IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 November 2024, accepted 17 December 2024, date of publication 23 December 2024,
date of current version 30 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3521226

RESEARCH ARTICLE

Gyroscope-Based Smartphone Model Identification via WaveNet and EfficientNetV2 Ensemble

ERKAN KIYMIK AND ALİ EMRE ÖZTÜRK
Department of Electrical and Electronics Engineering, Hasan Kalyoncu University, 27010 Gaziantep, Türkiye
Corresponding author: Erkan Kıymık (erkan.kiymik@hku.edu.tr)

ABSTRACT Smartphone model detection through sensor data is important for enhancing security protocols, preventing device fraud, and ensuring authorized service access. While extensive research has utilized sensors like cameras, microphones, accelerometers, and magnetometers for device fingerprinting, gyroscope data has remained largely unexplored for model detection due to its high susceptibility to noise from small vibrations and mechanical imperfections. This study investigates the use of gyroscope data alone for smartphone model detection. Leveraging the Google Smartphone Decimeter Challenge 2023-2024 dataset, which provides real-world gyroscope data from multiple smartphones mounted identically on vehicles during various driving tests, the challenging task of distinguishing between smartphone models under nearly identical motion conditions is addressed. A fine-tuned WaveNet model is employed to analyze the sequential nature of the gyroscope data, and an EfficientNetV2 model captures intricate time-frequency representations using Continuous Wavelet Transform (CWT) with the Morlet wavelet. Combining these models in an ensemble framework enhanced with an attention mechanism gives an accuracy of 99.01% using just 1–2 seconds of gyroscope data. This performance suggests that gyroscope data alone can be effective for model identification, even under challenging real-world conditions. These findings indicate the potential of gyroscope data in device fingerprinting and may provide a foundation for future advancements in mobile device security and authentication.

INDEX TERMS Gyroscope, accelerometer, fingerprint, smartphone, neural network, machine learning, model ensemble, feature extraction.

I. INTRODUCTION

Smartphones are continually rising because they enhance daily life by providing convenient access to various applications, including Internet of Things (IoT) device management, education, healthcare, sports, and financial services. Naturally, smartphones that enhance quality of life gather extensive personal information through various sensors, including cameras, microphones, accelerometers, magnetometers, gyroscopes, and radio frequency sensors. These sensors are also employed for device fingerprinting [1], a field with extensive research focused on enhancing security protocols, preventing device fraud, safeguarding sensitive data, and ensuring that only authorized devices can access specific services [2], [3], [4]. Smartphone fingerprinting encompasses two primary approaches: individual smartphone detection and smartphone model detection. In individual smartphone detection, sensor data is leveraged to recognize specific devices based on unique usage patterns and behavioral signatures of individual users. This approach utilizes data from sensors such as accelerometers, gyroscopes, and magnetometers to capture personal routines, physical handling, and other subtle interactions with the device, making it possible to identify the unique usage profile of a specific person's smartphone. In contrast, smartphone model detection aims to distinguish between different smartphone makes and models, often by identifying subtle variations in sensor outputs, manufacturing imperfections, and signal noise. This approach is

The associate editor coordinating the review of this manuscript and approving it for publication was Hasan Al-Marzouqi.

typically model-agnostic, relying on intrinsic device features to create unique signatures for each model. Both techniques are essential for enhancing security, preventing unauthorized access, and verifying devices in various applications, from personal authentication to model-based device classification in large-scale systems.

In general, smartphone device fingerprinting studies primarily focus on cameras, microphones, and accelerometer sensors. Extensive research has been conducted on these sensors, yielding successful results. These sensors exhibit unique patterns, manufacturing defects, and distinct noise characteristics, which can be leveraged as distinguishing features for device identification. Sensors can be utilized individually or in combination to enhance the robustness of device fingerprints, and research has explored both approaches.

In addition to the accelerometer, the Inertial Measurement Unit (IMU) in smartphones includes a magnetometer and a gyroscope, though fewer studies focus on these sensors. A gyroscope is a sensor that measures angular velocity across three axes (x, y, z), enabling the detection of rotational movement. In smartphones, gyroscopes provide critical orientation and movement tracking data, supporting features such as screen rotation, gaming, and augmented reality [5], [6]. The physical principle underlying smartphone gyroscopes is the Coriolis force [7]. When the gyroscope rotates around any axis, the Coriolis effect causes a slight vibration detected by capacitive pickups. This movement changes the capacitance, generating a signal proportional to the angular velocity. This signal is then amplified, demodulated, filtered, and digitized to produce an output in radians per second. The Coriolis force, defined by $F_k = 2m\omega\vartheta$ where m is the mass of the vibrating structure, ω the angular velocity, and ϑ the linear velocity along the driving direction—enables microelectromechanical systems (MEMS) gyroscopes to measure rotation efficiently. Due to their compact size, low cost, and low power consumption, MEMS gyroscopes are ideal for smartphone integration despite their susceptibility to noise compared to other sensors.

The gyroscope is more noise-sensitive than the accelerometer and magnetometer due to small vibrations and mechanical imperfections, which can result in inconsistent outputs across different smartphone brands and models. This sensitivity poses unique challenges for device fingerprinting. The paper ''Smartphone MEMS Accelerometer and Gyroscope Measurement Errors'' demonstrates that gyroscopes suffer more from noise than accelerometers [8]. This noise significantly impacts positioning accuracy, with gyro biases causing errors that grow cubically over time t^3, represented by $\delta_r = \frac{1}{6} b_g g t^3$, compared to accelerometer biases that grow quadratically t^2, represented by $\delta_r = \frac{1}{6} b_a g t^2$. This highlights the gyroscope's higher noise susceptibility. The parameters in the error growth equations are: δ_r for position error, b_g for gyroscope bias, b_a for accelerometer bias, g for gravitational acceleration (approximately 9.81 m/s^2), and t for time.

A. FINGERPRINTING OF SMARTPHONES WITH MACHINE LEARNING AND DEEP LEARNING

Machine learning algorithms, particularly deep learning, have emerged as crucial techniques for device fingerprint detection on smartphones, driven by the increasing availability of data and enhanced processing power. As smartphones become integral to our daily lives, secure and accurate identification is becoming increasingly important. Traditional identification methods often fall short in countering sophisticated identity fraud techniques, necessitating more advanced and reliable solutions. In this context, machine learning, especially deep learning methods, plays a crucial role.

Machine learning and deep learning algorithms offer robust solutions by extracting patterns from sensors such as accelerometers, gyroscopes, magnetometers, cameras, and microphones [1], [9]. The production errors in sensors, the data processing methods of smartphones, and the inherent noise characteristics contribute to a unique signature for each smartphone model. This distinctive signature can be effectively utilized as a device fingerprint. Machine learning algorithms like Support Vector Machines (SVM) [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], K-Nearest Neighbors (KNN) [10], [11], [28], [29], [30], [31], Decision Trees [28], [32], [33], [34], and Random Forests [11], [12], [35], Gaussian mixture model (GMM) [29], [36], [37], [38], [39], Gaussian supervector (GSV) [40], [41], as well as deep learning architectures such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, have been employed to analyze these data patterns.

CNNs are designed to extract features and patterns from images, making them particularly effective for image analysis. CNNs are the most commonly used algorithms for camera-based fingerprinting and identification of smartphones [21], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]. One-dimensional signals, such as audio signals, are also suitable for CNN models. Therefore, audio signals captured by smartphone microphones can be effectively utilized for device fingerprinting using CNN models [14], [53], [54], [55]. LSTM and Bidirectional LSTM (Bi-LSTM) are recurrent neural network layers designed for analyzing time series and sequence data. These models have been employed for microphone [55] and loudspeaker identification [20], [56].

Using these techniques with the appropriate sensor or sensors can detect devices, thereby improving security protocols, preventing device spoofing, and protecting user privacy. Machine learning and deep learning algorithms use large amounts of data to enable identification and verification with high precision and reliability. Although large data usage is considered a disadvantage, it helps solve security issues and pave the way for future innovations in mobile device management and security.

B. LITERATURE REVIEW ON ACCELEROMETER, GYROSCOPE, AND MAGNETOMETER DATA

Camera and microphone sensors are the most commonly used for device fingerprinting, with numerous reliable studies documented in the literature. The accelerometer, a key sensor in the IMU, has also been extensively studied, nearly as much as the camera and microphone. Table 1 and Figure 1 show several studies on smartphone fingerprinting with various transducers.

TABLE 1. Number of the works given in the literature [1].

Transducer	Number of papers for fingerprinting of smartphones
Camera	27
Microphone	21
Loudspeaker	5
Accelerometer	13
Gyroscope	7
Magnetometer	5
Other sensors	14

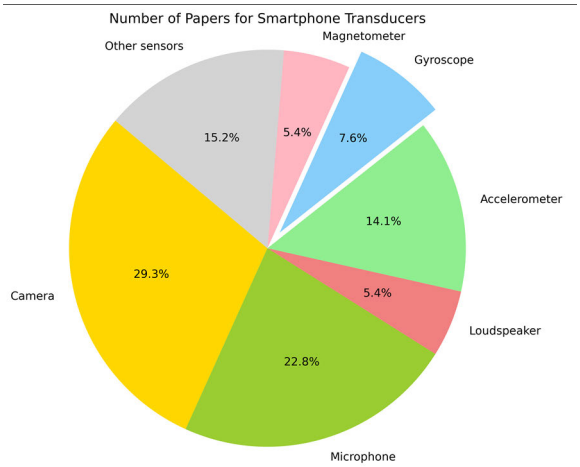

FIGURE 1. A number of papers are based on smartphone transducers for fingerprinting.

Accelerometers have been extensively studied and are considered a reliable option for device fingerprinting. In contrast, gyroscopes face challenges due to their higher noise levels, which are caused by sensitivity to small vibrations and bias instability. These challenges make gyroscopes less reliable for standalone device fingerprinting, and as a result, they are often used in combination with other sensors, such as magnetometers, to improve accuracy. The following two sections provide a summary of the relevant literature. A comparative overview of the literature on these sensors is presented in Table 2.

C. GYROSCOPE-BASED STUDIES OF INDIVIDUAL PHONE DETECTION

1) DAS et al. (2015)

This work used accelerometer and gyroscope data to extract 25 time and frequency features. The classifiers tested included SVM, Naive Bayes (NB), Multiclass Decision Tree (MDT), KNN, Quadratic Discriminant Analysis (QDA), and Bagged Decision Trees (BDT). The study achieved an F1 score of 96% using data from 30 devices [28].

2) DAS et al. (2016)

Similar to the previous study, this research also used accelerometer and gyroscope signals with 26 time and frequency features. The same classifiers were employed, resulting in an F1 score of 96% for 30 devices, demonstrating the robustness of these features and classifiers [32].

3) AMERINI et al. (2016)

This study extended fingerprinting by incorporating camera data alongside accelerometer and gyroscope signals. Using 21 time and frequency features for the accelerometer and gyroscope and Photo-Response Non-Uniformity (PRNU) for the camera, a decision tree classifier achieved an F1 score greater than 75% with data from 10 devices [57].

4) HUPPERICH et al. (2016)

Utilizing a broader range of sensors (acceleration, magnetic field, orientation, gyroscope, rotation vector, gravity, and linear acceleration), this work extracted time and frequency features and employed KNN, SVM, Bagged Tree, Random Forest (RF), and Extra Tree classifiers. The study achieved a remarkably high accuracy of 99.995% using data from 5000 devices [58].

5) BALDINI et al. (2016)

Focusing on accelerometer and gyroscope data, this study introduced novel features such as Threshold Entropy, Sure Entropy, and Norm Entropy. The SVM classifier achieved accuracy greater than 90% with data from 3 devices [59].

6) BALDINI et al. (2017)

Magnetometer data was used in this study, with time and frequency features being extracted. The classifiers included SVM, KNN, and Bagged Decision Tree, resulting in an F1 score of 90.7% with data from 9 devices [33].

7) AMERINI et al. (2017)

This study combined data from an accelerometer, gyroscope, magnetometer, and microphone, using time and frequency features and employing NB and RF classifiers. Using data from 20 devices, an accuracy of 90% was achieved [60].

8) DAS et al. (2018)

Using accelerometer and gyroscope data, this comprehensive study extracted time and frequency features and tested various classifiers, including SVM, RF, Extra Tree, Logistic Regression (LR), Gaussian Naive Bayes (GNB), Stochastic Gradient Descent (SGD), KNN, Bagging, Linear Discriminant Analysis (LDA), and Multi-Layer Perceptron (MLP).

The study achieved a high accuracy of 98% with data from 550 devices [11].

9) LI et al. (2019)

This study utilized accelerometer and gyroscope data with a novel multidimensional Balls-into-Bins approach and multi-LSTM classifiers, achieving accuracy greater than 98.8% using data from 117 devices [61].

10) ZHANG et al. (2019)

Focused on gyroscope and magnetometer data, this study introduced the sensorID technique for ADC Value Estimation, achieving 67 bits of entropy with data from 797 devices [62].

11) ZHANG et al. (2020)

Using accelerometer, gyroscope, and magnetometer data, this study employed the sensorID technique for ADC Value Estimation. The entropy achieved was 42 bits for the gyroscope and 25 bits for the magnetometer, using data from 1006 devices [63].

12) TIAN et al. (2021)

This recent study focused on gyroscope resonance peak features and utilized decision tree and regression tree classifiers, achieving an accuracy of 96.5% with data from 25 devices [34].

13) AL-MAHADEEN et al. (2023)

This study focused on smartphone user identification and authentication using accelerometer and gyroscope data from the HMOG dataset. Feature extraction included statistical metrics, such as skewness, kurtosis, and autocorrelation, and Recursive Feature Elimination (RFE) was identified as the optimal feature-selection method. Classifiers tested included ANN, SVM, XGBoost, and LSTM, with LSTM achieving the best results. LSTM attained an accuracy of 89% for identification and near-perfect results when both sensors were used for authentication, demonstrating the effectiveness of this approach for secure, personalized smartphone applications [64].

