Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, **Number** 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

Development of a Hybrid ConvNeXt-BiLSTM Model Based on Segment-Level Mel-Spectrograms for AI-Generated Song Detection

Dimas Aditya Saputra[1*], Nugroho Dwi Saputro[2], Ramadhan Renaldy[3]
[1,2,3]Informatics Engineering, Universitas PGRI Semarang, Indonesia
[1*]dimas.adit.sap@gmail.com, [2]nugputra@upgris.ac.id, [3]ramadhanrenaldy@upgris.ac.id

open access

***Corresponding Author**

Article History:
Submitted: 24-07-2026
Accepted: 01-08-2026
Published: 07-08-2026

Keywords:
AI-generated music; audio classification; bidirectional long short-term memory; ConvNeXt; mel-spectrogram.

ABSTRACT

AI-generated music increasingly resembles human-produced songs, creating a need for reliable source detection in audio classification. This study aims to develop and evaluate a segment-level hybrid model that combines ConvNeXt with bidirectional long short-term memory (BiLSTM) to capture local spectral patterns and temporal relationships across a song. Each song is represented by 24 five-second segments. Every segment is converted into a 128 x 128 *mel-spectrogram*, processed by ConvNeXt to obtain 768 features, and arranged as a sequence for BiLSTM modeling. The base dataset contains 22,844 songs divided into 16,522 training, 877 validation, and 5,445 test songs. Under the same data split and training protocol, the proposed ConvNeXt-BiLSTM model achieved accuracy of 0.8408, balanced accuracy of 0.8450, sensitivity of 0.7002, and an F1-score of 0.8190. The ConvNeXt-only *baseline* achieved 0.8228, 0.8276, 0.6613, and 0.7934 for the same metrics. *Fine-tuning* used 3,200 additional balanced samples, while final evaluation used a separate balanced test set of 200 songs. With a calibrated threshold of 0.484, the fine-tuned model achieved an F1-score of 0.8556 and balanced accuracy of 0.8700, correctly identifying 97 non-AI songs and 77 AI-generated songs. These results show that temporal modeling across *mel-spectrogram* segments improves detection, although evaluation using additional music generators, codecs, recording qualities, and adaptive audio contexts remains necessary.

INTRODUCTION

Artificial intelligence (AI) generated songs increasingly resemble human productions. MusicLM generates minutes of music from text (Agostinelli et al., 2023), while AudioLDM (H. Liu et al., 2023), Stable Audio Open (Evans et al., 2024), MusicGen (Copet et al., 2023), and AudioLDM 2 (H. Liu et al., 2024) use latent representations, audio tokens, or pretraining. Their originality, copyright, attribution, trust, and security implications require reliable source authentication (Khanjani, Watson, & Janeja, 2023). However, synthetic-audio detection has focused mainly on speech: ASVspoof 2021 evaluated channel and compression variations (Yamagishi et al., 2021), while ADD 2022 included low-quality and partially fake audio (Yi et al., 2022). Because songs contain instruments, harmony, dynamics, and long structures, SingFake (Zang et al., 2024), FSD (Xie et al., 2023), and SONICS (Rahman et al., 2024) indicate that speech detectors may not transfer effectively to complete songs. This mismatch motivates a representation that preserves long-range structure and local time-frequency cues.

Mel-spectrograms map frequency energy over time. The Audio Spectrogram Transformer achieved mean average precision of 0.485 on AudioSet and accuracies of 95.6% on ESC-50 and 98.1% on Speech Commands V2 (Gong, Chung, & Glass, 2021b). Song-based FSD training reduced average equal error rate by 38.58% relative to speech training (Xie et al., 2023). In *The AI Music Arms Race*, an SVM obtained F1-scores of 0.969 for AI and 0.935 for non-AI, while IRCAM Amplify reached 0.988 and 0.976 (Vila et al., 2025). FakeMusicCaps reported 0.90 for M5 and 0.88 for RawNet2 in closed-set testing, but only 0.48–0.56 for an SVM in open-set testing (Comanducci et al., 2025), confirming that performance in one setting does not ensure cross-generator generalization (Almutairi & Elgibreen, 2022).

This study therefore uses segment-level mel-spectrograms to learn local patterns and their sequence, following Rahman et al. (2024), Vila et al. (2025), and Comanducci et al. (2025). ConvNeXt extracts segment features, while bidirectional long short-term memory (BiLSTM) models both temporal directions; ConvNeXt-only serves as the baseline and ConvNeXt-BiLSTM as the proposed model (Z. Liu et al., 2022). Accuracy, balanced accuracy, precision, sensitivity, specificity, F1-score, and a confusion matrix assess its potential use in music platforms, digital archives, and audio forensics. The study first tests whether BiLSTM improves ConvNeXt-only detection under identical splits, preprocessing, training rules, and thresholds, then evaluates the adapted hybrid on a separate balanced test set. Keeping these evaluations distinct clarifies the contribution and prevents cross-protocol scores from being treated as directly comparable.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, Number 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

METHOD

This study adapts the Cross-Industry Standard Process for Data Mining (CRISP-DM), covering data understanding, preparation, modeling, evaluation, and reporting (Schroer, Kruse, & Gomez, 2021), with the experimentation practices of Pfob, Lu, & Sidey-Gibbons (2022). Both models use the same base-data split, and only ConvNeXt-BiLSTM is *fine-tuned* on additional data. Its 24 five-second segments retain local artifacts and temporal relationships across 120 seconds, as studied by Rahman et al. (2024), Vila et al. (2025), and Comanducci et al. (2025). This controlled design makes the added temporal layer, rather than a change in split or preprocessing, the main explanatory factor in the base comparison.

