun N.[e]

[a]*162112133030 Teknologi Sains Data, Fakultas Teknologi Maju dan Multidisiplin, Universitas Airlangga, Surabaya*
[b]*162112133033 Teknologi Sains Data, Fakultas Teknologi Maju dan Multidisiplin, Universitas Airlangga, Surabaya*
[c]*162112133044 Teknologi Sains Data, Fakultas Teknologi Maju dan Multidisiplin, Universitas Airlangga, Surabaya*
[d]*162112133048 Teknologi Sains Data, Fakultas Teknologi Maju dan Multidisiplin, Universitas Airlangga, Surabaya*
[e]*162112133098 Teknologi Sains Data, Fakultas Teknologi Maju dan Multidisiplin, Universitas Airlangga, Surabaya*

Abstrak
Pertumbuhan pesat alat yang menyerupai AI baru-baru ini telah menunjukkan kekuatan mereka dalam menghasilkan suara yang meyakinkan yang mengarah pada penyebaran disinformasi menggunakan audio di seluruh dunia. Pada studi kasus ini membahas perbandingan hasil dari beberapa metode pada *Machine Learning* dan *Deep Learning.* Studi kasus ini menggunakan algoritma diantaranya *Support Vector Machine* (SVM), *Decision Tree* (DT), *Random Forest, K-Nearest Neighbors (KNN),* dan *Multilayer Perceptron* (MLP). Dari hasil analisis yang didapatkan, model dengan algoritma *Support Vector Machine* memberikan metrik evaluasi tertinggi, sehingga dalam studi kasus ini lebih sesuai dengan menggunakan model tersebut. *Support Vector Machine* memiliki performa yang paling baik jika dibandingkan dengan empat model lainnya dengan akurasi mencapai 94%.
Kata Kunci : *Cross Validation*, audio mining, image mining, text mining

1. Pendahuluan

Pertumbuhan pesat alat yang menyerupai AI baru-baru ini telah menunjukkan kekuatan mereka dalam menghasilkan suara yang meyakinkan [1], yang mengarah pada penyebaran disinformasi menggunakan audio di seluruh dunia [2]. Teknologi baru yang digunakan untuk menirukan suatu media sedang berkembang pesat dan semakin mudah diakses yang memungkinkan pengguna untuk membuat klip video dan audio yang meyakinkan tentang individu yang melakukan dan mengatakan hal-hal yang tidak pernah mereka lakukan atau katakan. Pengguna dapat melakukan hal tersebut, misalnya, menyintesis suara individu tertentu berdasarkan transkrip, menukar wajah seseorang ke tubuh orang lain dalam video, atau mensintesis video baru sepenuhnya dari seseorang yang berbicara berdasarkan audio yang disinkronkan dengan bibir mereka [2].

Hal ini memiliki dampak yang positif pada kehidupan. Contohnya, membantu orang yang telah kehilangan suara mereka karena penyakit tenggorokan atau masalah medis lainnya dan mensimulasikan suara yang menenangkan di buku audio [1]. Meskipun awalnya ditujukan untuk meningkatkan kehidupan masyarakat, namun ada yang memanfaatkannya untuk hal negatif, sehingga membahayakan keselamatan publik. Teknologi ini tidak hanya digunakan untuk menargetkan individu, tetapi juga bank dan perusahaan. Oleh karena itu, metode *Machine Learning* (ML) dan *Deep Learning* (DL) telah dikembangkan untuk mendeteksi suara-suara yang ditiru atau dipalsukan secara sintetis.

Untuk mendeteksi *Audio Deepfake*, banyak metode *Machine Learning* dan *Deep Learning* telah dipublikasikan dalam literatur, dan model-model baru terus dikembangkan [1]. Metode-metode ini meliputi *Machine Learning* seperti *Support Vector Machine* (SVM) yang mana membagi data menjadi kelas-kelas yang berbeda berdasarkan fitur-fiturnya. Sedangkan *Deep Learning* contohnya seperti *Convolutional Neural Network* (CNN) yang terinspirasi oleh struktur visual korteks serebral. Dengan demikian, projek ini dilakukan untuk membandingkan hasil dari beberapa metode pada *Machine Learning* dan *Deep Learning*.

2. Landasan Teori

Audio mining adalah sebuah teknik dalam menganalisis konten sinyal audio secara otomatis yang umum digunakan pada bidang pengenalan suara otomatis. Pengenalan konten sinyal audio memerlukan fitur ekstraksi baik berbasis waktu seperti *energy*, *Zero Crossing Rate* (ZCR), dan *entropy of energy* ataupun berbasis frekuensi seperti *spectral centroid and spead*, *spectral entropy*, *spectral rolloff*, MFCC, dan *chroma vector* [3]. Sedangkan *deepfakes* adalah kombinasi dari kata "*deep learning*" dan "*fake*". *Deepfake audio*, secara ilmiah dikenal sebagai metode *spoofing* audio akses-logis, telah menjadi tantangan yang semakin besar karena terobosan cepat dalam teknologi pengenalan suara dan transfer suara [4]. Dengan berkembangnya bentuk-bentuk baru teknologi sintesis

pidato dan konversi suara, potensi untuk menggeneralisasi tindakan penanggulangan menjadi tantangan yang semakin kritis [4].