D. STUDIES BASED ON MODEL DETECTION

1) HANILÇI et al. (2012)

This study uses mel-frequency cepstrum coefficients (MFCCs) to identify cell phone brands and models from recorded speech signals. Using speech samples from 14 different phones, it achieved closed-set identification rates of 92.56% with vector quantization (VQ) and 96.42% with support vector machines (SVM). Each 3-second voice sample was recorded in a silent environment, and tests were conducted using the TIMIT database, ensuring speaker-independent performance. The results demonstrate the potential of MFCC features in distinguishing devices based on unique audio artifacts introduced during recording [65].

2) GARCÍA VILLALBA et al. (2016)

This paper presents a method for identifying smartphone brands and models from forensic video analysis. Using sensor pattern noise (SPN) and wavelet transform features extracted from keyframes, the study analyzes videos from five smartphone models, each with 2-minute samples at 1080p resolution. An SVM classifier was employed for identification, achieving an average success rate of 90.3% with higher success rates for larger frame crops, demonstrating the effectiveness of SPN in differentiating video sources in forensic investigations [66].

3) BALDINI et al. (2017)

This paper explores mobile phone identification using built-in magnetometers stimulated by a rotating platform with a fixed magnet. Statistical features, including variance, skewness, kurtosis, and entropy, were extracted from the magnetometer data across three axes, and an SVM was used for classification. The study involved ten mobile phones and recorded magnetometer readings over six days with a 60-second sample duration. The system achieved 98.07% accuracy for inter-model classification but had a lower intra-model accuracy of 54.15% [67].

4) HOSLER et al. (2019)

This study presents a video camera model identification system using deep learning, specifically a CNN trained on patches from video frames. The system extracts features from 256×256 pixel patches, chosen across multiple frames to increase temporal diversity. The CNN is combined with a fusion mechanism to improve identification accuracy. Testing on 20 different camera models, each with 5-second video samples, the system achieved an overall accuracy of 95.9%. The approach highlights that patches from I-frames yield more reliable identification than P-frames [68].

5) JIANG AND LEUNG (2019)

This study focuses on microphone recognition using GSV as feature vectors for speech recordings, representing microphone response and speech content. The model achieves improved recognition performance by applying a kernel-based projection method to separate microphone-specific artifacts from speech information. Tested on four microphone models from the Ahumada-25 dataset, with 2- to 5-second speech samples, the projected GSV outperformed raw GSV with accuracies up to 87.98% using Sparse Representation Classifier (SRC) and 89.04% using SVM, demonstrating the method's effectiveness in identifying microphone models under varying conditions [16].

6) LIN et al. (2020)

This study introduces a subband-aware CNN with channel attention for cell phone model identification from audio recordings. Spectrograms of 2-second audio clips,

sampled at 16 kHz, were split into eight subbands to capture device-specific frequency artifacts. Tested on recordings from 20 phone models across five brands, the model achieved over 95% accuracy for inter-brand classification and maintained robust performance for intra-brand classification with data augmentation. Performance under noise conditions improved using data augmentation and the attention mechanism, showing resilience even at low SNR levels [54].

7) COZZOLINO AND VERDOLIVA (2020)

This study presents ''Noiseprint,'' a CNN-based method for camera model identification by extracting unique noise residuals from images, designed to suppress scene content while enhancing camera-specific artifacts. Using a Siamese network architecture, Noiseprint is trained on patches of images from 70 camera models (125 cameras in total). This method achieved up to 99% accuracy in camera model identification on a diverse test set, even handling compressed JPEG images effectively. It also performs well in forgery localization by identifying manipulation inconsistencies in noise patterns within image patches [50].

8) ZHAO et al. (2020)

This study introduces a deep learning-based source camera identification system using coupling coding and an adaptive filter to enhance the identification of camera brands, models, and specific devices. The approach relies on image blocks (48×48 pixels) processed through a multi-layer CNN to extract features while reducing content noise. Tested on 74 devices across 14 brands using images from the Dresden database, the model achieved an accuracy of 96% for model detection. This method demonstrates scalability and robustness, effectively handling multi-class camera identification tasks [52].

9) DAL CORTIVO et al. (2021)

This study introduces a CNN-based multi-modal approach to identify smartphone camera models from video sequences. Using both visual patches (256×256 pixels) and audio features (Log-Mel Spectrogram patches) from 1-minute video samples, the model achieves up to 99% accuracy on native videos and maintains strong performance on compressed versions (YouTube and WhatsApp). Evaluations were conducted on 25 camera models from the Vision dataset, showing that the fusion of visual and audio data improves model identification over single-modality methods [44].

10) SALVI et al. (2024)

This paper presents the POLIPHONE dataset, designed for smartphone model identification from audio recordings, with data acquired in a controlled environment using 20 smartphone models. The dataset includes 50 hours of audio across various categories—speech, music, and environmental sounds—with recordings lasting up to 2 minutes each, captured at a 44.1 kHz sampling rate. Using a CNN-based classifier on POLIPHONE, the authors achieved nearly 100% accuracy in model identification on 44.1 kHz clean speech data, demonstrating the effectiveness of high-frequency content in distinguishing device models [69].

E. PROPOSED STUDY

In this study, we explore the novel application of gyroscope data for smartphone model detection, an area that has been largely overlooked in device identification research. While previous studies on individual phone detection have utilized gyroscope data, often in combination with accelerometers and magnetometers—to distinguish between individual devices [28], [32], [57], [58], [59], [60], [62], [63], the use of gyroscope data for differentiating between smartphone models remains relatively unexplored.

Most studies on smartphone model detection have relied on signals from cameras, microphones, and magnetometers [16], [44], [50], [52], [54], [65], [66], [67], [68], [69]. For instance, Baldini et al. [67] used magnetometer data to achieve 98.07% accuracy in inter-model detection over ten devices with 60 seconds of input duration. However, to our knowledge, gyroscope data has not been employed for model detection tasks in the literature.

Our proposed study aims to fill this gap by leveraging gyroscope data alone for smartphone model detection, highlighting the untapped potential of this sensor in distinguishing between different smartphone models. This is particularly challenging because gyroscopes are more susceptible to noise due to their sensitivity to small vibrations and mechanical imperfections, which can obscure subtle variations between different devices of the same model.

Our research utilizes the dataset from the Google Smartphone Decimeter Challenge (GSDC) 2023-2024 [70], which provides comprehensive real-world data collected from multiple smartphones mounted on vehicles during various driving tests. In this setup, all phones are aligned next to each other and securely mounted on the vehicle's chassis, ensuring they experience identical motion patterns. This configuration presents a significant challenge for model detection, as the gyroscope data from different devices are almost identical due to the uniform movement. Unlike individual phone detection studies that exploit variations in user activities—such as differentiating between a runner's phone and a non-athlete phone using IMU sensors—our study addresses the more challenging task of distinguishing between smartphone models under nearly identical motion conditions.

Our dataset includes gyroscope readings from six smartphone models and 118 devices recorded along the x, y, and z axes. The uniformity of the data collection conditions underscores the robustness required of our detection model to identify subtle differences in gyroscope signals attributable to hardware variations between different smartphone models.

To tackle this challenge, we employ a fine-tuned WaveNet model to analyze the sequential nature of the gyroscope data and an EfficientNetV2 model to capture intricate time-frequency representations of the data using Continuous

TABLE 2. Comparative studies of gyroscope-based individual phone detection and model detection techniques.

GYROSCOPE-BASED STUDIES OF INDIVIDUAL PHONE DETECTION								
No.	**Work**	**Year**	**Signal**	**Features**	**Classifier**	**Results**	**Devices**	**Input Duration**
1	[28]	2015	accelerometer + gyroscope	25 time and frequency features	SVM, NB, MDT, KNN, QDA, BDT	F1 96%	30	5-8 sec
2	[32]	2016	accelerometer + gyroscope	26 time and frequency features	SVM, NB, MDT, KNN, QDA, BDT	F1 96%	30	5-8 sec
3	[57]	2016	accelerometer + gyroscope + camera	21 time and freq. features for acc. and gyro., PRNU for camera	decision tree	F1>75%	10	2 sec
4	[58]	2016	acceleration, magnetic field, orientation, gyroscope, rotation vector, gravity, linear acceleration	time and frequency features	KNN, SVM, B Tree, RF, Extra Tree	99.995%	5000	10 secs
5	[59]	2016	accelerometer + gyroscope	Threshold Entropy, Sure Entropy, and Norm Entropyc	SVM	>90%	3	10 secs
6	[33]	2017	magnetometer	time and frequency features	SVM, KNN, Bagged Decision Tree	F1 90.7%	9	10 secs
7	[60]	2017	accelerometer + gyroscope + magnetometer + microphone	time and frequency features	NB, RF	90%	20	3 secs
8	[11]	2018	accelerometer + gyroscope	time and frequency features	SVM, RF, Extra tree, LR, GNB, SGD, KNN, Bagging, LDA, MLP	98%	550	5 secs
9	[61]	2019	accelerometer + gyroscope	multidimensional Balls-into-Bins	multi-LSTM	>98.8%	117	1-2 sec
10	[62]	2019	gyroscope + magnetometer	sensorID	ADC Value Estimation	67bits entropy	797	1-2 sec
11	[63]	2020	accelerometer + gyroscope + magnetometer	sensorID	ADC Value Estimation	gyro.:42 bits entropy, mag.: 25 bits	1006	1-2 sec
12	[34]	2021	gyroscope	resonance peak	decision tree, regression tree	96.5%	25	n/a
STUDIES BASED ON MODEL DETECTION								
No.	**Work**	**Year**	**Signal**	**Features**	**Classifier**	**Results**	**Devices**	**Input Duration**
1	[65]	2012	speech	MFCCs	VQ, SVM	VQ: 92.56%, SVM: 96.42%	14	3 sec
2	[66]	2016	video	SPN, wavelet transform	SVM	90.3%	5	2 min
3	[67]	2017	magnetometer	variance, skewness, kurtosis, entropy	SVM	inter-model: 98.07%, intra-model: 54.15%	10	60 sec
4	[68]	2019	video	frame patches (256x256 pixels)	CNN with fusion	95.9%	20	5 sec
5	[16]	2019	microphone	Gaussian Supervectors (GSVs)	SRC, SVM	SRC: 87.98%, SVM: 89.04%	4	2-5 sec
6	[54]	2020	audio	subband-aware spectrograms	CNN with channel attention	>95% inter-brand	20	2 sec
7	[50]	2020	image	noise residuals (Noiseprint)	Siamese CNN	99%	70	patches
8	[52]	2020	image	image blocks (48x48 pixels)	multi-layer CNN	96%	74	n/a

TABLE 2. *(Continued.)* **Comparative studies of gyroscope-based individual phone detection and model detection techniques.**

9	[44]	2021	video + audio	visual and Log-Mel Spectrogram patches	CNN-based multi-modal	99% on native, strong on compressed	25	1 min
10	[69]	2024	audio	high-frequency spectrograms	CNN	99% on clean speech	20	2 min

Wavelet Transform (CWT) with the Morlet wavelet [71], [72], [73]. Combining these models in an ensemble framework with an attention mechanism aims to achieve high accuracy in smartphone model detection under highly similar data from different devices.

Our approach demonstrates the feasibility of using gyroscope data for smartphone model detection and sets a new benchmark in the field. Since gyroscope data has not been previously used for this purpose, our study provides the first evidence that high accuracy can be achieved using only gyroscope data for smartphone model detection. We achieve an accuracy of 99.01% using just 1-2 seconds of gyroscope data. This result is particularly noteworthy compared to existing model detection studies that rely on other signals and often require longer input durations. For example, Hanilçi et al. [1] achieved 96.42% accuracy using 3 seconds of speech data, and Baldini et al. [67] achieved 98.07% accuracy using 60 seconds of magnetometer data.

Moreover, our study exclusively uses real-world datasets with devices mounted in identical positions experiencing identical motion patterns, emphasizing our model's robustness and predictive power in practical scenarios. The challenging conditions of our data collection underscore the effectiveness of our approach in detecting subtle hardware-induced variations between different smartphone models.

In summary, our contributions are as follows:

1) **Novel Application of Gyroscope Data**: To the best of our knowledge, this study is the first to demonstrate the potential of gyroscope data alone for smartphone model detection. It addresses an area in the literature where the use of gyroscope data has been limited.
2) **High Accuracy Under Challenging Conditions**: Our ensemble model achieves a remarkable accuracy of 99.01% using just 1-2 seconds of gyroscope data, outperforming existing model detection methods that often require longer input durations and additional sensor data.
3) **Robustness with Real-World Data**: By utilizing the GSDC 2023-2024 dataset, which includes devices mounted in identical positions experiencing identical motion patterns, we show that our model can detect subtle hardware-induced variations, demonstrating robustness and practical applicability.
4) **Efficient and Practical Approach**: Our method's brief input duration and reliance on a single sensor make it efficient and suitable for real-time applications. It offers a practical solution for smartphone model detection in various scenarios.

Our findings highlight the untapped potential of gyroscope data in device fingerprinting and open new avenues for research in smartphone model detection using IMU sensors.

II. DATASET AND FEATURE ENGINEERING

The dataset used in this study is from the GSDC 2023-2024 [70]. The GSDC competition provides a comprehensive dataset of accurately labeled GNSS and IMU data from multiple smartphones mounted on vehicles. The dataset is designed to drive innovation in mobile device localization and sensor-based identification.

GSDC collected data from 196 driving tests conducted in the San Francisco Bay Area and Los Angeles. These tests covered a variety of road types and environments to ensure the reliability and applicability of the findings. Each test involved multiple smartphones mounted near the vehicle's windshield alongside a high-precision real reference system, as shown in Figures 2 and 3.

