p-ISSN: 2301-5373
e-ISSN: 2654-5101

Pengaruh *Attention Pooling* dan Metode Ekstraksi Fitur Terhadap Performa LSTM dalam Klasifikasi Emosi Musik Piano

I Gusti Bagus Putrawan[a1], I Ketut Gede Suhartana[a2], Anak Agung Istri Ngurah Eka Karyawati[a3], I Made Widhi Wirawan[a4]
[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana, Jimbaran, Bali, Indonesia
Address, Country
[1]agunggusti95@gmail.com
[2]ikg.suhartana@unud.ac.id
[3]eka.karyawati@unud.ac.id
[4]made_widhi@unud.ac.id

Abstract

Music serves as a medium for emotional expression through acoustic elements such as pitch, rhythm, and timbre. This study focuses on music emotion classification based on solo piano instruments without vocal elements in the field of Music Emotion Recognition (MER). The objective of this research is to analyze the effect of pooling mechanisms and to compare the performance of MFCC and Log-Mel Spectrogram feature extraction methods on an LSTM model in classifying music emotions into four categories based on Thayer's quadrant model. The results demonstrate that the LSTM model with Attention Pooling consistently outperforms the LSTM model with Mean Pooling across all evaluation metrics, with accuracy improving from 63.97% to 66.91%, macro precision from 64.03% to 68.24%, macro recall from 64.53% to 66.65%, and macro F1-score from 64.14% to 66.81%. Furthermore, a comparison of feature extraction methods on the LSTM architecture with Attention Pooling reveals that the model based MFCC features outperforms the model based on Log-Mel Spectrogram features across all evaluation metrics, achieving a macro F1-score of 66.81% compared to 60.54%, with an accuracy gap of 5.15%.

Keywords: *Music Emotion Recognition*, Log-Mel Spectrogram, MFCC, LSTM, EMOPIA, *Thayer Quadrant, Attention Pooling*

1. Pendahuluan

Musik merupakan bentuk ekspresi seni yang tersusun dari elemen bunyi seperti nada, ritme, tempo, dan timbre yang disusun secara terstruktur untuk menyampaikan makna atau emosi tertentu. Instrumen musik berperan sebagai medium utama dalam menghasilkan bunyi, di mana setiap instrumen memiliki karakteristik akustik dan warna bunyi yang khas. Salah satu instrumen modern yang banyak digunakan dan memiliki rentang ekspresi emosional yang luas adalah piano sehingga berpotensi menjadi sumber data yang representatif dalam penelitian. Dalam bidang *Music Information Retrieval* (MIR) terdapat cabang kajian *Music Emotion Recognition* (MER) yang berfokus pada pengenalan emosi atau suasana (*mood*) dari musik. Hingga saat ini, sebagian besar penelitian MER masih berfokus pada lagu lengkap dengan vokal dan iringan musik sehingga kajian yang secara khusus menelaah emosi musik berbasis instrumen tunggal seperti piano masih relatif terbatas dan memiliki potensi kontribusi ilmiah.

Penelitian mengenai pemilihan ekstraksi fitur pada *Speech Emotion Recognition* (SER) dan *Music Emotion Recognition* (MER) masih menunjukkan hasil yang inkonsisten. Penelitian menggunakan dataset berisi 2.474 sampel dan model VGGb melaporkan bahwa MFCC menghasilkan akurasi yang lebih tinggi, yaitu 90.73% dibandingkan Log-Mel Spectrogram sebesar 81.05%, konsisten di berbagai kondisi pengujian dalam penelitian tersebut [1]. Sebaliknya, penelitian lain menunjukkan bahwa Log-Mel Spectrogram mengungguli MFCC ketika digunakan bersama arsitektur CNN [2]. Kedua studi tersebut tidak hanya berbeda pada hasil, tetapi juga pada ukuran sampel dan arsitektur model yang digunakan sehingga sulit untuk

menyimpulkan fitur yang lebih baik. Perbedaan hasil tersebut menunjukkan bahwa belum terdapat kesimpulan yang konsisten mengenai fitur yang paling efektif untuk kasus audio dan music emotion recognition. Selain pemilihan fitur akustik, komponen arsitektur juga mempengaruhi performa dari model. Pada task *speech intelligibility* penggunaan *attention pooling* setelah arsitektur LSTM juga memberikan akurasi 76.92% dimana hal ini lebih tinggi dibandingkan penggunaan *mean pooling* yang memberikan akurasi 74.82% [3].

Emosi pada *speech* dan *music* memiliki kemiripan akustik sehingga pengetahuan dari satu domain berpotensi ditransfer ke domain lainnya sebagaimana ditunjukkan oleh penelitian yang mentransfer pengetahuan dari SER ke MER [4]. Meskipun demikian, evaluasi perbandingan MFCC dan Log-Mel Spectrogram masih jarang dilakukan pada domain MER. Hal ini menjadi semakin relevan mengingat karakteristik musik piano, seperti struktur harmonik, dinamika nada, dan timbre. Berbeda halnya pada *speech* yang lebih banyak dipengaruhi oleh intonasi dan artikulasi sehingga hasil dari domain SER belum menjamin generalisasi ke domain MER. Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi efektivitas fitur MFCC dan Log-Mel Spectrogram pada Music Emotion Recognition berbasis musik piano.

Penelitian ini juga mengusulkan pendekatan klasifikasi emosi musik berbasis karakteristik akustik instrumen piano tanpa melibatkan unsur vokal dengan memanfaatkan dataset EMOPIA [5]. Pendekatan ini dilatarbelakangi oleh kebutuhan sistem yang mampu memberikan rekomendasi musik yang tepat kepada pendengar berdasarkan emosi yang terkandung dalam musik, mengingat salah satu alasan utama seseorang mendengarkan musik adalah untuk merasakan *mood* atau emosi.

2. Metodologi Penelitian

2.1 Pengumpulan dan Pembagian Data

Data yang digunakan merupakan data sekunder dan EMOPIA adalah sebuah *dataset multi-modal* yang berisi data audio dan MIDI, serta fokus pada persepsi emosi dalam musik piano pop. Data audio diperoleh dengan mengunduh file audio dari *platform* YouTube. Proses pengunduhan dilakukan menggunakan data berupa youtube_id yang berperan sebagai *key* pada berkas metadata.json. Setiap youtube_id kemudian dikonversi ke dalam format *URL* YouTube sehingga dapat digunakan untuk mengunduh file audio yang bersesuaian. Dalam satu youtube_id, terdapat beberapa segmen audio yang masing-masing telah diberikan label emosi oleh para ahli. Setelah proses ini, didapatkan jumlah data audio sebanyak 878 klip dengan proses *split* 85:15 distribusi *train* dan *test* berturut-turut menghasilkan 742 dan 136. Masing-masing label dan jumlah data yang terdapat pada *train set* berturut-turut, kelas Q1 (*Happy*) berjumlah 156, Q2 (*Angry*) berjumlah 211, Q3 (*Sad*) berjumlah 155, dan Q4 (*Relax*) berjumlah 193.