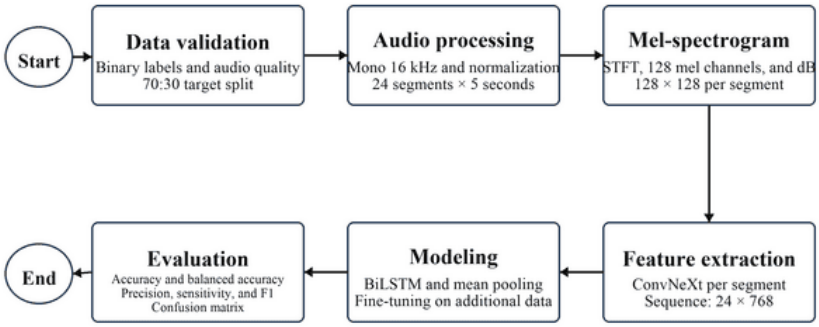

Figure 1. Research workflow

Figure 1 summarizes the workflow. Song labels, audio quality, and the 70:30 target split are validated at the song level. Audio is converted to mono at 16 kHz, normalized, and divided into 24 five-second segments. The Short-Time Fourier Transform (STFT) maps each segment to 128 mel channels on a decibel scale and a 128 × 128 input. ConvNeXt produces 768 features per segment, while BiLSTM and *mean pooling* produce 512 song-level features. Both architectures use the same base data, and only ConvNeXt-BiLSTM is *fine-tuned* on additional data before final evaluation.

Data Collection

The Kaggle base data are divided into non-AI and AI-generated songs, with one complete song as the observation unit. Following Pfob et al. (2022), metadata retain identity, duration, label, experimental group, and source for traceability. The AI class contains 7,818 Suno songs and 3,635 Udio songs, while the prediction target remains binary.

Table 1. Research data summary

Data source	Category	Number of songs	Description
Non-AI songs	Non-AI	11,391	Songs representing human-created works
AI-generated songs	AI	11,453	Songs generated by generative systems
Total	All classes	22,844	Research dataset

Table 1 summarizes 22,844 songs, comprising 11,391 non-AI and 11,453 AI-generated songs. The difference of 62 songs indicates a relatively balanced composition, so the study does not need to correct an extreme class imbalance. Both classes are therefore retained without resampling in the base experiment.

Data Processing

Using Torchaudio (Yang et al., 2022), audio is converted to mono at 16 kHz, normalized, and divided into 24 five-second segments. Training uses random segment positions for contextual variation, while validation and testing use deterministic positions. Audio shorter than 120 seconds is zero-padded. The procedure introduces positional variation only during learning while keeping validation and test inputs stable for reproducible architecture comparison.

Table 2. Data-processing stages

Stage	Input	Output	Parameter	Purpose
Audio loading	Digital song	Mono signal	Mono	Standardize audio channels
Sample-rate standardization	Audio signal	16 kHz signal	16 kHz	Standardize the time scale
Amplitude normalization	Audio signal	Normalized signal	Standard deviation	Reduce amplitude variation
Segmentation	Complete song	24 segments	5 seconds per segment	Form a 120-second audio context
Model-input assembly	Audio segments	Audio segment sequence	24 segments	Form segment-level input

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, Number 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

Table 2 shows that each five-second segment contains 80,000 samples at 16 kHz. Across 24 segments, the model receives local information from a 120-second song context while the evaluation unit remains one complete song. This representation keeps segment duration and song-level evaluation consistent across both architectures.

Feature Extraction

Each segment becomes a mel-spectrogram through STFT and mel-scale mapping, consistent with Almutairi & Elgibreen (2022), Yang et al. (2022), Zaman, Sah, Direkoglu, & Unoki (2023), and Gong, Chung, & Glass (2021a). The configuration uses a 2,048-point fast Fourier transform, a 2,048-sample window, a hop length of 512, 128 mel channels, and 20–8,000 Hz. Values are converted to decibels, normalized by mean and standard deviation, and standardized to 128 × 128 following Chen et al. (2022), Koutini, Schluter, Eghbal-zadeh, & Widmer (2022), Tak, Kamble, Patino, Todisco, & Evans (2022), Jung et al. (2022), and (Tak et al., 2021).

Data Labeling

Labels follow the data source. Non-AI songs receive class 0 and AI-generated songs class 1, with no additional manual labeling. Consistency is verified across source category, class value, and descriptive metadata following Pfob et al. (2022). The target is therefore auditable because each assigned class can be traced to an explicit source category instead of a subjective listening judgment.

Table 3. Data-labeling scheme

Class category	Class value	Operational definition	Label basis
Non-AI	0	Song representing a human-created work	Source song category
AI-generated	1	Song created or synthesized by a generative system	Source song category

Table 4. Label-consistency checks

Check	Number of items	Expected class value	Matched	Mismatched
Non-AI label	11,391	0	11,391	0
AI-generated label	11,453	1	11,453	0
Total	22,844	–	22,844	0

Table 3 defines the two classes, and Table 4 confirms consistency among source categories, values, and metadata. All 22,844 entries match their categories and can be used to train and evaluate the binary classifier. The provenance-based procedure therefore produces a reproducible binary target without adding subjective relabeling.

Data Validation

Before training, validation checks metadata, labels, splits, audio availability and readability, and cross-group overlap. These checks reduce *data leakage* risk according to Kapoor & Narayanan (2023) and follow the experimentation practices of Pargent, Schoedel, & Stachl (2023). The validation results are documented before model fitting so that data-quality findings remain separate from performance claims.

Table 5. Data-validation results

Dataset split	Number of songs	Technical errors	Label errors	Duplicate files across splits
Training	16,522	0	0	0
Validation	877	0	0	0
Test	5,445	0	0	0
Total	22,844	0	0	0

Table 5 shows no technical errors, inconsistent labels or splits, or identical audio across groups. The audit found 384 generative source IDs with different variants in validation and test data. They are not identical files and do not occur in training, but remain a limitation because the evaluation groups may be overly similar.

Data Balancing

The training data contain 8,356 non-AI songs (50.57%) and 8,166 AI-generated songs (49.43%). Because this difference is small, *oversampling* is not applied, preserving the original distribution and avoiding duplication bias as discussed by Ghosh et al. (2024) and Tarawneh, Hassanat, Altarawneh, & Almuhaimeed (2022). The same unaltered class counts are used by both initial architectures, supporting a fair comparison.

Table 6. Training data before and after oversampling

Class	Before oversampling	After oversampling	Method
Non-AI	8,356	8,356	Not applied
AI-generated	8,166	8,166	Not applied

Brilliance
Research of Artificial Intelligence
E-ISSN : 2807-9035
Volume 6, Number 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

Table 6 confirms that no songs are added through *oversampling*. Random training segments only vary their positions and do not duplicate songs, while deterministic validation and test positions support reproducible evaluation. Consequently, model differences cannot be attributed to unequal oversampling between the two architectures.