Deepfake terjadi ketika teknologi kecerdasan buatan (*Artificial Intelligence*/AI) membuat gambar dan suara palsu yang tampak nyata. *Deepfake* memungkinkan unruk memanipulasi gambar atau video dari seseorang untuk menggambarkan beberapa aktivitas yang sebenarnya tidak terjadi. Ada banyak spekulasi bahwa *deepfakes* mungkin pada akhirnya akan muncul sebagai ancaman keamanan siber utama yang digunakan untuk maksud jahat (W. Purbo, n.d.). *Deepfake* menghadirkan prosedur otomatis untuk menciptakan informasi palsu yang lebih sulit dideteksi oleh analis manusia [4].

Machine learning merupakan cabang ilmu bagian dari kecerdasan buatan (*artificial intelligence*), dengan pemrograman untuk memungkinkan komputer menjadi cerdas berperilaku seperti manusia dan dapat meningkatkan pemahamannya melalui pengalaman secara otomatis (P.D. Kusuma dalam Retnoningsih & Pramudita, n.d.). *Machine learning* memiiki fokus pada pengembangan sistem yang mampu belajar sendiri untuk memutuskan sesuatu tanpa harus berulang kali diprogram oleh manusia. Hal tersebut menjadikan mesin tidak hanya berperilaku mengambil keputusan, tetapi juga dapat beradaptasi terhadap perubahan yang terjadi. *Machine learning* bekerja apabila tersedia data sebagai *input* untuk dilakukan analisis terhadap kumpulan data besar (*big data*) sehingga dapat menemukan pola tertentu. Di dalam *machine learning* terdapat *data training* dan *data testing*, data training untuk melatih algoritma dalam *machine learning*, sedangkan *data testing* untuk mengetahui performa dari algoritma yang telah dilatih yaitu ketika menemukan data baru yang belum pernah diberikan dalam *data training* [5].

Support Vector Machine (SVM), *Logistic Regression* (LR), *Multilayer Perceptron* (MLP), *Adaptive Boosting* (AdaBoost), *eXtreme Gradient Boosting* (XGBoost), *K-Means clustering* (k-MN), *Random Forest* (RF), *Decision Tree* (DT), *Discriminant Analysis* (DA), *Naive Bayes* (NB), dan *Multiple Instance Learning* (MIL) digunakan sebagai model berbasis machine learning [6]. Berikut merupakan hasil dari perbandingan *output* pada SVM, MLP, DT, LR, NB, dan XGB [3]:

Tabel 1. Perbandingan Hasil Akurasi dari Setiap Dataset

Models	***for-2sec***	***for-norm***	***for-rerec***
SVM	97,57	71,54	98,83
MLP	94,69	86,82	98,79
DT	87,13	62,16	88,28
LR	89,92	82,80	88,31
XGB	94,52	92,60	93,40
NB	88,20	81,80	81,91

Dapat dilihat bahwa SVM menghasilkan akurasi tertinggi sebesar 97,57% pada dataset *For-2sec* dan 98,83% pada dataset *For-rerec*. Salah satu algoritma machine learning supervised yang paling kuat dan banyak digunakan untuk klasifikasi data adalah *Support Vector Machines* (SVM) [4]. Tujuan utama SVM adalah untuk menghasilkan *hyperplane marginal* maksimum dalam ruang multidimensi untuk memisahkan kelas-kelas yang berbeda. *Hyperplane* adalah sebuah bidang datar yang memisahkan dua kelas data dalam ruang multidimensi. Maka, *hyperplane* marginal adalah *hyperplane* dengan jarak terjauh dari titik data terdekat di setiap kelas. Kesalahan dapat diminimalkan dengan menghasilkan hyperplane secara berulang.

Pada penelitian ini algoritma *machine learning* yang akan digunakan sebagai perbandingan adalah *Random Forest*, SVM, *K-Nearest Neighbors* (KNN), dan *Decision Tree*. *Random forest* didasarkan pada teknik pohon keputusan sehingga mampu mengatasi masalah non-linier. Metode ini merupakan metode pohon gabungan. Untuk mengidentifikasi peubah penjelas yang relevan dengan peubah respon, *random forest* menghasilkan ukuran tingkat kepentingan (*variable importance*) peubah penjelas (Dewi & Syafitri, 2011).

Selain *machine learning*, penelitian ini juga akan menggunakan algoritma *deep learning*. *Deep Learning* (DL) adalah teknik dalam NN yang menggunakan teknik tertentu seperti *Restricted Boltzmann Machine* (RBM) untuk mempercepat proses pembelajaran dalam NN yang menggunakan lapis yang banyak atau lebih dari 7 lapis [7]. Algoritma *deep learning* dapat mengambil gambar atau spektrogram dan menetapkan bobot yang berbeda untuk setiap gambar untuk membedakannya dan melakukan tugas apa pun, seperti klasifikasi gambar.

3. Sumber Data dan Metodologi

3.1 Sumber Data

Penelitian ini menggunakan data sekunder tanpa label yang berupa audio *podcast* yang didapatkan dari berbagai channel di aplikasi *Spotify* dan *Youtube*. Rata-rata durasi audio *podcast* yang digunakan adalah 30 detik dengan *bitrate* 1411 kbps. *Bitrate* ini menunjukkan bahwa dalam setiap detiknya, ada sekitar 1411 kilobit data yang direpresentasikan. Pilihan *bitrate* yang tinggi seperti ini umumnya ditemukan dalam format audio tanpa kompresi, seperti file WAV yang digunakan pada data peneltian ini. Berikut variabel yang digunakan pada penelitian ini.

