Journal of Scientific & Industrial Research
Vol. 84, October 2025, pp. 1107-1122
DOI: 10.56042/jsir.v84i10.17030

Enhancing Environmental Sound Classification with EcoOptiNet: A CNN-based Approach for Low-Frequency Signals

Kumud Patel[1]*, J P Pandey[2] & Malay Kishore Dutta[3]

[1]Centre for Advanced Studies, Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh, India

[2]Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh, India

[3]Amity Centre for Artificial Intelligence, Amity University, Noida, Uttar Pradesh, India

Received 12 February 2025; revised 27 August 2025; accepted 21 October 2025

Time–frequency analysis is widely used to extract meaningful features from raw signals, but it is particularly challenging for low-frequency, non-stationary audio data. Deep neural networks offer strong feature-learning capabilities, yet their effectiveness depends on optimizers that can manage temporal variability and guide models toward stable convergence. This research introduces EcoOptiNet, an improved optimization algorithm designed for low-frequency time-series audio signals. EcoOptiNet incorporates bias correction, an adaptive learning rate, and a learning warm-up phase (ζ) to prevent abrupt weight updates during early training. In contrast, a cubed-gradient update strategy enhances learning for non-stationary signals. Audio signal data is transformed into Mel-spectrograms and delta features, capturing both spectral and temporal characteristics, and a Convolutional Neural Network (CNN) architecture is employed for classification. The algorithm is evaluated on two benchmark environmental sound datasets using standard splits: ESC-50 with 5-fold cross-validation and UrbanSound8K with 10-fold cross-validation. Experimental results indicate that EcoOptiNet achieves an average accuracy of 98.83% on ESC-50 and 91% on UrbanSound8K, outperforming commonly used optimizers such as Adam, RMSprop, and SGD, while maintaining low variance across folds. These findings demonstrate that EcoOptiNet provides an efficient and robust approach for optimizing deep neural networks on low-frequency, real-world audio signals. The study highlights the algorithm's ability to reliably extract discriminatory features from challenging datasets, offering a practical solution for environmental sound recognition applications where non-stationarity and low-frequency components can hinder traditional training approaches.

Keywords: Acoustic recognition, Deep-learning, Optimization algorithms, Signal processing, Time series analysis

Introduction

Classifying low-frequency time series signals poses unique challenges due to the inherent non-stationary and variable characteristics of these signals. In traditional approaches, optimization algorithms such as Stochastic Gradient Descent (SGD) and Adam are commonly used to estimate model parameters. However, these optimizers are often ill-suited for low-frequency signals where data points exhibit temporal variability, causing inefficiencies in model convergence and accuracy. This problem is exacerbated in tasks involving non-stationary signals, where common assumptions of data distribution and consistency do not hold. The optimization algorithm estimates the parameters in the neural network model. There are no direct solutions available mathematically. In general, all the data points are drawn from the same distribution, but in the case of low-frequency time series signals, this does not hold. In the deep neural network model, the optimization algorithm determines the time complexities of the learning process and the accuracy rate. Designing a CNN (Convolutional Neural Network) and selecting the best optimization algorithm for low-frequency time series data is a significant challenge.[1]

In recent years, Deep Neural Networks (DNNs) have significantly altered information science. The parameters of neural networks are estimated using optimization. Typically, it's assumed that every data point comes from the same distribution. This presumption is incorrect in low-frequency time series signal observations. The statistics of data points change over time when sequences are low-frequency time series signals, considerably affecting the parameter estimation optimization process. Arithmetical algorithms based on gradient descent can compute model parameters iteratively.[2] The loss function gradients update the parameter vector, with

*Author for Correspondence
E-mail: kumudcs14.academic@gmail.com

the learning data setting its contours. As a result, gradient categorizations also become random signals. The optimization procedure for all deep network models significantly impacts the learning process duration and outcome accuracy. The SGD (Stochastic Gradient Descent) method is the most fundamental and popular.[3] The Snake Optimizer (SO) is a newly developed meta-heuristic algorithm inspired by snake mating behaviours, focusing on balancing exploration and exploitation to improve convergence speed.[4] The Corona Virus Optimization (CVO) algorithm, inspired by COVID-19 virus characteristics, aims to demonstrate feasible, effective performance in solving continuous mathematical functions.[5] The optimization algorithm enhances protein generation from DNA synthesis, incorporating advanced searching techniques and deep composite optimization, reducing execution time and improving efficiency. Meta-heuristic neural network optimization is also used for loss function reduction and feature selection.[6] The optimization algorithm updates the parameter vector using only the gradient descent direction and a manually chosen constant step size.

The l_2 standard of past gradient vectors determines the step size of RMSprop, an exponentially weighted variant of AdaGrad [7]The most recent algorithm is Adam.[8] Other methods include Adadelta, SGD with Nesterov momentum AMSGrad. RMSprop and Adamare claimed to be efficient for low-frequency time series signals, considering that the current step's update path and size are calculated using all prior gradients.[9,10] Data change over time in low-frequency signals, making it unlikely that the current step relates to data from an extended period.

In time series analysis, low-frequency signals differ from image datasets. Image classification uses CNN, popular for various image datasets. RGB channels follow the color receptors in human eyes used in displays and scanners.[11] Many steps, such as convolutional, pooling, and fully connected neural networks, are used for fixed-dimension image datasets.[12,13] For low-frequency time series signals, dimensions are not selected but continuously varying. Predicting RGB channels is challenging because they're not equally separated, unlike image data.[13] Data and architecture are major issues for these signals. To resolve this, an enhanced optimization algorithm with a CNN-based model is designed to analyze data on different parameters for low-frequency time series signals.

This research addresses these limitations by introducing an enhanced EcoOptiNet optimization algorithm for low-frequency time series signals. It incorporates a bias correction mechanism and an adaptive learning rate that adjusts to the changing nature of the data. Unlike conventional optimization, often customized to stationary data, this method effectively manages temporal variability in low-frequency signals.

The main contributions of this research are:

1. An enhanced consideration module addressing intraclass inconsistency in low-frequency time series signals improves classification accuracy.
2. EcoOptiNet, an improved CNN-based optimization method, is proposed to enhance parameter optimization for low-frequency time-series signal classification.
3. A learning warm-up phase (ζ) is incorporated to stabilize early training, which helps stabilize abrupt weight updates, accelerates convergence, and enhances accuracy in classifying low-frequency signal data.
4. The proposed EcoOptiNet optimization overcomes limitations of traditional algorithms b introducing new learning dynamics, resulting in better performance and generalization in deep neural networks.

Literature Review

This research employed the ESC-50 dataset to analyze and classify low-frequency time series signals. ESC-50 contains 50 balanced environmental sound classes, making it diverse and relevant to our objectives. Its 16 kHz sampling rate enables analysis across both low- and high-frequency components, capturing indirect sound characteristics within the low-frequency spectrum. Environmental sound classification has applications in intelligent surveillance, healthcare, robot navigation, customer alert systems, disaster identification, and ecological monitoring.[14] Audio classification involves assigning short clips to appropriate categories, where understanding the context of surrounding sounds is essential for timely responses to potential risks.[15]

The prediction of non-stationary time series signals is prominent in three domains: Music Information Retrieval (MIR), Automatic Speech Recognition (ASR), and Environmental Sound Classification (ESC).[16–18] Unlike structured music and speech signals, environmental sounds are unstructured and often exhibit a lower Signal-to-Noise Ratio (SNR),

making ESC a more challenging task. The ESC-50 dataset covers natural, domestic, animal/bird, urban, and human sound classes, while the UrbanSound8K dataset provides additional urban sound samples such as sirens, engines, and street noise.[19–21] Together, these datasets cover a broad frequency spectrum, supporting the exploration of low-frequency components across diverse sound categories.

Early work by Piczak applied CNNs to ESC-50 using Log-Mel spectrograms and their deltas as two-dimensional representations, significantly improving performance over traditional handcrafted features.[22] Since then, multiple feature extraction methods, machine learning approaches, and deep learning models have been applied to both ESC-50 and UrbanSound8K CNN architectures, initially developed for image classification, have been effectively adapted to spectrogram-based audio classification, where robust feature extraction from audio samples plays a crucial role in determining accuracy, especially for low-frequency time series data.

Preliminaries

Low-Frequency Time Series Data

In time series data sequences, $X = \{x_1, x_2.....,x_t\}$ represents the entire time series where, x_t is its element at time t and t = 1, 2.....T. X is said to be a stationary frequency (weak or strict) time series signal if both the following conditions are verified; otherwise, x_t is said to be a non-stationary low-frequency time series data.

Condition 1: Expectations of signal frequency are not changing over time. $E(x_t)$ is constant for all time t. The mathematical representation of expectation x_t is $E(x_t)$.