Data Splitting

Song-level splitting prevents segments from one song from entering different groups, following Kapoor & Narayanan (2023), Pargent et al. (2023), and Rainio, Teuho, & Klen (2024). The target training-to-nontraining ratio is 70:30. After preserving source groups and canonical manifests, the actual proportions are 72.33% training, 3.84% validation, and 23.84% testing. Although the realized proportions differ from the nominal target, the fixed split is shared by both architectures and remains suitable for controlled comparison.

Table 7. Research data split

Split	Non-AI	AI-generated	Total	Ratio
Training	8,356	8,166	16,522	72.33%
Validation	392	485	877	3.84%
Test	2,643	2,802	5,445	23.84%
Total	11,391	11,453	22,844	100%

Table 7 shows both classes in every split. The 5,445-song test set is excluded from training and model selection, and all segments from one song remain in the same split. Additional data are separated specifically for *fine-tuning*, validation, and testing, as shown in Table 8.

Table 8. Additional data split

Use	Non-AI	AI-generated	Total
Fine-tuning	1,600	1,600	3,200
Additional validation	40	40	80
Additional test	100	100	200

Table 8 shows balanced additional data. File locations, source groups, YouTube IDs, and SHA-256 hashes have zero cross-group overlap. The additional test set is excluded from training, checkpoint selection, and threshold selection.

Model Training and Fine-Tuning

ConvNeXt Tiny converts each 128 × 128 mel-spectrogram into 768 features, forming a 24 × 768 sequence (Z. Liu et al., 2022). The *baseline* averages this sequence before dropout at 0.2 and classification. The proposed model uses one BiLSTM layer with 256 units per direction to produce 24 × 512 features, followed by *mean pooling*, dropout at 0.2, and classification. This combination is supported by Zhen & Changhui (2021), Ashraf et al. (2023), Muruganandham, Thangasamy, Jayaraman, & Dharmarajan (2025), and Wani, Qadri, Comminiello, & Amerini (2024).

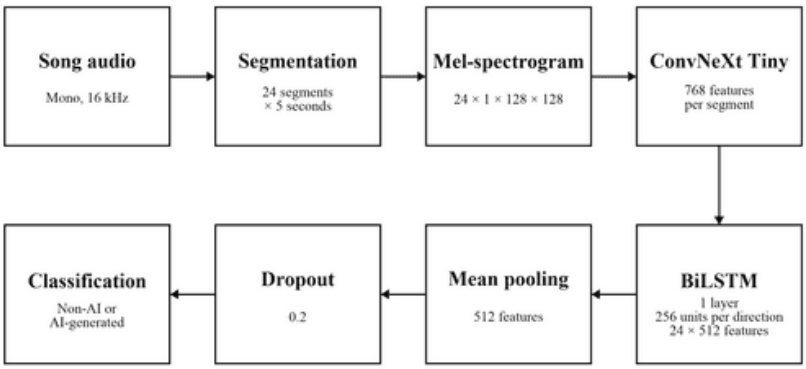

Figure 2. Proposed ConvNeXt-BiLSTM architecture

Figure 2 shows ConvNeXt converting each 128 × 128 mel-spectrogram into 768 features and a 24 × 768 song sequence. BiLSTM models bidirectional intersegment relationships and returns 24 × 512 features. Mean pooling then produces 512 song-level features for dropout at 0.2 and binary classification.

Initial training uses seed 42, random weights, AdamW (Zhuang, Liu, Cutkosky, & Orabona, 2022), learning rate 0.0005, weight decay 0.05, batch size 4, BCEWithLogitsLoss, and label smoothing 0.02 (Zhang et al., 2021). A cosine scheduler, maximum gradient norm of 5.0, and time-frequency masking probability of 0.5 are applied. Early stopping monitors validation F1 with patience 10 and minimum delta 0.002. ConvNeXt-BiLSTM is fine-tuned on 3,200 samples using seed 45, learning rate 0.000003, batch size 4, reset optimization states, and the same stopping rule. Checkpoint

Brilliance
Research of Artificial Intelligence
E-ISSN : 2807-9035
Volume 6, Number 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

selection uses mean F1 across the additional and base validation sets.

Model Evaluation

The best validation-F1 checkpoint is evaluated using accuracy, balanced accuracy, precision, sensitivity, specificity, F1-score, and a confusion matrix, following Ghosh et al. (2024) and Rainio et al. (2024). AI-generated songs are positive. True positive (TP) and true negative (TN) are correct AI and non-AI predictions, while false positive (FP) is non-AI predicted as AI and false negative (FN) is undetected AI. Balanced accuracy and F1-score complement raw accuracy by exposing performance across both classes and the trade-off between missed AI songs and false alarms.

Final Visualization and Validation

Final validation links the workflow, data distribution, signal and feature representations, architecture, validation curve, performance comparison, and confusion matrix under CRISP-DM (Schroer et al., 2021). The audit covers counts, file separation, metadata similarity, audio parameters, binary metrics, and the decision not to oversample (Tarawneh et al., 2022). Together, these checks connect each reported result to a documented preprocessing, training, or evaluation artifact.

RESULTS

The results distinguish the comparison between the baseline and the proposed model on the base test data from the evaluation of ConvNeXt-BiLSTM after *fine-tuning* on additional data. This separation is necessary because the two evaluations use different datasets and decision thresholds. Results are therefore interpreted within their own protocols before broader implications are discussed.

Research Data Results

The base dataset contains 22,844 songs, comprising 11,391 non-AI songs and 11,453 AI-generated songs. The additional data are balanced for *fine-tuning*, validation, and testing. These counts define the common comparison set for ConvNeXt-only and ConvNeXt-BiLSTM before fine-tuning.

Table 9. Experimental data summary

Dataset group	Non-AI	AI-generated	Total
Training	8,356	8,166	16,522
Validation	392	485	877
Test	2,643	2,802	5,445
Fine-tuning	1,600	1,600	3,200
Additional validation	40	40	80
Additional test	100	100	200

Table 9 distinguishes the base data used to compare the architectures from the additional data used for *fine-tuning*. The 3,200 additional training samples include replay and codec variants, not 3,200 entirely unique new songs. Validation data are used for model selection, while test data are used only for final evaluation. Audits of files, source groups, YouTube IDs, and SHA-256 hashes show zero overlap across groups.