Tabel 2. Variabel Penelitian

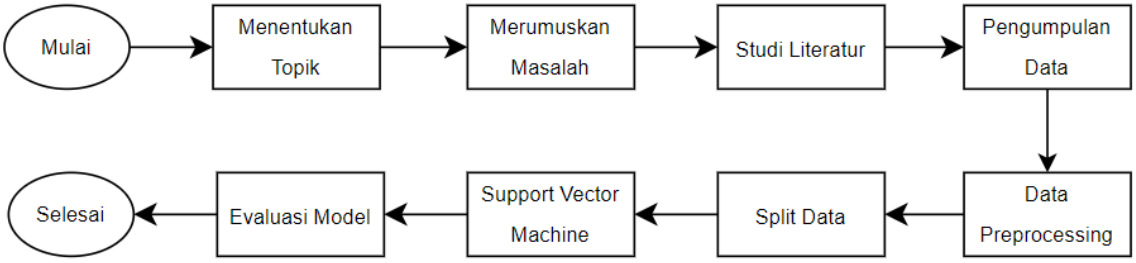
	chroma_ stft	**rms**	**spectral_ centroid**	**spectral_ bandwith**	**rolloff**	**zero_ cross**	**mfcc_1**	**…**	**mfcc_2**	**label**
0	0,324	0,022	2084,217	1218,087	3153,674	0,137	-491,607	…	-8,675	Real

1	0,327	0,023	1951,786	1441,157	3205,860	0,109	-488,438	…	-8,757	Real
2	0,337	0,029	1806,766	1376,169	2813,195	0,108	-458,505	…	-4,337	Real
…	…	…	…	…	…	…	…	…	…	…
299	0,381	0,026	2730,036	1147,013	3617,695	0,156	-486,637	…	-1,721	Fake
300	0,359	0,030	1926,038	1041,721	2764,667	0,136	-450,453	…	-1,626	Fake

Data audio yang digunakan terdiri dari 26 variabel X dan 1 variabel Y. Variabel X merupakan fitur audio yang digunakan untuk membedakan antara audio asli dan palsu. Sedangkan variabel Y merupakan label audio apakah asli atau palsu. Total sampel yang digunakan sebanyak 301 data yang terdiri dari 150 data audio dengan label "Real" dan 151 data audio dengan label "Fake".

3.2 Metodologi

Berikut merupakan skema analisis yang dilakukan pada penelitian ini.



Gambar 1. Langkah-Langkah Penelitian

Berikut tahapan analisis yang dilakukan berdasarkan skema di atas.

1. Pengumpulan data

   Data yang diperlukan untuk penelitian ini adalah data audio asli dan data audio palsu. Data audio asli dapat diperoleh dari berbagai sumber, seperti audio *podcast* dari *Spotify* dan *Youtube*. Sedangkan data audio palsu diperoleh dari audio asli yang dikonversi menjadi suara AI melalui salah satu situs web. Durasi satu data audio yang digunakan, baik audio asli maupun audio palsu adalah 30 detik dengan *bitrate* 1141 kbps.

2. *Data Preprocessing*

   Tahapan ini dilakukan agar proses *mining* berjalan lebih efektif dan efisien. Pada tahapan ini, diharapkan dapat memberikan informasi mengenai fitur-fitur data yang dapat digunakan dan membedakan antara audio asli dan audio palsu. Terdapat beberapa proses yang dilakukan pada tahapan ini, diantaranya :

   a. *Resampling* yang merupakan proses mengubah *sample rate* dari sinyal audio. Pada penelitian ini, *resampling* dilakukan dengan mengurangi *sampel rate* yaitu dari 44100 Hz menjadi 22050 hZ.
   b. Melakukan normalisasi audio, tujuannya untuk memastikan bahwa amplitude audio berada dalam rentang yang sama.
   c. Menghapus *noise*.
   d. Melakukan *feature extraction* menggunakan metode *Mel Frequency Cepstrum Coefficients* (MFCC) yang banyak digunakan untuk pengenalan suara.
   e. *Data cleaning*, meliputi pengecekan *missing value* dan penghapusan *outlier*.
   f. *Data transformation*, meliputi *feature encoding* untuk variabel label, untuk kategori Real ditandai dengan angka 1 dan kategori Fake ditandai angka 0. Kemudian, melakukan normalisasi menggunakan StandardScaler untuk memastikan bahwa semua variabel memiliki skala yang sama.

3. *Split Data*

   Split data adalah proses membagi data menjadi dua bagian, yaitu *data training* dan *data testing*. *Data training* digunakan untuk melatih model, sedangkan *data testing* digunakan untuk mengevaluasi kinerja model. Ukuran *data training* dan *data testing* harus proposional, dengan rasio yang umum digunakan adalah 80% untuk *data training* dan 20% untuk *data testing*. Jika ukuran *data testing* terlalu kecil, maka model akan sulit untuk menggeneralisasi kinerjanya ke data baru yang belum pernah dilihat sebelumnya. Pembagian dataset ini bertujuan untuk menghindari *overfitting*. Pada penelitian ini, pembagian dataset menggunakan rasio 80:20 di mana 80% untuk *data train* dan 20% untuk *data test*.