Condition 2: The variance of signal frequency does not change over time. The mathematical representation of variance x_t is $\sigma^2(x_t)$, where $\sigma^2(x_t)$ is constant for all time t.

In statistical literature, condition 1(expectation) and condition 2(variance) are known as the first and second moments, respectively. Here, X is called strict stationery if all the X frequencies remain constant over time for all joint probability distributions. The Stationary (weak or inflexible) time series data are easier to verify than non-stationary signals. To illustrate, Fig. 1 represents an audio recording of a chirping bird from the ESC-50 (environmental sound classification) datasets. This example is particularly insightful as the expectations and variance of the signal change over time, which means it is demonstrating a non-stationary low-frequency time series characteristic.

Convolutional Neural Network (CNN)

Classification of low-frequency time series signals can be formulated as a mapping problem between input and output classes, as given in Eq. 1
Let

$$X_{class} = (x_1, x_2.....x_n), Y_{class} = (y_1, y_2.....y_n) \quad \ldots (1)$$

where, X_{class} represents the input matrix and Y_{class} the output matrix, with each column indexed by n. The i^{th} element of the input vector x_n denotes x_n^i. It is crucial to note that X_{class} is distinct from the original time series X. Here, x_n corresponds to the parameter available for the low-frequency time series signals problem.

For forward classification, the output can be expressed as given in Eq. 2

$$Y_n = [y_{(n-1)}, y_{(n-2)}, \ldots\ldots\ldots\ldots y_{(n-d)}]^T \quad \ldots (2)$$

The relationship between the input and output is modelled as given in Eq. 3

$$Y_n = f(x_n) \quad \ldots (3)$$

where, f is approximated using a CNN architecture. A general mathematical representation of CNN is given in Eqs 4 and 5:

$$output = Activation(f)((w_i) * (x_n^i) + \theta) \quad \ldots (4)$$

$$Y_i = ReLu(f)(weight)((w_i) * (x_n^i) + \theta) \quad \ldots (5)$$

where, x_n^i The input vector is denoted as $i = (1, 2,..., n)$, w_i represents the weight matrix, θ is the bias, and Y_i represents the output of the CNN unit at the *i-th* iteration and also serves as input to subsequent CNN layers.

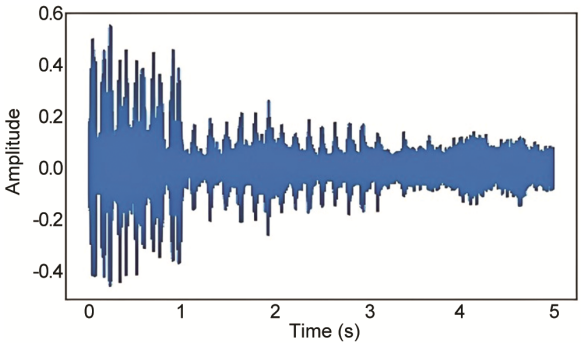

Fig. 1 — Audio signals of a chirping bird

The final output is computed as shown in Eq. 6 below.

$$(y_i) = g(Y_i) \quad \ldots (6)$$

where, $g_i(Y)_i$ represents the concatenation of Y_i elements with a constant bias b_i. The overall low-frequency time series classification model can then be represented as shown in Eq. 7

$$(y_n) = f(x_n, \theta) \quad \ldots (7)$$

Here, the outcome depends on the network weights y_n, inputs, and bias. To address the limitations of standard optimizers for non-stationary low-frequency signals, this work proposes the EcoOptiNet optimization algorithm, which improves convergence rate and classification accuracy when applied to CNN-based low-frequency time series data.

The Optimization Algorithm of Deep Neural Networks

Based on the given data *(X_n, Y_n),* determining the optimal value of θ is the next issue of low-frequency time series signal classification. As discussed in the introduction section, many optimization algorithms such as SGD, RMSprop, and Adam are used to minimize the loss function in neural networks. By incorporating the loss function, this problem can be framed as a standard optimization task.

The gradient is defined as shown in Eq. 8

$$g_t = \nabla_\theta Lf(\theta_{t-1}) \quad \ldots (8)$$

where, 'L' denotes the loss function, typically the MSE (Mean Squared Error). The parameter update using SGD is given by Eq. 9

$$\theta = \theta_{t-1} - \alpha * \nabla_\theta Lf(\theta_{t-1}) \quad \ldots (9)$$

where, g_t is the gradient descent, and α is the learning rate, which determines the step size at each iteration

In Adam, the updated value of θ is dependent on all the estimated gradients until the current time step t, rather than only the most recent gradient. The update is given in Eq. 10:

$$\theta_t \rightarrow (\theta_{t-1} - \alpha\, \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)) \quad \ldots (10)$$

where, ϵ is a small constant for numerical stability and$\hat{m}_t$ and $\hat{v}_t$ are calculated as given in Eqs 11 and 12:

$$\hat{m}_t = \frac{m_t}{1-\beta_{1t}} \quad \ldots (11)$$

$$\hat{v}_t = \frac{v_t}{1-\beta_{2t}} \quad \ldots (12)$$

Here, m_t and v_t are the first-and second-order momentum of the gradients β_1 and β_2 are the hyperparameters that control the exponential decay rates. The first-order momentum is updated as shown in Eq. 13, while the second-order momentum is updated as given in Eq. 14. Specifically, the second-order momentum v_t is calculated as the weighted sum of the current gradient squared and the previous v_{t-1}:

$$m_t = (\beta_1 * (m_{t-1} + (1-\beta_1)g_t) \quad \ldots (13)$$

$$v_t = (\beta_2 * (v_{t-1} + (1-\beta_2)g_t^2) \quad \ldots (14)$$

As discussed, the frequency of low-frequency time series signals changes with time. Therefore, bias correction may be irrelevant, as the updated value of θ based on the current step can be far from recent steps. To address this issue, this research proposes a novel optimization algorithm, which is discussed in the methodology section.

Methodology

The accurate classification of low-frequency time-series signals is a challenging task due to their non-stationary nature and the limited discriminative features in the raw data. Traditional approaches often struggle to capture both temporal and spectral characteristics effectively. To overcome these limitations, a structured four-stage methodology is designed that leverages spectrogram-based representation, CNN-driven feature extraction, and an enhanced optimization. This pipeline aims to enhance feature learning, reduce training complexity, and achieve robust classification performance.

The proposed framework consists of four main stages: data preprocessing, conversion to Mel-spectrograms, CNN-based feature extraction, and optimization using EcoOptiNet. The initial phase prepares audio samples for analysis, detailed in the Experiment, works, and results. The second stage transforms low-frequency time-series data into Mel-spectrograms, which visually represent frequency variations over time. The third stage extracts discriminative features from these spectrograms using a CNN. Finally, the fourth component, integrating with the CNN to improve convergence and classification accuracy, is detailed in this Section along with pseudocode. This structure approach provides a comprehensive solution for robust classification of low-frequency time-series signals, as shown in Fig. 2.

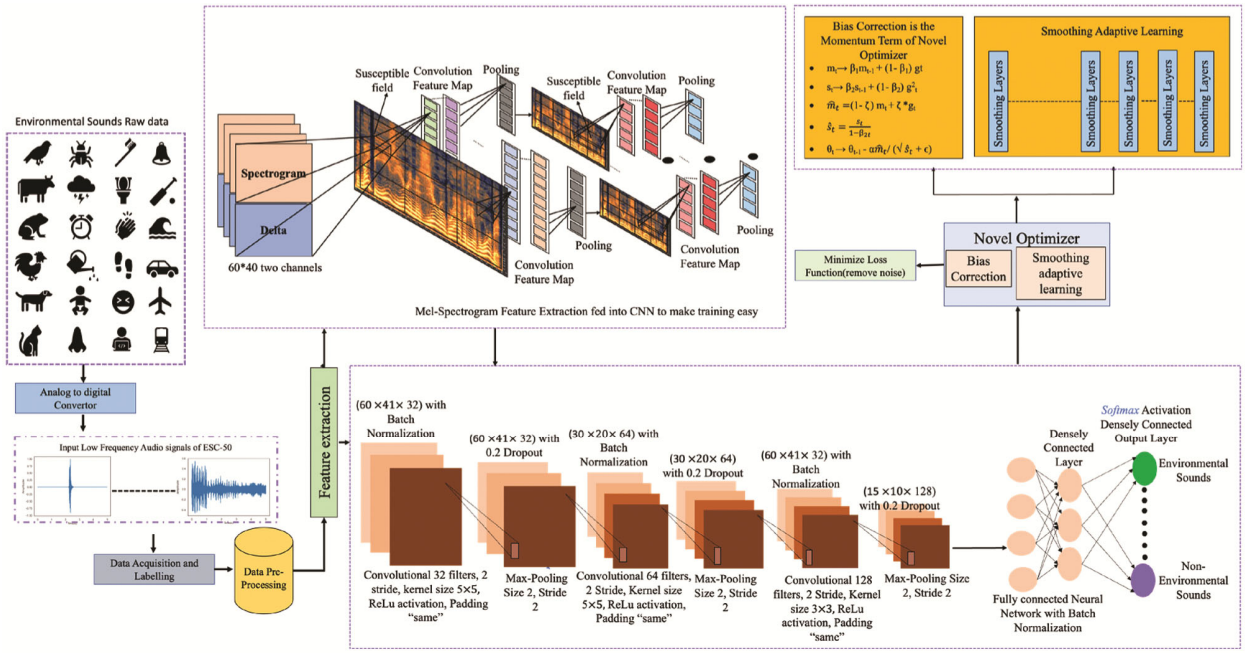

Fig. 2 — Architecture of the proposed EcoOptiNet with the layered structure of the CNN Model and the combination of feature extraction and a fully connected neural network fed as input data