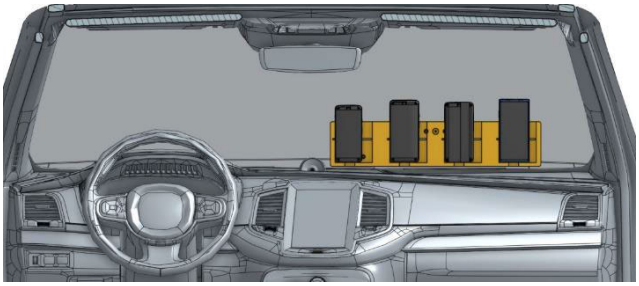

FIGURE 2. Schematic diagram of phone mounting to vehicle [70], [74].

196 driving tests provided by GSDC used 20 unique phone models. It's important to note that each driving test uses a different physical sample of the same phone model. For example, while one phone model was used on one trip, a separate, different phone model of the same type was used on another trip. This approach ensured that each phone model contributed unique data without being copied between trips.

In this study, six smartphone models were selected based on the availability of the largest datasets. Table 3 details the number of readings taken by each phone, the sampling frequencies for the gyroscope data for each smartphone, and the number of units used for each smartphone model. Two models were created for this study: fine-tuned WaveNet and the backbone of EfficientNetV2, which were then combined.

FIGURE 3. In-vehicle setup for gyroscope data collection using mounted smartphones [70], [74].

TABLE 3. Details of the gyroscope readings.

Phone Models	Readings (each x, y, z axes)	Smartphone count	Sampling Rate
Model-0	(1,956,421 readings)	22	50 Hz -75 Hz
Model-1	(1,360,804 readings)	15	50 Hz -75 Hz
Model-2	(1,505,327 readings)	13	100 Hz
Model-3	(4,159,190 readings)	39	100 Hz
Model-4	(1,575,958 readings)	15	100 Hz
Model-5	(2,384,974 readings)	14	100 Hz

A. INPUT OF WAVENET

Originally developed as a generative model for raw audio data, WaveNet has been adapted and fine-tuned to analyze gyroscope readings [71]. The input data for WaveNet consisted of 100-sample-long gyroscope sequences from the x, y, and z axes, corresponding to approximately 1–2 seconds of data, resulting in a model input shape of (100, 3). However, we decided to augment the feature set by incorporating additional informative attributes to achieve better results.

Recognizing that feature selection is a fundamental step in machine learning and deep learning across various domains, we aimed to identify the most impactful features to enhance model performance. In numerous fields, ranging from medical diagnosis [75], [76], text classification [77], [78], and image processing [79], [80] to security applications [81]—effective feature selection algorithms are essential for improving classification accuracy and computational efficiency. By selecting relevant features, models can focus on the most significant data patterns, leading to better generalization and faster convergence during training.

In the context of smartphone fingerprinting technologies, especially those based on accelerometer, gyroscope, and magnetometer data, selecting the right features is crucial to capturing the unique patterns of different devices. To this end, we introduced nine additional features, as detailed in the following table, commonly used in such applications to improve classification accuracy and robustness.

$$\text{Sequence} = A = a_1 + a_2 + \ldots + a_n \quad (1)$$

$$\text{Mean} = \frac{1}{n}\sum_{i=1}^{n} a_i \quad (2)$$

$$\text{Min} = \min(a_i), i \in \{1, 2, \ldots, n \quad (3)$$

$$\text{Max} = \max(a_i), i \in \{1, 2, \ldots, n \quad (4)$$

$$\text{Variance(Var)} = \frac{1}{n}\sum_{i=1}^{n} (a_i - Mean)^2 \quad (5)$$

$$\text{Standard Deviation(STD)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (a_i - Mean)^2} \quad (6)$$

$$\text{Range} = |max - min| \quad (7)$$

$$\text{Skewness} = \frac{1}{n}\sum_{i=1}^{n} \left(\frac{a_i - Mean}{Var}\right)^3 \quad (8)$$

$$\text{Kurtosis} = \frac{1}{n}\sum_{i=1}^{n} \left(\frac{a_i - Mean}{Var}\right)^4 \quad (9)$$

$$\text{Root Mean Square(RMS)} = \sqrt{\frac{1}{n}(a_1^2 + a_2^2 + \ldots + a_n^2)} \quad (10)$$

These features were selected based on their effectiveness in the literature on fingerprinting technologies [11], [33], [55], [57], [58], [60]. They have been shown to capture unique characteristics of sensor data that can distinguish between different devices. Each feature contributes to a comprehensive representation of the variations in the time-series data, enhancing the model's ability to identify distinct patterns in the gyroscope readings.

- **Sequence**: This refers to the windowed segments of the time series data, allowing the model to capture the temporal dynamics of the gyroscope signals. Using overlapping sequences ensures that temporal continuity is preserved while focusing on smaller segments that contain brand-specific characteristics.
- **Mean**: Represents the average value of the gyroscope readings within a sequence, providing a measure of the central tendency of the data.
- **Min and Max**: Capture the extreme values within each sequence, highlighting the range of variations in the sensor readings.
- **Variance and Standard Deviation (STD)**: Quantify the degree of variation or dispersion within the sequence, helping to differentiate devices based on the consistency or variability of their sensor outputs.
- **Range**: Calculated as the difference between the maximum and minimum values, this feature measures the spread of values, offering insight into the overall fluctuation within each sequence.
- **Skewness**: Measures the asymmetry of the data distribution, indicating whether the sensor readings are biased towards higher or lower values.

- **Kurtosis**: This assesses the peakedness or flatness of the data distribution, identifying sequences with extreme values or outliers.
- **Root Mean Square (RMS)**: This measure reflects the overall energy of the signal and provides a measure of the magnitude of fluctuations within the sequence.

These features were chosen because they provide a well-rounded description of the gyroscope data's statistical properties and dynamics. This combination enables the model to capture subtle device variations, which is critical for accurate smartphone model identification. By leveraging these statistical features, the study aims to achieve a more detailed and distinctive representation of each smartphone's gyroscope readings, ultimately improving the precision and reliability of the device fingerprinting process.

With nine new features derived from the x, y, and z gyroscope measurements, the input to the WaveNet model is structured as (100, 12), where 12 channels correspond to the three axes plus the additional statistical features. Each channel contains sequences of length 100, which represents a fixed window of 100 consecutive samples from the time series data. The choice of a 100-sample window length is informed by comparisons with other sensor fusion fingerprinting methods in the literature, ensuring consistency in evaluating the effectiveness of different approaches. This window length enables the model to capture the temporal dynamics within each sequence while providing a manageable input size for processing.

To enhance the model's ability to recognize patterns across overlapping time windows, we use a sequence overlap of 50 samples (time steps). This overlap preserves temporal continuity between adjacent sequences, allowing the model to capture smoother transitions in the data. Our selection of a 50% overlap is supported by previous studies that have demonstrated maximum accuracy at this level of overlap [64]. Increasing the overlap percentage beyond 50% leads to sequences intersecting not only with their immediate neighbors but also with non-adjacent sequences (e.g., the first and third sequences), which significantly intensifies RAM requirements due to the exponential growth in overlapping data that needs to be processed simultaneously. Also, empirical results indicated that convergence speed improves as the overlap percentage increases; however, the associated increase in memory consumption becomes a limiting factor. Therefore, a 50% overlap strikes an optimal balance between model performance, temporal continuity, convergence speed, and hardware constraints.

All data is normalized between 0 and 1 to ensure that the input values are on a consistent scale. This minimizes the risk of bias due to differences in the range of values across different devices. This normalization process helps the model focus on the underlying patterns rather than the scale differences.

Additionally, we employed 4-fold cross-validation to ensure robust evaluation of the model. This approach divides the dataset into 4 subsets, using 3 for training and one for testing in each iteration, thus improving the reliability of the results. Further details on the cross-validation process are discussed in the Results section.

The input tensor $X_{WaveNet}$ can be defined by equation 11

$$\begin{aligned} a &= x_i, y_i, z_i, mean_i, max_i, min_i, var_i \\ b &= std_i, range_i, skewness_i, kurtosis_i, rms_i \\ X_{WaveNet} &= \{(a_i, b_i)\}_{i=1}^{100} \end{aligned} \tag{11}$$

B. CONTINUOUS WAVELET TRANSFORM WITH MORLET WAVELET FOR EFFICIENTNETV2

CWT is a signal processing technique that analyzes localized power changes within a time series. Unlike the Fourier Transform, which provides frequency information but loses time information, CWT provides a time-frequency representation of the signal, making it particularly suitable for non-stationary signals such as gyroscope readings [82].

The CWT of a signal x(t) is defined by equation 12.

$$W(a, b) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt \tag{12}$$

where:

- $W(a, b)$ are the wavelet coefficients,
- a is the scale parameter controlling the width of the wavelet,
- b is the translation parameter, determining the location of the wavelet,
- $\psi(t)$ is the mother wavelet.

The mother wavelet $\psi(t)$ is a function used to generate wavelets through scaling and translation. In our case, we used the Morlet wavelet.

C. MORLET WAVELET

Morlet wavelet is a complex wavelet that combines a complex exponential value with a Gaussian window [73]. The complex exponential value represents the oscillations, enabling localization in time provided by the Gaussian window. The Morlet wavelet is defined by equation 13.

$$\psi(t) = \pi^{-\frac{1}{4}} e^{-j\omega_0 t} e^{-\frac{t^2}{2}} \tag{13}$$

where:

1) j is the imaginary unit,
2) w_0 is the central frequency of the wavelet,
3) $e^{-j\omega_0 t}$ represents the oscillatory component,
4) $e^{-\frac{t^2}{2}}$ represents the Gaussian window.

D. SCALE AND FREQUENCY RELATIONSHIP

In CWT, the scale parameter a corresponds inversely to the frequency. Smaller scales correspond to higher frequencies, providing finer time resolution, while larger scales correspond to lower frequencies, providing coarser time resolution. This relationship allows CWT to capture both high-frequency events and low-frequency trends in the signal.

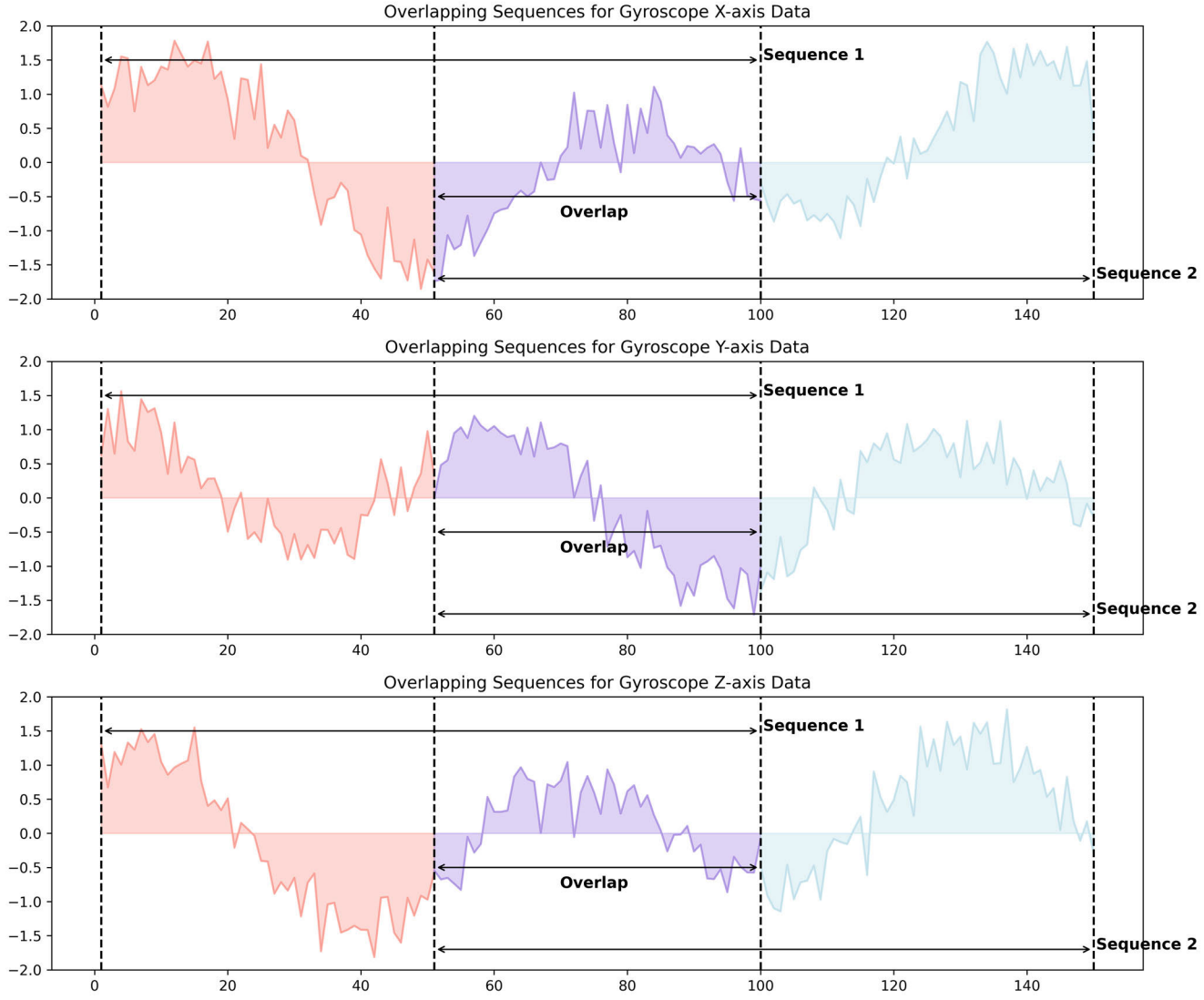

FIGURE 4. Sequence creation and overlapping process.