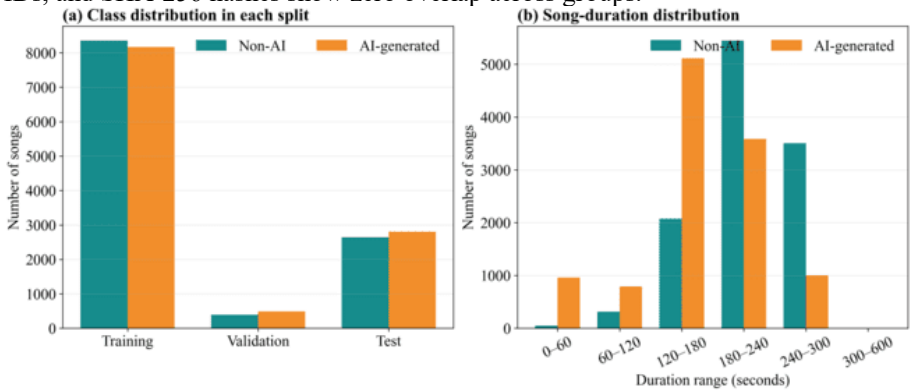

Figure 3. Class distribution and song duration

Figure 3(a) shows that the class counts in the training, validation, and test sets are relatively balanced. Figure 3(b) shows that song durations are distributed differently across the two classes. The duration chart only describes dataset characteristics and is not used as a labeling rule. Because the model input must be uniform, every song is still represented by a 120-second context.

Brilliance
Research of Artificial Intelligence
E-ISSN : 2807-9035
Volume 6, Number 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

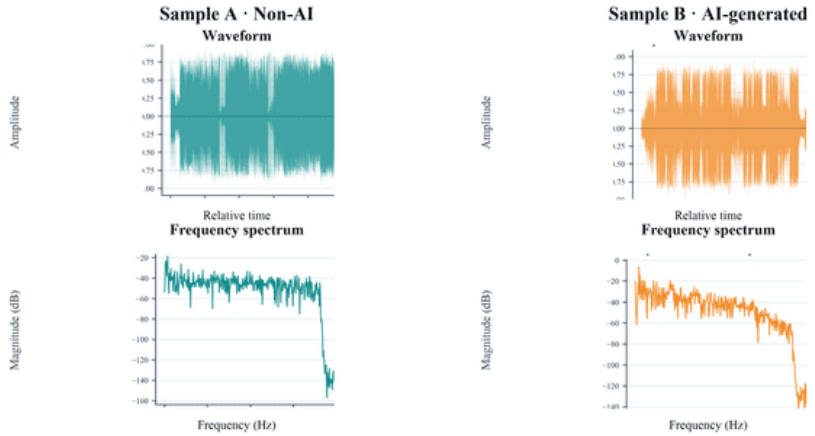

Figure 4. Waveforms and frequency spectra of non-AI and AI-generated songs

In Figure 4, the Sample A non-AI waveform shows relatively dense amplitude throughout the song with several local decreases, while the Sample A non-AI spectrum contains energy up to nearly 8 kHz. The Sample B AI-generated waveform varies more strongly and weakens near the end, while the Sample B AI-generated spectrum shows decreasing energy as frequency increases. The 8 kHz boundary results from the 16 kHz sample rate and is not a class characteristic. These two examples only illustrate the data representation and do not define a rule for identifying AI-generated songs.

Audio-Processing Results

Table 10 shows that audio is standardized to mono at 16 kHz, normalized, and divided into 24 five-second segments. Multiplying 16,000 samples per second by five seconds produces 80,000 values in each segment. One song therefore provides a 120-second context in a consistent input representation. The identical representation ensures that later performance differences arise after preprocessing rather than from unequal sample rates, segment durations, or song contexts.

Table 10. Audio-processing results

Stage	Input	Output	Purpose
Conversion	Original audio	Mono 16 kHz	Standardize channels and sample rate
Normalization	Amplitude	Standard deviation 1	Standardize the amplitude scale
Segmentation	One song	24 × 5 seconds	Form a 120-second context
Transformation	80,000 values	128 × 128 mel-spectrogram	Form ConvNeXt input

For Sample A, the amplitude standard deviation (SD) changes from 0.13812 to 1.00000, while the mean remains close to zero, as shown in Figure 5. This comparison verifies that normalization changes the amplitude scale rather than the relative timing of waveform peaks. The paired display is therefore used to inspect preprocessing behavior before feature extraction.

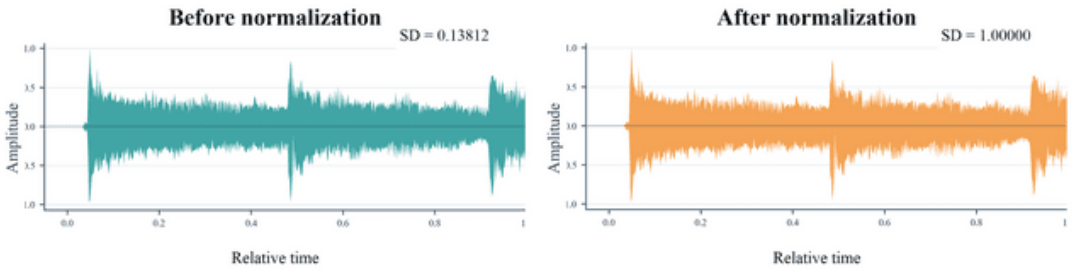

Figure 5. Signal before and after normalization

Figure 5 shows that the amplitude standard deviation changes from 0.13812 to 1.00000, while the mean remains close to zero. The peak positions and waveform sequence also remain unchanged. Normalization therefore standardizes the input scale without changing the audio content or class label.

Feature-Extraction Results

Table 11 traces how one song is transformed into numerical features. The number 24 always denotes the number of segments, while 768 and 512 denote the feature dimensions learned by ConvNeXt and BiLSTM. These numbers do not represent frequency, tempo, or the number of classes.

Brilliance
Research of Artificial Intelligence
E-ISSN : 2807-9035
Volume 6, Number 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

Table 11. Data-shape changes during feature extraction

Stage	Input	Output	Meaning
Segmentation	One song	24 × 80,000	Audio signal sequence
Mel-spectrogram	24 × 80,000	24 × 128 × 128	Time-frequency energy
ConvNeXt	24 × 128 × 128	24 × 768	Features for each segment
BiLSTM	24 × 768	24 × 512	Intersegment relationships
Mean pooling	24 × 512	1 × 512	Song-level features

The tensor changes in Table 11 establish the dimensions used in the visual comparison. Figure 6 then places one non-AI and one AI-generated representation side by side. The comparison is descriptive and does not treat a single color pattern as a decision rule.