CNN Architecture and Feature Extraction

In this research work, the proposed CNN architecture is applied to both the ESC-50 and UrbanSound8K datasets, with only the output layer adapted to the number of classes (50 for ESC-50 and 10 for UrbanSound8K). The choice of CNN is motivated by its ability to learn local spectral temporal patterns from spectrograms, which are the most informative representations of environmental sounds. Unlike traditional handcrafted feature-based methods, CNNs automatically capture hierarchical structures in audio data, making them well-suited for non-stationary and noisy low-frequency signals.

The input Mel-spectrograms of size 60 × 40 × 32 serve as the starting point. The first convolutional layer uses 32 filters of size 5 × 5 with a stride of 2 and 'same' padding, followed by batch normalization and ReLU activation, effectively capturing low-level frequency patterns while ensuring stability in training. Subsequent convolutional layers with 64 and 128 filters (3 × 3 kernels, stride 2, 'same' padding) progressively extract higher-level features, enabling the network to represent both short-term transients (such as bird chirps) and long-term tonal structures. The CNN Layers architecture and the hyperparameters of EcoOptiNet are presented in Tables 1 & 2.

After flattening, two fully connected layers (512 and 128 neurons) are used to learn abstract audio representations. Finally, a Softmax layer outputs class probabilities, adapted according to the dataset size. By maintaining the same architecture for both datasets and modifying only the output dimension, the framework ensures consistency in feature learning while preserving fairness in evaluation.

In the CNN paradigm, ReLU and SoftMax activation functions are tremendously essential. They essentially decide whether the neuron should be activated or not. ReLU activation is shown in Eq. 15 for both positive and negative input x values by multiplying matrices and making use of incremental bias offset; it is feasible to ascertain the weights and bias of the neurons included inside a layer.

$$\text{ReLU} = \begin{cases} 0, \text{if} x \leq 0 \\ 1, \text{if} x > 0 \end{cases} \quad \ldots (15)$$

The SoftMax activation function determines the probability distribution of the event across 'n' distinct events. This function will evaluate the likelihood for every class across overall potential target classes. A target class for the supplied inputs is later determined using the calculated probability, which is SoftMax's primary benefit. The probabilities will be between 0 and 1, with the total being equal to 1. The odds of each class are decreased when the SoftMax function is applied in a multiclassification model, and the target class will have a high probability, as shown in Eq. 16

Table 1 — Architecture of CNN with Layers and Parameters

S. No	Layers	Output Shape	Kernel Size	Stride	Filters	Activation Function	Parameters
1	Convolution	(60,41,32)	5 × 5	2	32	Relu	288
2	Batch Normalization	(60,41,32)	—	—	—	—	128
3	Dropout	(60,41,32)	—	—	—	—	0
4	Convolution	(60,41,32)	—	—	—	Relu	4128
5	Max Pooling	(30,20,32)	—	—	—	—	0
6	Convolution	(30,20,64)	—	2	64	Relu	8256
7	Dropout	(30,20,64)	—	—	—	—	0
8	Convolution	(30,20,64)	—	—	—	—	16448
9	Max Pooling	(15,10,64)	—	—	—	—	0
10	Convolution	(15,10,128)	—	2	128	Relu	32896
11	Dropout	(15,10,128)	—	—	—	—	0
12	Convolution	(15,10,128)	—	—	—	—	65664
13	Max Pooling	(7,5,128)	—	—	—	—	0
14	Flatten	4480	—	—	—	—	0
15	Dropout	4480	—	—	—	—	0
16	Dense	1024	3 × 3	—	—	Relu	45885
17	Dropout	1024	—	—	—	—	0
18	Dense	512	3 × 3	—	—	Relu	52480
19	Dropout	512	—	—	—	—	0
20	Dense	50	—	—	—	Softmax	25650
Trainable Parameters							5,266,738
Non-Trainable Parameters							64
		Total Parameters					5,266,802

Table 2 — Hyperparameters of Proposed EcoOptiNet

Name	Parameters
Input data	The dataset consists of many "wav" files of Environmental sounds
Input shape	(60,41,2)
Loss type	Categorical Cross-Entropy
Activation function	ReLU and Softmax
No. epochs	100
Optimization function	EcoOptiNet algorithm
Learning rate	0.001
β_1	0.99
β_2	0.999
ζ	0.05
epsilon	1e-8

$$\mathrm{SoftMax}(x_i) = \frac{\exp(x_i)}{\Sigma_j \exp(x_j)} \quad \ldots (16)$$

The equation calculates the exponential (e-Power) of the specified input value as well as the total exponential value. The output of the SoftMax function will then most likely be the ratio of the exponential input values to the sum of exponential values. A neuron's output in a fully connected layer with inputs of x, w, b, and f, as well as weight w and bias b is Eq.17.

$$Y = \mathrm{Activation(f)}\left(\Sigma\left(\mathrm{weight(w)} * \mathrm{Input(x)}\right) + \mathrm{Bias(b)}\right) \quad \ldots (17)$$

Finally, the CNN contains convolutional layers that include the first layer 32 filters, the second layer 64 filters, and the third 128 filters. The model's accuracy, specificity, sensitivity, precision, and f1 score have all improved significantly after using batch normalization.

EcoOptiNet: An Enhanced Optimization Algorithm for Low-Frequency Time Series Signals

Based on the given Eqs.13,14, It can be observed that the low-frequency time series signal sequences are non-stationary. This implies that their frequency characteristics vary with time. To address this, Eq.18 represents an Exponential Moving Average (EMA) of the previous time step m_{t-1} and the squared current gradient (g_t^2). Similarly, Eq. 19 calculates an EMA of the cube of the current gradient $(g_t * g_t * g_t)$, which forms the root of the optimization algorithm.

$$m_t = (\beta_1 * (m_{t-1} + (1-\beta_1) g_t^2) \quad \ldots (18)$$

$$s_t = (\beta_2 * (s_{t-1} + (1-\beta_2) g_t^3) \quad \ldots (19)$$

The motivation for using the cube of the gradient (g_t^3) arises from the nature of low-frequency signals,

where gradient values are often small due to slow variations. Cubing amplifies these subtle changes, enabling the optimizer to respond more effectively to minor but meaningful variations in the signal. The weighted sum of the current step and the cubed gradient to obtain s_t are computed via Eq. (16) further computes.

Finally, the parameter θ is updated, inspired by the Adam optimizer algorithm, but modified to enhance adaptivity for non-stationary signals as shown in Eq. 20.

$$\theta_t \rightarrow (\theta_{t-1} - \alpha\, \hat{m}_t/(\sqrt{\hat{s}_t} + \epsilon)) \quad \ldots (20)$$

This proposed work introduces a new statistical method, where s_t, Modifies the traditional Adam update rule. This design enhances performance for low-frequency time series signals by incorporating two primary adaptations:

1. Learning warm-up phase (ζ) to stabilize early training.
2. Redefined the velocity update term using cubed gradients.

These adaptations address the challenges of non-stationary signals while improving convergence stability. To further strengthen adaptability, EcoOptiNet integrates a bias correction mechanism that dynamically reduces the impact of outdated gradient information, particularly during early training when fluctuations are high. The bias correction formulations are given in Eqs 21 & 22:

$$\hat{m}_t = \{(1-\zeta)m_t + (\zeta * g_t)\} \quad \ldots (21)$$

$$\hat{s}_t = \frac{s_t}{1-\beta_{2t}} \quad \ldots (22)$$

Here, Eq. 21 incorporates the learning warm-up (ζ) phase parameter (ζ = 0.05), which minimizes instability during the initial iterations by balancing the historical average with the current gradient. Similarly, Eq. 22 normalizes the exponential moving average of the cubed gradient, ensuring stable variance control during training.