The relationship between the scale a and the equivalent Fourier frequency f is given by:

$$f = \frac{w_0}{2\pi a} \tag{14}$$

We chose a 100-point scale range for our study after experimenting with scales of 10, 50, and 100. Smaller scales, such as 10, tend to emphasize high-frequency components, capturing fine details but also introducing significant high-frequency noise inherent in gyroscope signals. Larger scales, like 50 and 100, focus more on lower-frequency components, helping to smooth out high-frequency noise and capture broader patterns. Our experiments showed that while a scale of 10 captured too much noise, a scale of 50 provided a good balance but still missed some broader patterns. The scale of 100 provided the optimal balance between time and frequency resolution, effectively capturing the essential characteristics of our gyroscope data and leading to high model accuracy. This choice allows us to thoroughly analyze the frequency components in the gyroscope signals, ensuring both fine details and broader patterns are well-represented.

E. WHY CHOOSE CWT WITH MORLET WAVELET?

The Morlet wavelet is particularly suitable for our application due to its time-frequency localization properties. The benefits of using CWT with Morlet wavelets are:

1) TIME-FREQUENCY LOCALIZATION

CWT provides high resolution in both time and frequency domains. Thanks to this feature, CWT is ideal for analyzing the non-stationary nature of gyroscope signals.

2) MULTI-SCALE ANALYSIS

CWT can analyze the signal at multiple scales, allowing the detection of features that vary with time and frequency.

3) ENHANCED FEATURE EXTRACTION

Morlet wavelet CWT converts gyroscope readings into a time-frequency representation, allowing EfficientNetV2 to extract more important features, which improves classification performance.

F. APPLICATION TO GYROSCOPE DATA

Gyroscope readings vary due to different user movements and phone orientations. Converting raw gyro data into a time-frequency representation by applying CWT with the Morlet wavelet allows our model to capture complex patterns and variations in the signal. The conversion process includes the following steps:

1) RAW DATA COLLECTION

Gyroscope readings are collected along the x, y, and z axes, each consisting of arrays 100 data points long. This preprocessing step was done during the preparation of the training dataset for the WaveNet model.

2) CWT TRANSFORMATION

Each sequence of gyroscope readings was transformed using CWT with Morlet wavelet. This process results in a time-frequency representation for each axis.

3) 3D REPRESENTATION

The converted data for each axis (x, y, z) are combined to create a 3D representation with dimensions (100,100,3). These dimensions correspond to time, scale, and axis, respectively. This 3D representation was prepared as input to the EfficientNetV2 model.

The CWT transformation of a random sample, including all three channels (x, y, and z axes), is illustrated in Figure 5.

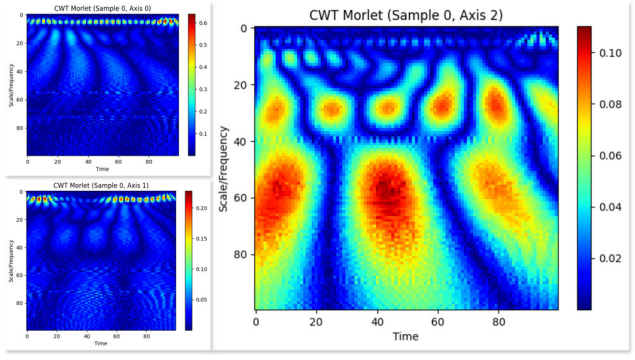

FIGURE 5. Transformed sequences based on CWT Morlet.

The transformed input tensor $X_{EffNetV2}$ is represented formula 15.

$$X_{EffNetV2} = \{(CWT_{x_i}, CWT_{y_i}, CWT_{z_i})\}_{i=1}^{100} \quad (15)$$

where $CWT_{x_i}, CWT_{y_i}, CWT_{z_i}$ are the CWT-transformed sequences for the x, y, and z axes, respectively.

III. METHODOLOGY

The proposed ensemble model leverages the strengths of both WaveNet and EfficientNetV2 and combines with an attention mechanism to effectively classify phone models based on gyroscope readings [71], [72], [83], [84]. This section details the model's components and their integration.

A. WAVENET

WaveNet is a generative model designed for sequential data processing. WaveNet was developed for sound synthesis and has been adapted to various applications, including time series analysis. WaveNet leverages dilated convolutions to effectively capture temporal dependencies in sequential data without requiring repeated layers [85]. This makes it particularly suitable for tasks involving long-term dependencies and complex patterns in time series data. The traditional WaveNet architecture is illustrated in Figure 6.

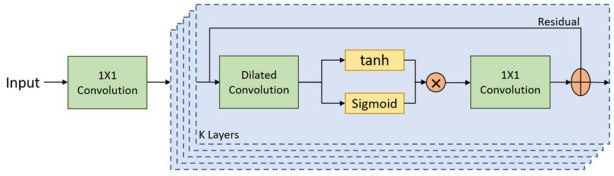

FIGURE 6. WaveNet architecture.

The WaveNet model is designed to process sequential data, making it ideal for time-series analysis. It is tailored to handle the input shape of (100, 12), where 100 represents the sequence length (time steps), and 12 represents the number of input channels. These channels include readings from the gyroscope's x, y, and z axes and additional derived features.

1. Input Layer:

- The input to the WaveNet model consists of 100-length sequences (time steps) of gyroscope readings across the x, y, and z axes and derived features, resulting in 12 input channels. The input tensor is permuted to (batch_size, 12, 100) to match the expected input format for 1D convolutional layers, enabling the model to treat each feature as a separate channel.

2. Wave Blocks:

- The model architecture includes four Wave_Block layers. Each Wave_Block is composed of dilated convolutions with increasing dilation rates, which allow the model to capture both short-term and long-term dependencies:
  - Wave_Block 1: Takes 12 input channels and produces 16 output channels using 7 dilation rates.
  - Wave_Block 2: Takes 16 input channels and produces 32 output channels with 4 dilation rates.
  - Wave_Block 3: Takes 32 input channels and outputs 64 channels with 3 dilation rates.

– Wave_Block 4: Takes 64 input channels and produces 128 output channels with 1 dilation rate.

3. Causal Convolutions:

- Each convolutional layer within a Wave_Block is causal, meaning that the output at any time step t depends only on inputs from time t and earlier. This ensures the temporal order is preserved, making the model suitable for sequential data where future values should not influence the past.

4. Gated Activations:

- Each Wave_Block uses a gated activation mechanism to control information flow:
  - Filter Convolution: Applies a *tanh* activation to emphasize specific feature values.
  - Gate Convolution: Uses a *sigmoid* activation to produce values between 0 and 1, acting as a gate.
  - Gated Combination: Combines these outputs through element-wise multiplication, allowing the network to pass through important information while suppressing less relevant signals selectively.

5. Output and Final Fully Connected Layer:

- After the input data passes through the series of Wave_Blocks, the resulting tensor has the shape (batch_size, 100, 128), representing the learned features across the 100-time steps.
- The tensor is then reshaped and flattened into a 12800-dimensional vector (128 channels * 100-time steps) for each sample.
- A fully connected (linear) layer maps this vector to a 512-dimensional feature vector. This step reduces the data dimensionality while retaining the essential features, making the output suitable for combination with other models in the ensemble approach.

6. Design Choices:

- No Residual Connections: This model does not include residual connections, often used to facilitate gradient flow in deep networks. This design choice was made because the gated activations and structure of each Wave_Block ensure stable training without the added complexity of residual connections. Omitting these connections reduces the model's complexity and computational requirements while achieving robust learning performance.
- Dilated Convolutions: Dilated convolutions allow the model to exponentially expand its receptive field, capturing dependencies across varying time scales without needing excessive layers.

The architectural hyperparameters are summarized in Table 4.

The finetuned WaveNet architecture is given in Figure 7.

WaveNet's architecture effectively captures temporal dependencies in sequential gyroscope data through the use of causal and dilated convolutions. The model's ability to process multiple levels of temporal information through its hierarchical block structure allows it to learn complex patterns and dependencies, making it a powerful component in the phone model detection task. The combination of fine-grained detail capture and long-range dependency modeling ensures that WaveNet can accurately represent the dynamics of gyroscope signals.

TABLE 4. The layer specifications for the WaveNet model architecture.

Layer	Input Channels	Output Channels	Dilation Rates	Kernel Size
Wave_Block 1	12	16	$[2^0 \cdots 2^6]$	3
Wave_Block 2	16	32	$[2^0 \cdots 2^3]$	3
Wave_Block 3	32	64	$[2^0 \cdots 2^2]$	3
Wave_Block 4	64	128	$[2^0]$	3
Fully Connected	12,800	512	N/A	N/A

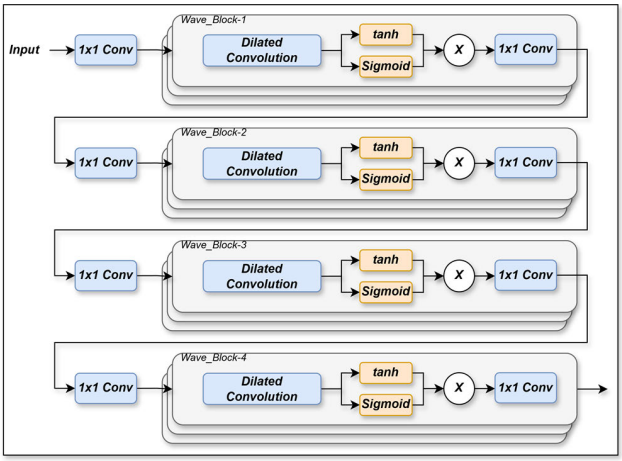

FIGURE 7. Finetuned WaveNet architecture for gyroscope sequences.

B. EfficientNetV2

EfficientNetV2 is a highly efficient and powerful convolutional neural network architecture that has demonstrated significant performance on various computer vision tasks [72], [83]. It builds on the strengths of the original EfficientNet model by combining improvements in network architecture and training techniques to achieve superior performance. Key Features of EfficientNetV2:

Compound Scaling Method: EfficientNetV2 uses composite scaling, which scales network depth, width, and resolution in a balanced manner. This approach allows the model to maintain efficiency while achieving high accuracy.

- Width Scaling: Used. The model increases the number of channels per layer.
- Depth Scaling: Used. The model increases the number of layers or block repeats.
- Resolution Scaling: The input image size is increased according to scaling coefficients.

Enhanced Architecture, Optimized Convolutional Layers: The network consists of optimized convolutional blocks designed to capture complex patterns and features from input

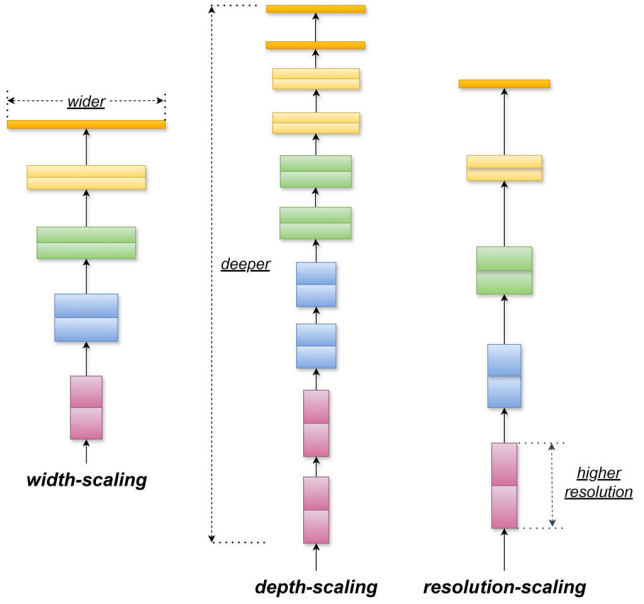

FIGURE 8. Compound scaling dimensions of EfficientNetV2: width, depth, and resolution scaling.

data. These blocks contain depth-separable convolutions that reduce the number of parameters and computational complexity without sacrificing performance.

Progressive Learning: EfficientNetV2 uses a progressive learning strategy that gradually increases image size and editing power during training. This helps achieve faster convergence and better generalization.

Pre-trained Weights and Transfer Learning: By utilizing pre-trained weights on large-scale image datasets, EfficientNetV2 can transfer learned features to specific tasks. This increases the model's ability to generalize from training data, even if the dataset is relatively small.

1) OUR MODEL ARCHITECTURE

a: INITIAL CONVOLUTIONAL LAYEr

The model begins with a Conv2d layer, which takes in the input data and applies a series of 2D convolutions. This layer is followed by a *BatchNorm* layer for normalization and a *SiLU* activation function, which introduces non-linearity into the network.

b: SEQUENTIAL BLOCKS

The main body of the model comprises a series of sequential blocks that progressively extract features from the input data. Each sequential block corresponds to Fused-mobile inverted bottleneck convolution (*Fused* − *MBConv*) or mobile inverted bottleneck convolution (*MBConv*) layers, as shown in Figure 10. These blocks consist of convolutional layers, Squeeze-and-Excitation (SE) modules, batch normalization, and activation functions, allowing the model to capture patterns at various scales and spatial resolutions.

For instance, the model's output shape transitions from [1,24,50,50] at the initial layers to [1,208,4,4]in the deeper layers. Each block down-samples the input while focusing on extracting increasingly abstract and high-level features. The *Fused* − *MBConv* blocks combine depthwise and pointwise operations using a single convolution, making them computationally efficient in the early stages. In contrast, the *MBConv* blocks apply a depthwise convolution followed by a pointwise convolution, providing a more expressive capacity in later stages. Both types of blocks incorporate *SE* modules that recalibrate channel-wise feature responses, enhancing the model's ability to focus on essential features.

The SE module within each *MBConv* and *Fused*−*MBConv* block consists of global average pooling followed by two fully connected layers. The first layer uses a *SiLU* activation, and the second layer employs a *sigmoid* function. This recalibration mechanism allows the model to learn the importance of each channel dynamically, contributing to its efficiency and accuracy. The detailed internal structures of the *MBConv*, *Fused* − *MBConv*, and SE modules are illustrated in Figure 10, showcasing each component's specific configuration and flow within the architecture.

c: ADAPTIVE POOLING AND FLATTENING

After the convolutional blocks, an adaptive average pooling layer reduces the spatial dimensions to a fixed size of [1, 1024, 1, 1]. This step ensures that the output features are of a consistent size, regardless of the input dimensions. The Flatten layer then converts the pooled features into a 1024-dimensional vector, preparing them for further processing.