2.2 Augmentasi Data

Data augmentation dilakukan untuk memperbanyak jumlah data maupun menangani ketidakseimbangan kelas (*class imbalance*) dengan melakukan *oversampling* pada kelas minoritas. Hal ini bertujuan agar model tidak berfokus pada kelas mayoritas selama proses pelatihan. Berdasarkan penelitian sebelumnya [6], melakukan terlalu banyak transformasi tidak akan menambah akurasi secara signifikan sehingga pada penelitian ini hanya dilakukan empat jenis transformasi terhadap data original pada seluruh kelas. Transformasi yang diterapkan meliputi kombinasi *time stretching* dan *pitch shifting* untuk mengubah kecepatan audio serta menggeser nada audio, *time shifting* untuk menggeser posisi sinyal audio secara temporal, dan penambahan *Gaussian noise* untuk meningkatkan ketahanan model terhadap variasi sinyal.

2.3 Preprocessing

Tahap setelah dilakukan pengumpulan dan augmentasi data, tahap selanjutnya dilakukan *preprocessing* data audio yang dijabarkan sebagai berikut.

a. *Resampling* dan *Mono Convertion*
   Sinyal audio yang dikonversi ke dalam format .wav, dilanjutkan ke tahap *audio signal processing* dengan melakukan penyeragaman *resampling* sebesar 22.050 Hz dan audio di-*load* dengan *mono channel*. Pemilihan frekuensi sampling 22.050 Hz didasarkan pada nilai

p-ISSN: 2301-5373
e-ISSN: 2654-5101

default librosa yang umum digunakan dalam penelitian MIR karena mampu merepresentasikan rentang frekuensi yang relevan dengan persepsi pendengaran manusia.

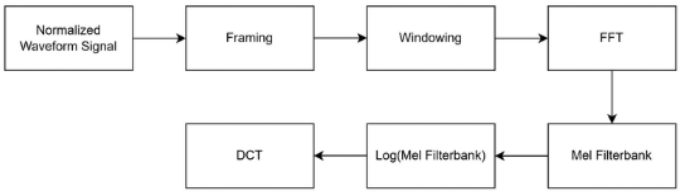
b. *Loudness Normalization* (*Waveform Normalization*)
Di tahap ini, dilakukan proses normalisasi audio lebih spesifiknya dengan *Loudness Normalization* berbasis LUFS (ITU-R BS.1770) menggunakan *pyloudnorm library* dengan tujuan untuk menyamakan tingkat *loudness* antar sinyal audio sehingga menghindari bias persepsi akibat perbedaan volume. Adapun target LUFS yang ditetapkan adalah sebesar -23 LUFS mengikuti standar dari EBU R128 [7].

c. Audio Segmentation
Segmentasi audio dilakukan dengan membagi sinyal menjadi beberapa segmen berdurasi 30 detik. Pemilihan durasi ini didasarkan pada temuan bahwa model *music emotion recognition* (MER) mampu menangkap karakteristik emosi secara efektif dalam rentang waktu 15 hingga 30 detik [8]. Selain itu, digunakan overlap sebesar 50% untuk menjaga kontinuitas temporal antar segmen dan mengurangi potensi kehilangan informasi pada batas segmen [9]. Mengingat beberapa audio dalam dataset EMOPIA memiliki durasi lebih dari 45 detik sehingga segmentasi dengan *overlap* memungkinkan memberikan representasi yang lebih baik.

d. *Feature Extraction* dan *Standardization* dengan StandardScaler
Pada penelitian ini, digunakan Log-Mel Spectrogram dan MFCC sebagai ekstraksi fitur. Fitur yang diekstrak kemudian dinormalisasi menggunakan Z-Score, yaitu dengan menggeser distribusi fitur sehingga memiliki *mean* 0 dan standar deviasi 1. Normalisasi dilakukan sebelum *padding* agar nilai statistik tidak terdistorsi oleh nilai dummy [3].

2.4 Mel Frequency Cepstral Coefficient

Pada penelitian ini, sinyal audio hasil *preprocessing* memiliki durasi 30 detik dengan *sampling rate* sebesar 22.050 Hz sehingga total jumlah sampel yang didapatkan sebesar 661.500 sampel. Proses *framing* dilakukan secara implisit melalui parameter transformasi Fourier jangka pendek, dengan panjang *frame* ditentukan oleh nilai n_fft sebesar 2.048 sampel yang setara dengan durasi, serta pergeseran antar *frame* (*hop length*) sebesar 512 sampel. Pemilihan parameter ini memungkinkan sinyal audio diasumsikan bersifat tetap (*stasioner*) dalam setiap *frame* sekaligus mempertahankan keseimbangan antara resolusi waktu dan resolusi frekuensi.



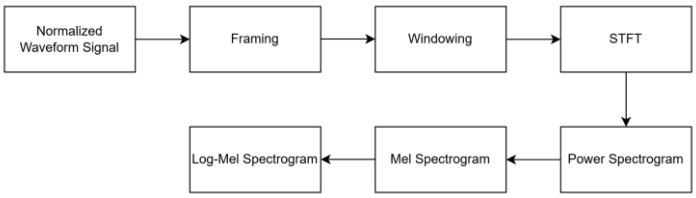
Gambar 1. Tahapan MFCC

Proses ekstraksi fitur MFCC terdiri dari beberapa tahapan yang saling berkesinambungan. Tahapan pertama adalah *framing*, yaitu sinyal audio dibagi menjadi sejumlah *frame* yang lebih kecil di mana setiap *frame* merepresentasikan sejumlah sampel audio dengan panjang tetap sebesar 25ms. Antar *frame* diterapkan mekanisme *overlap* atau tumpang tindih untuk mencegah hilangnya informasi pada batas antar *frame*. Tahapan kedua adalah *windowing*, yang bertujuan untuk meredam efek *aliasing* yang dapat muncul akibat proses *framing*. *Aliasing* merupakan fenomena munculnya sinyal baru dengan frekuensi yang berbeda dari sinyal aslinya sehingga dapat menyebabkan diskontinuitas sinyal. Pada tahap ini, setiap *frame* dikalikan dengan fungsi *Hann window* untuk memperhalus transisi pada tepi *frame*.