The combination of bias correction and cubed-gradient updates enables EcoOptiNet to maintain stability and robustness when handling non-stationary, low-frequency signals.

For clarity, the main differences between Adam and EcoOptiNet are given below in Table 3.

When tested on the ESC-50 dataset, EcoOptiNet achieved an overall accuracy of 99%, with a macro F1-score of 98.83%. The weighted precision and recall values also remained consistently above 98.5%, indicating balanced performance across all classes. In contrast, Adam achieved 71%, while SGD reached 62%. Moreover, EcoOptiNet reduced the loss function by 81%, demonstrating its superior ability to handle non-stationary, low-frequency signals effectively. Similarly, on the UrbanSound8K dataset, EcoOptiNet achieved an overall accuracy of 91.87%, with a macro F1-score of 89.62% and a weighted F1-score of 89.00%. These results confirm that EcoOptiNet not only excels in relatively balanced datasets like ESC-50 but also generalizes effectively to large-scale, real-world urban sound datasets where class distribution and signal characteristics are more complex.

Experimental Work and Results

For the experimental work, the ESC-50 dataset comprising 2000 audio clips across 50 diverse

Table 3 — Stepwise formulation of Adam vs. EcoOptiNet

The following notations are used for the new proposed EcoOptiNet with bias correction. Default values suggested for decay-hyper-parameters are β1 = 0.99, β2 = 0.999, ζ = 0.05, α = 1e-4 and ϵ= 10^{-8}

Adam Optimizer (Algorithm 1)	Proposed EcoOptiNet (Algorithm 2)
Initialize θ_0, $m_0 \rightarrow 0$, $v_0 \rightarrow 0$ $t_0 \rightarrow 0$	Initialize θ_0, $m_0 \rightarrow 0$, $v_0 \rightarrow 0$ $t_0 \rightarrow 0$,
Learning rate $(\alpha) = 0.001$, $\beta_1 \rightarrow 0.9$, $\beta_2 \rightarrow 0.999$, $\epsilon = 10^{-7}$	Learning rate $(\alpha) = 0.001$, $\beta_1 \rightarrow 0.99$, $\beta_2 \rightarrow 0.999$, $\zeta \rightarrow 0.05$, $\epsilon = 10^{-8}$
θ_t it did not converge	θ_t it did not converge
$t \rightarrow t+1$	$t \rightarrow t+1$
$g_t \rightarrow \nabla_\theta f_t(\theta_{t-1})$	$g_t \rightarrow \nabla_\theta f_t(\theta_{t-1})$
$m_t \rightarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$	$m_t \rightarrow \beta_1 m_{t-1} + (1-\beta_1) g^2_t$
$v_t \rightarrow \beta_2 v_{t-1} + (1-\beta_2) g^2_t$	$s_t \rightarrow \beta_2 s_{t-1} + (1-\beta_2) g^3_t$
Bias correction	Bias correction
$\hat{m}_t = \frac{m_t}{1-\beta_{1t}}$	$\hat{m} = \{(1-\zeta)m_t + (\zeta * g_t)\}$
$\hat{v}_t = \frac{v_t}{1-\beta_{2t}}$	$\hat{s}_t = \frac{s_t}{1-\beta_{2t}}$
Update	Update
$\theta_t \rightarrow (\theta_{t-1} - \alpha\, \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon))$	$\theta_t \rightarrow (\theta_{t-1} - \alpha\, \hat{m}_t/(\sqrt{\hat{s}_t} + \epsilon))$

environmental sound classes (e.g., dog bark, car horn, crying baby, wind) is used. To further strengthen the generalizability of the proposed EcoOptiNet, experiments are also conducted on the Urban Sound8K dataset, which contains 8732 labeled clips of 10 urban sound classes (e.g., siren, drilling, engine idling, gunshot).Both datasets are widely recognized benchmarks for environmental sound classification, ensuring a robust and comprehensive evaluation of EcoOptiNet. The tools and Technology used in this research work are shown in Table 4.

Datasets

In this research work, two widely used environmental sound datasets are used to evaluate the proposed EcoOptiNet optimizer: ESC-50 and UrbanSound8K.

- ESC-50: This dataset consists of 2,000 labeled environmental audio recordings of 5 seconds each, equally distributed across 50 balanced classes (40 samples per class). The datasets are already organized into 5 pre-defined folds, each containing 400 samples, enabling consistent 5-fold cross-validation. For training and testing, an 80–20 split was used resulting in 1600 samples for training and 400 samples for testing. ESC-50 is widely used as a benchmark for environmental sound classification due to its balance across diverse sound categories such as animals, natural soundscapes, human non-speech sounds, domestic sounds, and urban noises[23]
- UrbanSound8K: This dataset contains 8,732 audio clips (up to 4 seconds) spanning 10 everyday urban sound classes like sirens, engine idling, car horns, drilling, and children playing. The dataset is organized into 10 pre-defined folds, allowing direct use of 10-fold cross-validation without additional partitioning. Following the same 80–20 split, 6986 samples were used for training and 1746 for testing. UrbanSound8K is more diverse and larger than ESC-50, making it suitable for validating the generalizability and robustness of the proposed EcoOptiNet optimizer.

Table 4 — Tools and technology

Model	Intel Core i5 8th Generation CPU (model 8750H)
Coding environment	Jupyter Notebook Anaconda
Common Python libraries	TensorFlow, Keras, and librosa
Files	wav files,
Libraries	NumPy, matplotlib, os, pandas, glob, seaborn, Sklearn
Programming language	Python 3.9.12
RAM	16 GB of DDR4 RAM

Preprocessing:

To prepare the audio signals for classification, a systematic preprocessing pipeline was applied. The process involved the following key stages:

Segmentation

Each audio recording was divided into fixed-size frames of 60 × 40 dimensions. This segmentation converted variable-length audio clips into uniform representations, making them suitable for further feature extraction.

Mel-Spectrogram and Delta Features

For every segmented frame, a log-scaled Mel-spectrogram was generated, which provides a time–frequency representation of the signal on a perceptual scale aligned with human auditory sensitivity. In addition, delta features were computed to capture temporal dynamics, i.e., the rate of change in the spectral information over time, as shown in Fig. 3. The combination of Mel-spectrograms and deltas offered a more informative two-channel representation of low-frequency signals.

Splitting Strategy

To ensure robust evaluation, dataset-specific cross-validation strategies were adopted. ESC-50 was evaluated using its pre-defined 5-fold cross-validation, while UrbanSound8K followed a 10-fold cross-validation protocol. These splits divided the data into training and testing subsets as shown in Table 5, balancing model learning and generalization while preventing overfitting.

The resulting two-channel inputs (Mel-spectrograms with deltas) were treated as image-like data, which aligned well with the convolutional neural network (CNN) framework. This representation enabled the CNN model to learn spatial and temporal relationships effectively, thereby improving classification performance on non-stationary, low-frequency environmental signals.

The complete preprocessing pipeline is illustrated in Fig. 4, beginning with raw audio signals and progressing through segmentation, feature extraction, and splitting strategies before the final input into the CNN model.

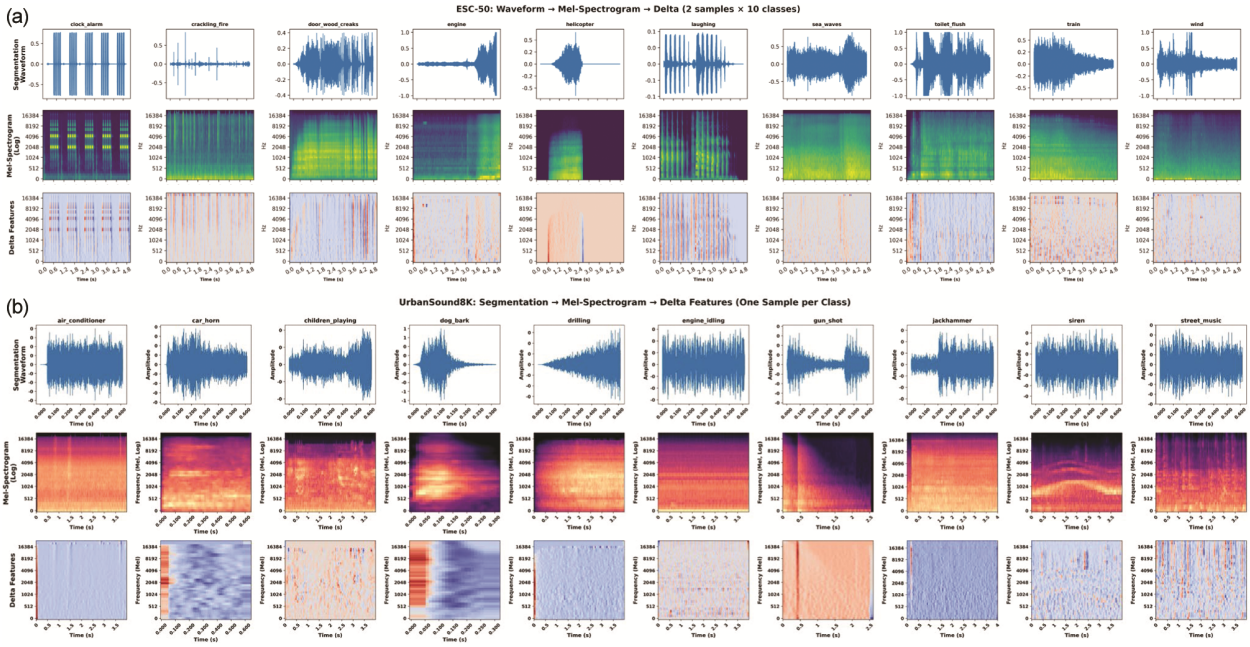

Fig. 3 — Preprocessing of environmental sound datasets: (a) UrbanSound8k, (b) ESC-50: Waveform, Mel Spectrogram, and delta features of one sample per class and a random sample, respectively