2) OUTPUT ADJUSTMENT

a: FULLY CONNECTED LAYER

The final classifier layer is replaced with a fully connected (linear) layer that outputs a 512-dimensional feature vector to adapt the pre-trained model for the current task. This dimensionality reduction serves two purposes: It compresses the extracted spatial features into a compact representation that integrates easily with outputs from other models (e.g., in an ensemble). It preserves essential feature information while reducing computational load during further processing. The 512-dimensional feature vector is then used in the ensemble model, where it is combined with the output from the WaveNet model and further processed using an attention mechanism.

3) DESIGN CONSIDERATIONS

a: EFFICIENCY AND PERFORMANCE

EfficientNetV2 strikes a balance between performance and computational efficiency. It is designed to achieve high accuracy while minimizing the number of parameters. As seen in the model summary, EfficientNetV2 processes the input in stages, reducing spatial dimensions and increasing feature richness, which is crucial for tasks where different levels of spatial detail matter.

```
==========================================================================================
Layer (type:depth-idx)                             Output Shape              Param #
==========================================================================================
EfficientNetV2                                     [1, 512]                  --
├─EfficientNet: 1-1                                [1, 512]                  --
│    └─Conv2d: 2-1                                 [1, 24, 50, 50]           648
│    └─BatchNormAct2d: 2-2                         [1, 24, 50, 50]           48
│    │    └─Identity: 3-1                          [1, 24, 50, 50]           --
│    │    └─SiLU: 3-2                              [1, 24, 50, 50]           --
│    └─Sequential: 2-3                             [1, 208, 4, 4]            --
│    │    └─Sequential: 3-3                        [1, 24, 50, 50]           10,464
│    │    └─Sequential: 3-4                        [1, 40, 25, 25]           218,048
│    │    └─Sequential: 3-5                        [1, 48, 13, 13]           343,616
│    │    └─Sequential: 3-6                        [1, 104, 7, 7]            607,790
│    │    └─Sequential: 3-7                        [1, 128, 7, 7]            2,240,634
│    │    └─Sequential: 3-8                        [1, 208, 4, 4]            8,988,100
│    └─Conv2d: 2-4                                 [1, 1024, 4, 4]           212,992
│    └─BatchNormAct2d: 2-5                         [1, 1024, 4, 4]           2,048
│    │    └─Identity: 3-9                          [1, 1024, 4, 4]           --
│    │    └─SiLU: 3-10                             [1, 1024, 4, 4]           --
│    └─SelectAdaptivePool2d: 2-6                   [1, 1024]                 --
│    │    └─AdaptiveAvgPool2d: 3-11                [1, 1024, 1, 1]           --
│    │    └─Flatten: 3-12                          [1, 1024]                 --
│    └─Linear: 2-7                                 [1, 512]                  524,800
==========================================================================================
```

FIGURE 9. Finetuned EfficientNetV2 architecture for transformed gyroscope sequences based on CWT Morlet.

b: FINE-TUNING FOR SPECIFIC FEATURES

By fine-tuning the model on domain-specific data, the EfficientNetV2 model learns to extract particularly relevant features to the problem, such as patterns in the image-like data generated from time-series sequences.

4) SUMMARY OF MODEL PARAMETERS

Total Parameters: The EfficientNetV2 model contains many parameters, ensuring a high capacity for learning complex features from the data.

a: OUTPUT STRUCTURE

The model's final output is a 512-dimensional vector, which balances the need for detailed feature extraction with the computational considerations of combining with other model outputs.

Our model architecture, which incorporates a pre-trained model, is illustrated in Figure 9.

C. ENSEMBLE MODEL ARCHITECTURE

To improve classification performance, we used an ensemble model that combines the strengths of the WaveNet and EfficientNetV2 architectures. The ensemble model took advantage of the unique features extracted by both models, and an attention mechanism was used to effectively combine their outputs.

D. ATTENTION MECHANISM

The attention mechanism plays an important role in selecting the most appropriate features from the combined outputs of WaveNet and EfficientNetV2 models [84].

1) ATTENTION LAYER

The attention layer calculates attention scores using a linear transformation followed by a *tanh* activation function [86], [87].

Attention weights are derived using a *softmax* function applied to the attention scores, highlighting the importance of each feature [86], [87].

The weighted average of the features is then calculated, producing a context vector that emphasizes the most informative features.

The attention mechanism uses query (Q), key (K), and value (V) matrices to calculate the attention scores. Given the combined feature vector $F_{combined}$:

$$Q = W_q F_{combined} \tag{16}$$

$$K = W_k F_{combined} \tag{17}$$

$$V = W_v F_{combined} \tag{18}$$

$$Attention_{scores} = \frac{QK^T}{\sqrt{d_k}} \tag{19}$$

$$Attention_{weights} = softmax(\tanh(Attention_{scores})) \tag{20}$$

$$Context_{vector} = Attention_{weights}.V \tag{21}$$

E. ENSEMBLE MODEL ARCHITECTURE

The architecture of our combined model is designed to integrate the outputs of the WaveNet and EfficientNetV2 models and then create an attention mechanism to improve the combined features. The key components and their roles are described below:

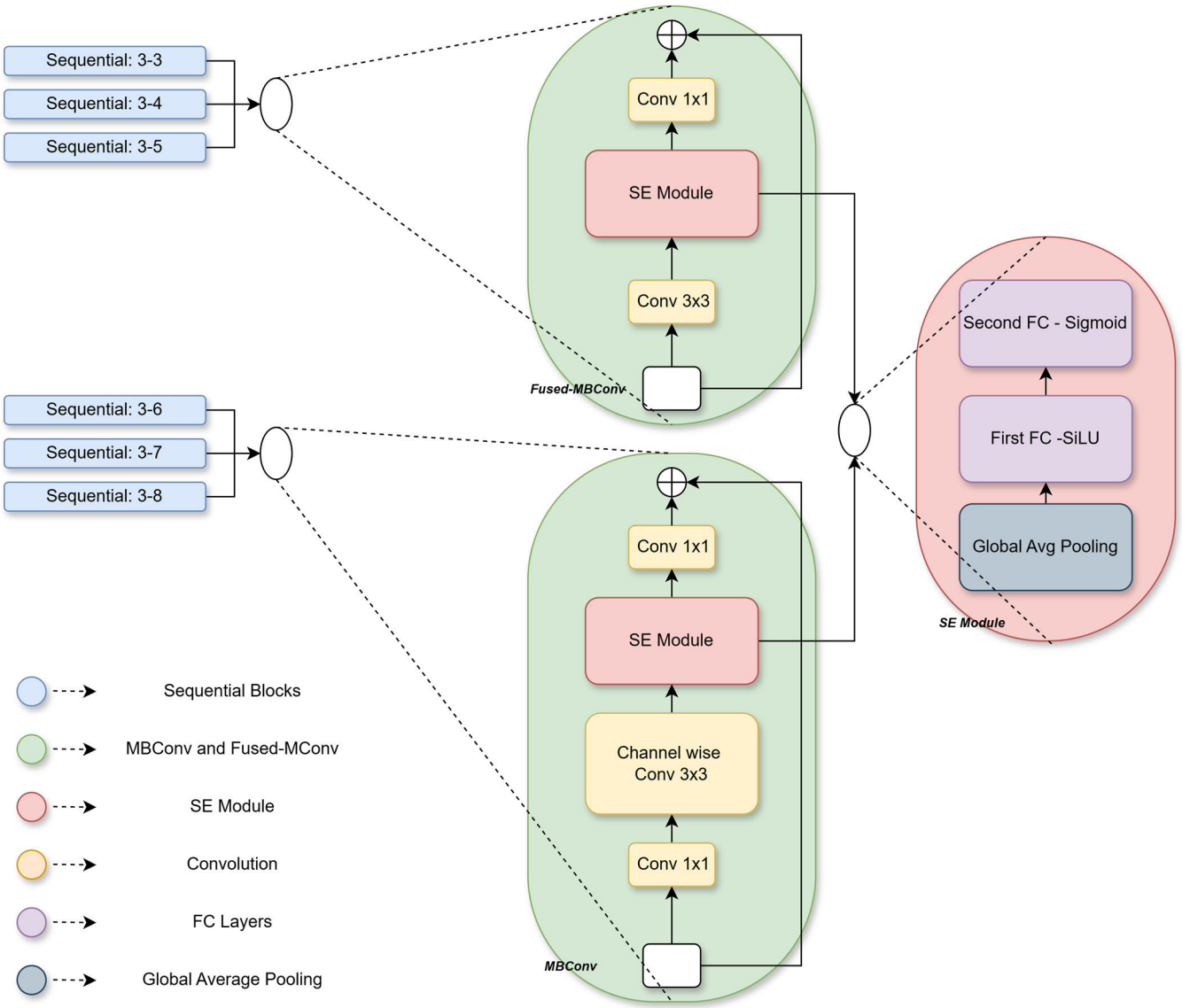

FIGURE 10. The internal structure of six sequential blocks in EfficientNetV2 architecture highlights MBConv, Fused-MBConv, and SE Modules.

1) FEATURE COMBINATION

The outputs from the WaveNet and EfficientNetV2 models are combined to form the composite feature vector with the shape of (1,1024).

2) ATTENTION MECHANISMS

The combined feature vector is first processed by the attention mechanism so that it can highlight the most relevant features.

The highlighted features are then passed through a fully connected layer, reducing the dimensionality to (1, 512).

A second attention mechanism is applied to further improve feature representation.

3) CLASSIFICATION LAYER

The final output layer is a fully connected layer that matches features optimized for phone model detection to class probabilities. The output form is (1, 6) and corresponds to the probability of having the six phone models in our study.

4) ARCHITECTURE DIAGRAM

The ensemble model architecture, which includes both WaveNet and EfficientNetV2 models with attention mechanisms, is given in Figure 11.

This ensemble approach allows us to capture both temporal and spatial patterns in gyroscope data, improving classification performance. The ensemble model, which takes advantage of the complementary strengths of WaveNet and EfficientNetV2 models and improves its outputs with attention mechanisms, can detect phone models with high performance.

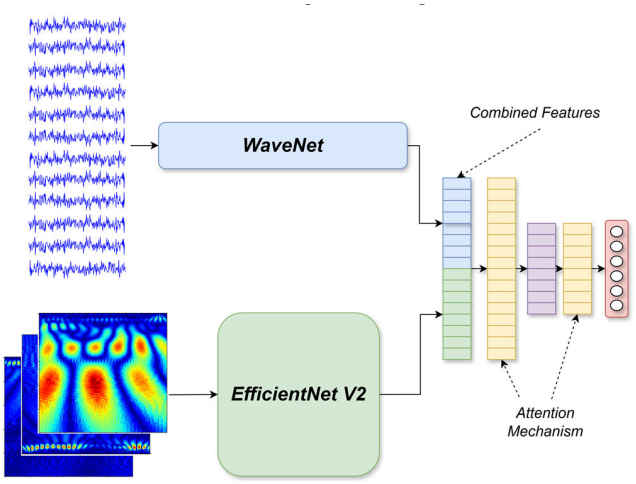

FIGURE 11. Ensemble model architecture.

Table 5 presents the pseudocode for the detection algorithm below to provide a comprehensive understanding of our approach.

F. PARAMETERS USED FOR THE ANALYSIS

The key parameters and settings used for training, evaluation, and model architecture are summarized in Table 6. These parameters were carefully fine-tuned through experiments to achieve a balanced performance across accuracy, training efficiency, and inference speed.

IV. RESULT

A. MODELS PERFORMANCE

The dataset for this study is divided into training and testing sets, with 75% allocated to training and 25% to testing, as detailed in Table 7. This split balances the need for a substantial amount of data to train the model effectively while reserving enough data to evaluate its performance robustly. A 75/25 division is commonly used in machine learning tasks to provide a reliable assessment of the model's generalization capabilities without overly restricting the data available for training. Both the WaveNet and EfficientNetV2 models use 207056 training samples and 51764 testing samples. WaveNet inputs are sequences of (100, 12), while EfficientNetV2 inputs have dimensions of (100, 100, 3). A 4-fold cross-validation approach is applied to ensure robust evaluation across different subsets of the data.

Note: All performance metrics reported in the Result section represent averages across the K-fold cross-validation.

The most successful models in our experiments were WaveNet and EfficientNetV2 models. The ensemble model was created by combining these two models to leverage their strengths and achieve better performance.

Initially, both WaveNet and EfficientNetV2 were fine-tuned and trained separately. Performance was then increased by combining the models. Three different experiments were conducted to improve performance:

TABLE 5. Pseudocode for smartphone detection with ensemble model.

