Machine Learning Analysis on Gunshot Recognition

Siddat B. Nesar
Electrical and Computer Engineering
Montana State University
Bozeman, Montana, USA
siddatnesar@montana.edu

Bradley M. Whitaker
Electrical and Computer Engineering
Montana State University
Bozeman, Montana, USA
bradley.whitaker1@montana.edu

Robert C. Maher
Electrical and Computer Engineering
Montana State University
Bozeman, Montana, USA
rmaher@montana.edu

***Abstract*—The ability to recognize a gunshot has significance in reinforcing public safety, assisting in crime scene investigations, and preventing gun violence. This paper investigates the efficiency of various machine learning models for gunshot recognition. We present a model to identify the type of pistol or rifle discharged by analyzing only an audio signal of the gunshot. Among the array of methods explored, AdaBoost performed the best achieving an accuracy of 99.9% and sustaining over 80% accuracy with 40 dB conditions. Additionally, we experimented with the importance level of the features to identify the most relevant variables that boost the performance of the algorithms.**

***Index Terms*—gunshot recognition, adaboost, ensemble, gaussian noise, feature selection**

I. Introduction

Public safety is a significant issue across any city around the globe. To ensure quick response by law enforcement authorities, a reliable and realistic gunshot detection system is necessary. To speed up the investigation process, a crime scene must be understood and the authorities should have the ability to recreate the scene. A robust gunshot recognition system will become useful by aiding in the crime scene reconstruction, estimating the shooter's position and the trajectory of the projectile, and verifying the details provided by witnesses.

Audiovisual surveillance systems are becoming more popular with the increase in crime. ShotSpotter [1] is a gunshot detection system in the SafetySmart platform introduced by SoundThinking. ShotSpotter is used by law enforcement agencies by strategically placing a network of audio and video sensors in city areas. The system captures all the surrounding sounds; if it detects any gunfire sounds, it triangulates the location and alerts the proper authorities. The system collects and analyzes the data to create a map of areas prone to gun violence. However, the system does not give information about the gun used for the violence and may raise false alarms for sounds like car backfires or fireworks.

Most of the research on gunshot detection systems uses training data either from a very controlled environment using multiple microphones or a traditional recording device [2]. This type of data usually contains spectral information only, and thus several pattern recognition approaches are used to implement gunshot detection systems without any spatial information. The input time domain signal is usually divided into multiple short-windowed frames and some widely used features—such as Mel-Frequency Cepstrum Coefficient (MFCC), Linear Prediction Coefficients (LPC), Linear Prediction Cepstral Coefficients (LPCC), and temporal features—are extracted. The features are then merged and fed into different classification models for gunshot detection. [3, 4] proposed a hierarchical classification method using the Gaussian Mixture Model (GMM), whereas [5] proposed a Hidden Markov Model (HMM) with autoregressive source densities using a nonparametric Bayesian technique-to classify gunshots. Integrating power spectral density with MFCC emerged as a promising feature array for gunshot detection in noisy environments [6]. Kiktova et. al. in [7] suggested a combination of a Hidden Markov Model (HMM) and Viterbi decoding algorithm to recognize gunshots collected in an urban environment. A Viterbi algorithm finds the candidates with the maximum likelihood of the hidden states in a HMM. A combination of MFCC, LPC, Gammatone cepstral coefficients, and spectral centroid are used in [8] as features for bagged tree ensemble and support vector machine classifiers. Different SNR settings were also tested in this work and the arrival time difference is considered to localize the firearm.