Tahapan ketiga adalah *Fast Fourier Transform* (FFT) yang digunakan untuk mengkonversi sinyal dari domain waktu ke domain frekuensi pada setiap *frame* [8] sehingga menghasilkan spektrum yang merepresentasikan amplitudo dari berbagai komponen frekuensi dalam sinyal audio. Tahapan keempat adalah *Mel Filterbank*, di mana hasil FFT dilewatkan melalui serangkaian *triangular filter* yang dirancang untuk mengukur energi pada rentang frekuensi tertentu. Setiap nilai FFT dikalikan dengan nilai filter yang bersesuaian dan hasilnya dijumlahkan untuk memperoleh representasi energi sinyal dalam skala mel yang lebih sesuai dengan karakteristik persepsi pendengaran manusia [10]. Skala mel bersifat linier untuk frekuensi di bawah 1000 Hz

dan logaritmik untuk frekuensi di atasnya. Tahapan kelima sekaligus terakhir adalah *Discrete Cosine Transform* (DCT) yang mengonversi spektrum magnitudo ke domain cepstral sehingga menghasilkan vektor koefisien *Mel-Frequency Cepstral Coefficients* [11]. Jumlah koefisien cepstral yang umum digunakan dalam penelitian adalah berada dari rentang 20 hingga 40 koefisien, serta penetapan dari 25 koefisien MFCC sebagai ekstraksi fitur sudah menjadi praktik yang umum digunakan. Sehingga, dalam penelitian ini akan digunakan koefisien MFCC sebanyak 25 dengan nilai mel sebesar 40. Adapun tahapan dari MFCC digambarkan sebagai berikut.

2.5 Log-Mel Spectrogram

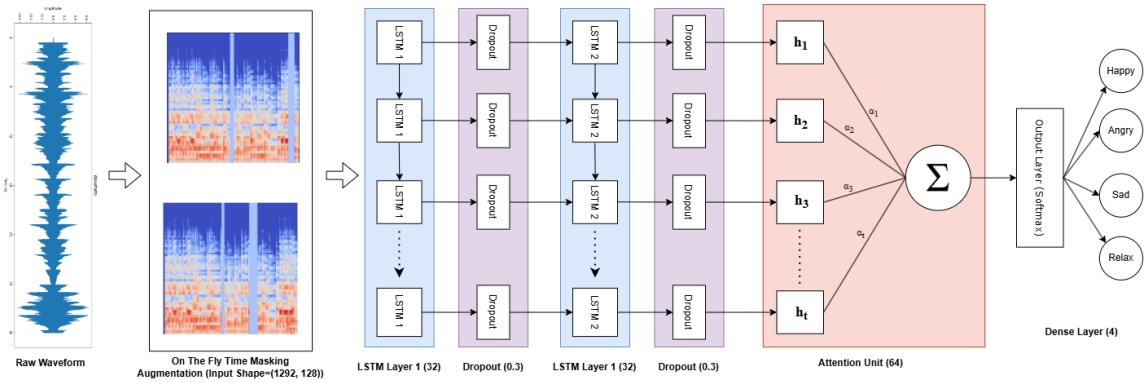


Gambar 2. Tahapan Log-Mel Spectrogram

Setelah proses *framing*, setiap *frame* diproses melalui *windowing*, yaitu pengalian sinyal *frame* dengan fungsi *window* seperti *Hanning window* yang digunakan untuk mengurangi diskontinuitas pada batas awal dan akhir *frame*, yaitu perubahan amplitudo sinyal yang terjadi secara tiba-tiba akibat pemotongan sinyal kontinyu menjadi segmen diskrit, serta untuk meminimalkan kebocoran spektral sebelum dilakukan transformasi ke domain frekuensi. Selanjutnya, setiap *frame* yang telah di-*window* ditransformasikan dari domain waktu ke domain frekuensi menggunakan *Short-Time Fourier Transform* (STFT) untuk memperoleh representasi frekuensi sinyal terhadap waktu. Hasil STFT kemudian dihitung nilai magnitudo atau dayanya sehingga menghasilkan *power spectrogram* yang merepresentasikan distribusi energi sinyal pada setiap frekuensi linear dalam satuan hertz (Hz). *Power spectrogram* ini selanjutnya dipetakan ke skala Mel menggunakan *Mel filter bank* dengan jumlah pita sebanyak 128 Mel Filterbank yang merupakan nilai *default* pada librosa, disusun lebih rapat pada frekuensi rendah dan semakin renggang pada frekuensi tinggi dengan frekuensi maksimum sebesar 8.000 Hz untuk memfokuskan representasi pada rentang frekuensi yang paling relevan dengan karakteristik persepsi pendengaran manusia.

2.6 Long Short Term Memory

Long Short Term Memory (LSTM) adalah versi yang sudah dimodifikasi dari *Recurrent Neural Network* (RNN) yang berdasarkan definisinya membuatnya lebih mudah untuk mengingat data masa lalu dalam memori. Dijelaskan juga bahwa LSTM lebih efektif dibandingkan RNN *standard* dalam mengatasi *vanishing gradient* [12]. Permasalahan ini muncul ketika model RNN dihadapkan dengan ukuran data yang sangat panjang sehingga model tidak dapat mengingat informasi dalam waktu yang lama.



Gambar 3. Arsitektur Model LSTM

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Berdasarkan arsitektur model pada gambar, model yang digunakan adalah stacked LSTM layer 1 dan 2 dengan masing-masing dropout sebesar 0.3 setelah LSTM layer. Dropout bekerja dengan cara mengeliminasi sejumlah unit secara acak pada tiap layer selama proses pelatihan sehingga unit-unit yang tersisa tidak bergantung satu sama lain, yang pada akhirnya membantu model terhindar dari *overfitting* [13, 14]. Parameter-parameter yang digunakan dalam block LSTM *Layer* 1 dan LSTM *Layer* 2 adalah sejumlah unit LSTM. Jumlah unit yang digunakan akan dipilih melalui proses *hyperparameter tuning* untuk memperoleh nilai akurasi dan F1-Score terbaik. Parameter berikut adalah kernel_regularizer L2 pada LSTM yang digunakan dengan nilai 0.0001 dengan tujuan untuk mengontrol kompleksitas bobot input yang masuk ke dalam layer LSTM. *Regularizer* L2 membantu model untuk mempelajari nilai bobot yang lebih kecil sehingga model tidak menghafalkan data pelatihan dan mampu melakukan generalisasi ke data baru [12].