Algorithm

Begin
1. Data Preprocessing
  a. Load Data
    For each gyroscope data file in the dataset D:
      - Load gyroscope readings $G(t)$ over time t.
  b. Normalization
    For each gyroscope reading $G(t)$:
      - Normalize the readings across x, y, and z axes to obtain $G_{norm}(t)$.
  c. Segmentation
    - Segment the normalized data $G_{norm}(t)$ into fixed-length sequences:
      $S = \{S_1, S_2, \cdots, S_n\}$, where each S_i represents a sequence.
2. Feature Engineering
  a. Compute Statistical Features
    For each sequence S_i in S_i:
      - Compute statistical features F_i:
        - Decided features: $F_i = Feature\_Extraction(S_i)$
  b. Append Features
    - Append the feature vector F_i to the corresponding sequence S_i:
      $S_i' = S_i \cup F_i$
3. CWT-Morlet Transform
  a. Apply Continuous Wavelet Transform
    For each sequence S_i in S:
      - Apply CWT using the Morlet wavelet:
        $T_i = CWT_{Morlet}(S_i)$
      - Obtain the time-frequency representation S_i.
4. Model Training
  a. Train WaveNet Model
    - Use the feature-engineered sequences $\{S_i'\}$ to train the WaveNet model:
      $M_{WaveNet} = TrainWaveNet(\{S_i'\}, labels)$
  b. Train EfficientNetV2 Model
    - Use the time-frequency representations $\{T_i\}$ to train the EfficientNetV2:
      $M_{EfficientNetV2} = TrainEfficientNetV2(\{T_i\}, labels)$
5. Ensemble Model Training
  a. Extract Features from Models
    For each sequence S_i:
      - Obtain features from WaveNet:
        $Features_{WaveNet}(S_i') = ExtractFeatures(Model_{WaveNet}, S_i')$
      - Obtain features from EfficientNetV2:
        $Features_{EfficientNetV2}(T_i) = ExtractFeatures(Model_{EfficientNetV2}, T_i)$
  b. Combine Features
    - Combine the extracted features into a single feature vector O_i:
      $O_i = Concat(Features_{WaveNet}(S_i'), Features_{EfficientNetV2}(T_i))$
  c. Apply Attention Mechanism
    - Apply an attention mechanism A to emphasize relevant features:
      $O_i' = A(O_i)$
  d. Train Ensemble Model
    - Use the attention-enhanced feature vectors $\{O_i'\}$ to train the Ensemble Model:
      $Model_{Ensemble} = TrainEnsemble(\{O_i'\}, labels)$
6. Prediction
  For each input sequence S_i:
    a. Feature Extraction
      - Obtain features:
        $Features_{WaveNet}(S_i') = ExtractFeatures(Model_{WaveNet}, S_i')$
        $Features_{EfficientNetV2}(T_i) = ExtractFeatures(Model_{EfficientNetV2}, T_i)$
      - Combine features:
        $O_i = Concat(Features_{WaveNet}(S_i'), Features_{EfficientNetV2}(T_i))$
      - Apply attention mechanism:
        $O_i' = A(O_i)$
    b. Final Prediction
      - Generate the final prediction Y_i for the smartphone model:
        $Y_i = Model_{Ensemble}(O_i')$
End Algorithm

Experiment 1: The ensemble model was trained with all parameters of both models frozen.

Experiment 2: The ensemble model was trained with only the last layer parameters of both models set as trainable.

Experiment 3: The ensemble model was trained with all parameters of both models set as trainable.

TABLE 6. The parameter settings for the proposed ensemble model.

Parameter	Value
WaveNet Input Channels	12 (sequence length input as channels)
WaveNet Kernel Size	3
WaveNet Blocks	4 (16, 32, 64, 128 output channels)
WaveNet Dilation Rates	7, 4, 3, 1 for successive blocks
WaveNet Fully Connected Output	512
EfficientNetV2 Model	Pretrained EfficientNetV2 (efficientnetv2_rw_t)
EfficientNetV2 Output	512
Ensemble Model Attention Pooling	Linear layer: Input 1024, Output 512
Attention Mechanism Dimensions	Input 512, Output 512
Final Classification Layer	512 to 6 classes
Loss Function	CrossEntropyLoss
Optimizer	AdamW
Learning Rate	5e-4
Weight Decay	1e-2
Scheduler	StepLR (step size = 10, gamma = 0.5)
Early Stopping Patience	5 epochs
Batch Size	32
Number of Epochs	200
Input Size - WaveNet	(100, 12)
Input Size – EfficientNetV2	(100, 100, 3)

TABLE 7. The dataset split and input dimensions.

Model	Training Set Size	Test Set Size	Input Dimensions	K-Fold
WaveNet	207056 samples	51764 samples	(100, 12) – 100 sequence length, 12 channels	4
EfficientNetV2	207056 samples	51764 samples	(100, 100, 3) - 3 frames, each of size 100x100	4

The class-wise support metrics, representing the distribution of instances across different classes, are illustrated in Figure 12.

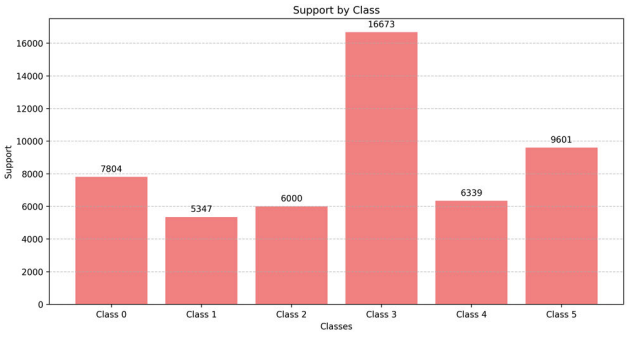

FIGURE 12. Support by class.

Table 8 presents the performance metrics, including accuracy, cross-entropy loss, mean absolute error (MAE), and F1 scores.

TABLE 8. The comparison of accuracy, loss, and F1 score of models.

	Accuracy	Cross Entropy Loss	MAE	F1
EfficientNetV2 (scale of 10)	0.7762	0.6612	0.4123	0.72
EfficientNetV2 (scale of 50)	0.8905	0.2503	0.2376	0.88
WaveNet	0.8977	0.2613	0.2201	0.88
EfficientNetV2 (scale of 100)	0.9335	0.1795	0.1633	0.90
Ensemble Model (Frozen Parameters)	0.9382	0.1976	0.1871	0.92
Ensemble Model (Frozen Parameters except the last layer)	0.9764	0.0750	0.0439	0.95
Ensemble Model	0.9901	0.0448	0.0202	0.99

The detailed classification report for each class is given in Table 9 and the visual representation of the report is shown in Figure 13. This dual representation provides both precise numerical accuracy and an easy-to-understand visual overview of the model's performance in different classes. Overall accuracy and F1 score are 0.99; The macro and weighted averages for precision, and recall are all 0.98. This shows that it provides consistent and reliable performance across all classes in prediction quality.

TABLE 9. Classification report of ensemble model (Averaged Across K-Folds).

	Precision	Recall	F1-score	Support
Class 0	0.99	0.99	0.99	7804
Class 1	0.97	0.96	0.97	5347
Class 2	0.98	0.96	0.97	6000
Class 3	0.97	0.99	0.98	16673
Class 4	0.98	0.98	0.98	6339
Class 5	1.00	0.99	1.00	9601
Accuracy		0.99		51764
Macro avg.	0.98	0.98	0.98	51764
Weighted avg.	0.98	0.98	0.98	51764

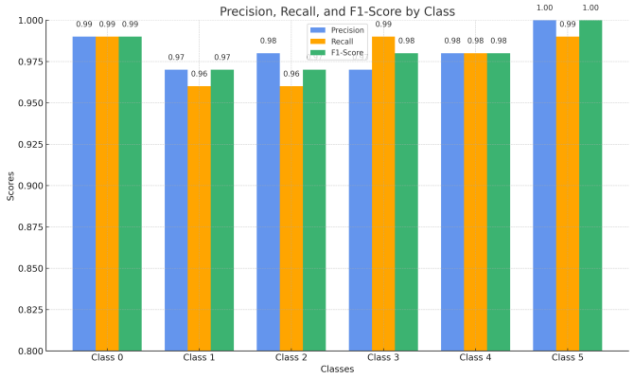

FIGURE 13. Precision, recall, and F1-score by class.

The confusion matrix is depicted in Figure 14. When Figure 14 is examined, the fact that the ensemble model shows high accuracy with mostly correct classifications and a very small number of misclassifications reveals the effectiveness of the model in distinguishing different classes.

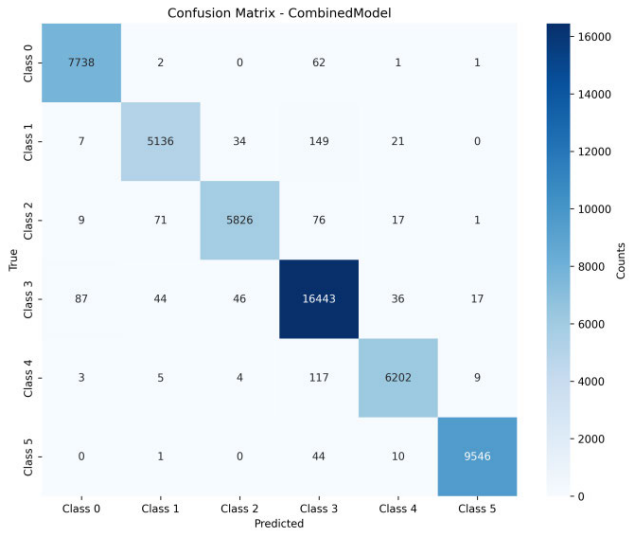

FIGURE 14. Confusion Matrix – Ensemble Model.

Class-wise accuracy is also shown in Figure 15.

Table 10 presents the performance metrics for the ensemble model across each class, detailing both the Area Under the ROC Curve (AUC-ROC) and the Average Precision (AP) scores from the precision-recall analysis. An AUC-ROC score of 1.00 for all classes indicates high sensitivity and specificity, demonstrating that the model distinguishes each smartphone model with minimal false positives and false negatives, as observed in the confusion matrix. The high AP scores, with values ranging from 0.97 to 1.00, confirm the model's strong precision and recall across all classes. The micro-average scores of 1.00 for AUC-ROC and 0.99 for AP further reinforce the model's overall reliability and robustness in multi-class classification tasks, ensuring consistently high performance in smartphone model identification.

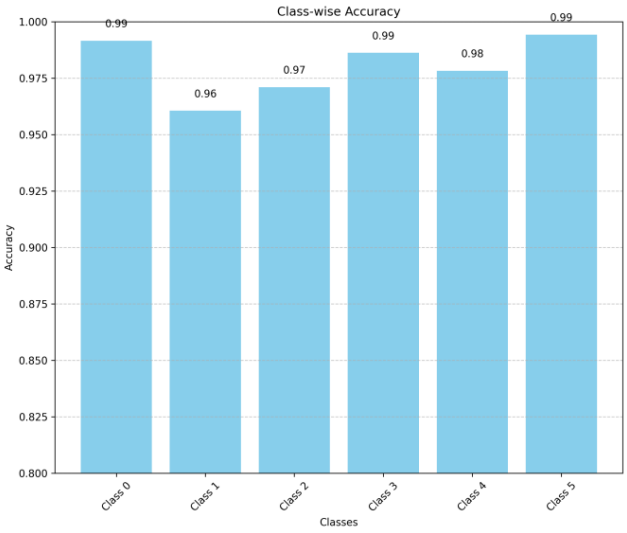

FIGURE 15. Class-wise accuracy – ensemble model.

TABLE 10. The AUC-ROC and average precision (AUC-AP) scores by class for the ensemble model.

Class	AUC-ROC	AUC-AP
Class 0	1.00	0.99
Class 1	1.00	0.97
Class 2	1.00	0.98
Class 3	1.00	0.99
Class 4	1.00	0.99
Class 5	1.00	1.00
Micro-average	**1.00**	**0.99**

The error distribution plot in Figure 16 shows that the majority of predictions are accurate, with most errors centered around zero, indicating high prediction accuracy. Only a few instances exhibit significant errors.

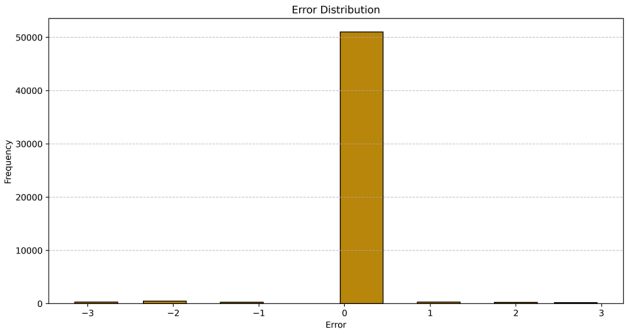

FIGURE 16. Error distribution.

B. TRAINING AND INFERENCE TIME

The training times for each model were measured using a system equipped with 32 GB RAM, an Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, and an NVIDIA RTX 2060 GPU

with 6 GB VRAM. The models were trained using PyTorch version 2.3.0 and CUDA version 12.1, allowing efficient GPU-accelerated training. The average training time per fold was 143.08 minutes for the WaveNet model, 580.29 minutes for the EfficientNetV2 model, and 321.11 minutes for the ensemble model. Early stopping with a patience of 5 epochs was applied, which allowed the models to continue training for up to 5 additional epochs to achieve optimal accuracy. Inference times were also evaluated, with the WaveNet model averaging 0.00659 seconds per batch, EfficientNetV2 averaging 0.01844 seconds per batch, and the ensemble model averaging 0.02543 seconds per batch. These measurements highlight both the training investment and the efficiency of each model during deployment.

TABLE 11. Computational environment and timing metrics for model execution.

Parameter	Value
PyTorch Version	2.3.0
CUDA Version	12.1
CUDA Device	NVIDIA RTX 2060 (6 GB VRAM)
CPU	Intel(R) Core(TM) i7-10750H @ 2.60GHz
RAM	32 GB
Learning Rate	5e-4
Optimizer	AdamW
Batch Size	32
Early Stopping Patience	5 epochs
Training Time - WaveNet	143.08 minutes per fold
Training Time - EfficientNetV2	580.29 minutes per fold
Training Time - Ensemble	321.11 minutes per fold
Inference Time - WaveNet	0.00659 seconds per batch
Inference Time - EfficientNetV2	0.01844 seconds per batch
Inference Time - Ensemble	0.02543 seconds per batch

V. DISCUSSION

This study demonstrates that gyroscope data alone can effectively distinguish between smartphone models, achieving an accuracy of 99.01% and an F1 score of 0.99. Despite gyroscopes' inherent noise susceptibility, our advanced deep learning models successfully extract distinguishing features, highlighting the gyroscope's potential as a robust source for device fingerprinting.