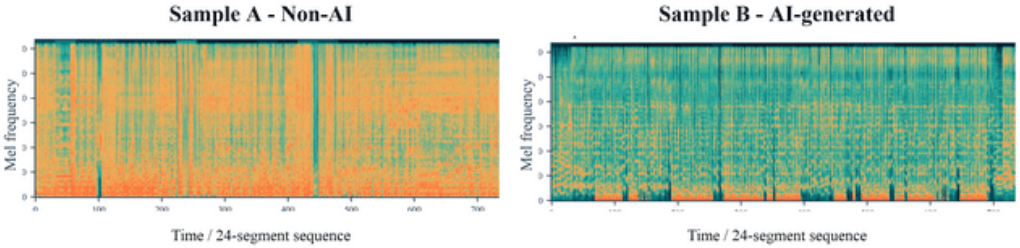

Figure 6. Mel-spectrogram comparison between non-AI and AI-generated songs

Figure 6 compares one non-AI mel-spectrogram with one AI-generated example. The horizontal axis shows the temporal sequence of 24 segments, the vertical axis shows mel frequency, and color represents relative energy intensity. Brighter colors indicate higher energy. Differences between the two examples are illustrative and do not define a fixed class rule. The model bases its decision on features combined across all segments rather than on one color or one region.

Table 12. Dimensions and sample numerical features

Stage	Output	First four values	Mean	SD
ConvNeXt	24 × 768	0.024 / 0.400 / 0.068 / 0.085	−0.001	0.334
BiLSTM	24 × 512	−0.017 / 0.214 / 0.001 / −0.033	−0.005	0.452
Mean pooling	1 × 512	−0.643 / 0.129 / −0.022 / −0.035	−0.005	0.411

Table 12 contains examples of activations learned by the model, not frequency, tempo, volume, or class-probability values. ConvNeXt produces 768 features per segment, BiLSTM produces 512 features per segment, and *mean pooling* summarizes 24 segments into 512 song-level features. The first four values show only a small subset of the features. The mean and standard deviation describe the center and spread of the activations at each stage.

The staged values in Table 12 provide a numerical trace of the learned representation. Figure 7 visualizes the same flow from mel-spectrograms to pooled song-level features. It is a feature-flow illustration rather than a class-probability plot.

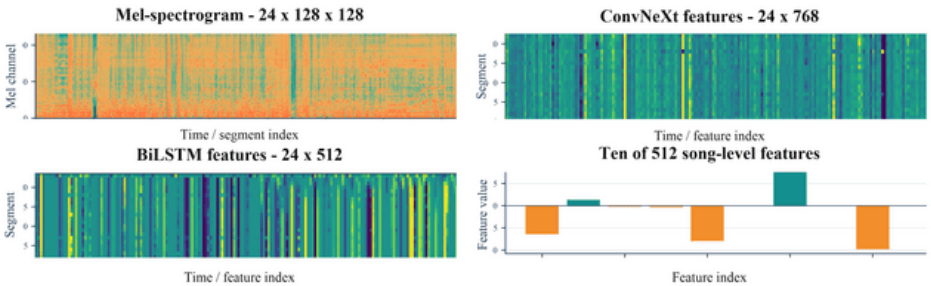

Figure 7. Transformation of audio into numerical features

Figure 7 shows the transformation from mel-spectrograms to ConvNeXt and BiLSTM features. ConvNeXt produces 768 features for each segment, after which BiLSTM organizes 512 features per segment according to their sequential relationships. The final chart displays only 10 of the 512 song-level features obtained after *mean pooling*. All 512 features are still passed to the classification layer. The plotted values are model activations, not class probabilities.

Training Results and Model Selection

With early stopping, the baseline stops at epoch 20, with its best checkpoint at epoch 10. The proposed model

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, **Number** 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

stops at epoch 24, with its best checkpoint at epoch 14. Fine-tuning stops at epoch 26, with its best checkpoint at epoch 16. These selected checkpoints, rather than the final stopping epochs, supply the weights used for the reported test metrics.

Table 13. Training summary and best checkpoints

Phase	Model	Learning rate	Best epoch	Test F1-score
Initial training	ConvNeXt-only baseline	0.0005	10	0.7934
Initial training	ConvNeXt-BiLSTM	0.0005	14	0.8190
Fine-tuning	ConvNeXt-BiLSTM	0.000003	16	0.8556

Table 13 identifies the best checkpoint epoch rather than the total number of training epochs. The baseline checkpoint is selected at epoch 10, the proposed-model checkpoint at epoch 14, and the *fine-tuning* checkpoint at epoch 16. The fine-tuned F1-score comes from the additional test data and is therefore not directly compared with the two initial-training results obtained from the base test data.

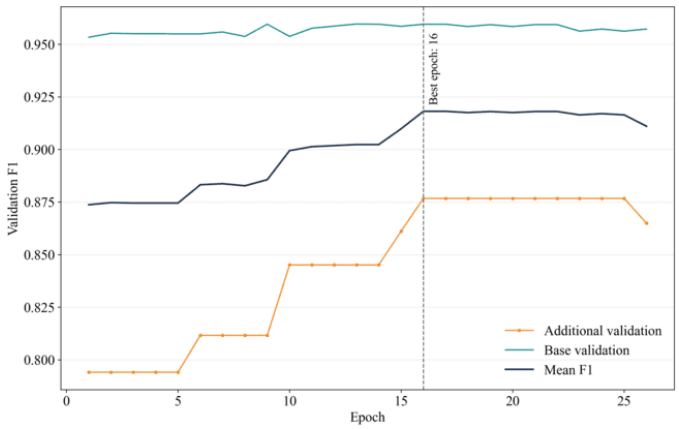

Figure 8. Validation F1 during fine-tuning

Figure 8 shows the progression of F1 during *fine-tuning*. The orange line denotes additional validation, the green line denotes base-data validation, and the blue line denotes their mean. The dashed line marks the best checkpoint at epoch 16. The mean of the two validation results is used so that model selection does not depend on a single data source and continues to preserve performance on the base data. After epoch 16, the validation mean does not surpass the value at that checkpoint.