Arsitektur *deep learning* yang sudah dibangun akan dilakukan *hyperparameter tuning* dengan membandingkan beberapa nilai, seperti unit LSTM *layer* pertama, unit LSTM *layer* kedua dan *learning rate* yang digunakan. Penentuan nilai *learning rate* didasarkan dari seberapa cepat model tersebut dapat belajar. Learning rate yang terlalu besar dapat menyebabkan model belajar lebih cepat, tetapi di satu sisi loss dari model bisa tidak stabil dan sulit konvergen. Learning rate yang lebih rendah sebagian besar menyebabkan validation loss selama training lebih stabil, hanya saja pada kondisi tertentu model dapat mengalami underfitting sekaligus memerlukan lebih banyak iterasi sehingga meningkatkan biaya komputasi [15]. Oleh karena itu, pemilihan learning rate yang tepat tidak hanya membantu model belajar lebih baik, tetapi juga berkontribusi pada efisiensi waktu pelatihan dengan mempercepat proses konvergensi. Sama halnya dengan pemilihan pada unit LSTM, pemilihan jumlah unit yang tepat untuk komparasi fitur membantu model menghindari terjadinya baik *overfitting* maupun *underfitting*.

Nilai *batch size* yang digunakan pada penelitian ini sebesar 32 dan *epoch* sebesar 50. Adapun untuk menghentikan training model sebelum terjadi *overfitting* digunakan *early stopping* dengan memonitoring nilai dari *validation loss* dengan 10 *epoch* sebagai nilai *patience* mengindikasikan model akan berhenti melakukan *training* jika setelah 10 *epoch,* nilai *validation loss* terbaik sebelumnya tidak mengalami penurunan. Optimizer yang digunakan adalah algoritma Adam dengan parameter *learning rate* yang akan disesuaikan pada tahap *hyperparameter tuning*. Adam merupakan salah satu optimizer adaptif yang paling banyak digunakan dalam pelatihan deep neural network karena kemampuannya menyesuaikan learning rate secara dinamis untuk setiap parameter model [16].

2.7 Attention Pooling

Istilah attention pada layer ini terinspirasi dari penggunaan mekanisme attention yang banyak digunakan dalam bidang *Natural Language Processing* (NLP). Setiap kata dalam suatu kalimat akan diberikan *attention score* yang menggambarkan tingkat kepentingan makna dalam kalimat tersebut [17]. Dengan demikian, kata dengan *score* yang paling besar memiliki kontribusi lebih banyak dalam representasi akhir dokumen.

Dalam konteks *task speaker recognition*, bobot atau skor akan dihitung berdasarkan *frame level* atau *timesteps* dan menghitung memilih *frame* dengan menghitung skor dari tiap frame. *Frame* yang lebih informatif sebagai contohnya bagian audio yang kaya akan informasi spektral akan memiliki bobot lebih besar dibandingkan *frame* yang kurang informatif. Hal ini menjadi alasan attention pooling lebih adaptif dalam situasi tertentu dibandingkan dengan average pooling yang mengukur seluruh *frame* secara sama [18].

Secara matematis, jika diberikan sekumpulan input dari *hidden layer* sebelumnya h_t untuk setiap *timesteps* atau *frame* t dan untuk menghitung *hidden representation* u_t dari h_t diperlukan satu *forward layer* MLP yang terdiri dari bobot W, bias b, dan fungsi aktivasi tanh, sehingga persamaan untuk menentukan *hidden representation* dirumuskan sebagai berikut.

$$u_t = \phi(W h_t + b) \quad (1)$$

Attention score diperoleh dengan melakukan *dot product* antara *hidden representation* u_t dengan *query vector* q yang berfungsi sebagai *query* untuk menilai seberapa penting setiap kata dalam kalimat [19]. Dalam konteks penelitian *audio processing, query* digunakan untuk menghitung seberapa penting *frame* sinyal dalam suatu urutan waktu atau *timesteps*. *Attention score* dinormalisasi pada seluruh *timesteps* menggunakan fungsi softmax sehingga *score* atau

disebut juga dengan *attention weight* α_t akan memiliki rentang nilai dari 0 hingga 1. Persamaan *attention weight* dirumuskan sebagai berikut.

$$\alpha_t = \frac{exp(u_t q)}{\sum_t \quad exp(u_t q)} \tag{2}$$

Sequence vector diperoleh dengan menghitung *weighted sum* dari seluruh *hidden representation* h_t menggunakan *attention weight* α_t. *Sequence vector* merepresentasikan informasi penting dalam urutan waktu atau *timesteps* dengan memberikan bobot atau *skor* lebih besar pada timestep yang dianggap lebih informatif. Persamaan dari *sequence vector* adalah sebagai berikut.

$$context = \sum_t \quad \alpha_t h_t \tag{3}$$

2.8 Metrik Pengujian

Presisi merupakan metrik yang mengukur proporsi prediksi positif yang benar-benar sesuai dengan kelas aktualnya terhadap keseluruhan prediksi positif yang dihasilkan model. Precision dihitung menggunakan persamaan (4), di mana TP merupakan true positive dan FP merupakan false positive.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Recall merupakan metrik yang mengukur kemampuan model dalam mendeteksi seluruh data positif yang sesungguhnya ada dalam dataset. Secara spesifik, recall menghitung perbandingan antara jumlah prediksi positif yang benar dengan total keseluruhan data positif yang tersedia. Recall dihitung menggunakan persamaan (5), di mana TP merupakan true positive dan FN merupakan false negative.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

F1-Score merupakan metrik evaluasi yang diperoleh dengan mengintegrasikan nilai precision dan recall ke dalam satu ukuran tunggal yang seimbang. Metrik ini banyak digunakan pada kasus imbalanced data karena mampu mengevaluasi performa model terhadap kelas minoritas secara lebih baik dibandingkan accuracy [20]. Nilai F1-Score dihitung menggunakan persamaan (6).

$$F1 - score = 2\ x\ \frac{presisi\ x\ recall}{presisi + recall} \tag{6}$$

Akurasi merupakan metrik yang mengukur perbandingan secara menyeluruh antara jumlah prediksi yang benar pada seluruh kelas terhadap total keseluruhan data yang dievaluasi. Berbeda dengan precision, recall, dan F1-Score yang berfokus pada kelas tertentu, akurasi memberikan gambaran umum mengenai performa model secara keseluruhan. Nilai akurasi dihitung menggunakan persamaan (7).

$$Accuracy = \frac{TP+TN}{TP + FN+TN+FP} \tag{7}$$

3. Hasil dan Pembahasan

Bagian ini berisi pembahasan mengenai komparasi perbandingan performa antara fitur MFCC dengan fitur Log-Mel Spectrogram yang diinputkan ke dalam model LSTM. Selain itu, pada bagian ini akan dibahas *hyperparameter tuning* berdasarkan *validation accuracy* juga diterapkan untuk mencari performa dengan model terbaik, evaluasi *loss* dan akurasi berdasarkan proses *training,* dan hasil evaluasi model menggunakan *confusion matrix*.