4. *Deepfake Detection Model (SVM)*

   Support Vector Machine (SVM) adalah metode *supervised learning* yang memiliki dua asumsi utama. Asumsi pertama adalah mengubah data ke dalam ruang berdimensi tinggi yang dapat mengurangi masalah klasifikasi kompleks dengan keputusan yang kompleks menjadi masalah yang lebih sederhana dengan pemisahan secara linier. Asumsi kedua adalah hanya pola pelatihan yang memberikan keputusan detail untuk klasifikasi [8]. SVM dipilih karena memiliki beberapa kelebihan, yaitu efektif dalam data berdimensi tinggi, efektif dalam kasus dengan jumlah dimensi yang lebih banyak daripada jumlah

sampelnya, dan menggunakan subset *data train* dalam mengambil keputusan yang efisien. SVM dapat bekerja pada data non-linear dengan pendekatan kernel. Pendekatan kernel bertujuan untuk memetakan dimensi awal kumpulan data, yaitu dimensi yang lebih rendah ke dimensi yang relatif tinggi [9].

Fungsi kernel yang digunakan pada penelitian ini adalah Kernel *Radial Basic Function* (RBF) dengan parameter Cost dan Gamma. Fungsi kernel RBF didapatkan melalui persamaan berikut.

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (1)$$

5. Evaluasi Model

Evaluasi dilakukan untuk memilih pemodelan terbaik yang dilihat melalui ukuran klasifikasi. Ukuran kinerja klasifikasi pada penelitian ini menggunakan *confussion matrix*, yang merupakan tabel yang menampilkan hasil prediksi model dengan hasil actual. Confussion matrix dapat digunakan untuk menganalisis seberapa baik atau akurat sebuah model dapat mengenali objek pengamatan dari kelas yang berbeda.

Tabel 3. Confussion Matrix

		Prediksi	
		Positive	**Negative**
Aktual	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Dalam penelitian ini, indikator yang digunakan untuk evaluasi adalah *precision*, *recall*, *f1-score*, dan *accuracy*. Keempat indikator tersebut didapatkan dari persamaan berikut.

$$precision = \frac{TP}{TP+FP} \quad (2)$$

$$recall = \frac{TP}{TP+FN} \quad (3)$$

$$f1\ score = 2 \times \frac{recall \times precision}{recall + precison} \quad (4)$$

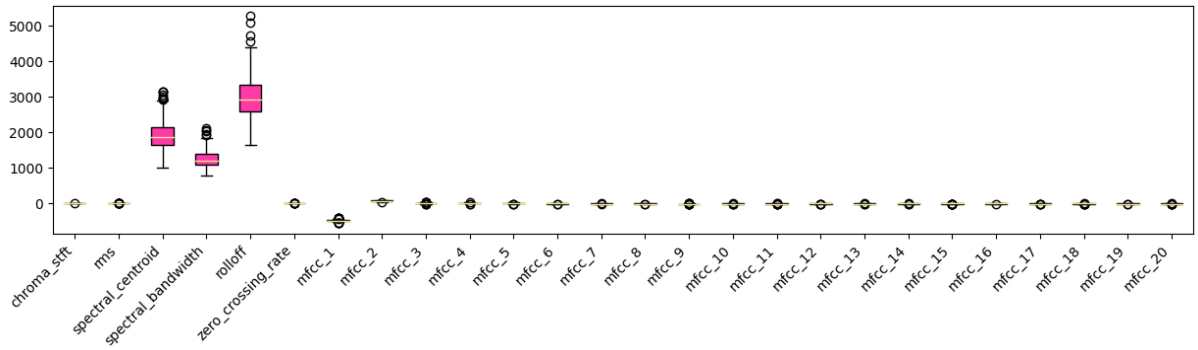
$$accuracy = \frac{TP+TN}{TP+FN+FP+FN} \quad (5)$$

Pada penelitian ini, hasil untuk evaluasi matriks didapatkan dengan menggunakan fungsi *accuracy_score* dan *classification_report* dari *library* sklearn,metrics.

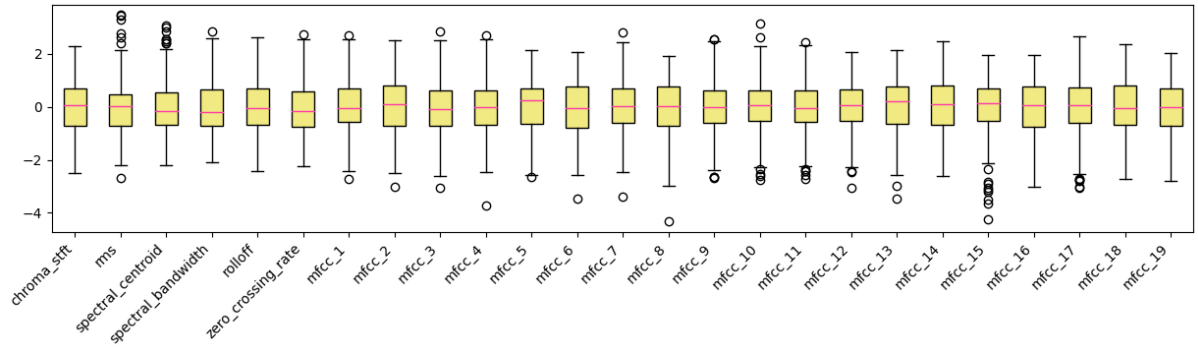
4. Analisis dan Pembahasan

4.1 Visualisasi Data

Data prerprocessing dilakukan untuk mempersiapkan data mentah agar siap dianalisis. Proses ini dilakukan untuk mengatasi berbagai masalah yang dapat mengganggu proses analisis data, seperti data yang memiliki distribusi tidak simetris atau memiliki *outlier* dan data yang memiliki format atau skala yang berbeda. Data mentah yang digunakan pada penelitian ini dapat dilihat pada Gambar 2 berikut.