Convolutional Neural Networks (CNN) [9] and transfer learning methods [10] have become popular in recent times for gunshot detection, enhancing classification and localization of the firearm. [11] proposed a wireless gunshot recognition algorithm referred to as EfficientNetTime, a lightweight deep learning model that leverages 1D convolution network and knowledge distillation. The name EfficientNetTime received its name because it reduces the computational cost of convolution. [12] introduces a low-cost and high-accuracy gunshot detection system using a Raspberry Pi. Morehead *et al.* deployed both 1D and 2D CNNs for training the model using spectrograms as features. The performance of the 2D CNN is slightly better than the 1D CNN. MFCC of gun sounds collected from YouTube is used in [13] to create new ensemble features utilizing a Discrete Wavelet Transform Random Forest Probabilistic (DWT-RFP) approach, which was then fed to a meta-learner referred to as Meta-RF-KN (MRK). [14] explored several established deep learning models such as Inception-ResnetV2, Inception-V3, YOLOv4, VGG16, etc. for weapon detection from CCTV cameras. Self-attention-based transformer architectures and convolutional neural networks are experimented in [15], with the transformer learning model performing better.

In this work, we explored different machine-learning models for recognizing gunshot data for pistols and rifles. We experimented with the robustness of the algorithms by introducing noise to the data and lastly reduced the computational cost

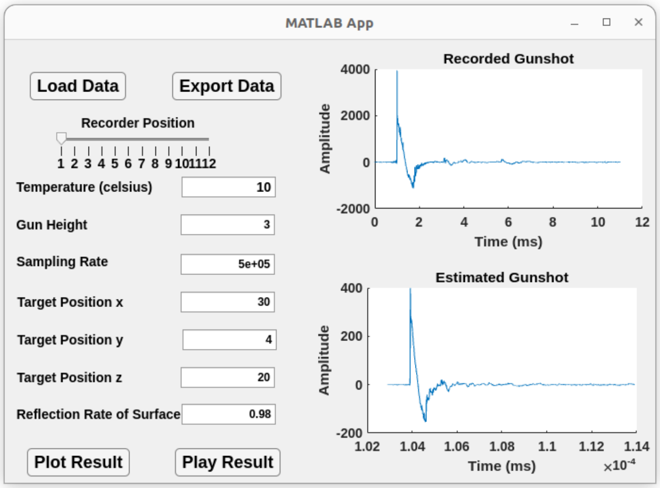

Fig. 1. Gunshot Generator App built with MATLAB. First load the recorded data, then enter the parameters: temperature, gun height while recording, sampling rate of the recording, target or microphone position (x, y, z), and reflection rate of the surface. Then hit 'plot result' to generate the data at your desired azimuth and reflection rate. The generated data considers the echoes and displays the delayed signal.

by utilizing only the important and relevant features without affecting the optimal performance of the models. First, we take a peek at the unique dataset used in this work in Section II. Then, section III discusses three different machine learning algorithms briefly that are used in this work, followed by the effect of adding noise to the data and performing feature selection techniques. Section IV illustrates the performance evaluation of the algorithms for various instances. Finally, section V wraps up the work with an overall summary.

II. A Unique Dataset

Most of the research in gunshot classification uses recordings from YouTube or sound effects from libraries rather than the true-to-life high-quality recordings we have available. Since the muzzle blast sound is only 3-4 milliseconds in duration, researchers who train with reverberant recordings hundreds of milliseconds or more in length are learning more about the acoustical impulse response of the location where the recording was made than they are about the sound of the gun itself. Synthetic data using a geometric approach is generated utilizing quasi-anechoic recordings [16] from three different firearms: a Glock-19 pistol, an AR15 rifle, and a 308R rifle. The audio was sampled at 500 kHz to obtain the supersonic sound waves. Ten shots from each gun were recorded using 12 different microphones. We generate a unique data set considering a single-ground reflection, different azimuths, and simulated distances that represent a more realistic and comprehensive set than the typically limited scope of gunshot sounds often available. Fig 1 illustrates the data generator app built using MATLAB and an example generated signal for Glock-19. The high-quality anechoic recordings were used to generate 12100 samples of synthetic data for each of the firearms. The data was split 80-20 for the train and test sets, and a 5-fold cross-validation was performed on the training data. The training and validation samples total up to 29040, while 7260 samples are used for testing.