Table 5 — Performance matrix of proposed EcoOptiNet optimization for UrbanSound8K datasets

No. of Classes	Audio Clips	Precision	Recall	F1-Score	Support
0	Air conditioner	0.9701	0.9606	0.9653	203
1	Car horn	0.8667	0.9070	0.8864	86
2	Children playing	0.8800	0.8415	0.8603	183
3	Dog bark	0.9399	0.8557	0.8958	201
4	drilling	0.8826	0.9126	0.8974	206
5	Engine idling	0.9794	0.9845	0.9819	193
6	Gun shot	0.9853	0.9306	0.9571	72
7	jackhammer	0.9561	0.9423	0.9492	208
8	siren	0.8889	0.9697	0.9275	165
9	Street music	0.8613	0.8613	0.8761	230
Accuracy				0.9187	1747
Macro average		0.9210	0.9196	0.9197	1747
Weighted average		0.9198	0.9187	0.9187	1747

Results

Dataset-wise Performance

To evaluate the effectiveness of EcoOptiNet, experiments were conducted on two benchmark datasets: ESC-50 and UrbanSound8K.On ESC-50, EcoOptiNet achieved 99% overall accuracy, with macro and weighted F1-scores of 0.99. The average micro-weighted accuracy across five pre-defined folds was 98.83% ± 0.4, confirming stable performance. Class-wise results are summarized in Table 6.

On UrbanSound8K, EcoOptiNet achieved 91.87% accuracy, with macro and weighted F1-scores of 0.8962 and 0.8900. The 10-fold cross-validation yielded an average accuracy of 91.87% ± 0.6, with results summarized in Table 5. The corresponding equations are provided below. Classes with overlapping acoustic patterns of Children playing and Street music were relatively more complex to classify. Most categories maintained F1-scores above 0.90, demonstrating strong generalizability.

$$\text{Accuracy} = \frac{\text{TruePositive} + \text{TrueNegative}}{\text{TruePositive} + \text{TrueNegative} + \text{FalsePositive} + \text{FalseNegative(Total)}}$$

$$\text{Precision} = \frac{\text{TruePositive}}{\text{Predicated}}$$

$$\text{Recall} = \frac{\text{TruePositive}}{\text{Actual}}$$

$$\text{f1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

In this dataset, environmental sounds like thunderstorm, rain, wind, sea waves, and fire are well

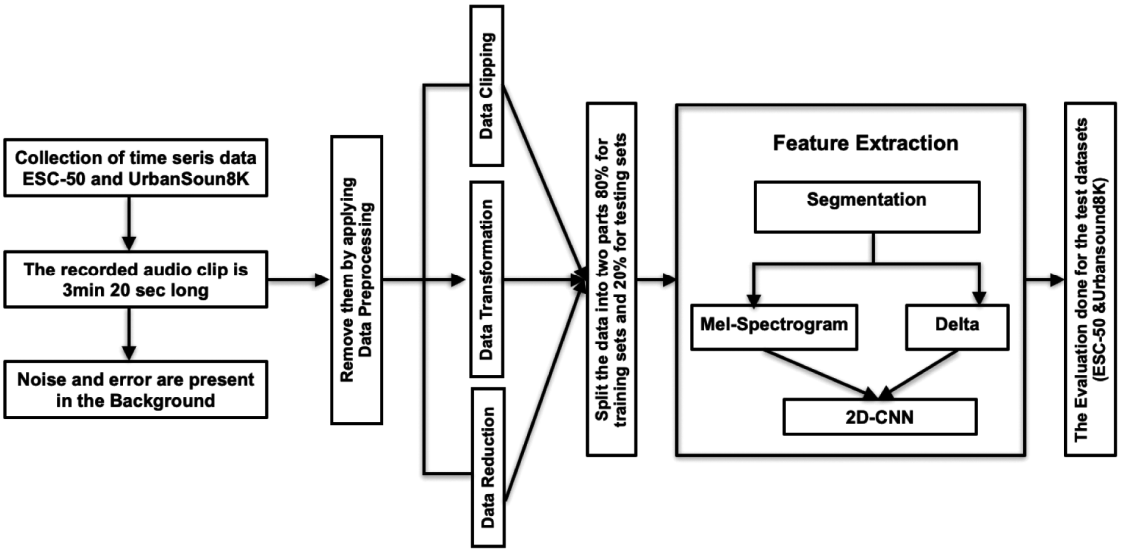

Fig. 4 — Flow Diagram of Data preprocessing for Low-frequency time series signals for ESC-50 datasets

Table 6 — Performance matrix of proposed EcoOptiNet optimization for ESC-50 datasets

No. of Classes	Audio clips	Precision	Recall	F1-Score	Support
0	Dog	1.00	0.97	0.98	59
1	Rooster	1.00	0.97	0.98	59
2	Pig	1.00	0.94	0.97	72
3	Cow	0.98	0.94	0.96	65
4	Frog	0.98	1.00	0.99	56
5	Cat	1.00	0.89	0.94	65
6	Hen	0.97	1.00	0.99	70
7	Insects	0.94	0.97	0.95	65
8	Sheep	1.00	1.00	1.00	60
9	Crow	1.00	1.00	1.00	63
10	Rain	1.00	1.00	1.00	66
11	Sea waves	0.86	0.96	0.91	70
12	Crackling_fire	0.98	0.98	0.98	63
13	Crickets	1.00	1.00	1.00	55
14	Chirping birds	1.00	0.94	0.97	72
15	Water drops	0.98	1.00	0.99	61
16	Wind	1.00	1.00	1.00	51
17	Pouring water	0.97	1.00	0.98	61
18	Toilet_flush	0.99	0.97	0.98	70
19	Thunderstorm	1.00	0.94	0.97	65
20	Crying baby	0.90	0.96	0.93	69
21	Sneezing	0.96	0.94	0.95	78
22	Clapping	0.99	0.95	0.97	87
23	Breathing	0.97	0.98	0.98	66
24	Coughing	0.98	0.96	0.97	67
25	Footsteps	1.00	0.97	0.98	59
26	Laughing	0.96	0.96	0.96	51
27	Brushing teeth	0.97	1.00	0.98	65
28	Snoring	1.00	1.00	1.00	69
29	Drinking sipping	0.98	1.00	0.99	80
30	Door wood knock	0.97	0.97	0.97	63

(*Contd.*)

Table 6 — Performance matrix of proposed EcoOptiNet optimization for ESC-50 datasets (*Contd.*)

No. of Classes	Audio clips	Precision	Recall	F1-Score	Support
31	Mouse click	0.93	0.99	0.96	67
32	Keyboard typing	0.92	1.00	0.96	49
33	Door wood creaks	0.87	0.98	0.93	63
34	Can opening	1.00	1.00	1.00	56
35	Washing machine	1.00	1.00	1.00	59
36	Vacuum cleaner	0.95	0.95	0.95	75
37	Clock alarm	1.00	1.00	1.00	65
38	Clock tick	1.00	0.97	0.99	73
39	Glass breaking	1.00	1.00	1.00	63
40	Helicopter	1.00	1.00	1.00	53
41	Chainsaw	1.00	1.00	1.00	66
42	Siren	1.00	1.00	1.00	59
43	Car horn	0.97	1.00	0.98	64
44	Engine	1.00	1.00	1.00	62
45	Train	1.00	1.00	1.00	65
46	Church bells	1.00	1.00	1.00	62
47	Airplane	1.00	1.00	1.00	61
48	Fireworks	0.98	0.89	0.93	53
49	Hand saw	1.00	1.00	1.00	63
Accuracy		—	—	0.98	3200
Macro average		0.99	0.99	0.99	3200
Weighted average		0.99	0.99	0.99	3200

classified with high accuracy due to their distinct low-frequency characteristics. Similarly, animal sounds such as cow, frog, sheep, and crow exhibit excellent performance as they produce clear sounds. Human-related low-frequency sounds, such as breathing, snoring, and coughing, are also effectively detected. Lastly, mechanical sounds such as engine and train noise are detected perfectly as they tend to produce consistent deep vibrations that make them stand out in the low-frequency spectrum.