3.1. *Hyperparameter Tuning 5-Fold Cross Validation*

Pengujian parameter dilakukan menggunakan *Grid Search* dengan *Stratified K-Fold Cross Validation*. Nilai K yang terlalu kecil dapat mengurangi jumlah data *training* secara signifikan

p-ISSN: 2301-5373
e-ISSN: 2654-5101

sedangkan nilai K yang terlalu besar membutuhkan waktu komputasi yang lama untuk menentukan model terbaik dan menghasilkan *validation set* yang terlalu kecil sehingga kurang representatif terutama pada dataset yang terbatas. Kondisi ini juga berpotensi meningkatkan standar deviasi antar fold akibat ketidakstabilan estimasi performa model. Oleh karena itu, nilai K=5 dipilih sebagai keseimbangan representasi data dan efisiensi komputasi pada penelitian ini.

Tabel 1. Perbandingan Konfigurasi Hyperparameter Terbaik

Parameter	Nilai
Unit LSTM 1	[64, 32]
Unit LSTM 2	[64, 32]
Learning Rate	[0.001, 0.0001]

Berdasarkan tabel 1, adapun beberapa nilai *learning rate* diuji, yaitu 0.001 dan 0.0001. Arsitektur model yang digunakan terdiri atas dua lapisan LSTM dengan jumlah unit masing-masing 64 dan 32. Pemilihan jumlah unit ini mempertimbangkan bahwa unit yang terlalu besar dapat meningkatkan risiko *overfitting* serta memperpanjang waktu *training*. Selain itu, dua nilai *learning rate* yang diuji, yaitu 0,001 dan 0,0001. Mengingat nilai yang terlalu besar dapat menyebabkan *training* tidak stabil atau gagal konvergen sedangkan nilai yang terlalu kecil dapat menyebabkan *training* berjalan sangat lambat dan terjebak pada *local minimum*. Learning rate dan jumlah unit hidden layer termasuk hyperparameter dengan pengaruh terbesar terhadap performa model deep learning dibanding parameter lain [21]. Meskipun demikian, penambahan jumlah *hyperparameter* yang diuji dapat menyebabkan biaya komputasi meningkat secara eksponensial mengingat kompleksitas *grid search* adalah O(n^k) [22]. Oleh karena itu, *hyperparameter* dalam penelitian ini dibatasi pada kombinasi jumlah unit LSTM dan *learning rate* saja untuk menjaga waktu komputasi tetap terkendali.

Tabel 2. KFold pada Metode MFCC dan LSTM dengan *Attention Pooling*

Parameter	Fold					Unweighted Average F1-Score
	1	2	3	4	5	
64, 64, 0.001	0.7051	0.6603	0.6180	0.6341	0.6830	0.6601 ± 0.0316
64, 64, 0.0001	0.7219	0.6476	0.6682	0.6350	0.6991	0.6744 ± 0.0322
64, 32, 0.001	0.7125	0.6490	0.6704	0.6615	0.7160	0.6819 ± 0.0273
64, 32, 0.0001	0.6904	0.6204	0.6234	0.6207	0.7047	0.6519 ± 0.0375
32, 64, 0.001	0.6526	0.6689	0.6468	0.6225	0.6575	0.6496 ± 0.0154
32, 64, 0.0001	0.6928	0.6437	0.6741	0.6154	0.6978	0.6636 ± 0.0329
32, 32, 0.001	0.7036	0.6411	0.6315	0.6193	0.6946	0.6580 ± 0.0344
32, 32, 0.0001	0.6831	0.6298	0.6400	0.5872	0.6539	0.6388 ± 0.0317

Hasil pengujian *5-fold cross validation* pada tabel 2 menunjukkan parameter terbaik diperoleh dengan nilai unit pada LSTM 1 adalah 64 dan unit LSTM 2 adalah 32 serta nilai *learning rate* 0.001 memberikan nilai rata-rata *macro f1-score* sebesar 0.6819 dengan nilai standar deviasi 0.0273.

Tabel 3. KFold pada Metode MFCC dan LSTM dengan *Mean Pooling*

Parameter	Fold					Unweighted Average F1-Score
	1	2	3	4	5	
64, 64, 0.001	0.6846	0.6474	0.6314	0.6155	0.6618	0.6481 ± 0.0240

64, 64, 0.0001	0.7071	0.6511	0.6461	0.6216	0.7183	0.6688 ± 0.0373
64, 32 0.001	0.6585	0.6695	0.6253	0.6222	0.7194	0.6590 ± 0.0354
64, 32, 0.0001	0.7183	0.6882	0.6353	0.6351	0.7211	0.6796 ± 0.0390
32, 64, 0.001	0.7421	0.6313	0.6185	0.6329	0.6836	0.6617 ± 0.0460
32, 64, 0.0001	0.6790	0.6227	0.6518	0.5696	0.6535	0.6353 ± 0.0374
32, 32, 0.001	0.6468	0.6043	0.6152	0.5714	0.6915	0.6258 ± 0.0401
32, 32, 0.0001	0.6512	0.6124	0.6198	0.5751	0.6886	0.6294 ± 0.0387

Hasil pengujian *5-fold cross validation* pada tabel 3 menunjukkan parameter terbaik diperoleh dengan unit LSTM 1 adalah 64 dan unit LSTM 2 adalah 32 serta nilai *learning rate* 0.0001 memberikan nilai rata-rata *macro f1-score* sebesar 0.6796 dengan nilai standar deviasi 0.0390.