From the gunshot data, 19 features are extracted. These features encompass a broad spectrum of characteristics, including MFCC with 13 coefficients, spectral spread, spectral flatness, spectral entropy, spectral skewness, harmonic ratio, and kurtosis. MFCC is a widely used audio feature that encapsulates the short-term power spectrum. Spectral spread measures the width of the spectrum across the frequencies and how the energy is distributed, while spectral flatness examines the uniformity of the energy levels across the bandwidth. The spectral entropy gauges the degree of uncertainty in the spectral energy distribution based on the principles of Shannon entropy. Spectral skewness delves into the asymmetry of the power spectrum and is the third standardized moment of the power spectrum, while harmonic ratio discerns the ratio of harmonic to non-harmonic components in the signal. Lastly, kurtosis expresses the flatness or peakedness of the power spectrum and is the fourth standardized moment of the power spectrum.

III. Analysis

A. Models Used

Numerous machine learning and deep learning models have been proposed for gunshot recognition in audio forensics. This section explores a few state-of-the-art models and the hyperparameter optimization of the models.

1) Ensemble Classifiers: Ensemble learning is a machine-learning approach that combines multiple base models to improve performance and prediction capability [17]. Instead of relying on the predictions of one single model, ensemble learning techniques leverage the collective predictions to decide on a final prediction. Bagging [18], Boosting [19], and Stacking [20] are three main variations of ensemble learning. Any machine learning model can be used for ensemble learning, however, in this section, we will consider a decision tree as the base model for ensemble learning. Bootstrap Aggregating or Bagging in short involves training several decision trees of randomly selected subset of the data, and either averaging or voting on the predictions of all the trained models. As an example, let X be the input data and Y be the target variable, and the set of the data with n subsets is

$$(X_1, Y_1), \ldots, (X_n, Y_n)$$

For each decision tree model T_i, each X_i is trained to obtain the predictions Y_i. The final prediction is then averaged as:

$$Y_{ensemble} = \frac{1}{N} \sum_{i=1}^{N} Y_i$$

Boosting is the process of sequentially training several weak learners focusing on the mistakes of the previously trained models. The goal is to build a strong learner by learning from the mistakes of all the weak learners. The final prediction is a weighted combination of all the predictions, where the weak learners performing better are assigned a higher weight. On the other hand, stacking uses a meta-model (usually a different higher-level machine-learning model) to train on the predictions of the base model to obtain the final prediction.

2) Support Vector Machines: Support vector machine (SVM) is a versatile tool for classification or regression tasks. SVM finds a hyperplane leveraging the support vectors in an N-dimensional space that best separates the data points of different classes. The support vectors define the boundaries to the closest data points and the hyperplane [21]. The goal of SVM is to maximize the boundary and various kernels are usually deployed for non-linear data. To simply explain the SVM, the classifier can be expressed as:

$$h_{w,b}(x) = f(w^T x + b)$$

where x is the feature vector, y are the labels, w is the weight vector, b is the bias, $f(z) = 1$ if $z \geq 0,$ and $f(z) = 0$ otherwise. The optimal margin can be obtained by minimizing the weight vector.

$$\begin{aligned} &\min_{w,b} \frac{1}{2}||w||^2 \\ &\text{s.t.} \;\; y^i(w^T x + b) \geq 1, i = 1, \ldots, n \end{aligned}$$

SVM is generally a two-class model. However, the algorithm can be extended for multiclass via the one-vs-one or one-vs-rest method. The one-vs-one approach trains a binary classifier for each of the pairs of the classes, while one-vs-rest trains separate binary classifiers for each of the classes.

3) Neural Networks: A neural network is a machine-learning process inspired by the connections of neurons in the human brain to learn the complex patterns in a given data. Neural networks consist of three main layers: input, hidden, and output layers [22]. The hidden layers can be fully connected and have a weight matrix W and biases b. As an example, the fifth neuron in the third layer can be computationally defined as:

$$z_5^{[3]} = {W_5^{[3]}}^T x + b_5^{[3]} \quad \text{and} \quad \sigma_5^{[3]} = g(z_5^{[3]})$$

Here, $g(z)$ is the activation function that determines whether a neuron should be activated or not. Based on the applications, and the number of layers and connections between the neurons, neural networks can get very complex with an enormous amount of parameters.