To provide a clearer understanding of the training dynamics and classification outcomes, several visualizations were generated. The accuracy and loss curves for both ESC-50 are shown in Fig. 5 and UrbanSound8K as given in Fig. 6 demonstrate that EcoOptiNet achieved faster convergence with smoother trends compared to baseline optimizers. The confusion matrices for UrbanSound8K, as given in Fig. 7 and ESC-50, as shown in Fig. 8, further highlight strong per-class discrimination, with only minor overlaps in acoustically similar classes. A confusion matrix is a tabular representation of actual versus predicted values, forming the basis for metrics such as accuracy, precision, recall, and F1-score. The confusion matrix in Fig. 8 shows near-perfect class separation, with several classes achieving 100% precision and recall.[24,25]

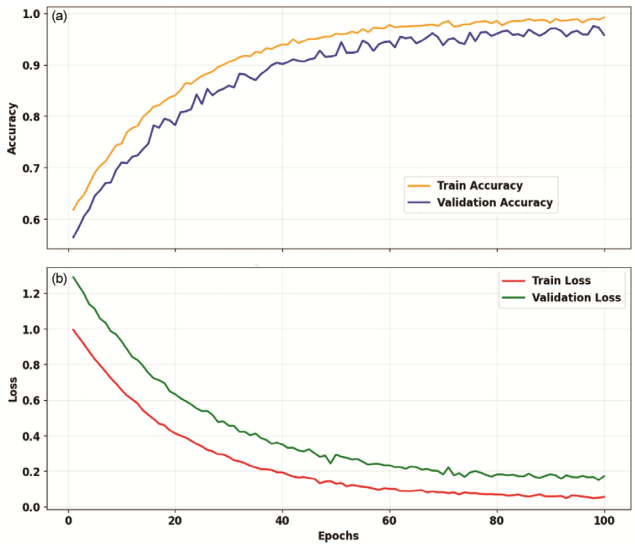

Fig. 5 — Training and validation over epochs of ESC-50: (a) accuracy, (b) loss

Comparison Results with Baseline Optimization

A comparison was conducted between the proposed EcoOptiNet and commonly used optimizers in deep learning, namely SGD, RMSprop, and Adam, with their respective default learning rate values of 0.001, and a fixed number of epochs set to 100.The state-of-the-art performance of the proposed EcoOptiNet optimizer on low-frequency time series data is shown in Table 7. All optimizers were evaluated on both datasets.

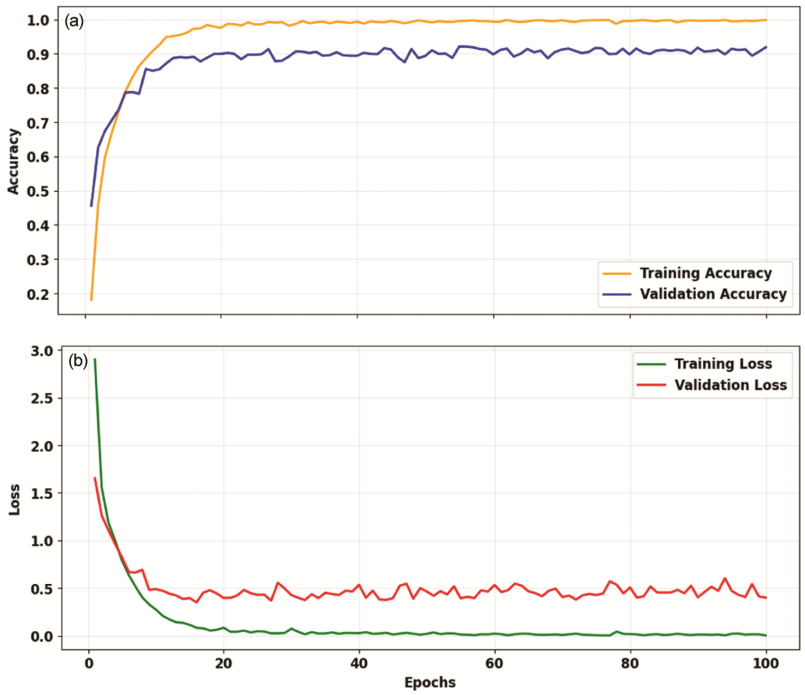

Fig. 6 — Training and validation over epochs of UrbanSound8K: (a) accuracy, (b) loss

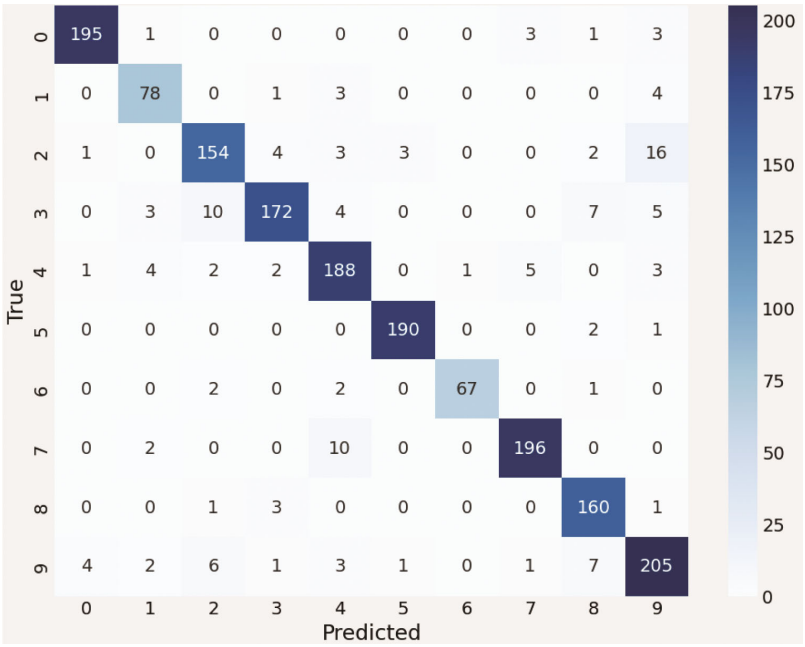

Fig. 7 — Confusion matrix obtained using the proposed EcoOptiNet optimization algorithm for UrbanSound8K datasets

The results indicate that the proposed EcoOptiNet on ESC-50 achieved an average accuracy rate of 99%, significantly outperforming SGD (62%), RMSprop (49%), and Adam (71%), and UrbanSound8k achieved 63% from SGD, 90% from RMSprop, 88% from Adam, and 91% from EcoOptiNet. The superior performance can be attributed to two key theoretical aspects: (i) the incorporation of a bias correction mechanism and adaptive learning rate, which stabilizes training for non-stationary signals, and (ii) the use of an exponential moving average of cube gradients instead of squared gradients, which allows for better gradient adaptation and faster convergence. These enhancements