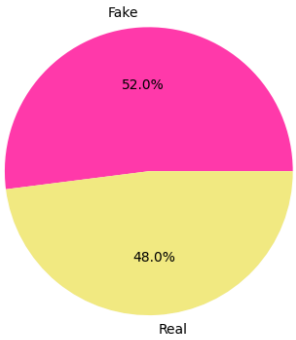


Gambar 2. Boxplot Sebelum Preprocessing



Gambar 3. Boxplot Setelah Preprocessing

Dari Gambar 3 di atas, terlihat bahwa terdapat outlier pada semua variabel numerik dan juga terlihat masing-masing variabel memiliki rentang data yang berbeda. Untuk itu, pada tahap preprocessing akan dilakukan penghapusan outlier pada masing-masing variabel dan juga normalisasi menggunakan StandardScaler untuk mengubah rentang data. Hasil dari tahapan preprocessing dapat dilihat pada Gambar 2, dari visualisasi boxplot tersebut dapat dilihat data sudah berada pada rentang yang sama walaupun masih terdapat outlier.



Gambar 4. Perbandingan Jumlah Data Fake dan Real

Dari Gambar 4 di atas, diketahui jumlah data yang akan digunakan pada tahap analisis berikutnya sejumlah 250 data yang terdiri dari audio berlabel Real sejumlah 120 dan audio berlabel Fake sejumlah 130 dengan persentase jumlah masing masing adalah 48% dan 52%

4.2 Model

4.2.1 Random Forest

Random forest adalah suatu jenis algoritma *machine learning* yang termasuk dalam kategori *ensemble learning*. *Ensemble learning* mengacu pada konsep menggabungkan beberapa model untuk meningkatkan kinerja dan kestabilan prediksi secara keseluruan. Model ini dipilih pada analisis kali ini karena kemampuan generalisasi yang baik sehingga diharapkan dapat mendeteksi deepfake suara di mana terdapat variasi data sangat kompleks. *Random Forest* juga mampu menangani data dengan fitur banyak, *Random Forest* dapat menangani keragaman data tanpa terlalu rentan terhadap *overfitting*. Pada data yang kami gunakan, terdapat 26 fitur sehingga model ini cocok untuk digunakan dalam analisis kali ini.

Tabel 4. Parameter Model Random Forest

Max Depth	Max Features	Min Sample Split	n_estimators
6	auto	4	38

Dalam membuat model dengan menggunakan *Random Forest*, kami melakukan *tuning* parameter untuk menentukan nilai optimal dari setiap parameter yang digunakan sehingga dapat menghasilkan model dengan hasil yang bagus. Pada tahap tuning parameter kami menggunakan metode *GridSearch Cross Validation*. Metode ini akan mencoba semua kombinasi parameter yang diberikan dan didapatkan parameter terbaik yang akan digunakan untuk membangun model. Pada proses *tuning* parameter ini kami menggunakan kombinasi parameter dengan rentang yang beragam, kemudian kami *trial and error* untuk mendapatkan hasil yang maksimal dan didapatkan untuk kedalaman setiap pohon (**max_depth**) sebesar 6, jumlah fitur yang harus dipertimbangkan untuk melakukan *split* (***max_features***) menggunakan *default*nya yaitu *auto*, jumlah sampel minimum yang diperlukan untuk membagi simpul *internal* (**min_sample_split**) sebesar 4, dan jumlah pohon keputusan yang akan dibuat dalam *ensemble* (***n_estimator***) sebesar 38.

Dalam membangun model ini, *data train* dimasukkan ke dalam model *Random Forest* sesuai dengan parameter yang telah didapatkan, kemudian dilakukan perbandingan antara y_predict dan y_test untuk mengetahui apakah suatu audio yang dikategorikan real benar-benar berlabel Real atau justru Fake, begitu juga sebaliknya. Dari hasil analisis, didapatkan beberapa hasil prediksi label suatu audio yang tidak sesuai dalam bentuk *confussion matrix*.

Tabel 5. Confussion Matrix Random Forest

		Prediksi	
		Real	**Fake**
Aktual	**Real**	23	3
	Fake	1	23

Berikut penjelasan *confussion matrix* pada Tabel 5 di atas.

- Jumlah data audio yang diklasifikasikan dengan benar sebagai audio Real (TP) sebanyak 23 audio.
- Jumlah data audio yang seharusnya diklasifikasikan sebagai audio Real, tetapi salah diklasifikasikan sebagai audio Fake (FP) sebanyak 1 audio.
- Jumlah data audio yang seharusnya diklasifikasikan sebagai audio Fake, tetapi salah diklasifikasikan sebagai audio Real (FN) sebanyak 3 audio.
- Jumlah data audio yang diklasifikan dengan benar sebagai audio Fake (TN) sebanyak 23 audio.