Baseline and Proposed Model Comparison

The comparison uses the same 5,445 base test songs, a threshold of 0.5, data split, and training rules, allowing the two models to be compared directly. This controlled setting isolates the architectural comparison from changes in the evaluation set. The fine-tuned result is reported separately because it uses additional data and a calibrated threshold.

Table 14. Performance comparison on the base test data

Metric	Baseline	Proposed model
Accuracy	0.8228	0.8408
Balanced accuracy	0.8276	0.8450
Precision	0.9914	0.9864
Sensitivity	0.6613	0.7002
Specificity	0.9939	0.9898
F1-score	0.7934	0.8190

Table 14 provides a direct comparison because both models use the same 5,445 test songs, a threshold of 0.5, data split, and training protocol. The proposed model improves F1 from 0.7934 to 0.8190 and sensitivity from 0.6613 to 0.7002. The higher sensitivity indicates that more AI-generated songs are detected, at the cost of slightly lower precision and specificity than the baseline. Operationally, this trade-off favors detecting more AI-generated songs while introducing only a small increase in non-AI false alarms.

Brilliance
Research of Artificial Intelligence
E-ISSN : 2807-9035
Volume 6, Number 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

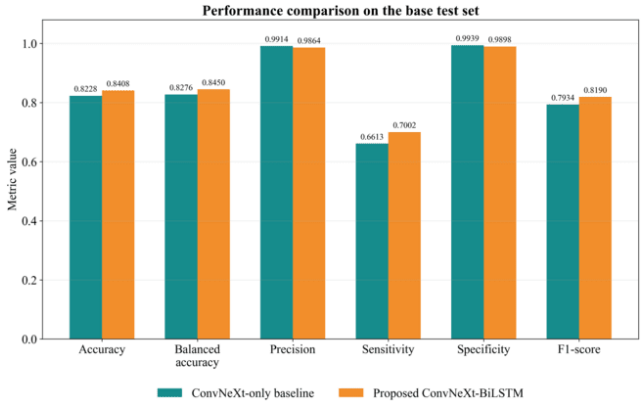

Figure 9. Performance comparison between the baseline and proposed model

Figure 9 compares six metrics on the 5,445 base test songs. The proposed model achieves higher accuracy, balanced accuracy, sensitivity, and F1-score, while the baseline achieves slightly higher precision and specificity. F1 increases by 0.0256 and sensitivity by 0.0389. The increase in sensitivity means that more AI-generated songs are identified correctly. Each pair of bars is directly comparable because both models use the same test data and protocol.

Results after Fine-Tuning

Fine-tuning uses 3,200 balanced samples, comprising 1,600 non-AI and 1,600 AI samples. Testing uses 200 separate songs, comprising 100 non-AI and 100 Suno AI songs. The threshold of 0.484 is selected from 957 validation items and locked before testing. Because threshold selection uses validation data only, the 200 test songs remain isolated from model and decision-rule selection.

Table 15. Proposed-model results on the additional test data

Metric	Value	Component	Count
Accuracy	0.8700	TN	97
Balanced accuracy	0.8700	FP	3
Precision	0.9625	FN	23
Sensitivity	0.7700	TP	77
Specificity	0.9700		
F1-score	0.8556		

Table 15 shows that 97 of 100 non-AI songs and 77 of 100 AI-generated songs are classified correctly. Three non-AI songs are incorrectly predicted as AI, and 23 AI-generated songs are not detected. Because the two classes are balanced, accuracy and balanced accuracy are both 0.8700. A sensitivity of 0.7700 indicates that the principal weakness remains the 23 undetected AI-generated songs.

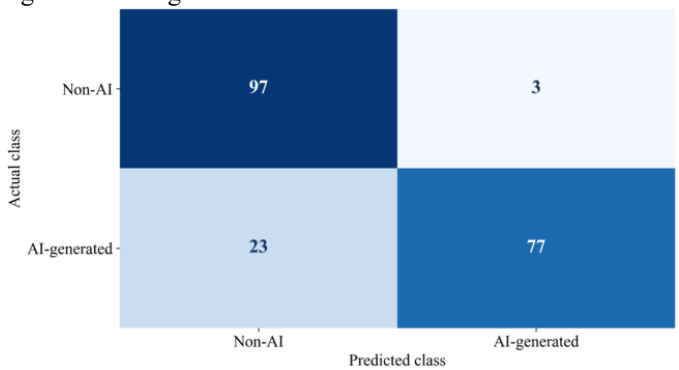

Figure 10. Fine-tuned ConvNeXt-BiLSTM confusion matrix

In Figure 10, rows represent actual classes and columns represent predicted classes. After *fine-tuning*, the model correctly classifies 97 non-AI songs and 77 AI-generated songs. Three non-AI songs are incorrectly predicted as AI, while 23 AI-generated songs remain undetected. This matrix represents only ConvNeXt-BiLSTM on the 200 additional test songs and is therefore not directly compared with the base test data, which use a different protocol.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 6, **Number** 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

DISCUSSION

On the same 5,445 song base test set, ConvNeXt-BiLSTM improves accuracy from 0.8228 to 0.8408, balanced accuracy from 0.8276 to 0.8450, sensitivity from 0.6613 to 0.7002, and F1-score from 0.7934 to 0.8190. Both architectures use the same split, threshold of 0.5, and training protocol, making the comparison internally fair. The improvement supports using BiLSTM to model relationships among the 24 segment-level ConvNeXt feature vectors. However, this conclusion is limited to the controlled experiment and does not establish that the proposed model is superior across every dataset or generator.

The most meaningful change is the 0.0389 increase in sensitivity, accompanied by a 0.0256 increase in F1-score. Higher sensitivity means that more AI-generated songs are detected, reducing false-negative risk in screening applications. However, precision decreases from 0.9914 to 0.9864 and specificity from 0.9939 to 0.9898, indicating a small trade-off in rejecting non-AI songs. This trade-off has different practical implications: a music platform may prioritize sensitivity to reduce undetected synthetic uploads, whereas copyright or archival workflows may preserve precision to avoid incorrectly flagging human-created works. Because threshold changes redistribute false negatives and false positives, deployment should select the threshold using a representative validation set and document the operational cost of each error type. Consequently, the calibrated threshold of 0.484 is supported only for the evaluated domain and must be revalidated when generators, codecs, or recording conditions change.