Tabel 4. KFold pada Metode Log-Mel Spectrogram dan LSTM dengan *Attention Pooling*

Parameter	Fold					Unweighted Average F1-Score
	1	2	3	4	5	
64, 64, 0.001	0.6474	0.6534	0.6470	0.6028	0.6645	0.6430 ± 0.0211
64, 64, 0.0001	0.6508	0.5861	0.6036	0.6016	0.6579	0.6200 ± 0.0288
64, 32, 0.001	0.6062	0.6384	0.6261	0.6600	0.7019	0.6465 ± 0.0327
64, 32, 0.0001	0.6697	0.6264	0.5812	0.6123	0.6471	0.6273 ± 0.0304
32, 64, 0.001	0.6498	0.6191	0.6247	0.6118	0.6814	0.6374 ± 0.0253
32, 64, 0.0001	0.6349	0.5967	0.6078	0.5844	0.6537	0.6155 ± 0.0286
32, 32, 0.001	0.6792	0.6141	0.6223	0.6361	0.6765	0.6456 ± 0.0272
32, 32, 0.0001	0.6224	0.6380	0.6055	0.6049	0.6762	0.6294 ± 0.0264

Hasil pengujian *5-fold cross validation* pada tabel 4 menunjukkan parameter terbaik diperoleh dengan unit LSTM 1 adalah 64 dan unit LSTM 2 adalah 32 serta nilai *learning rate* 0.001 memberikan nilai rata-rata *macro f1-score* sebesar 0.6465 dengan nilai standar deviasi 0.0327.

Tabel 5. Perbandingan Performa Model dengan 5-Fold Cross Validation

Feature Extraction	***Model***	***UA F1-Score***
MFCC	LSTM + Attention Pooling	0.6819 ± 0.0273
MFCC	LSTM + Mean Pooling	0.6796 ± 0.0390
Log-Mel Spectrogram	LSTM + Attention Pooling	0.6465 ± 0.0327

Berdasarkan tabel 5, model LSTM dengan *Attention Pooling* dan metode ekstraksi fitur MFCC mencapai *Unweighted Average F1-Score* tertinggi sebesar 0.6819. Perbandingan metode ekstraksi fitur pada model yang sama, yakni LSTM dengan *Attention Pooling*, menunjukkan bahwa MFCC mengungguli *Log-Mel Spectrogram* sebesar 3.54%. Selain itu, metode ekstraksi fitur MFCC dengan model LSTM berbasis *Attention Pooling* menunjukkan standar deviasi terendah (±0.0273) dibandingkan *Mean Pooling* (±0.0390) dan *Log-Mel Spectrogram* (±0.0327) yang mengindikasikan bahwa model ini memiliki performa yang lebih stabil dan konsisten di seluruh *fold* validasi. Hasil ini menunjukkan bahwa kombinasi MFCC sebagai metode ekstraksi

p-ISSN: 2301-5373
e-ISSN: 2654-5101

fitur dan *Attention Pooling* sebagai mekanisme agregasi menghasilkan representasi yang paling optimal pada kasus ini.

3.2. Analisis Performa Model

Tabel 6 membahas perbandingan penggunaan metode *pooling* pada model LSTM sedangkan pada tabel 7 membandingkan penggunaan ekstraksi fitur menggunakan model yang sama, yakni LSTM berbasis *Attention*. Adapun untuk pengujian digunakan metrik *macro average precision, recall, f1-score*, dan nilai akurasi dengan dilakukan pengujian pada *test set.*

Tabel 6. Perbandingan Performa *Pooling* pada Model LSTM

Pooling Method	*Accuracy*	*Precision*	*Recall*	*F1-Score*
LSTM + Mean Pooling	0.64	0.64	0.65	0.64
LSTM + Attention Pooling	0.67	0.68	0.67	0.67

Penggunaan *attention pooling* setelah layer LSTM menghasilkan peningkatan sebesar 3% pada *accuracy* dan *F1-score*. Attention pooling menghasilkan performa yang lebih baik karena mekanisme attention memberikan bobot yang berbeda pada setiap representasi temporal yang dihasilkan LSTM. *Frame* musik yang mengandung informasi emosi yang lebih kuat memperoleh bobot yang lebih besar sedangkan *frame* yang kurang relevan memperoleh bobot yang lebih kecil. Bobot yang lebih besar ini umumnya berasal dari segmen yang merepresentasikan bagian klimaks atau *chorus* lagu, yaitu bagian yang secara emosional paling menonjol serta paling merepresentasikan keseluruhan isi lagu. Hal ini juga terjadi pada penelitian yang menunjukkan bahwa *attention* dapat membantu mendeteksi bagian *chorus* dari sebuah lagu [23]. Dengan demikian, representasi akhir yang digunakan untuk proses klasifikasi menjadi lebih informatif karena berpusat pada segmen yang paling merepresentasikan emosi keseluruhan lagu dibandingkan mean pooling yang hanya menghitung rata-rata seluruh frame tanpa mempertimbangkan tingkat kepentingannya.

Tabel 7. Perbandingan Performa Model LSTM-Attention Berdasarkan Ekstraksi Fitur

Feature Extraction	*Accuracy*	*Precision*	*Recall*	*F1-Score*
Log-Mel Spectrogram	0.62	0.64	0.61	0.61
MFCC	0.67	0.68	0.67	0.67

Perbedaan performa antara kedua metode ekstraksi fitur terlihat cukup signifikan dimana MFCC mencapai nilai akurasi dan *F1-score* yang lebih tinggi dibandingkan Log-Mel Spectrogram pada dataset EMOPIA. Hal ini menandakan bahwa fitur yang dihasilkan MFCC lebih efektif dalam menangkap karakteristik sinyal audio pada dataset yang digunakan. Keunggulan ini disebabkan kemampuan dari MFCC sendiri dalam merepresentasikan karakteristik spektral yang berkaitan dengan timbre dan dinamika permainan piano secara lebih ringkas karena MFCC pada dasarnya merupakan versi terkompresi dan terdekorelasi dari Mel-Spectrum. Representasi yang lebih ringkas tersebut sudah cukup untuk membedakan karakteristik emosi antar kelas. Sebaliknya, Log-Mel Spectrogram memiliki dimensi fitur yang jauh lebih tinggi sehingga membutuhkan data pelatihan yang lebih banyak agar pola-pola spektralnya dapat dipelajari secara optimal oleh model. Mengingat ukuran dataset EMOPIA yang relatif terbatas, representasi MFCC menjadi lebih efektif dibandingkan Log-Mel Spectrogram. Selain itu, MFCC cenderung menghindari *overfitting* karena dimensinya lebih rendah, sementara Log-Mel spectrogram lebih unggul ketika data tersedia lebih banyak. Hal ini juga ditunjukkan pada penelitian sebelumnya yang menunjukkan bahwa MFCC seringkali berkinerja lebih baik ketika data yang tersedia terbatas [24], serta penelitian lain yang melaporkan bahwa MFCC menunjukkan efektivitas lebih tinggi dibandingkan spektrogram Mel pada arsitektur LSTM [25].