Tabel 6. Evaluasi Model Random Forest

Precision	**Recall**	**f1-score**	**Accuracy**
0,92	0,92	0,92	0,92

Dari Tabel 6 didapatkan hasil untuk *precision* yang digunakan untuk mengukur sejauh mana prediksi positif model yang benar dan didapatkan sebesar 92%, *recall* yang digunakan untuk mengukur sejauh mana model mampu mendeteksi semua *instance positif* yang sebenarnya dan meminimalkan *false negative* dan didapatkan sebesar 92%, *f1-score* yang digunakan untuk membetikan keseimbangan antara *precision* dan *recall* dan didapatkan sebesar 92%, dan *accuracy* model yang digunakan untuk mengukur sejauh mana model mampu memprediksi dengan benar dari seluruh prediksi yang dibuat dan didapatkan sebesar 92%.

4.2.2 Support Vector Machine (SVM)

Support Vector Machine adalah algoritma pembelajaran mesin yang bekerja dengan mencari *hyperplane* pemisah optimal antara dua kelas dalam ruang fitur. *Hyperplane* ini dipilih untuk memiliki margin maksimum, yaitu jarak terdekat antara titik-titik data dari kedua kelas ke *hyperplane*. Algoritma SVM dipilih pada analisis ini karena keberhasilan SVM dalam tugas klasifikasi berasal dari fokusnya menemukan *hyperplane* yang memaksimalkan *margin* antara kelas, yang dapat meningkatkan kemampuan model untuk menggeneralisasi pada data yang belum pernah dilihat sebelumnya. SVM juga dapat bekerja efektif pada data dengan jumlah fitur yang besar.

Tabel 7. Parameter Model SVM

C	***Gamma***	***Kernel***
0.1	0,01	rbf

Dalam membangun model ini, *data train* dimasukkan ke dalam model *Support Vector Machine* (SVM), kemudian dilakukan perbandingan antara y_predict dan y_test untuk mengetahui apakah suatu audio yang dikategorikan Real benar-benar berlabel Real atau justru Fake, begitu juga sebaliknya. Dari hasil analisis, didapatkan beberapa hasil prediksi label suatu audio yang tidak sesuai dalam bentuk *confussion matrix*.

Tabel 8. Confussion *Matrix SVM*

		Prediksi	
		Real	***Fake***
Aktual	***Real***	23	3
	Fake	0	24

Berikut penjelasan *confussion matrix* pada Tabel 8 di atas.

- Jumlah data audio yang diklasifikasikan dengan benar sebagai audio Real (TP) sebanyak 24 audio.
- Tidak ada data audio yang seharusnya diklasifikasikan sebagai audio Real, tetapi salah diklasifikasikan sebagai audio Fake (FP).
- Jumlah data audio yang seharusnya diklasifikasikan sebagai audio Fake, tetapi salah diklasifikasikan sebagai audio Real (FN) sebanyak 2 audio.

- Jumlah data audio yang diklasifikan dengan benar sebagai audio Fake (TN) sebanyak 24 audio.

Tabel 9. Evaluasi Model SVM

Precision	Recall	f1-score	Accuracy
0,95	0,94	0,94	0,94

Dari Tabel 9 didapatkan hasil untuk *precision* yang digunakan untuk mengukur sejauh mana prediksi positif model yang benar dan didapatkan sebesar 95%, *recall* yang digunakan untuk mengukur sejauh mana model mampu mendeteksi semua *instance positif* yang sebenarnya dan meminimalkan *false negative* dan didapatkan sebesar 94%, *f1-score* yang digunakan untuk membetikan keseimbangan antara *precision* dan *recall* dan didapatkan sebesar 94%, dan *accuracy* model yang digunakan untuk mengukur sejauh mana model mampu memprediksi dengan benar dari seluruh prediksi yang dibuat dan didapatkan sebesar 94%.

4.2.3 K-NN

K-Nearest Neighbors (KNN) adalah algoritma *machine learning* yang bekerja dengan cara menyimpan seluruh data train di memori dan memilih k tetangga terdekat dari titik data yang akan diprediksi. Prediksi kemudian dilakukan dengan mayoritas kelas dari tetangga terdekat. Algoritma K-NN dipilih pada analisis ini karena K-NN dapat bekerja dengan baik pada dataset dengan dimensi tinggi, dan dapat beradaptasi dengan variasi pola yang tidak pasti.

Tabel 10. Parameter K-NN

Metric	*N_neighbors*	p
euclidean	6	1

Dalam membangun model ini, *data train* dimasukkan ke dalam model K-NN, kemudian dilakukan perbandingan antara y_predict dan y_test untuk mengetahui apakah suatu audio yang dikategorikan Real benar-benar berlabel Real atau justru Fake, begitu juga sebaliknya. Dari hasil analisis, didapatkan beberapa hasil prediksi label suatu audio yang tidak sesuai dalam bentuk confussion matrix.

Tabel 11. Confussion Matrix K-NN

		Prediksi	
		Real	***Fake***
Aktual	***Real***	21	5
	Fake	0	24

Berikut penjelasan *confussion matrix* pada Tabel 11 di atas.

- Jumlah data audio yang diklasifikasikan dengan benar sebagai audio Real (TP) sebanyak 21 audio.
- Tidak ada data audio yang seharusnya diklasifikasikan sebagai audio Real, tetapi salah diklasifikasikan sebagai audio Fake (FP).
- Jumlah data audio yang seharusnya diklasifikasikan sebagai audio Fake, tetapi salah diklasifikasikan sebagai audio Real (FN) sebanyak 5 audio.
- Jumlah data audio yang diklasifikan dengan benar sebagai audio Fake (TN) sebanyak 24 audio.