The result is consistent with hybrid convolutional-recurrent architectures, in which convolutional layers extract local time-frequency patterns and recurrent layers integrate sequential context, as explored by Ashraf et al. (2023) and Wani et al. (2024). Direct numerical comparison with *The AI Music Arms Race* (Vila et al., 2025) or FakeMusicCaps (Comanducci et al., 2025) would be inappropriate because their datasets, generators, features, and evaluation protocols differ. FakeMusicCaps also reports a substantial decline from closed-set to open-set performance, showing that strong results on known generators do not guarantee cross-generator robustness. Accordingly, this study's defensible contribution is the controlled improvement of ConvNeXt-BiLSTM over its ConvNeXt-only baseline under an identical protocol.

After fine-tuning, the model obtains an F1-score of 0.8556, balanced accuracy of 0.8700, precision of 0.9625, sensitivity of 0.7700, and specificity of 0.9700 on 200 additional test songs. The confusion matrix contains 97 true negatives, 3 false positives, 23 false negatives, and 77 true positives, indicating that missed AI songs remain the dominant error. The threshold of 0.484 was selected from 957 validation items and locked before testing, thereby separating threshold selection from final evaluation. Because this test set and threshold differ from those used in the base comparison, the fine-tuned F1-score represents domain-adapted performance rather than a direct continuation of the 0.8190 base-test score.

Several limitations constrain the scope of these findings. The fixed representation of 24 five-second segments provides 120 seconds of context but may omit information outside the selected segments in longer songs. The base evaluation includes 384 source IDs with distinct variants in the validation and test sets; although the files do not overlap and the IDs are absent from training, these groups may remain more similar than fully source-disjoint sets. Moreover, all 100 AI-generated songs in the additional test set come from Suno, while the 3,200 fine-tuning samples include replay and codec variants rather than 3,200 entirely unique songs. The external result therefore supports performance within the evaluated domain but does not establish generator-agnostic generalization. Future evaluations should include generator-disjoint data from Udio and other systems, broader recording and codec conditions, repeated seeds, and uncertainty estimates.

CONCLUSION

Using 24 mel-spectrogram segments, ConvNeXt-BiLSTM achieves accuracy 0.8408, balanced accuracy 0.8450, sensitivity 0.7002, and F1-score 0.8190 on 5,445 base test songs. After *fine-tuning* on 3,200 samples, it reaches F1-score 0.8556 and balanced accuracy 0.8700 on 200 separate songs, correctly classifying 97 non-AI and 77 AI-generated songs. The matched improvement shows that intersegment temporal modeling helps the classifier detect more AI-generated songs without changing the base split, preprocessing, or decision threshold. The fine-tuned result further demonstrates strong performance after domain adaptation, but it is interpreted separately because the test set and threshold differ from the controlled base comparison. Overall, the study contributes a traceable segment-to-song pipeline and evidence that temporal integration adds value beyond ConvNeXt-only feature averaging.

These conclusions remain bounded by the fixed 120-second representation, the presence of related source variants in validation and test groups, and the Suno-only AI class in the additional test set. Future work should compare attention, audio transformers, and pretrained audio representations, then test generators beyond Suno and Udio, diverse codecs and recording qualities, and longer or adaptive contexts.

REFERENCES

Agostinelli, A., Denk, T. I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., … Frank, C. (2023). *MusicLM: Generating Music From Text*. arXiv. https://doi.org/10.48550/arXiv.2301.11325

E-ISSN : 2807-9035
Volume 6, **Number** 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

Almutairi, Z., & Elgibreen, H. (2022). A Review of Modern Audio Deepfake Detection Methods: Challenges and Future Directions. *Algorithms*, *15*(5), 155. https://doi.org/10.3390/a15050155

Ashraf, M., Abid, F., Din, I. U., Rasheed, J., Yesiltepe, M., Yeo, S. F., & Ersoy, M. T. (2023). A Hybrid CNN and RNN Variant Model for Music Classification. *Applied Sciences*, *13*(3), 1476. https://doi.org/10.3390/app13031476

Chen, K., Du, X., Zhu, B., Ma, Z., Berg-Kirkpatrick, T., & Dubnov, S. (2022). HTS-AT: A Hierarchical Token-Semantic Audio Transformer for Sound Classification and Detection. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 646–650. IEEE. https://doi.org/10.1109/ICASSP43922.2022.9746312

Comanducci, L., Bestagini, P., & Tubaro, S. (2025). FakeMusicCaps: A Dataset for Detection and Attribution of Synthetic Music Generated via Text-to-Music Models. *Journal of Imaging*, *11*(7), 242. https://doi.org/10.3390/jimaging11070242

Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., … Defossez, A. (2023). *Simple and Controllable Music Generation*. arXiv. https://doi.org/10.48550/arXiv.2306.05284

Evans, Z., Parker, J. D., Carr, C. J., Zukowski, Z., Taylor, J., & Pons, J. (2024). *Stable Audio Open*. arXiv. https://doi.org/10.48550/arXiv.2407.14358

Ghosh, K., Bellinger, C., Corizzo, R., Branco, P., Krawczyk, B., & Japkowicz, N. (2024). The Class Imbalance Problem in Deep Learning. *Machine Learning*, *113*(7), 4845–4901. https://doi.org/10.1007/s10994-022-06268-8

Gong, Y., Chung, Y.-A., & Glass, J. (2021a). AST: Audio Spectrogram Transformer. *Interspeech 2021*, 571–575. ISCA. https://doi.org/10.21437/Interspeech.2021-698

Gong, Y., Chung, Y.-A., & Glass, J. (2021b). PSLA: Improving Audio Tagging With Pretraining, Sampling, Labeling, and Aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3292–3306. https://doi.org/10.1109/TASLP.2021.3120633

Jung, J., Heo, H.-S., Tak, H., Shim, H., Chung, J. S., Lee, B.-J., … Evans, N. (2022). AASIST: Audio Anti-Spoofing Using Integrated Spectro-Temporal Graph Attention Networks. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6367–6371. IEEE. https://doi.org/10.1109/ICASSP43922.2022.9747766

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, *4*(9), 100804. https://doi.org/10.1016/j.patter.2023.100804

Khanjani, Z., Watson, G., & Janeja, V. P. (2023). Audio deepfakes: A survey. *Frontiers in Big Data*, *5*, 1001063. https://doi.org/10.3389/fdata.2022.1001063