4) Hyperparameter Tuning: The hyperparameters of any machine learning model need to be tuned for optimal performance and generalization. The process of trying out different parameter settings iteratively based on a specific target variable to determine the performance level of the model in that specific setting is referred to as hyperparameter tuning. Mathematically hyperparameter optimization [23] can be represented as:

$$x' = \underset{x \in X}{\operatorname{argmin}} f(x)$$

Here, x' is the set of tuned hyperparameters, x can be any value from the space X while $f(x)$ id the objective function that needs to be minimized. The hyperparameter varies based on the models used, such as learning rate, kernel type, number of learners, number of layers and neurons, regularization strength, etc. After the variable parameter bounds are set, various algorithms can be utilized for the tuning. Some of the common hyperparameter tuning algorithms are grid search, random search, and Bayesian optimization [23]. Grid search takes all the possible combinations of the parameters into account and thus is very tedious and computationally expensive. As the name implies, random search randomly selects a group of hyperparameters for each iteration. On the contrary, the Bayesian optimization algorithm is based on Bayes' theorem using a probabilistic approach to steer the hyperparameter search.

B. Adding Gaussian Noise

White Gaussian noise, also known as white noise or Gaussian noise, is a simple noise model with a flat frequency spectrum across the entire bandwidth and follows a Gaussian distribution [24]. Adding Gaussian noise to the input data helps in mimicking real-world conditions and makes the model more robust towards the tasks assigned. Adding the noise to the data is a form of augmentation, that exposes the model to diverse input situations. Additionally, introducing noise helps in generalizing a model better to unseen data.

C. Feature Selection

Feature selection is the process of selecting a subset of the features based on their importance on the performance of a machine learning model. This technique reduces the dimensionality of the data and computational costs and speeds up the training process while maintaining optimal performance. Feature selection can be categorized into three main types: filter type, wrapper type, and embedded type [25]. Filter-type feature selection uses statistical measures such as feature variance and feature relevance to measure the importance of the features. This method can be considered as a part of the data preprocessing step and is not related to the training model. A wrapper-type feature selection algorithm uses a subset of features to evaluate the performance of a pre-selected model and continues training iterations until the stopping criteria are met. The importance of the features is directly obtained during the model training phase in the case of embedded-type feature selection algorithms. All the feature selection algorithms find a relevant set of features and can be applied to either numerical or categorical features.

IV. Results

This section elaborates on the experimental results of different machine learning algorithms and evaluates their performance in optimized settings, added noise, and with a subset of features.

A. Optimized Hyperparameters

We have used ensemble classifiers, support vector machines, and shallow neural networks for the machine learning analysis. The models are also hyperparameter-tuned for optimal