Tabel 12. Evaluasi Model K-NN

Precision	*Recall*	*f1-score*	*Accuracy*
0,92	0,90	0,90	0,90

Dari Tabel 12 didapatkan hasil untuk *precision* yang digunakan untuk mengukur sejauh mana prediksi positif model yang benar dan didapatkan sebesar 92%, *recall* yang digunakan untuk mengukur sejauh mana model mampu mendeteksi semua *instance positif* yang sebenarnya dan meminimalkan *false negative* dan didapatkan sebesar 90%, *f1-score* yang digunakan untuk membetikan keseimbangan antara *precision* dan *recall* dan didapatkan sebesar 90%, dan *accuracy* model yang digunakan untuk mengukur sejauh mana model mampu memprediksi dengan benar dari seluruh prediksi yang dibuat dan didapatkan sebesar 90%.

4.2.4 Decision Tree

Decision tree adalah model prediktif yang mewakili struktur berhingga dari keputusan berdasarkan fitur-fitur audio. Model ini memecah data audio menjadi serangkaian keputusan berhierarki berdasarkan fitur-fitur seperti frekuensi, amplitudo, atau karakteristik lainnya. Setiap simpul dalam pohon keputusan mewakili suatu keputusan atau pemisahan berdasarkan suatu fitur, yang memungkinkan model ini untuk secara efektif mengidentifikasi pola atau tren dalam data audio.Algoritma decision tree dipilih pada analisis ini karena

mampu menangani fitur kompleks, memerlukan jumlah data pelatihan yang relatif kecil, dan memberikan hasil yang interpretabil.

Tabel 13. Parameter *Decision Tree*

ccp_alpha	*criterion*	*max_depth*	*max_features*	*min_samples_leaf*	*min_sample_split*
0,02	entropy	10	auto	6	5

Dalam membangun model ini, *data train* dimasukkan ke dalam model *Decision Tree*, kemudian dilakukan perbandingan antara y_predict dan y_test untuk mengetahui apakah suatu audio yang dikategorikan Real benar-benar berlabel Real atau justru Fake, begitu juga sebaliknya. Dari hasil analisis, didapatkan beberapa hasil prediksi label suatu audio yang tidak sesuai dalam bentuk *confussion matrix*.

Tabel 14. *Confussion Matrix Decision Tree*

		Prediksi	
		Real	***Fake***
Aktual	***Real***	22	4
	Fake	3	21

Berikut penjelasan *confussion matrix* pada Tabel 14 di atas.

- Jumlah data yang diklasifikasikan dengan benar sebagai audio Real (TP) sebanyak 22 audio.
- Jumlah data yang seharusnya diklasifikasikan sebagai audio Real, tetapi salah diklasifikasikan sebagai audio Fake (FP) sebanyak 3 audio.
- Jumlah data yang seharusnya diklasifikasikan sebagai audio Fake, tetapi salah diklasifikasikan sebagai audio Real (FN) sebanyak 4 audio.
- Jumlah data yang diklasifikan dengan benar sebagai audio Fake (TN) sebanyak 21 audio.

Tabel 15. Evaluasi Model *Decision Tree*

Precision	*Recall*	*f1-score*	*Accuracy*
0,86	0,86	0,86	0,86

Dari Tabel 15 didapatkan hasil untuk *precision* yang digunakan untuk mengukur sejauh mana prediksi positif model yang benar dan didapatkan sebesar 86%, *recall* yang digunakan untuk mengukur sejauh mana model mampu mendeteksi semua *instance positif* yang sebenarnya dan meminimalkan *false negative* dan didapatkan sebesar 86%, *f1-score* yang digunakan untuk membetikan keseimbangan antara *precision* dan *recall* dan didapatkan sebesar 86%, dan *accuracy* model yang digunakan untuk mengukur sejauh mana model mampu memprediksi dengan benar dari seluruh prediksi yang dibuat dan didapatkan sebesar 86%.

4.2.5 Multilayer Perception (MLP)

Multilayer Perceptron (MLP) adalah jenis jaringan saraf tiruan yang terdiri dari lapisan input, lapisan tersembunyi, dan lapisan *output*. Lapisan input menerima fitur data, lapisan tersembunyi melakukan transformasi linier dengan bobot dan fungsi aktivasi, sementara lapisan *output* menghasilkan *output* untuk tugas klasifikasi atau regresi. Pelatihan MLP melibatkan penyesuaian bobot melalui *backpropagation* untuk mengurangi kesalahan antara prediksi dan target. Algoritma MLP dipilih pada analisis ini karena kemampuannya memproses fitur tingkat tinggi, kemudahan implementasinya, dan potensinya dalam memahami pola kompleks dalam data terbatas.

Tabel 16. Parameter MLP

alpha	*Hidden_layer_sizes*	*Learning_rate*	*Max_iter*
0,01	50	invscaling	150

Dalam membangun model ini, *data train* dimasukkan ke dalam model *Multilayer Perception* (MLP), kemudian dilakukan perbandingan antara y_predict dan y_test untuk mengetahui apakah suatu audio yang dikategorikan *Real* benar-benar berlabel *Real* atau justru *Fake*, begitu juga sebaliknya. Dari hasil analisis, didapatkan beberapa hasil prediksi label suatu audio yang tidak sesuai dalam bentuk *confussion matrix*.