Koutini, K., Schluter, J., Eghbal-zadeh, H., & Widmer, G. (2022). Efficient Training of Audio Transformers with Patchout. *Interspeech 2022*, 2753–2757. ISCA. https://doi.org/10.21437/Interspeech.2022-227

Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., … Plumbley, M. D. (2023). *AudioLDM: Text-to-Audio Generation with Latent Diffusion Models*. arXiv. https://doi.org/10.48550/arXiv.2301.12503

Liu, H., Yuan, Y., Liu, X., Mei, X., Kong, Q., Tian, Q., … Plumbley, M. D. (2024). AudioLDM 2: Learning Holistic Audio Generation With Self-Supervised Pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *32*, 2871–2883. https://doi.org/10.1109/TASLP.2024.3399607

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11966–11976. IEEE. https://doi.org/10.1109/CVPR52688.2022.01167

Muruganandham, P., Thangasamy, G. R., Jayaraman, S., & Dharmarajan, R. (2025). LSTM Autoencoder Based Parallel Architecture for Deepfake Audio Detection with Dynamic Residual Encoding and Feature Fusion. *Scientific Reports*, *15*(1). https://doi.org/10.1038/s41598-025-08198-6

Pargent, F., Schoedel, R., & Stachl, C. (2023). Best Practices in Supervised Machine Learning: A Tutorial for Psychologists. *Advances in Methods and Practices in Psychological Science*, *6*(3). https://doi.org/10.1177/25152459231162559

Pfob, A., Lu, S.-C., & Sidey-Gibbons, C. (2022). Machine learning in medicine: a practical introduction to techniques for data pre-processing, hyperparameter tuning, and model comparison. *BMC Medical Research Methodology*, *22*(1), 282. https://doi.org/10.1186/s12874-022-01758-8

Rahman, M. A., Hakim, Z. I. A., Sarker, N. H., Paul, B., & Fattah, S. A. (2024). *SONICS: Synthetic Or Not – Identifying Counterfeit Songs*. arXiv. https://doi.org/10.48550/arXiv.2408.14080

Rainio, O., Teuho, J., & Klen, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, *14*(1), 6086. https://doi.org/10.1038/s41598-024-56706-x

Schroer, C., Kruse, F., & Gomez, J. M. (2021). A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, *181*, 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Tak, H., Kamble, M., Patino, J., Todisco, M., & Evans, N. (2022). RawBoost: A Raw Data Boosting and Augmentation Method Applied to Automatic Speaker Verification Anti-Spoofing. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6382–6386. IEEE. https://doi.org/10.1109/ICASSP43922.2022.9746213

E-ISSN : 2807-9035
Volume 6, **Number** 3, August 2026
https://doi.org/10.47709/brilliance.v6i3.9372

Tak, H., Patino, J., Todisco, M., Nautsch, A., Evans, N., & Larcher, A. (2021). End-to-End anti-spoofing with RawNet2. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6369–6373. IEEE. https://doi.org/10.1109/ICASSP39728.2021.9414234

Tarawneh, A. S., Hassanat, A. B., Altarawneh, G. A., & Almuhaimeed, A. (2022). Stop Oversampling for Class Imbalance Learning: A Review. *IEEE Access*, *10*, 47643–47660. https://doi.org/10.1109/ACCESS.2022.3169512

Vila, L. C., Sturm, B. L. T., Casini, L., & Dalmazzo, D. (2025). The AI Music Arms Race: On the Detection of AI-Generated Music. *Transactions of the International Society for Music Information Retrieval*, *8*(1), 179–194. https://doi.org/10.5334/tismir.254

Wani, T. M., Qadri, S. A. A., Comminiello, D., & Amerini, I. (2024). Detecting Audio Deepfakes: Integrating CNN and BiLSTM with Multi-Feature Concatenation. *Proceedings of the 2024 ACM Workshop on Information Hiding and Multimedia Security*, 271–276. ACM. https://doi.org/10.1145/3658664.3659647

Xie, Y., Zhou, J., Lu, X., Jiang, Z., Yang, Y., Cheng, H., & Ye, L. (2023). *FSD: An Initial Chinese Dataset for Fake Song Detection*. arXiv. https://doi.org/10.48550/arXiv.2309.02232

Yamagishi, J., Wang, X., Todisco, M., Sahidullah, M., Patino, J., Nautsch, A., … Delgado, H. (2021). ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection. *2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 47–54. ISCA. https://doi.org/10.21437/ASVSPOOF.2021-8

Yang, Y.-Y., Hira, M., Ni, Z., Astafurov, A., Chen, C., Puhrsch, C., … Quenneville-Belair, V. (2022). Torchaudio: Building Blocks for Audio and Speech Processing. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6982–6986. IEEE. https://doi.org/10.1109/ICASSP43922.2022.9747236

Yi, J., Fu, R., Tao, J., Nie, S., Ma, H., Wang, C., … Li, H. (2022). ADD 2022: the first Audio Deep Synthesis Detection Challenge. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 9216–9220. IEEE. https://doi.org/10.1109/ICASSP43922.2022.9746939

Zaman, K., Sah, M., Direkoglu, C., & Unoki, M. (2023). A Survey of Audio Classification Using Deep Learning. *IEEE Access*, *11*, 106620–106649. https://doi.org/10.1109/ACCESS.2023.3318015

Zang, Y., Zhang, Y., Heydari, M., & Duan, Z. (2024). SingFake: Singing Voice Deepfake Detection. *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 12156–12160. IEEE. https://doi.org/10.1109/ICASSP48485.2024.10448184

Zhang, C.-B., Jiang, P.-T., Hou, Q., Wei, Y., Han, Q., Li, Z., & Cheng, M.-M. (2021). Delving Deep Into Label Smoothing. *IEEE Transactions on Image Processing*, *30*, 5984–5996. https://doi.org/10.1109/TIP.2021.3089942

Zhen, C., & Changhui, L. (2021). Music Audio Sentiment Classification Based on CNN-BiLSTM and Attention Model. *2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE)*, 156–160. IEEE. https://doi.org/10.1109/RCAE53607.2021.9638811

Zhuang, Z., Liu, M., Cutkosky, A., & Orabona, F. (2022). *Understanding AdamW through Proximal Methods and Scale-Freeness*. arXiv. https://doi.org/10.48550/arXiv.2202.00089