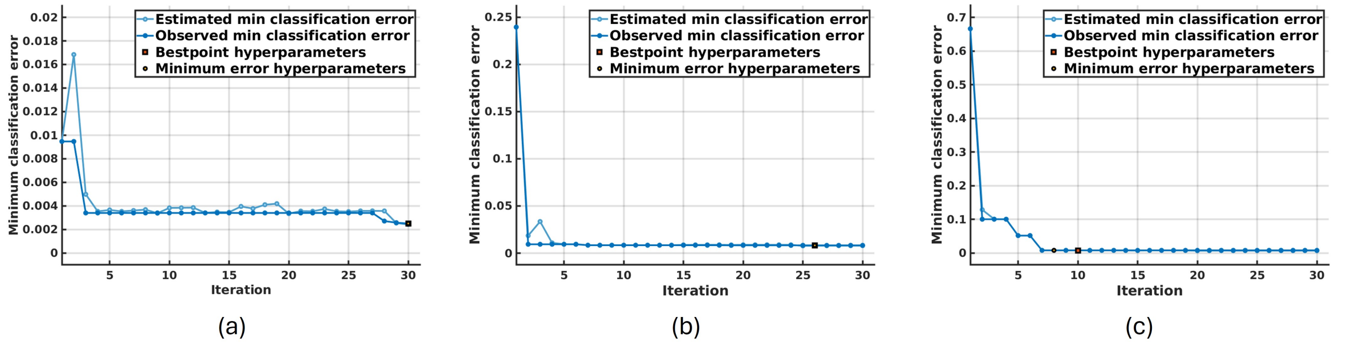

Fig. 2. Minimum Classification Error plots. The legends are Estimated min classification error, Observed min classification error, Bestpoint hyperparameters, and Minimum error hyperparameters. The plots are for (a) ensemble classifiers, (b) support vector machines, and (c) neural networks.

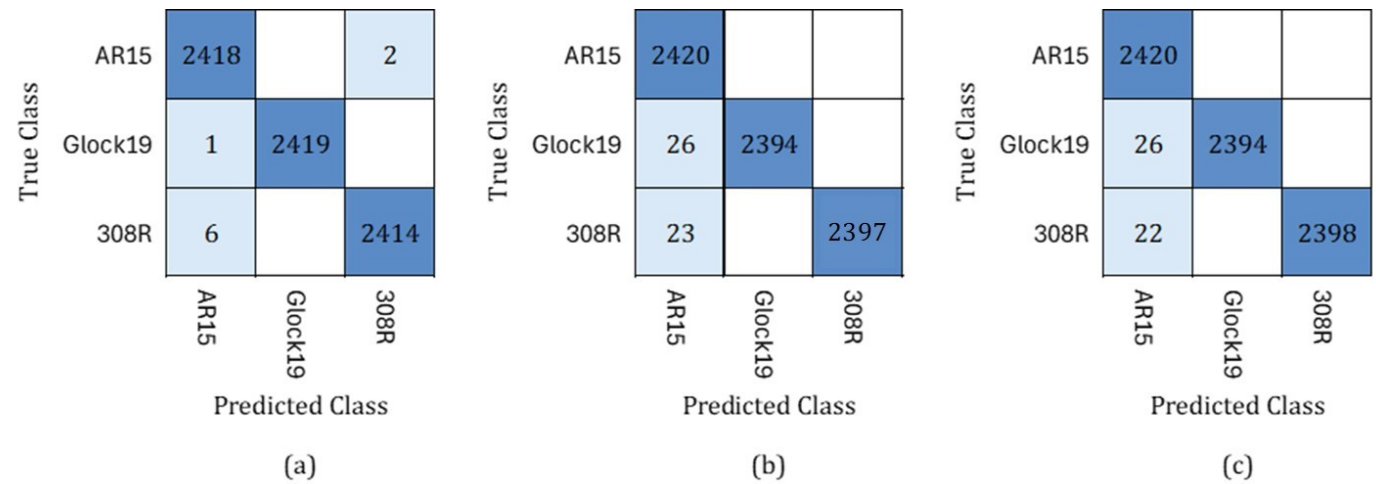

Fig. 3. Confusion Matrices for (a) ensemble classifier, (b) support vector machine, and (c) neural network. Note, that the models are hyperparameter optimized and report the confusion matrix on the unseen test set for each of the models.