A. COMPARISON WITH PREVIOUS STUDIES

Previous research on smartphone model detection has predominantly utilized signals from cameras, microphones, and magnetometers, often requiring longer data collection durations and complex processing techniques [16], [44], [50], [52], [54], [65], [66], [67], [68], [69]. For example, Baldini et al. [67] achieved 98.07% accuracy using 60 seconds of magnetometer data, and Hanilçi et al. [65] reached 96.42% accuracy with 3 seconds of speech data. To our knowledge, gyroscope data has not been previously employed for smartphone model detection.

Our study sets a new benchmark by achieving higher accuracy using only 1–2 seconds of raw gyroscope data. This not only surpasses the performance of existing methods but also offers greater efficiency and practicality. Our approach simplifies the data acquisition process while maintaining high accuracy by relying on a single sensor and a significantly shorter input duration compared to the 3-second to 2-minute durations in previous studies.

B. METHODOLOGICAL STRENGTHS

Utilizing the GSDC 2023–2024 dataset allowed us to work with real-world data collected from smartphones mounted uniformly during driving tests. This uniform alignment presents a challenging scenario due to the minimal differences in motion experienced by the devices. Our methodology addresses this by employing CWT with the Morlet wavelet for detailed time-frequency analysis, enabling the EfficientNetV2 model to capture complex patterns in the gyroscope data. Combining this with a fine-tuned WaveNet model in an ensemble framework enhanced by an attention mechanism allowed us to effectively analyze sequential and time-frequency representations, leading to high accuracy despite the challenging conditions.

C. EFFICIENCY AND PRACTICALITY

Our approach offers a distinct advantage in terms of efficiency, requiring only approximately 1–2 seconds of gyroscope readings, which is ideal for real-time applications where rapid and accurate identification is essential. Unlike other studies that rely on longer data collection durations, some lasting up to several minutes, or the integration of multiple sensors, our method stands out for its simplicity and resource efficiency. This reduced input duration accelerates the identification process and minimizes data processing demands, making the model lighter and more adaptable to various hardware environments. By eliminating the need for complex sensor fusion, our approach is practical for wide-scale deployment in scenarios that demand high-speed, reliable device identification.

D. LIMITATIONS

While the results are promising, there are several limitations to consider. First, the gyroscope's sensitivity to noise means that variations in user handling and external conditions could impact performance, potentially affecting the model's robustness in diverse real-world settings. Second, our dataset is limited to specific brands and models provided by the GSDC 2023–2024 competition, which may constrain the generalizability of our findings to other smartphone models not represented in this dataset. Third, since data collection was conducted with smartphones uniformly mounted on vehicles, the study may not fully capture the variability

introduced by typical user interactions, such as handheld use with varying orientations and movements. Lastly, the computational demands of the CWT Morlet transform and ensemble model training require substantial RAM, which could limit scalability in memory-constrained environments or on lower-specification hardware, especially during training and processing of large datasets.

E. IMPLICATIONS AND FUTURE DIRECTIONS

Our findings highlight the untapped potential of gyroscope data in smartphone model detection and open new avenues for research in device fingerprinting using inertial sensors. Future work could explore the integration of additional sensors like accelerometers or magnetometers to further enhance identification capabilities through multi-modal sensor fusion. Assessing the method's robustness under varied real-world conditions and user behaviors would also be valuable. Additionally, investigating privacy implications and developing strategies to protect user data while utilizing gyroscope readings for device authentication are important considerations. Extending this methodology to other devices, such as wearables or IoT sensors, could broaden its applicability across different technological domains.

VI. CONCLUSION

This study has demonstrated the effectiveness of using gyroscope data alone for smartphone model detection, achieving a remarkable accuracy of 99.01% and an F1 score of 0.99. By leveraging advanced deep learning models, a fine-tuned WaveNet and EfficientNetV2 combined in an ensemble framework with an attention mechanism, we successfully captured subtle differences in gyroscope readings, even under challenging conditions where smartphones were uniformly aligned during data collection.

Our findings highlight the untapped potential of gyroscope data as a robust and reliable fingerprint for smartphone model detection, surpassing previous studies that relied on combinations of sensors or longer data collection durations. CWT with the Morlet wavelet for detailed time-frequency analysis further enhanced the model's ability to extract complex patterns from the gyroscope data. A significant advantage of our method is its reliance on short-duration (approximately 1–2 seconds) gyroscope readings, making it efficient and practical for real-world applications where rapid and accurate device identification is essential.

The robustness of our models in processing real-world data suggests that similar methodologies can be applied to other types of sensors and contexts, potentially improving mobile device security and authentication. Building upon these promising results, future research could explore incorporating additional sensors to enhance device identification capabilities through multi-modal sensor fusion techniques. Evaluating the method's robustness under diverse real-world conditions, including varying user behaviors, device orientations, and environmental factors, would further validate its applicability. Investigating the privacy implications and developing mechanisms to protect user data while leveraging gyroscope readings for device authentication are also important considerations. Extending this methodology to other domains, such as wearable devices or IoT sensors, could broaden the impact of our findings across different technological landscapes.

By addressing these areas, we can continue to advance the field of device fingerprinting, enhancing security and authentication methods while ensuring user privacy and adapting to the evolving technological environment. Our work sets a new standard in smartphone model detection using gyroscope data alone, paving the way for future advancements and applications in this field.

ACKNOWLEDGMENT

The authors declare no conflict of interest. The grammar and readability of this article were refined with the assistance of AI systems.

REFERENCES

[1] A. Berdich, B. Groza, and R. Mayrhofer, "A survey on fingerprinting technologies for smartphones based on embedded transducers," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14646–14670, Apr. 2023, doi: 10.1109/JIOT.2023.3277883.

[2] M. Taleby, Q. Li, M. Rabbani, and A. Raza, "A survey on smartphones security: Software vulnerabilities, malware, and attacks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 10, pp. 30–45, 2017, doi: 10.14569/ijacsa.2017.081005.

[3] F. Breitinger, R. Tully-Doyle, and C. Hassenfeldt, "A survey on smartphone user's security choices, awareness and education," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101647, doi: 10.1016/j.cose.2019.101647.

[4] M. H. Khan and M. Ali Shah, "Survey on security threats of smartphones in Internet of Things," in *Proc. 22nd Int. Conf. Autom. Comput. (ICAC)*, Sep. 2016, pp. 560–566, doi: 10.1109/IConAC.2016.7604979.

[5] S. Caro-Alvaro, E. Garcia-Lopez, A. Brun-Guajardo, A. Garcia-Cabot, and A. Mavri, "Gesture-based interactions: Integrating accelerometer and gyroscope sensors in the use of mobile apps," *Sensors*, vol. 24, no. 3, p. 1004, Feb. 2024, doi: 10.3390/s24031004.

[6] S. Song, Y. Wang, S. Wang, C. Wang, and Y. Yao, "A Unity3D engine-based virtual reality game for improving cervical mobility using gyroscope controls," in *Frontiers in Artificial Intelligence and Applications*, vol. 385. Amsterdam, The Netherlands: IOS Press, 2024, pp. 257–265, doi: 10.3233/FAIA240160.

[7] H. Said and S. M. A'Zmi, "The Coriolis Effect: Four centuries of conflict between common sense and mathematics, Part I: A history to 1885," *Fak. Pendidik. UTM*, vol. 2, pp. 1–24, Jan. 2001.

[8] V. Capuano, L. Xu, and J. Estrada Benavides, "Smartphone MEMS accelerometer and gyroscope measurement errors: Laboratory testing and analysis of the effects on positioning performance," *Sensors*, vol. 23, no. 17, p. 7609, Sep. 2023, doi: 10.3390/s23177609.

[9] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.

[10] Z. Ding and M. Ming, "Accelerometer-based mobile device identification system for the realistic environment," *IEEE Access*, vol. 7, pp. 131435–131447, 2019, doi: 10.1109/ACCESS.2019.2939205.

[11] A. Das, N. Borisov, and E. Chou, "Every move you make: Exploring practical issues in smartphone motion sensor fingerprinting and countermeasures," *Proc. Privacy Enhancing Technol.*, vol. 2018, no. 1, pp. 88–108, Jan. 2018.

[12] M. H. Al Banna, M. Ali Haider, M. J. Al Nahian, M. M. Islam, K. A. Taher, and M. S. Kaiser, "Camera model identification using deep CNN and transfer learning approach," in *Proc. Int. Conf. Robot., Elect. Signal Process. Techn. (ICREST)*, Jan. 2019, pp. 626–630, doi: 10.1109/ICREST.2019.8644194.

[13] X. Li, D. Yan, L. Dong, and R. Wang, "Anti-forensics of audio source identification using generative adversarial network," *IEEE Access*, vol. 7, pp. 184332–184339, 2019, doi: 10.1109/ACCESS.2019.2960097.

[14] G. Baldini and I. Amerini, "Smartphones identification through the built-in microphones with convolutional neural network," *IEEE Access*, vol. 7, pp. 158685–158696, 2019, doi: 10.1109/ACCESS.2019.2950859.

[15] G. Baldini, I. Amerini, and C. Gentile, "Microphone identification using convolutional neural networks," *IEEE Sensors Lett.*, vol. 3, no. 7, pp. 1–4, Jul. 2019, doi: 10.1109/LSENS.2019.2923590.

[16] Y. Jiang and F. H. F. Leung, "Source microphone recognition aided by a kernel-based projection method," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 11, pp. 2875–2886, Nov. 2019, doi: 10.1109/TIFS.2019.2911175.

[17] C. Jin, R. Wang, and D. Yan, "Source smartphone identification by exploiting encoding characteristics of recorded speech," *Digit. Invest.*, vol. 29, pp. 129–146, Jun. 2019, doi: 10.1016/j.diin.2019.03.003.

[18] G. Baldini and I. Amerini, "An evaluation of entropy measures for microphone identification," *Entropy*, vol. 22, no. 11, p. 1235, Oct. 2020, doi: 10.3390/e22111235.

[19] D. Bykhovsky, "Recording device identification by ENF harmonics power analysis," *Forensic Sci. Int.*, vol. 307, Feb. 2020, Art. no. 110100, doi: 10.1016/j.forsciint.2019.110100.

[20] T. Qin, R. Wang, D. Yan, and L. Lin, "Source cell-phone identification in the presence of additive noise from CQT domain," *Information*, vol. 9, no. 8, p. 205, Aug. 2018, doi: 10.3390/info9080205.

[21] D. Cozzolino, F. Marra, D. Gragnaniello, G. Poggi, and L. Verdoliva, "Combining PRNU and noiseprint for robust and efficient device source identification," *EURASIP J. Inf. Secur.*, vol. 2020, no. 1, pp. 1–12, Dec. 2020, doi: 10.1186/s13635-020-0101-7.

[22] N. Zandi and F. Razzazi, "Source camera identification using WLBP descriptor," in *Proc. Int. Conf. Mach. Vis. Image Process. (MVIP)*, Feb. 2020, pp. 1–6, doi: 10.1109/MVIP49855.2020.9187484.

[23] K. S. Choi, E. Y. Lam, and K. K. Y. Wong, "Source camera identification using footprints from lens aberration," *Proc. SPIE*, vol. 6069, no. 852, 2006, Art. no. 60690J, doi: 10.1117/12.649775.

[24] B. Xu, X. Wang, X. Zhou, J. Xi, and S. Wang, "Source camera identification from image texture features," *Neurocomputing*, vol. 207, pp. 131–140, Sep. 2016, doi: 10.1016/j.neucom.2016.05.012.

[25] V. U. Sameer, A. Sarkar, and R. Naskar, "Source camera identification model: Classifier learning, role of learning curves and their interpretation," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2017, pp. 2660–2666, doi: 10.1109/WiSPNET.2017.8300246.

[26] A. Rashidi and F. Razzazi, "Single image camera identification using I-vectors," in *Proc. 7th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, Oct. 2017, pp. 406–410, doi: 10.1109/ICCKE.2017.8167913.

[27] Y. Huang, L. Cao, J. Zhang, L. Pan, and Y. Liu, "Exploring feature coupling and model coupling for image source identification," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 12, pp. 3108–3121, Dec. 2018, doi: 10.1109/TIFS.2018.2838079.

[28] A. Das, N. Borisov, and M. Caesar, "Exploring ways to mitigate sensor-based smartphone fingerprinting," 2015, *arXiv:1503.01874*.

[29] A. Das, N. Borisov, and M. Caesar, "Do you hear what I hear?: Fingerprinting smart devices through embedded acoustic components," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 441–452, doi: 10.1145/2660267.2660325.

[30] R. Buchholz, C. Kraetzer, and J. Dittmann, "Microphone classification using Fourier coefficients," in *Proc. Int. Workshop Inf. Hiding*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 5806, Jun. 2009, pp. 235–246, doi: 10.1007/978-3-642-04431-1_17.

[31] H. Q. Vu, S. Liu, X. Yang, Z. Li, and Y. Ren, "Identifying microphone from noisy recordings by using representative instance one class-classification approach," *J. Netw.*, vol. 7, no. 6, pp. 908–917, Jun. 2012, doi: 10.4304/jnw.7.6.908-917.

[32] A. Das, N. Borisov, and M. Caesar, "Tracking mobile web users through motion sensors: Attacks and defenses," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2016, pp. 21–24, doi: 10.14722/ndss.2016.23390.

[33] G. Baldini, G. Steri, I. Amerini, and R. Caldelli, "The identification of mobile phones through the fingerprints of their built-in magnetometer: An analysis of the portability of the fingerprints," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2017, pp. 1–6, doi: 10.1109/CCST.2017.8167855.

[34] J. Tian, J. Zhang, X. Li, C. Zhou, R. Wu, Y. Wang, and S. Huang, "Mobile device fingerprint identification using gyroscope resonance," *IEEE Access*, vol. 9, pp. 160855–160867, 2021, doi: 10.1109/ACCESS.2021.3131408.

[35] D. P. Chowdhury, S. Bakshi, P. K. Sa, and B. Majhi, "Wavelet energy feature based source camera identification for ear biometric images," *Pattern Recognit. Lett.*, vol. 130, pp. 139–147, Feb. 2020, doi: 10.1016/j.patrec.2018.10.009.