Tabel 17. *Confussion Matrix MLP*

		Prediksi	
		Real	***Fake***
Aktual	***Real***	23	3
	Fake	0	24

Berikut penjelasan *confussion matrix* pada Tabel 17 di atas.

- Jumlah data audio yang diklasifikasikan dengan benar sebagai audio *Real* (TP) sebanyak 23 audio.
- Tidak ada data audio yang seharusnya diklasifikasikan sebagai audio *Real*, tetapi salah diklasifikasikan sebagai audio *Fake* (FP).
- Jumlah data audio yang seharusnya diklasifikasikan sebagai audio *Fake*, tetapi salah diklasifikasikan sebagai audio *Real* (FN) sebanyak 3 audio.
- Jumlah data audio yang diklasifikan dengan benar sebagai audio *Fake* (TN) sebanyak 24 audio.

Tabel 18. Evaluasi Model MLP

Precision	***Recall***	***f1-score***	***Accuracy***
0,93	0,92	0,92	0,92

Dari Tabel 18 didapatkan hasil untuk *precision* yang digunakan untuk mengukur sejauh mana prediksi positif model yang benar dan didapatkan sebesar 93%, *recall* yang digunakan untuk mengukur sejauh mana model mampu mendeteksi semua *instance positif* yang sebenarnya dan meminimalkan *false negative* dan didapatkan sebesar 92%, *f1-score* yang digunakan untuk membetikan keseimbangan antara *precision* dan *recall* dan didapatkan sebesar 92%, dan *accuracy* model yang digunakan untuk mengukur sejauh mana model mampu memprediksi dengan benar dari seluruh prediksi yang dibuat dan didapatkan sebesar 92%.

4.3 Perbandingan Akurasi Model

Tabel 19. Perbandingan Akurasi

	Random Forest	***SVM***	***K-NN***	***Decision Tree***	***MLP***
Accuracy	0,92	0,94	0,90	0,86	0,92

Setelah dilakukan pemodelan dengan empat model *machine learning* dan satu model *deep learning* didapatkan bahwa salah satu model *machine learning*, yaitu *Support Vector Machine* memiliki performa yang paling baik jika dibandingkan dengan empat model lainnya dengan akurasi mencapai 94%.

5. Kesimpulan

Berdasarkan hasil dan pembahasan yang telah dilakukan dalam penelitian ini, maka dapat dibuat kesimpulan bahwa penerapan SVM untuk mendeteksi *audio deepfake* merupakan algoritma yang baik karena manusia dapat terbantu untuk mendeteksi suara-suara yang ditiru atau dipalsukan secara sintetis. Dari hasil analisis yang didapatkan, model dengan *machine learning*, yaitu *Support Vector Machine* (SVM) memiliki performa yang paling baik jika dibandingkan dengan empat model lainnya dengan akurasi sebesar 94%, sehingga dalam studi kasus ini algoritma SVM merupakan algoritma *machine learning* yang paling sesuai untuk digunakan. Dalam penelitian selanjutnya, kami menyarankan sebaiknya dataset yang digunakan dapat lebih diperbanyak dan bervariasi untuk mendapatkan hasil yang lebih akurat dan relevan.

Daftar Pustaka

[1] Z. M. Almutairi and H. Elgibreen, "Detecting Fake Audio of Arabic Speakers Using Self-Supervised Deep Learning," *IEEE Access*, vol. 11, pp. 72134–72147, 2023, doi: 10.1109/ACCESS.2023.3286864.

[2] N. Diakopoulos and D. Johnson, "Anticipating and addressing the ethical implications of deepfakes in the context of elections," *New Media Soc*, vol. 23, no. 7, pp. 2072–2098, Jul. 2021, doi: 10.1177/1461444820925811.

[3] F. Iqbal, A. Abbasi, A. Rehman Javed, Z. Jalil, and J. Al-Karaki, "Deepfake Audio Detection via Feature Engineering and Machine Learning," *CEUR Workshop Proc*, 2022.

[4] H. Agarwal, A. Singh, and R. D, "Deepfake Detection Using SVM," in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, IEEE, Aug. 2021, pp. 1245–1249. doi: 10.1109/ICESC51422.2021.9532627.

[5] E. Retnoningsih and R. Pramudita, "Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python," *BINA INSANI ICT JOURNAL*, vol. 7, no. 2, p. 156, Dec. 2020, doi: 10.51211/biict.v7i2.1422.

[6] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, "Deepfake Detection: A Systematic Literature Review," *IEEE Access*, vol. 10, pp. 25494–25513, 2022, doi: 10.1109/ACCESS.2022.3154404.

[7] M. Mcuba, A. Singh, R. A. Ikuesan, and H. Venter, "The Effect of Deep Learning Methods on Deepfake Audio Detection for Digital Investigation," *Procedia Comput Sci*, vol. 219, pp. 211–219, 2023, doi: 10.1016/j.procs.2023.01.283.

[8] A. Hamza *et al.*, "Deepfake Audio Detection via MFCC Features Using Machine Learning," *IEEE Access*, vol. 10, pp. 134018–134028, 2022, doi: 10.1109/ACCESS.2022.3231480.

[9] J. A. K. Suykens, “Support Vector Machines: A Nonlinear Modelling and Control Perspective,” *Eur J Control*, vol. 7, no. 2–3, pp. 311–327, Jan. 2001, doi: 10.3166/ejc.7.311-327.