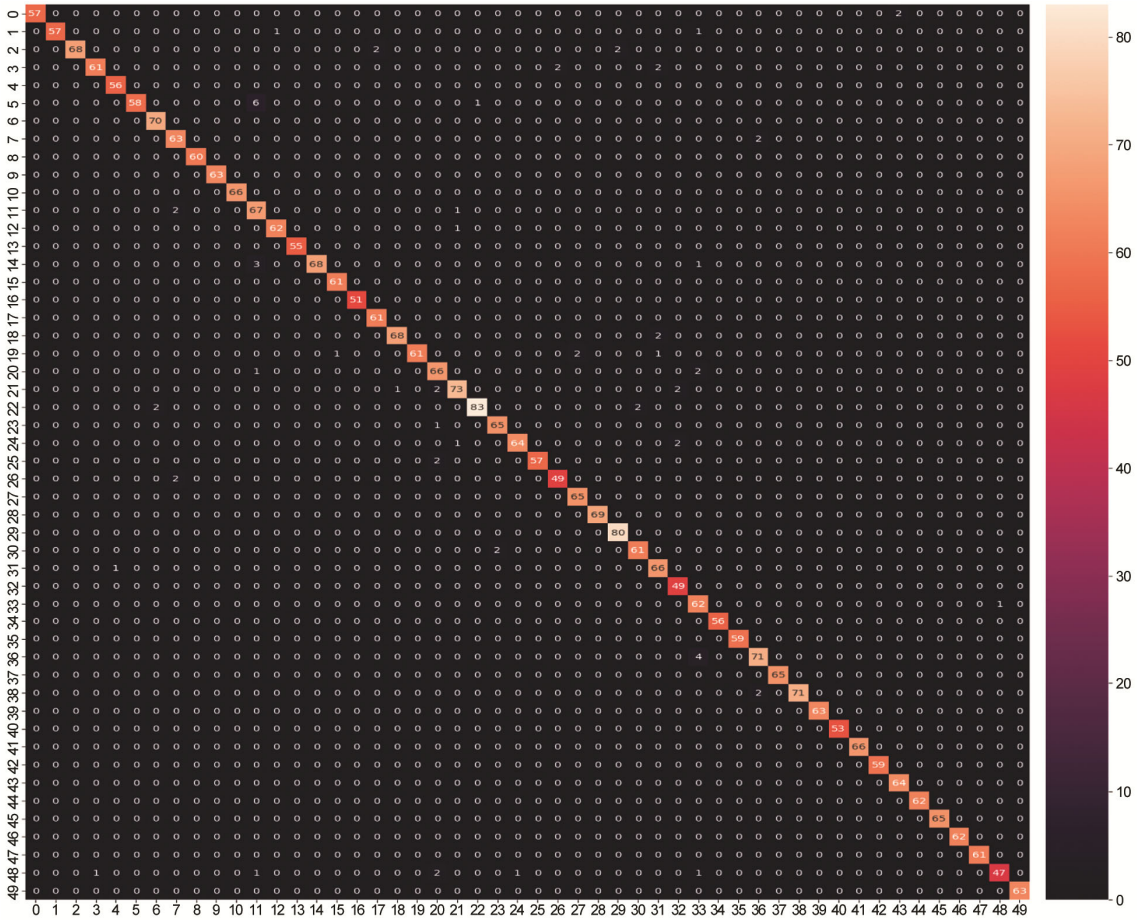

Fig. 8 — Confusion matrix obtained using the proposed EcoOptiNet optimization algorithm for ESC-50 dataset

Table 7 — State–of–the–State-of-the-art comparison for the proposed EcoOptiNet on ESC-50 Datasets and UrbanSound8K

Datasets	Optimizer	Loss Function	Accuracy (%)	Precision(%)	Recall(%)	F1-score(%)
ESC-50	SGD	1.3579	62	65	62	63
	RMSprop	1.9402	49	57	49	50
	Adam	1.0214	71	74	71	72
	EcoOptiNet	0.8100	99	99	99	99
UrbanSound8K	SGD	1.1109	63	66	63	63
	RMSprop	0.0016	90	90	90	90
	Adam	0.0218	88	89	88	89
	EcoOptiNet	0.0032	91	91	91	91

enabled EcoOptiNet to achieve both higher accuracy and lower loss compared to conventional optimizers.

Statistical Robustness of EcoOptiNet

To assess the statistical robustness of EcoOptiNet, the mean and standard deviation (SD) of accuracy and loss were computed across the validation folds for both datasets. On ESC-50, EcoOptiNet achieved an average accuracy of 98.83 % ± 0.40, with a mean validation loss of 0.82 ± 0.03, indicating stable learning across all five folds. On UrbanSound8K, the optimizer obtained an average accuracy of 91.70% ± 0.60and a mean validation loss of 1.15 ± 0.05over ten folds. These minor standard deviations confirm that the model maintains consistent performance and is less prone to overfitting, even on datasets with higher intra-class variability, as shown in Table 8.

Comparison Results among the Different Methods Applied to CNN

Classifying the low-frequency audio signal time series data cannot be accomplished using RGB channels. Instead, the Mel-spectrogram is employed for feature extraction, which divides the features into two components: the spectrogram and the delta. These

features are extracted from the low-frequency time-series signal and applied to a CNN, as CNNs are widely used for data prediction in deep learning. Various methods applied to CNN, such as data augmentation, Adam optimizer, and batch normalization, are compared in Table 9.

Discussion

This Section discusses the performance of the proposed EcoOptiNet optimizer in comparison with existing methods. The results obtained on ESC-50 and UrbanSound8K demonstrate that EcoOptiNet consistently outperforms baseline optimizers in terms of accuracy, loss reduction, and stability, confirming its suitability for low-frequency environmental sound classification.

The updated value of θ in the EcoOptiNet exhibits the highest accuracy rate compared to previous works. As shown in Table 10, the proposed work is compared with earlier results.

The superior performance of EcoOptiNet can be attributed to:

- Abias-correction mechanism, which minimizes the impact of outdated gradients and stabilizes learning.
- The cubed-gradient update strategy, which enhances stability when optimizing non-stationary, low-frequency signals.

Table 8 — Statistical robustness of EcoOptiNet across validation folds

Dataset	Accuracy (mean SD)	Loss (mean SD)
ESC-50	98.83% ± 0.40	0.82 ± 0.03
UrbanSound8K	91.00% ± 0.60	0.0032 ± 0.0004

Table 9 — Comparison among the different methods applied to CNN for both datasets ESC-50 and UrbanSound8K

Datasets	Model	Accuracy	Loss	F1 Score	Precision	Recall
ESC-50	Data augmentation +CNN +Adam optimizer	71.07%	1.1470	0.72	0.75	0.71
	Data augmentation+ CNN +Adam optimizer+ Batch normalization	72.85%	0.9647	0.75	0.78	0.74
	Data augmentation +CNN + Adam optimizer +Batch normalization +Max pooling +same padding	76.71%	0.8268	0.78	0.81	0.77
	Data augmentation +CNN + Adam optimizer+ Batch normalization + Max pooling + same padding + Transfer learning	82.29%	0.6590	0.83	0.86	0.82
	Data augmentation +CNN+ EcoOptiNet + Batch normalization + Max pooling + same padding	99%	0.0800	0.99	0.99	0.99
UrbanSound8K	Data augmentation +CNN +Adam optimizer	65.08%	1.3500	0.66	0.66	0.65
	Data augmentation+ CNN +Adam optimizer+ Batch Normalization	65.88%	1.2200	0.66	0.69	0.66
	Data augmentation +CNN + Adam optimizer +Batch normalization +Max pooling +same padding	72.50%	1.0500	0.74	0.76	0.74
	Data augmentation +CNN + Adam optimizer+ Batch normalization + Max pooling + same padding + Transfer learning	78.80%	0.8900	0.79	0.81	0.79
	Data augmentation +CNN+ EcoOptiNet + Batch normalization + Max pooling + same padding	91%	0.3900	0.91	0.91	0.91

Table 10 — The state-of-the-art in comparison with the proposed EcoOptiNet Optimization

Paper name[Ref]	Year	Methodology	Accuracy
An Ensemble of Convolutional Neural Networks for Audio Classification[26]	2021	Pre-trained CNN	88.65%
BEATs: Audio Pre-Training with Acoustic Tokenizers[27]	2022	Pre-trained with acoustic tokenizers	98.10%
A High-Accuracy and Low-Power CNN-Based Environmental Sound Classification Processor[28]	2023	a big-small CNN-based reconfigurable	84.5%
A CNN Sound Classification Mechanism Using Data Augmentation [29]	2023	CNN with data augmentation	97%
ESC-NAS: Environment Sound Classification Using Hardware-Aware Neural Architecture Search for the Edge [30]	2024	HW-NAS process	81%
Robust technique for environmental sound classification using a convolutional recurrent neural network[31]	2024	MFCCs and CRNN	93.58%
BRIDLE: Generalized Self-supervised Learning with Quantization[32]	2025	Bidirectional Residual Quantization Interleaved Discrete Learning Encoder	94.95
Mobile Acoustic Net: A novel early detection model for wood-boring pests[33]	2025	lightweight CNN	96.136
Proposed work: Enhancing Environmental Sound Classification with EcoOptiNet: A CNN-Based Approach for Low-Frequency Signals	2025	CNN with EcoOptiNet Optimization	91-99%

- Robustness across datasets– EcoOptiNet achieved higher accuracy and lower variance than

Overall, these findings validate the effectiveness of EcoOptiNet in handling the challenges of low-frequency environmental sound data, making it a promising optimization approach for deep learning applications in similar domains.