3.3. Evaluasi Metode *Pooling*

Tabel 8. *Classification Report* Model LSTM dan *Attention Pooling*

	Precision	Recall	F1-Score	Support
Happy	0.5952	0.7143	0.6494	35
Angry	0.7059	0.6154	0.6575	39
Sad	0.7619	0.5714	0.6531	28
Relax	0.6667	0.7647	0.7123	34
Macro Average	0.6824	0.6655	0.6681	136

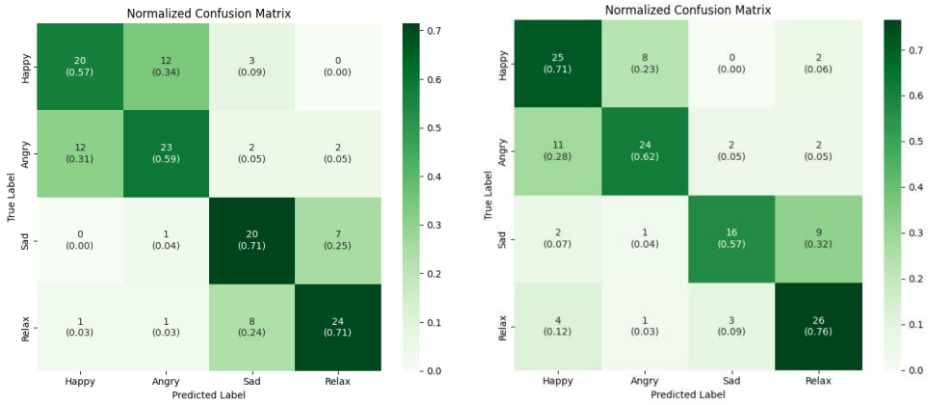
Evaluasi model pada data uji yang terdiri dari 136 sampel dengan empat kelas emosi menghasilkan akurasi keseluruhan sebesar 66,91% dan *Macro* (*Unweighted*) *Average F1-Score* sebesar 66.81%. Model memperoleh *F1-score* sebesar 0.65 untuk kelas *happy* dengan *precision* 0.60 dan *recall* 0.71. *F1-Score* 0.66 untuk kelas *angry* dengan *precision* 0.71 dan *recall* 0.62. *F1-Score* 0.65 pada kelas *sad* memiliki *precision* 0.76 serta *recall* 0.57. *F1-Score* 0.71 untuk kelas *relax* dengan *precision* 0.67 dan *recall* 0.76. Kelas *relax* memiliki F1-score tertinggi sedangkan kelas *sad* memiliki *precision* tertinggi namun *recall* yang rendah yang mengindikasikan bahwa model cenderung dapat mengklasifikasikan dengan benar dan kadang melewatkan data yang sebenarnya termasuk kelas tersebut. Sebaliknya, kelas *happy* memiliki nilai *recall* yang tinggi namun *precision* rendah yang berarti model terlalu sering memprediksi kelas *happy* sehingga menghasilkan lebih banyak *false positive*.

Tabel 9. *Classification Report* Model LSTM dan *Mean Pooling*

	Precision	Recall	F1-Score	Support
Happy	0.6061	0.5714	0.5882	35
Angry	0.6216	0.5897	0.6053	39
Sad	0.6061	0.7143	0.6557	28
Relax	0.7273	0.7059	0.7164	34
Macro Average	0.6403	0.6453	0.6414	136

Berdasarkan tabel 9. hasil akurasi yang diperoleh adalah 63.97% dan *Macro Average F1-Score* sebesar 64.14%. Model memperoleh *F1-score* sebesar 0.59 untuk kelas *happy* dengan *precision* 0.61 dan *recall* 0.57. *F1-Score* 0.61 untuk kelas *angry* dengan *precision* 0.62 dan *recall* 0.59. *F1-Score* 0.66 pada kelas *sad* dengan *precision* 0.61 serta *recall* 0.71. *F1-Score* 0.72 untuk kelas *relax* dengan *precision* 0.73 dan *recall* 0.71. Kelas *relax* memiliki *F1-score* tertinggi sedangkan kelas *sad* memiliki *recall* tertinggi namun *precision* yang rendah, mengindikasikan bahwa model cenderung terlalu sering memprediksi kelas *sad* sehingga menghasilkan lebih banyak *false positive*. Sebaliknya, kelas *happy* memiliki nilai *recall* dan *precision* yang sama-sama rendah, menjadikannya kelas dengan performa terburuk dan mengindikasikan model kesulitan dalam mengenali karakteristik emosi tersebut. Gambar 4 menunjukkan visualisasi *confusion matrix* dari kedua metode *pooling*.

p-ISSN: 2301-5373
e-ISSN: 2654-5101



Gambar 4. Confusion Matrix dengan Mean Pooling (Kiri) dan Attention Pooling (Kanan)

4. Kesimpulan

Berdasarkan hasil evaluasi yang diperoleh, model LSTM dengan *Attention Pooling* menunjukkan performa yang lebih baik dibandingkan model LSTM dengan *Mean Pooling* pada seluruh metrik evaluasi. Peningkatan performa terlihat pada nilai *F1-score* yang meningkat dari 64.14% menjadi 66.81%, precision dari 64.03% menjadi 68.24%, recall dari 65.34% menjadi 66.55%, serta akurasi dari 63.97% menjadi 66.91%. Hasil ini menunjukkan bahwa mekanisme *attention* mampu menghasilkan representasi fitur yang lebih informatif dengan memberikan bobot yang berbeda pada setiap *hidden state* yang dihasilkan oleh LSTM. Dengan demikian, informasi yang lebih relevan terhadap proses klasifikasi dapat memperoleh kontribusi yang lebih besar dibandingkan informasi yang kurang penting. Kemampuan tersebut memungkinkan model untuk menangkap pola temporal yang lebih kompleks sehingga menghasilkan prediksi yang lebih akurat dibandingkan pendekatan *Mean Pooling* yang hanya melakukan agregasi secara merata terhadap seluruh timestep. Berdasarkan ekstraksi fitur yang digunakan, MFCC terbukti memberikan performa yang lebih baik dibandingkan Log-Mel Spectrogram ketika digunakan sebagai input pada model LSTM dengan *Attention Pooling*. Meskipun MFCC hanya menggunakan 25 fitur sedangkan Log-Mel Spectrogram menggunakan 128 fitur, MFCC mampu menghasilkan akurasi sebesar 66.91%, sementara Log-Mel Spectrogram memperoleh akurasi sebesar 61.76%. Selain itu, nilai *F1-score* yang dihasilkan MFCC juga lebih tinggi dengan selisih sebesar 5.15%. Hasil ini memperlihatkan bahwa dimensi fitur yang lebih rendah berpotensi mengurangi informasi yang redundan atau kurang relevan. Oleh karena itu, MFCC dapat dianggap sebagai representasi fitur yang lebih efektif untuk dataset yang digunakan dalam penelitian ini, terutama ketika dikombinasikan dengan mekanisme *Attention Pooling* pada arsitektur LSTM.