performance. We used a Bayesian optimization algorithm for hyperparameter tuning with 30 iterations. The hyperparameter search range for ensemble classifiers was: ensemble methods = Bagging, AdaBoost, RUSBoost, number of learners = 10-500, learning rate = 0.001-1, and maximum number of splits = 1-29039. The optimized parameters include the AdaBoost ensemble method, with 14 learners, a learning rate of 0.8793, and a maximum number of splits of 145. For the support vector machine, the hyperparameter search range was: multiclass method = one-vs-all, one-vs-one, box constraint level = 0.001-1000, kernel scale = 0.001-1000, and kernel function = Gaussian, Linear, Quadratic, Cubic. The optimization results favor for one-vs-one multiclass method with a Gaussian kernel function, a box constraint level of 329.5042, and a kernel scale of 5.2877. The neural network was optimized within the following hyperparameter spaces: number of fully connected layers = 1-3, activation function = ReLU, Tanh, Sigmoid, None, regularization strength = 3.4435e-10-3.4435, and layer sizes = 1-300. The optimized neural network has 1 fully connected layer with Relu as the activation function. The layer had 54 units and the regularization strength was 2.8911e-07.

Fig 2 shows the minimum classification error plots for the three different algorithms respectively. The minimum classification error plots the minimum errors observed against all the iterations. Each of the figures displays errors estimated, errors observed, best point hyperparameter, and minimum error hyperparameter. In the case of ensemble classifiers, the optimal hyperparameters are obtained in the 30th iteration as illustrated in Fig 2(a). From Fig 2(b), it can be observed that the best hyperparameter settings for the support vector machines were obtained in the 26th iteration, and for the neural networks the best hyperparameters were obtained in the 10th iteration as seen in Fig 2(c).

Confusion matrix is a tool used to evaluate the accuracy of a machine learning model based on the true positives and negatives, and false positives and negatives. Fig 3(a) shows the confusion matrix on the test data set for the ensemble classifier and obtained an accuracy of 99.9%, while the validation accuracy was 99.7%. Fig 3(b) reports the confusion matrix on the test set for the support vector machine with a 99.3% accuracy and 99.2% validation accuracy. The neural network achieved similar results as can be seen from Fig 3(c). The test

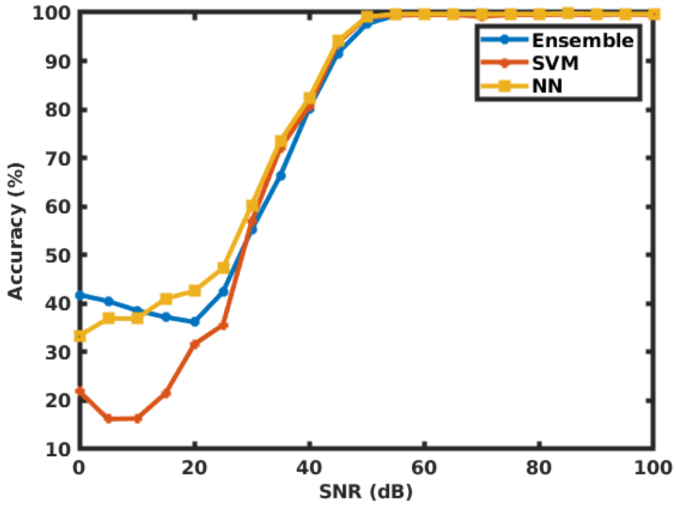

Fig. 4. Comparisons of different machine learning models with different signal-to-noise-ratio data in terms of accuracy. The blue starred curve represents the accuracies for the ensemble classifier, the red circled curve represents the support vector machine, and the orange squared curve represents the neural network.

accuracy was 99.3% while the validation accuracy was 99.2%. The ensemble classifier performed a bit better compared to the rest of the algorithms, however, all the results are very close.

B. Added Gaussian Noise

To increase the robustness in the real world, we trained the models with noiseless data, and the test data was exposed to Additive White Gaussian Noise of different levels from 0 dB to 100 dB with an increment of 5 dB. The same set of features was extracted from the noiseless training data and the noisy test data. A comparison of the performance of the ensemble classifier, support vector machine, and neural network is illustrated in Fig 4 with respect to different SNR values. All the models achieve over 90% accuracy at 45 dB SNR test data which supports the robustness of the system.