Conclusions

The research work introduces the proposed EcoOptiNet, an optimization algorithm developed for low-frequency and non-stationary time series signals. It incorporates bias-correction and cubed-gradient updates, which improve training stability and support reliable learning in challenging signal environments. The algorithm demonstrates strong capability in extracting meaningful spectral–temporal features from environmental sound data when combined with a CNN framework. Overall, EcoOptiNet provides an effective and robust solution for optimizing deep neural networks in real-world audio recognition tasks, where traditional approaches often face limitations due to signal variability and instability. EcoOptiNet can also be applied in different areas like biomedical signal analysis, speech and speaker recognition, and smart monitoring systems in IoT-based environments. Further research will involve extending EcoOptiNet with adaptive learning-rate schedules, momentum annealing, or layer-wise parameter scaling to improve its efficiency on highly non-stationary datasets. Hybrid architectures, such as CNN–Transformer or CNN–RNN pipelines, will be explored to leverage both local and global temporal dependencies. Incorporating regularization mechanisms like gradient clipping or weight decay tuned for cubed-gradient updates further enhances generalization.

Datasets link:
ESC-50:
https://www.kaggle.com/datasets/mmoreaux/environmental-sound-classification-50
Urbansound8K:
https://urbansounddataset.weebly.com/urbansound8k.

References

1 Huzaifah M, Comparison of time-frequency representations for environmental sound classification using convolutional neural networks, *arXiv preprint*, (2017) https://doi.org/10.48550/arXiv.1706.07156.

2 Frank R J, Davey N & Hunt S P, Time series prediction and neural networks, *J Intell Robot Syst*, 31(1–3) (2001) 91–103, https://doi.org/10.1023/A:1012074215150.

3 Besbes O, Gur Y & Zeevi A, Non-stationary stochastic optimization, *Oper Res*, 63(5) (2015) 1227–1244, https://doi.org/10.1287/opre.2015.1408.

4 Hashim F A & Hussien A G, Snake optimizer: A novel meta-heuristic optimization algorithm, *Knowl Based Syst*, 242 (2022) 108320, https://doi.org/10.1016/j.knosys. 2022. 108320.

5 Salehan A & Deldari A, Corona virus optimization (CVO): a novel optimization algorithm inspired from the corona virus pandemic, *J Supercomput*, 78(4) (2022) 5712–5743, https://doi.org/10.1007/s11227-021-04100-z.

6 Abdelhamid A A, El-Kenawy E S M, Khodadadi N, Mirjalili S, Khafaga D S, Alharbi A H, Ibrahim A, Eid M M & Saber M, Classification of monkeypox images based on transfer learning and the Al-Biruni Earth radius optimization algorithm, *Mathematics* 10(19) (2022) 3614, https://doi.org/10.3390/math10193614.

7 Duchi J & Singer Y, Adaptive subgradient methods for online learning and stochastic optimization, *J Mach Learn* Res, 12 (2011) 2121–2159, https://jmlr.org/papers/v12/duchi11a.html.

8 Reddi S J, Kale S & Kumar S, On the convergence of adam and beyond, *Proc 6th Int Conf Lear Represent (ICLR)* 2018, https://arxiv.org/pdf/1904.09237.

9 Zeiler M D, ADADELTA: An adaptive learning rate method, *arXiv preprint*, 2012, https://arxiv.org/abs/1212.5701.

10 Mitchell D, Ye N & De Sterck H, Nesterov acceleration of alternating least squares for canonical tensor decomposition: Momentum step size selection and restart mechanisms, *Numer Linear Algebra Appl*, 27(4) (2020) e2297, https://doi.org/10.1002/nla.2297.

11 P R & Azhagiri M, Deep learning based predicting urban traffic congestion with RGB-coded images using GRU-CNN and LSTM, *Multimedia Tools Appl*, 83(38) (2024) 86261–8680, https://doi.org/10.1007/s11042-024-20376-8.

12 Gal Y & Ghahramani Z, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, *Proc 29th Int Conf Neural Inf Process Systs (NeurIPS)*, (2016) 5–10, https://proceedings.mlr.press/v48/gal16.html.

13 Kuo C C J, Understanding convolutional neural networks with a mathematical model, *J Vis Commun Image Represent*, 41 (2016) 406–413, https://doi.org/10.1016/j.jvcir.2016.11.003.

14 Chen Y, Guo Q, Liang X, Wang J & Qian Y, Environmental sound classification with dilated convolutions, *Appl Acoust,* 148 (2019) 123–132, https://doi.org/10.1016/j.apacoust. 2018.12.019.

15 Gong Y, Chung Y A & Glass J, AST: Audio spectrogram transformer, *Proc Annu Conf Intl Speech Commun Assoc*, *(INTERSPEECH)*, 2021, 1–60, https://arxiv.org/pdf/2104.01778.

16 Ramírez J & Flores M J, Machine learning for music genre: Multifaceted review and experimentation with audioset, *J Intell Inf Syst*, 55(3) (2020) 469–499, https://doi.org/10.1007/s10844-019-00582-9.

17 Bian W, Wang J, Zhuang B, Yang J, Wang S & Xiao J, Audio-based music classification with DenseNet and data augmentation, *Proc Int Conf Artif Intell Stat (AISTATS)*, 11672 (2019) 56–65, https://doi.org/10.1007/978-3-030-29894-4_5.

18 Jing L, Liu B, Choi J, Janin A, Bernd J & Mahoney M W, DCAR: A discriminative and compact audio representation for audio processing, IEEE Trans Multimedia, 19(12) (2017) 2637–2650, 10.1109/TMM.2017.2703939.

19 Ye J, Kobayashi T, Wang X, Tsuda H & Murakawa M, Audio data mining for anthropogenic disaster identification: An automatic taxonomy approach, *IEEE Trans Emerg Top Comput*, 8(1) (2020) 126–136, 0.1109/TETC.2017.2700843

20 Green M & Murphy D, Environmental sound monitoring using machine learning on mobile devices, *App Acoust*, 159 (2020) 107041, https://doi.org/10.1016/j.apacoust.2019.107041.

21 Crocco M, Cristani M, Trucco A & Murino V, Audio Surveillance: A systematic review, *ACM Comput Surv (CSUR),* 48(4) (2016) 1–46, https://doi.org/10.1145/2871183.

22 Piczak K J, Environmental sound classification with convolutional neural networks, *IEEE Int Workshop Mach Lear Signal Process (MLSP)* (2015) 1–6, https://ieeexplore.ieee.org/document/7324330.

23 Nasef M M, Nabil M M & Sauber A M, Multiclass environmental sound classification model based on adding residual connections to self-attention layers, *Multimedia Tools Appl*, 83(28) (2024) 71359–71377, https://doi.org/10.1007/s11042-024-18421-7.

24 Xu J, Zhang Y & Miao D, Three-way confusion matrix for classification: A measure driven view, *Inf Sci (N Y)*, 507 (2020) 772–794, https://doi.org/10.1016/j.ins.2019.06.064.

25 Yacouby R & Axman D, Probabilistic extension of precision, recall, and F1 score for more thorough evaluation of classification models, *Multimedia Tools Appl*, 79 (2020) 79–91, 10.18653/v1/2020.eval4nlp-1.9.

26 Nanni L, Maguolo G, Brahnam S & Paci M, An ensemble of convolutional neural networks for audio classification, *Appl Sci*, 11(13) (2021) 5796, https://doi.org/10.3390/app11135796.

27 Chen S, Wu Y, Wang C, Liu S, Tompkins D, Chen Z & Liu Q, BEATs: Audio pre-training with acoustic tokenizers, *Proc Mach Learn Res*, 202 (2022) 4672–4712, https://doi.org/10.48550/arXiv.2212.09058.

28 Peng L, Yang J, Chen Z, Yan L, Jiao X, Xiao J & Wang F, A high accuracy and low power CNN-based environmental sound classification processor, *IEEE Trans Circuits and Syst I Regul Pap*, 70(12) (2023) 4865–4876, 10.1109/TCSI.2023.3299823.

29 Chu H C, Zhang Y L & Chiang H C, A CNN sound classification mechanism using data augmentation, *Sensors*, 23(15) (2023) 6972, https://doi.org/10.3390/s23156972.

30 Ranmal D, Ranasinghe P, Paranayapa T, Meedeniya D & Perera C, ESC-NAS: Environment sound classification using hardware-aware neural architecture search for the edge, *Sensors*, 24(12) (2024) 3749, https://doi.org/10.3390/s24123749.

31 Bansal A & Garg N K, Robust technique for environmental sound classification using convolutional recurrent neural network, *Multimedia Tools Appl*, 83(18) (2024) 54755–54772, https://doi.org/10.1007/s11042-023-17066-2.

32 Nguyen H M, Shukla S N, Zhang Q, Yu H, Roy SD, Tian T & Li Y, BRIDLE: Generalized self-supervised learning with quantization, *arXiv preprint*, (2025) https://doi.org/10.48550/arXiv.2502.02118.

33 Min W, Zhai M, Chen S, Huang L, Wang F & Zhu T, Mobile acoustic net: A novel early detection model for wood-boring pests, *Comput Electron Agric*, 229 (2025) 109699, https://doi.org/10.1016/j.compag.2024.109699.