References

[1] Retta, E. A., Almekhlafi, E., Sutcliffe, R., Mhamed, M., Ali, H., & Feng, J. (2023). A new amharic speech emotion dataset and classification benchmark. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(1), 1–22. https://doi.org/10.1145/3529759

[2] Valadares, P. B., Rosero Jácome, K. G., dos Santos, A. N., & Sanches Masiero, B. (2022). Song emotion recognition: a performance comparison between audio features and artificial neural networks. *Revista Eletrônica de Iniciação Científica Em Computação*, *20*(4), 55–62. https://doi.org/10.5753/reic.2022.2766

[3] M. Fernández-Díaz and A. Gallardo-Antolín, “An attention Long Short-Term Memory based system for automatic classification of speech intelligibility,” Engineering Applications of Artificial Intelligence, vol. 96, p. 103976, Nov. 2020, doi: 10.1016/j.engappai.2020.103976.

[4] Coutinho, E., Deng, J., & Schuller, B. (2014). Transfer learning emotion manifestation across music and speech. *2014 International Joint Conference on Neural Networks (IJCNN)*, 3592–3598. https://doi.org/10.1109/ijcnn.2014.6889814

[5] H.-T. Hung, J. Ching, S. Doh, N. Kim, J. Nam, and Y.-H. Yang, “EMOPIA: A Multi-Modal Pop Piano Dataset for Emotion Recognition and Emotion-Based Music Generation,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 318–325.
[6] R. Mignot and G. Peeters, “An Analysis of the Effect of Data Augmentation Methods: Experiments for a Musical Genre Classification Task,” Transactions of the International Society for Music Information Retrieval, vol. 2, no. 1, pp. 97–110, 2019, doi: 10.5334/tismir.26.
[7] V. Franzoni, G. Biondi, and A. Milani, “Emotional sounds of crowds: spectrogram-based analysis using deep learning,” Multimedia Tools and Applications, vol. 79, no. 47–48, pp. 36063–36075, Aug. 2020, doi: 10.1007/s11042-020-09428-x.
[8] R. Delbouys, R. Hennequin, F. Piccoli, J. Royo-Letelier, and M. Moussallam, “Music Mood Detection on Audio and Lyrics with Deep Neural Network,” in Proceedings of the 19th ISMIR Conference, pp. 370–375.
[9] J. Grekow, “Music emotion recognition using recurrent neural networks and pretrained models,” Journal of Intelligent Information Systems, vol. 57, pp. 531–546, doi: https://doi.org/10.1007/s10844-021-00658-5.
[10] I. D. G. y A. Wibawa and I. D. M. B. A. Darmawan, “Implementation of audio recognition using mel frequency cepstrum coefficient and dynamic time warping in wirama praharsini,” Journal of Physics: Conference Series, vol. 1722, no. 1, p. 012014, Jan. 2021, doi: 10.1088/1742-6596/1722/1/012014.
[11] S. P. Dewi, A. L. Prasasti, and B. Irawan, “Analysis of LFCC Feature Extraction in Baby Crying Classification using KNN,” in 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), IEEE, Nov. 2019, pp. 86–91. Available: https://doi.org/10.1109/iotais47347.2019.8980389
[12] S.-H. Noh, “Analysis of Gradient Vanishing of RNNs and Performance Comparison,” *Information*, vol. 12, no. 11, p. 442, Oct. 2021, doi: 10.3390/info12110442.
[13] B. H. Pansambal and A. B. Nandgaokar, “Integrating Dropout Regularization Technique at Different Layers to Improve the Performance of Neural Networks,” International Journal of Advanced Computer Science and Applications, vol. 14, no. 4, 2023, doi: 10.14569/ijacsa.2023.0140478.
[14] I. Salehin and D.-K. Kang, “A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain,” Electronics, vol. 12, no. 14, p. 3106, Jul. 2023, doi: 10.3390/electronics12143106.
[15] Y. Wu and L. Liu, “Selecting and Composing Learning Rate Policies for Deep Neural Networks,” *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 2, pp. 1–25, Feb. 2023, doi: 10.1145/3570508.
[16] M. Reyad, A. M. Sarhan, and M. Arafa, “A modified Adam algorithm for deep neural network optimization,” *Neural Computing and Applications*, vol. 35, no. 23, pp. 17095–17112, Apr. 2023, doi: 10.1007/s00521-023-08568-z.
[17] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical Attention Networks for Document Classification,” in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 1480–1489. Available: https://doi.org/10.18653/v1/n16-1174
[18] H. Nam and Y.-H. Park, “Temporal Attention Pooling for Frequency Dynamic Convolution in Sound Event Detection,” arXiv preprint arXiv:2504.12670, 2025.
[19] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, “End-To-End Memory Networks,” in NIPS’15: Proceedings of the 29th International Conference on Neural Information Processing Systems, pp. 2440–2448. Available: https://doi.org/10.48550/arXiv.1503.08895.
[20] F. Kamalov, F. Thabtah, and H. H. Leung, “Feature Selection in Imbalanced Data,” *Annals of Data Science*, vol. 10, no. 6, pp. 1527–1541, Jan. 2022, doi: 10.1007/s40745-021-00366-5.
[21] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
[22] Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061
[23] A. Deep Attentive Network authors, "A deep attentive network for music emotion recognition

using the chorus part of the music," *Research Square*, 2023. [Online]. Available: https://www.researchsquare.com/article/rs-3130450/v1

[24] G. Banerjee, E. Huang, and A. Lettiere, "Understanding emotion classification in audio data," Stanford University. Accessed: Jul. 08, 2026. [Online]. Available: https://web.stanford.edu/class/cs224n/reports/final_reports/report043.pdf

[25] Jorrin-Coz, J., Nakano, M., Perez-Meana, H., & Hernandez-Gonzalez, L. (2025). Multi-Corpus benchmarking of CNN and LSTM models for speaker gender and age profiling. *Computation*, *13*(8), 177. https://doi.org/10.3390/computation13080177