[36] Ö. Eskidere, "Source microphone identification from speech recordings based on a Gaussian mixture model," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 22, pp. 754–767, Jan. 2014, doi: 10.3906/elk-1207-74.

[37] C. Hanilçi and T. Kinnunen, "Source cell-phone recognition from recorded speech using non-speech segments," *Digit. Signal Process.*, vol. 35, pp. 75–85, Dec. 2014, doi: 10.1016/j.dsp.2014.08.008.

[38] Ö. Eskidere and A. Karatutlu, "Source microphone identification using multitaper MFCC features," in *Proc. 9th Int. Conf. Electr. Electron. Eng. (ELECO)*, Nov. 2015, pp. 227–231, doi: 10.1109/ELECO.2015.7394482.

[39] V. A. Hadoltikar, V. R. Ratnaparkhe, and R. Kumar, "Optimization of MFCC parameters for mobile phone recognition from audio recordings," in *Proc. 3rd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Jun. 2019, pp. 777–780, doi: 10.1109/ICECA.2019.8822177.

[40] L. Zou, Q. He, and X. Feng, "Cell phone verification from speech recordings using sparse representation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 1787–1791, doi: 10.1109/ICASSP.2015.7178278.

[41] X. Zhou, X. Zhuang, H. Tang, M. Hasegawa-Johnson, and T. S. Huang, "Novel gaussianized vector representation for improved natural scene categorization," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 702–708, Jun. 2010, doi: 10.1016/j.patrec.2009.12.010.

[42] S. Mandelli, D. Cozzolino, P. Bestagini, L. Verdoliva, and S. Tubaro, "CNN-based fast source device identification," *IEEE Signal Process. Lett.*, vol. 27, pp. 1285–1289, 2020, doi: 10.1109/LSP.2020.3008855.

[43] A. M. Rafi, T. I. Tonmoy, U. Kamal, Q. M. J. Wu, and M. K. Hasan, "RemNet: Remnant convolutional neural network for camera model identification," *Neural Comput. Appl.*, vol. 33, no. 8, pp. 3655–3670, Apr. 2021, doi: 10.1007/s00521-020-05220-y.

[44] D. Dal Cortivo, S. Mandelli, P. Bestagini, and S. Tubaro, "CNN-based multi-modal camera model identification on video sequences," *J. Imag.*, vol. 7, no. 8, p. 135, Aug. 2021, doi: 10.3390/jimaging7080135.

[45] R. Rouhi, F. Bertini, and D. Montesi, "No matter what images you share, you can probably be fingerprinted anyway," *J. Imag.*, vol. 7, no. 2, p. 33, Feb. 2021, doi: 10.3390/jimaging7020033.

[46] P. R. Mendes Jr., L. Bondi, P. Bestagini, S. Tubaro, and A. Rocha, "An in-depth study on open-set camera model identification," *IEEE Access*, vol. 7, pp. 180713–180726, 2019, doi: 10.1109/ACCESS.2019.2921436.

[47] H. Yao, T. Qiao, M. Xu, and N. Zheng, "Robust multi-classifier for camera model identification based on convolution neural network," *IEEE Access*, vol. 6, pp. 24973–24982, 2018, doi: 10.1109/ACCESS.2018.2832066.

[48] D. Freire-Obregón, F. Narducci, S. Barra, and M. Castrillón-Santana, "Deep learning for source camera identification on mobile devices," *Pattern Recognit. Lett.*, vol. 126, pp. 86–91, Sep. 2019, doi: 10.1016/j.patrec.2018.01.005.

[49] P. Yang, R. Ni, Y. Zhao, and W. Zhao, "Source camera identification based on content-adaptive fusion residual networks," *Pattern Recognit. Lett.*, vol. 119, pp. 195–204, Mar. 2019, doi: 10.1016/j.patrec.2017.10.016.

[50] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-based camera model fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 144–159, 2020, doi: 10.1109/TIFS.2019.2916364.

[51] X. Ding, Y. Chen, Z. Tang, and Y. Huang, "Camera identification based on domain knowledge-driven deep multi-task learning," *IEEE Access*, vol. 7, pp. 25878–25890, 2019, doi: 10.1109/ACCESS.2019.2897360.

[52] M. Zhao, B. Wang, F. Wei, M. Zhu, and X. Sui, "Source camera identification based on coupling coding and adaptive filter," *IEEE Access*, vol. 8, pp. 54431–54440, 2020, doi: 10.1109/ACCESS.2019.2959627.

[53] V. K. Verma and N. Khanna, "CNN-based system for speaker independent cell-phone identification from recorded audio," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, Jan. 2019, pp. 53–61.

[54] X. Lin, J. Zhu, and D. Chen, "Subband aware CNN for cell-phone recognition," *IEEE Signal Process. Lett.*, vol. 27, pp. 605–609, 2020, doi: 10.1109/LSP.2020.2985594.

[55] M. A. Qamhan, H. Altaheri, A. H. Meftah, G. Muhammad, and Y. A. Alotaibi, "Digital audio forensics: Microphone and environment classification using deep learning," *IEEE Access*, vol. 9, pp. 62719–62733, 2021, doi: 10.1109/ACCESS.2021.3073786.

[56] A. Berdich, B. Groza, R. Mayrhofer, E. Levy, A. Shabtai, and Y. Elovici, "Sweep-to-unlock: Fingerprinting smartphones based on loudspeaker roll-off characteristics," *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2417–2434, Apr. 2023, doi: 10.1109/TMC.2021.3119987.
[57] I. Amerini, P. Bestagini, L. Bondi, R. Caldelli, M. Casini, and S. Tubaro, "Robust smartphone fingerprint by mixing device sensors features for mobile strong authentication," in *Proc. IS&T Int. Symp. Electron. Imaging Sci. Technol.*, 2016, pp. 1–8, doi: 10.2352/ISSN.2470-1173.2016.8.MWSF-088.
[58] T. Hupperich, H. Hosseini, and T. Holz, "Leveraging sensor fingerprinting for mobile device authentication," in *Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, Jan. 2016, pp. 377–396, doi: 10.1007/978-3-319-40667-1_19.
[59] G. Baldini, G. Steri, F. Dimc, R. Giuliani, and R. Kamnik, "Experimental identification of smartphones using fingerprints of built-in micro-electro mechanical systems (MEMS)," *Sensors*, vol. 16, no. 6, p. 818, Jun. 2016, doi: 10.3390/s16060818.
[60] I. Amerini, R. Becarelli, R. Caldelli, A. Melani, and M. Niccolai, "Smartphone fingerprinting combining features of on-board sensors," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2457–2466, Oct. 2017, doi: 10.1109/TIFS.2017.2708685.
[61] X.-Y. Li, H. Liu, L. Zhang, Z. Wu, Y. Xie, G. Chen, C. Wan, and Z. Liang, "Finding the stars in the fireworks: Deep understanding of motion sensor fingerprint," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 1945–1958, Oct. 2019, doi: 10.1109/TNET.2019.2933269.
[62] J. Zhang, A. R. Beresford, and I. Sheret, "SensorID: Sensor calibration fingerprinting for smartphones," in *Proc. IEEE Symp. Secur. Privacy*, May 2019, pp. 638–655, doi: 10.1109/sp.2019.00072.
[63] J. Zhang, A. R. Beresford, and I. Sheret, "Factory calibration fingerprinting of sensors," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1626–1639, 2021, doi: 10.1109/TIFS.2020.3039685.
[64] E. Al-Mahadeen, M. Alghamdi, A. S. Tarawneh, M. A. Alrowaily, M. Alrashidi, I. S. Alkhazi, A. Mbaidin, A. A. Alkasasbeh, M. A. Abbadi, and A. B. Hassanat, "Smartphone user identification/authentication using accelerometer and gyroscope data," *Sustainability*, vol. 15, no. 13, p. 10456, Jul. 2023, doi: 10.3390/su151310456.
[65] C. Hanilci, F. Ertas, T. Ertas, and Ö. Eskidere, "Recognition of brand and models of cell-phones from recorded speech signals," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 625–634, Apr. 2012, doi: 10.1109/TIFS.2011.2178403.
[66] L. J. G. Villalba, A. L. S. Orozco, R. R. López, and J. H. Castro, "Identification of smartphone brand and model via forensic video analysis," *Expert Syst. Appl.*, vol. 55, pp. 59–69, Aug. 2016, doi: 10.1016/j.eswa.2016.01.025.
[67] G. Baldini, F. Dimc, R. Kamnik, G. Steri, R. Giuliani, and C. Gentile, "Identification of mobile phones using the built-in magnetometers stimulated by motion patterns," *Sensors*, vol. 17, no. 4, p. 783, Apr. 2017, doi: 10.3390/s17040783.
[68] B. Hosler, O. Mayer, B. Bayar, X. Zhao, C. Chen, J. A. Shackleford, and M. C. Stamm, "A video camera model identification system using deep learning and fusion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 8271–8275, doi: 10.1109/ICASSP.2019.8682608.
[69] D. Salvi, D. U. Leonzio, A. Giganti, C. Eutizi, S. Mandelli, P. Bestagini, and S. Tubaro, "POLIPHONE: A dataset for smartphone model identification from audio recordings," 2024, *arXiv:2410.06221*.
[70] A. Chow, D. Orendorff, M. Fu, M. Khider, S. Dane, and V. Gulati. (2023). *Google Smartphone Decimeter Challenge 2023–2024*. Kaggle. [Online]. Available: https://kaggle.com/competitions/smartphone-decimeter-2023
[71] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.
[72] M. Tan and Q. V. Le, "EfficientNetV2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2021, pp. 10096–10106.
[73] B. Russell and J. Han, "Jean Morlet and the continuous wavelet transform," CREWES, Univ. Calgary, Calgary, AB, Canada, Tech. Rep. 1946, 2016, vol. 28, no. 1946, p. 1.
[74] F. Van Diggelen, "Global differential GPS system (GDGPS) & Galileo high accuracy service (HAS) comparison," Nat. Space-Based Positioning, Navigation, Timing Advisory Board, League City, TX, USA, Tech. Rep. 29, Dec. 2023. [Online]. Available: https://www.gps.gov/governance/advisory/meetings/2023-12/van-diggelen.pdf
[75] G. Mostafa, H. Mahmoud, T. Abd El-Hafeez, and M. E. ElAraby, "Feature reduction for hepatocellular carcinoma prediction using machine learning algorithms," *J. Big Data*, vol. 11, no. 1, pp. 1–15, Jun. 2024, doi: 10.1186/s40537-024-00944-3.
[76] G. Mostafa, H. Mahmoud, T. Abd El-Hafeez, and M. E. ElAraby, "The power of deep learning in simplifying feature selection for hepatocellular carcinoma: A review," *BMC Med. Informat. Decis. Making*, vol. 24, no. 1, p. 287, Oct. 2024, doi: 10.1186/s12911-024-02682-1.
[77] H. Mamdouh Farghaly and T. Abd El-Hafeez, "A high-quality feature selection method based on frequent and correlated items for text classification," *Soft Comput.*, vol. 27, no. 16, pp. 11259–11274, Aug. 2023, doi: 10.1007/s00500-023-08587-x.
[78] H. M. Farghaly and T. A. El-Hafeez, "A new feature selection method based on frequent and associated itemsets for text classification," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 25, pp. 1–17, Nov. 2022, doi: 10.1002/cpe.7258.
[79] B. Singh and A. P. Singh, "Edge detection in gray level images based on the Shannon entropy," *J. Comput. Sci.*, vol. 4, no. 3, pp. 186–191, Mar. 2008, doi: 10.3844/jcssp.2008.186.191.
[80] M. A. El-Sayed and T. A.-E. Hafeez, "New edge detection technique based on the Shannon entropy in gray level images," 2012, *arXiv:1211.2502*.
[81] E. H. I. Eliwa, A. M. E. Koshiry, T. A. El-Hafeez, and A. Omar, "Secure and transparent lung and colon cancer classification using blockchain and Microsoft Azure," *Adv. Respiratory Med.*, vol. 92, no. 5, pp. 395–420, Oct. 2024, doi: 10.3390/arm92050037.
[82] S. Scholl, "Fourier, Gabor, Morlet or wigner: Comparison of time-frequency transforms," 2021, *arXiv:2101.06707*.
[83] M. Louis, "20:21," *CJEM*, vol. 15, no. 3, p. 191, May 2013, doi: 10.2310/8000.2013.131108.
[84] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017. Accessed: Aug. 15, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
[85] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2016, *arXiv:1511.07122*.
[86] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018, *arXiv:1811.03378*.
[87] T. Szandała, "Review and comparison of commonly used activation functions for deep neural networks," in *Bio-Inspired Neurocomputing* (Studies in Computational Intelligence), vol. 903, A. K. Bhoi, P. K. Mallick, C.-M. Liu, and V. E. Balas, Eds. Singapore: Springer, 2021, pp. 203–224, doi: 10.1007/978-981-15-5495-7_11.

ERKAN KIYMIK was born in Gaziantep, Türkiye. He received the B.Sc. degree in electrical-electronics engineering from Hasan Kalyoncu University, Gaziantep, in 2015, and the M.Sc. and Ph.D. degrees in electrical-electronics engineering from Gaziantep University, Gaziantep, in 2018 and 2022, respectively. Since 2016, he has been a member of the Academic Staff with the Department of Electrical-Electronics Engineering, Hasan Kalyoncu University. His research interests include artificial intelligence, machine learning, and signals.

ALİ EMRE ÖZTÜRK was born in Ankara, Türkiye. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical-electronics engineering from Gaziantep University, Gaziantep, Türkiye, in 2009, 2014, and 2021, respectively. Since 2011, he has been an Academic Staff in electrical-electronics engineering with Hasan Kalyoncu University, Gaziantep. His research interests include machine learning, and signal and systems.