C. Selected Features

We explored the stretch of feature selection in our experiment. Feature selection is preferred compared to feature transformation as it is suitable for maintaining the feature space as it was. Feature selection reduces the dimensionality of the feature space and thus reduces the computational cost. We deployed a wrapper-type feature selection method called sequential feature selection and used the ensemble bagged tree algorithm as the base evaluating model. A subset of 2000 samples was used for the feature selection purpose. The algorithm ended up with 9 features out of 19, reducing the feature dimension by over 52%. The selected features with more importance are some MFCC coefficients (1, 3, 4, 6, 7, 8, 13), spectral skewness, and kurtosis. Table I shows the performance evaluation of the three explored machine learning algorithms with the selected 9 features. For this test, the complete training set was used for training, validation, and testing.

TABLE I

RESULTS OF DIFFERENT MACHINE LEARNING ALGORITHMS WITH SELECTED FEATURES. ENSEMBLE CLASSIFIERS PROVIDE THE BEST PERFORMANCE WITH A SUBSET OF THE RELEVANT FEATURES.

Model Name	Validation Accuracy	Test Accuracy
Ensemble Tree	**99.4**	99.2
Support Vector Machine	98.7	99.0
Neural Network	99.0	**99.3**

V. CONCLUSION

Through rigorous experimentation and analysis, we demonstrate the efficacy of the AdaBoost ensemble classifier in gunshot recognition. The spectral characteristics of the audio signal from the gunshots were used for training different algorithms. We exhibited the evaluations of ensemble classifiers, support vector machines, and neural networks across varying Gaussian noise levels. To optimize the performance of the algorithms, we tuned the hyperparameters for each of the algorithms. To reduce the dimensionality we have deployed a wrapper-type sequential feature selection algorithm that reduces the feature space by over 52%. The model still shows similar performance with a few important features compared to the complete set of features. In summary, we present practical insights into the selection and optimization of machine learning models for a robust gunshot recognition system.

ACKNOWLEDGEMENT

This work was funded by the U.S. Department of Justice's National Institute of Justice Grant #2019-BU-DX-0019.

REFERENCES

[1] SoundThinking, "Leading gunshot detection system," https://www.soundthinking.com/law-enforcement/leading-gunshot-detection-system/, Feb 2024, Accessed on 20th February 2024.

[2] Simone Raponi, Gabriele Oligeri, and Isra Mohamed Ali, "Sound of guns: digital forensics of gun audio samples meets artificial intelligence," *Multimedia Tools and Applications*, vol. 81, pp. 30387–30412, 9 2022.

[3] Mustapha Djeddou and Tayeb Touhami, "Classification and modeling of acoustic gunshot signatures," *Arabian journal for science and engineering*, vol. 38, no. 12, pp. 3399–3406, 2013.

[4] Luigi Gerosa, Giuseppe Valenzise, Marco Tagliasacchi, Fabio Antonacci, and Augusto Sarti, "Scream and gunshot detection in noisy environments," in *2007 15th European Signal Processing Conference*. IEEE, 2007, pp. 1216–1220.

[5] Kenneth D Morton Jr, Peter A Torrione, and Leslie Collins, "Classification of acoustic gunshot signatures using a nonparametric bayesian signal model," in *Sensors, and command, control, communications, and intelligence (C3I) technologies for homeland security and homeland defense X*. SPIE, 2011, vol. 8019, pp. 151–161.

[6] Md Siddat Bin Nesar and Bradley M Whitaker, "Audio event recognition in noisy environments using power spectral density and dimensionality reduction," in *2022 Intermountain Engineering, Technology and Computing (IETC)*. IEEE, 2022, pp. 1–5.

[7] Eva Kiktova, Martin Lojka, Matus Pleva, Jozef Juhar, and Anton Cizmar, "Gun type recognition from gunshot audio recordings," in *3rd international workshop on biometrics and forensics (IWBF 2015)*. IEEE, 2015, pp. 1–6.

[8] Muhammad Salman Kabir, Junid Mir, Caleb Rascon, Muhammad Laiq Ur Rahman Shahid, and Furqan Shaukat, "Machine learning inspired efficient acoustic gunshot detection and localization system," *University of Wah Journal of Computer Science*, vol. 3, no. 1, 2021.

[9] Lydia KD Katsis, Andrew P Hill, Evelyn Pina-Covarrubias, Peter Prince, Alex Rogers, C Patrick Doncaster, and Jake L Snaddon, "Automated detection of gunshots in tropical forests using convolutional neural networks," *Ecological Indicators*, vol. 141, pp. 109128, 2022.

[10] John Irungu, Jamelia Ancel, Wagdy Mahmoud, and Max Denis, "Gunshot detection from audio excerpts of urban sounds using transfer learning," in *Proceedings of Meetings on Acoustics*. AIP Publishing, 2023, vol. 51.

[11] Zhicong Chen, Haoxin Zheng, Jingchang Huang, Lijun Wu, Shuying Cheng, Qianwei Zhou, and Yang Yang, "A wireless gunshot recognition system based on tri-axis accelerometer and lightweight deep learning," *IEEE Internet of Things Journal*, 2023.

[12] Alex Morehead, Lauren Ogden, Gabe Magee, Ryan Hosler, Bruce White, and George Mohler, "Low cost gunshot detection using deep learning on the raspberry pi," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 3038–3044.

[13] Ali Raza, Furqan Rustam, Bhargav Mallampati, Pradeep Gali, and Imran Ashraf, "Preventing crimes through gunshots recognition using novel feature engineering and meta-learning approach," *IEEE Access*, 2023.

[14] Muhammad Tahir Bhatti, Muhammad Gufran Khan, Masood Aslam, and Muhammad Junaid Fiaz, "Weapon detection in real-time cctv videos using deep learning," *IEEE Access*, vol. 9, pp. 34366–34382, 2021.

[15] Rahul Nijhawan, Sharik Ali Ansari, Sunil Kumar, Fawaz Alassery, and Sayed M El-Kenawy, "Gun identification from gunshot audios for secure public places using transformer learning," *Scientific reports*, vol. 12, no. 1, pp. 13300, 2022.

[16] Tushar K Routh and Robert C Maher, "Recording anechoic gunshot waveforms of several firearms at 500 kilohertz sampling rate," in *Proceedings of Meetings on Acoustics*. AIP Publishing, 2016, vol. 26.

[17] Thomas G Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[18] Quan Sun and Bernhard Pfahringer, "Bagging ensemble selection," in *AI 2011: Advances in Artificial Intelligence: 24th Australasian Joint Conference, Perth, Australia, December 5-8, 2011. Proceedings 24*. Springer, 2011, pp. 251–260.

[19] Vladimir Svetnik, Ting Wang, Christopher Tong, Andy Liaw, Robert P Sheridan, and Qinghua Song, "Boosting: An ensemble learning tool for compound classification and qsar modeling," *Journal of chemical information and modeling*, vol. 45, no. 3, pp. 786–799, 2005.

[20] Yifeng Yang, Long Wei, Ying Hu, Yan Wu, Liangyun Hu, and Shengdong Nie, "Classification of parkinson's disease based on multi-modal features and stacking ensemble learning," *Journal of Neuroscience Methods*, vol. 350, pp. 109019, 2021.

[21] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[22] Mohaiminul Islam, Guorong Chen, and Shangzhu Jin, "An overview of neural network," *American Journal of Neural Networks and Applications*, vol. 5, no. 1, pp. 7–11, 2019.

[23] Li Yang and Abdallah Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.

[24] Wei Liu and Weisi Lin, "Additive white gaussian noise level estimation in svd domain for images," *IEEE Transactions on Image processing*, vol. 22, no. 3, pp. 872–883, 2012.

[25] Alan Jović, Karla Brkić, and Nikola Bogunović, "A review of feature selection methods with applications," in *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*. Ieee, 2015, pp. 1200–1205.